

UNCLASSIFIED

AD NUMBER

ADB006740

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies only; Proprietary Information; 30 SEP 1975. Other requests shall be referred to U.S. Army Command and General Staff College, Attn: ATSW-DD, Fort Leavenworth, KS 66027.

AUTHORITY

ODDR&E ltr, 20 Jan 1976

THIS PAGE IS UNCLASSIFIED

THIS REPORT HAS BEEN DELIMITED
AND CLEARED FOR PUBLIC RELEASE
UNDER DOD DIRECTIVE 5200.20 AND
NO RESTRICTIONS ARE IMPOSED UPON
ITS USE AND DISCLOSURE.

DISTRIBUTION STATEMENT A

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

7
ADB006740

An Interactive Language Query System for Retrieving Alphanumeric Data from
an Army Tactical Data System

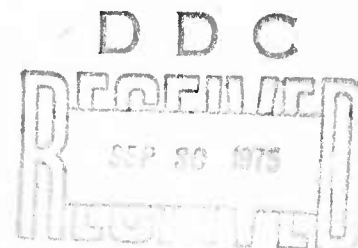
Ian W. Larson, MAJ, USA
U.S. Army Command and General Staff College
Fort Leavenworth, Kansas 66027

Final report 6 June 1975

80 SEP 1975

Distribution limited to U.S. Government agencies only; proprietary information.
Other requests for this document must be referred to U.S. Army Command and
General Staff College, ATTN: ATSW-DD, Fort Leavenworth, Kansas 66027.

A thesis presented to the faculty of the U.S. Army Command and General Staff
College, Fort Leavenworth, Kansas 66027



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Interactive Language Query System for Retrieving Alphanumeric Data from an Army Tactical Data System		5. TYPE OF REPORT & PERIOD COVERED Final report 6 Jun 75
7. AUTHOR(s) Larson, Ian W., MAJ, USA		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Student at the U.S. Army Command and General Staff College, Fort Leavenworth, Kansas 66027		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Command and General Staff College ATTN: ATSW-DD Fort Leavenworth, Kansas 66027		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE 6 Jun 75
		13. NUMBER OF PAGES 90
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Distribution limited to U.S. Government agencies only: Proprietary Information. Other requests for this document must be referred to U.S. Army Command and General Staff College, Fort Leavenworth, Kansas 66027. 30 SEP 1975		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Master of Military Art and Science (MMAS) Thesis prepared at CGSC in partial fulfillment of the Masters Program requirements, U.S. Army Command and General Staff College, Fort Leavenworth, Kansas 66027		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) See reverse.		

DDC
RECEIVED
SEP 30 1975
C

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

The purpose of this study is to describe a query system, based on an interactive query language (IQL), as an alternative to this use of formats.

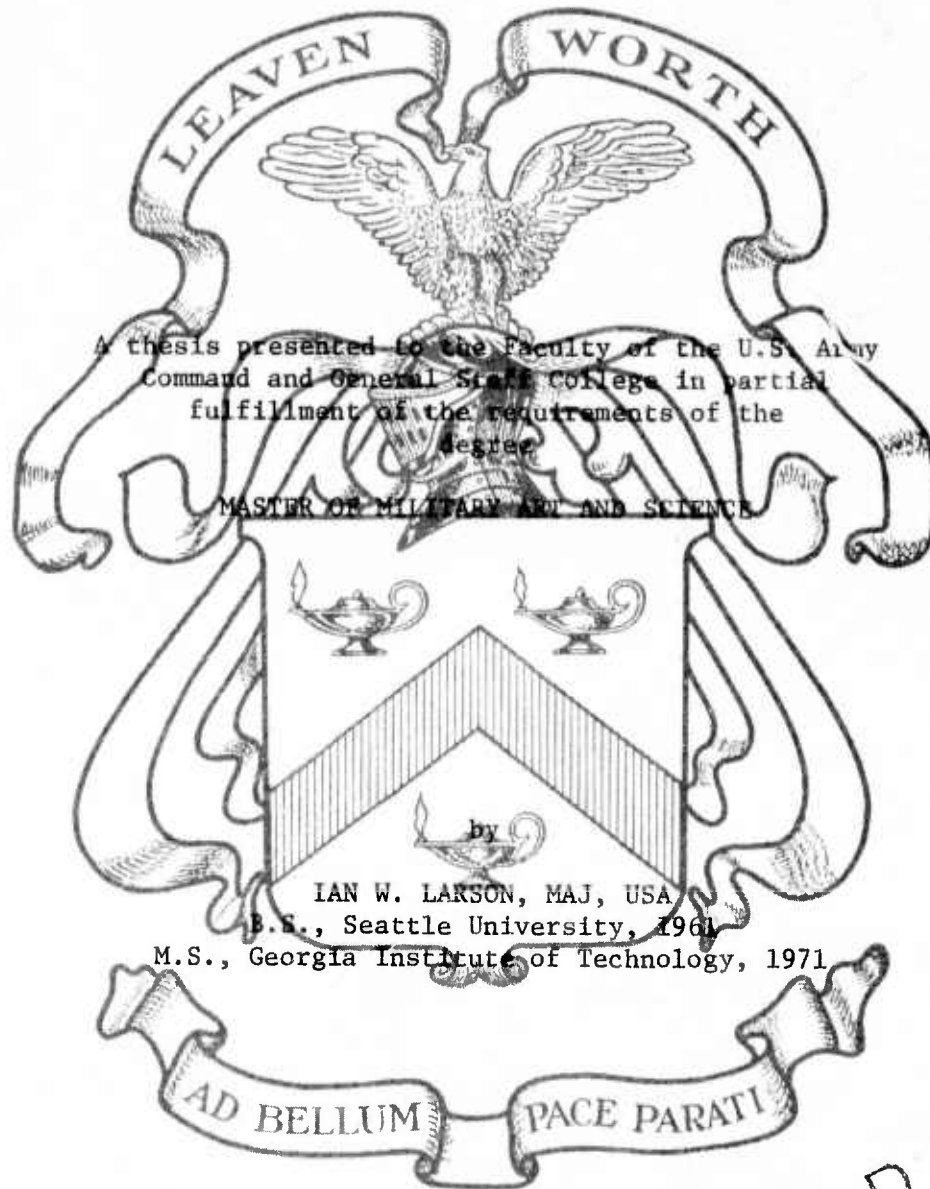
The methodology encompassed four steps. First, user requirements for interactive query were stated, based on an analysis of data flow in a division tactical operations center. Second, the assumed hardware and software capabilities of a tactical data system which would use the proposed query system were stated. Third, research of the literature and of existing systems having IQLs was conducted to determine the state of the art in IQL design. Fourth, a proposed query system was described. The query system was based on an IQL and designed to meet the stated user requirements.

The study recommends implementation of the proposed query system on an experimental basis for evaluation by field users. It also recommends that related studies be conducted in three areas impacting upon query system design: the structure of the tactical data system's data base, the standardization of data terms and abbreviations, and the design of the tactical data system's command language, of which the query system language is a subset.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

AN INTERACTIVE LANGUAGE QUERY SYSTEM FOR
RETRIEVING ALPHANUMERIC DATA FROM
AN ARMY TACTICAL DATA SYSTEM



Fort Leavenworth, Kansas
1975



ABSTRACT

For a decade the United States Army has been investigating the uses of computers in the area of tactical operations. While demonstrating that computer based tactical data processing systems can provide useful assistance, field tests have uncovered some operational problems. One major problem has been the inability of prestructured formats to meet user requirements of flexibility and generality in querying the tactical data system for alphanumeric data. The purpose of this study was to describe a query system, based on an interactive query language (IQL), as an alternative to this use of formats.

The methodology encompassed four steps. First, user requirements for interactive query were stated, based on an analysis of data flow in a division tactical operations center. Second, the assumed hardware and software capabilities of a tactical data system which would use the proposed query system were stated. Third, research of the literature and of existing systems having IQLs was conducted to determine the state of the art in IQL design. Fourth, a proposed query system was described. The query system was based on an IQL and designed to meet the stated user requirements.

The stated user requirements for interactive query specify, above all, that the query system be user-oriented. They also state that

the query system be easy to learn and use, permit selective file and data retrieval, permit ad hoc queries to be formulated, and be designed for use by the non-programmer.

The research conducted indicates that the Data Base Management System (DBMS) represents the state of the art in IQL technology. A significant technical feature of the DBMS is the use of a command language designed specifically for querying and updating data files. The "self-contained" DBMS uses a high level, task-oriented language with a vocabulary intended for use by the non-programmer. While many of the features of DBMS languages would meet the requirements of the tactical data system user, the sophistication and complexity of the capabilities they generally provide would overwhelm all but the very experienced user.

The proposed query system describes a simple IQL with which the user can formulate queries to the tactical data system in two ways. The first way is to formulate the query in one step as a legal sentence of the language. The second is to formulate the query in a series of steps with prompting by the system at each step. The prompting instructions can be minimal or very detailed, depending on the user's choice, based on his desire or experience. The proposed query system also includes a number of other features such as the ability of the user to call on the system for assistance in formulating a query and the ability to save queries for later recall and use.

The study recommends implementation of the proposed query system

on an experimental basis for evaluation by field users. It also recommends that related studies be conducted in three areas impacting upon query system design: the structure of the tactical data system's data base, the standardization of data terms and abbreviations, and the design of the tactical data system's command language, of which the query system language is a subset.

TABLE OF CONTENTS

THESIS APPROVAL PAGE	ii
ABSTRACT	iii
LIST OF FIGURES	vii
CHAPTER	
I. INTRODUCTION	1
The Problem and Its Significance	3
Interactive Query Languages	6
The Study	7
Assumptions	8
Organization of Remainder of Thesis	9
II. USER QUERY REQUIREMENTS AND TACTICAL DATA SYSTEM DESCRIPTION	10
User Requirements	10
Assumed Tactical Data System Capabilities	25
III. INTERACTIVE QUERY LANGUAGES: A SURVEY	33
Background	34
Design Approaches to Interactive Query Languages	40
Formulation of Queries Using Interactive Languages	45
Ease of Learning and Use	52
Hardware	55
IV. A PROPOSED QUERY SYSTEM	57
Selection of General Design Approach	58
Hardware Aids	60
Language Syntax	61
Description of Query Procedure	71
Saving of Queries	76
Other User Assistance	77
V. SUMMARY AND RECOMMENDATIONS	79
BIBLIOGRAPHY	84

LIST OF FIGURES

Figure	Page
1. Summary of Required User-Oriented Query System Capabilities	26
2. Interactive Query Languages Studied	35

CHAPTER 1

INTRODUCTION

The past decade has seen an ever-increasing United States Army reliance on computer based data systems. The majority of these systems were developed to assist in the management of functions such as personnel administration, logistics, and finance, in which the procedures were relatively standard, readily defined, and adaptable to automation. The systems were largely implemented with commercial off-the-shelf hardware and existing software techniques.

While using computer technology for the more routine applications noted above, the Army has also been investigating the uses of this technology in the area of tactical operations.¹ While it is generally agreed that automatic data processing systems can provide useful assistance in this area, controversy has existed concerning which current tactical functions should be automated and the degree to which they would be; the echelons (e.g., battalion, brigade, division) at which the functions are to be automated; and the hardware and software configuration of the tactical data processing system. The main goal of the continuing investigation has been to assist the battlefield commander

¹U.S. Army Computer Systems Command [USACSCOM], Letter CSCS-TSP-C, Subject: Narrative History of DEVTOS (23 March 1974), pp. 1-7.

and his staff in making tactical decisions by using the computer to provide data which are more accessible, more accurate, more timely, and more useful than existing (manual) means provide. A family of Army tactical data systems (ARTADS) has been conceptualized for field army employment in the 1980s. This ARTADS family includes systems to provide automated assistance for artillery fire direction and control, air defense, air traffic management, and tactical command and control of maneuver units.

The tactical command and control system of the ARTADS family, the tactical operations system (TOS), is designed for use in a combat unit's tactical operations center (TOC). The TOS will be first used at division level. It will consist of computer hardware and software for the input, storage, processing, and retrieval of data needed by the division commander and his staff in the planning and control of tactical operations.²

Hardware and software design concepts for the TOS are undergoing testing and evaluation to establish the final design specifications. Experiments, studies, and field tests using developmental systems consisting of commercial hardware have been ongoing to determine those command and control functions that should be automated and incorporated into the TOS.³ The data required to support the selected functions will determine the content of the TOS data base. A segment of the TOS, using

²USACSCOM, TOS Fact Sheet (1970), pp. 4-8.

³USACSCOM, Letter CSCS-TSP-C, p. 5.

militarized hardware, called the TOS operable segment (TOS²), is scheduled for field testing during calendar year 1976. Results of the tests will directly influence the final TOS design.

Both the TOS² and the eventual TOS to be fielded will be interactive data systems. An interactive system allows 2-way, on-line communication between the system and its users. The primary aspect of this communication is the feedback which permits the system and the user to acknowledge whether the last message of the other was understood and to provide results for the action requested.⁴

The TOS will be able to present both graphic (e.g., military unit symbols displayed on a map background) and alphanumeric data to the user in the division TOC. The proposed means by which the user will query the TOS² (and, at this point, the TOS) to retrieve alphanumeric data is through prespecified, structured query formats. A specific number of user query formats will be available for callup and display on a cathode ray tube (CRT) console. The particular format to be called up will depend upon the nature of the data to be retrieved. When displayed on the CRT console, a format will essentially be a structured blank form which the user, using the CRT console keyboard, will fill in by typing appropriate entries according to a set of rules he must know.

The Problem and Its Significance

Modern Army Selected Systems Test Evaluation and Review

⁴Charles T. Meadow, Man-Machine Communication (New York: John Wiley & Sons, Inc., 1970), p. 3.

(MASSTER) has conducted tests at Fort Hood, Texas, with developmental tactical data systems using command and staff personnel from Active Army divisions.⁵ Results of the tests have raised serious questions as to the acceptability of prestructured formats as the method for alphanumeric data query. The four major disadvantages of formats discussed below were observed during the tests.

- The user (i.e., the individual or group of individuals who needed the data in order to make a decision) who had not been extensively trained was normally incapable of direct interaction with the system because of the amount of detail involved.⁶ The user was dependent upon a system specialist to perform the interaction. The user's data requirements were provided to the specialist, who then determined which format to call up and fill in to query the system. The specialist was often unfamiliar with tactical concepts, and describing the data requirements so he could understand them and properly query the system was a time-consuming and frustrating user exercise. Users regarded the necessity to go through a specialist to communicate with the system as a serious drawback. (The use of an intermediary to formulate query requests has also been found generally unacceptable by other users.)⁷

⁵USACSCOM, Letter CSCS-TSP-C, pp. 1-7.

⁶Modern Army Selected Systems Test Evaluation and Review [MASSTER], Tactical ADP Test Experience Review (June 1973), p. 2-4.

⁷Richard G. Canning, "Progress in Information Retrieval," EDP Analyzer (January 1970), p. 12; and U.S. Air Force Systems Command, Rome Air Development Center, On-Line Intelligence Processing System, Report RADC-TR-66-2 (July 1966), I, 14-15.

• While many data retrieval requirements of individual users (or groups of users) were recurring or periodic, such as the daily intelligence summary, the majority were relatively simple ad hoc requirements the user developed as he monitored and reacted to the tactical situation. These ad hoc requirements normally required an immediate reply. Examples of requirements of this nature might be stated by a user as follows: "What are the current locations of the 1st Brigade lead units?" or "Find out how many contacts with enemy units of company size or larger we've had in Area X-Ray in the past 72 hours." The use of query formats did not have the flexibility to cope satisfactorily with many of these situation-related requirements.⁸ In some cases no query format existed for the requirement. In other cases the requirement was met by using a combination of several formats. In still other cases the data provided in response to a query format exceeded that which the user actually needed and used, but the format was the only way of retrieving the data.

• Just as the query formats themselves were prestructured, the data outputs provided in response to the queries were also prestructured formats, restricting system query replies to the limited set of output formats. The user did not have the option of specifying or modifying the form of the data output presented to him. Very often, voluminous output burdened the user with unnecessary detail which required

⁸MASSTER, Staff Organization and Procedures, Test Report No. 113 (September 1972), p. 2-18.

synthesis to obtain meaningful information.⁹

• The procedure of filling out the formats was tedious and relatively slow.¹⁰ In many instances the user felt the data he needed could be obtained more rapidly using the existing (manual) methods.

In summary, the use of formats as the means for querying a tactical data system was not suitably oriented to the requirements of the system user. The procedure was slow, complicated, and could not normally be performed by user alone. The prespecified format structure was too inflexible to meet the variety of user queries. User acceptance of an Army tactical data system such as the TOS will come about only after the user becomes convinced the system can satisfy his needs better than the manual system with which he is familiar. For this to occur, the user himself must interact with the automated system in order to appreciate its capabilities. Thus, the means of interaction with the system becomes extremely important, for it is in fact the key to user acceptance.

Interactive Query Languages

The disadvantages relating to the use of prestructured formats pointed to the requirement for investigation of alternative methods by which the user could more easily and more flexibly communicate his alphanumeric data requirements to the tactical data system. The

⁹MASSTER, Tactical ADP Test Experience Review, p. 2-9.

¹⁰MASSTER, IBCS IIA: Experiment 3 Results, Test Report No. 108 (January 1972), p. 84.

prevalent means with which users define queries for interactive data systems is the use of a specific interactive query language and associated hardware aids. The syntax of the query language is often a subset of the system's "command" language. The command language is the means by which the user communicates interactively with the system. It includes, in addition to the query capability, the syntax to allow data base creation and update.

Hardware aids are normally associated with and complement the query language. These hardware aids (e.g., function push buttons) elicit specified system responses when activated.

The Study

The purpose of this study was to describe a query system consisting of an interactive query language (and associated hardware aids) for retrieving alphanumeric data. The query system was tailored to meet a given set of user requirements for data retrieval from an Army tactical data system.

The fourfold general sequence of the methodology was:

- A statement of the tactical data system user requirements for interactive query. These were the requirements that the proposed query system was tailored to meet. A baseline set of requirements was determined from an analysis of data flow in the intelligence (G2) and operations (G3) staff sections of an Army division, the initial intended users of the TOS. It should be noted that these requirements relate only to how the user defines the output desired from the system. The

determination of the inputs required to establish and update the data base from which data will be output is a related problem and was not the subject of this study. (See the second assumption.)

- Concurrent with the statement of user data query requirements, a statement of assumed hardware and software capabilities of the tactical data system with which interaction occurs.

- Research of the literature to determine interactive query language (and associated hardware aids) features, capabilities, and design considerations.

- Description of a proposed query system tailored for the given tactical data system and stated user requirements.

Assumptions

Four assumptions necessary to limit the scope of this study were:

- Automatic data processing assistance is, and will continue to be, required in the division TOC.

- The data base structure of the system with which interaction occurs is compatible with the query method used. The study does not concern itself with how the system data are structured and updated. Rather, the focus is on the means by which the user specifies retrieval of stored data.

- The interaction occurs with a tactical data system whose hardware and software capabilities are technically feasible within the existing state of the art. The description of such a system and its

hardware and software capabilities is based upon the TOS design concepts and changes recommended by various tests and studies relating to the TOS.

- The interactive query language defined as a result of this study will be a subset of the tactical data system's command language. The command language will include the appropriate syntax to enable data base update (i.e., adding, changing, or deleting data items) as well as special syntax for retrieval, display, and update of graphical data.

Organization of Remainder of Thesis

Chapter II of this thesis describes user requirements for data query and states the assumed tactical data system's hardware and software capabilities. Chapter III discusses the features, capabilities, and design considerations of interactive query languages and available hardware aids. Chapter IV describes the query system tailored to meet the user requirements stated in the second chapter, while Chapter V summarizes the major points of the thesis and makes recommendations based on those points.

CHAPTER II

USER QUERY REQUIREMENTS AND TACTICAL DATA SYSTEM DESCRIPTION

The twofold purpose of this chapter is to state user requirements for querying to retrieve data stored in a computer based tactical data system and to describe the broad characteristics and capabilities of an assumed tactical data system from which the data are to be retrieved. The user requirements and the tactical data system description together establish the starting point for defining the query system to be used in the data retrieval process.

User Requirements

What does the user of an Army computer-based tactical data system want a query system to do for him? To arrive at an answer more useful than the obvious "to retrieve data from the tactical data system," it is necessary to define what is meant by the terms "user" and "query system."

For purposes of this study, the user of the query system is defined to be a member of the G2 (intelligence) or G3 (operations) staff section of an Army division as specified by current tables of organization and equipment (TOE). The restriction of the term to these two predominant users of data in the division tactical operations center

(TOC) is in consonance with the envisioned use of the tactical operations system (TOS) initially at division by the G2 and G3 staffs.

In broad terms, a query system is defined to be a method by which a user communicates with the tactical data system to identify and specify both the alphanumeric data that will be retrieved by the tactical data system from its data base and the output format in which the data will be presented to the user.

With both the purpose of the query system and the user of the system established, key aspects of the definition of the query system can be further analyzed. From the analysis, implications of the definition as they relate to the tactical data system, the user, and the query system are determined. These implications are used to derive requirements for the query system as seen from a user viewpoint.

Query System As a User Tool

The first part of the definition states that a query system "is a method by which" "Method," as used in the context of the definition, is synonymous with "tool." Thus, a query system is a tool designed to be used for a given purpose by a particular user or group of users having identifiable characteristics and requirements. Its acceptability and usefulness as a tool is directly dependent upon how well its design accommodates the characteristics of the user and meets the user requirements for its stated purpose. In other words, the query system must be user-oriented. While this conclusion is rather obvious, it is one whose application cannot be treated lightly in the design of the

query system, particularly in the matching of the query capabilities to the user characteristics. The requirement that the query system be user-oriented is the most general and, at the same time, the most important, because it permeates and influences all other, though more specific, requirements.

Query System Design To Incorporate User Characteristics

The second aspect of the definition is that of the "user." The identification of specific types of users implies the existence of specific user characteristics.

The division G2 and G3 staff composition is specified by a TOE which states the position, rank, and military occupational specialty (MOS) of each individual on the staff. For example, the TOE for a mechanized infantry division shows 13 potential individual users (officer and enlisted) in the G3 section and 12 in the G2 section. The rank structure of these potential users varies from private first class (E3) to lieutenant colonel (O5). The general characteristics of the G2 and G3 staff sections are as described below.

- Different potential users of a tactical data system exist not only between the G2 and the G3 sections but also within each of the staff sections. Each individual potential user has data requirements unique to the task he performs. Some of the potential users may spend most of their working hours monitoring or interacting with the tactical data system (e.g., much as radio-telephone operators do in the manual

system), while other users will use the system less frequently.

- The potential user, upon first being exposed to a tactical data system, has little or no previous hands-on experience with automatic data processing systems.

- The degree of individual user skill and experience when using the query system can vary from relatively little to the ability for fully exploiting the capabilities of the system. This is due partly to the variation in individual learning ability and sophistication among the users, partly to the continual turnover of personnel through reassignments or replacement, and partly to the amount of time the individual's function requires him to use the query system.

- The users are military personnel, and they use military terminology in accomplishing their functions.

- The primary functions of the G2 and G3 staff users are distinct from the use of the tactical data system, i.e., the tactical data system aids the users in accomplishing their functions.

The following required query system capabilities derive from the stated user characteristics:

- The query system should be general and flexible enough to allow retrieval of unique data required by different potential users. This implies that the system procedures should be independent of the unique user data retrieved.

- The query system should be designed to be as easy to learn and use as is practicable. Its operation should assist, rather than

burden, the G2 and G3 staffs.

- The query system, while designed to be easy to use, should include procedural options for both the beginner and the more experienced user (e.g., a structured procedure for the beginner and options for "shortcuts" by more experienced users).

- The query system should minimize the use of new or unusual terminology and should use, whenever possible, military terms and English words or phrases with which the user is familiar.

Query System Interactive Capability

Analyzing the definition further, the "user communicates with the . . . system." This aspect implies that interaction occurs between the user and the tactical data system. "To identify and specify" imply a procedure that the user follows. Finally, communication implies a language or a technique to let the user and the system be understood by each other.

The required query system capabilities derived from the above are tempered by the overall requirement for user-orientation, requirements relating to user characteristics, and lessons learned. First, by definition, the query system is interactive. Second, the procedures for interaction must be simple to learn and to use, must accommodate both the beginner and the more experienced user, and must allow the user himself to perform this interaction. The user himself has the best concept of the output he requires from the system. Reliance upon a system specialist to perform the data query is time-consuming and

potentially difficult, especially if the specialist does not fully understand the tactical situation (even though the specialist may fully understand the system data files and the query procedure).

A third requirement is that the language or method used for communicating user data requirements to the system should not be so complex that its use burdens rather than assists the user. Military terminology and/or symbology and English-type words or phrases should be used in lieu of strange or unusual terminology. Fourth, the query system should be responsive. It should provide the user immediate feedback during the interaction to acknowledge user completion of a step in the procedure or to notify him of an error. Fifth, the variety of potential users and the further variety of unique data required by these users demand that the query system be as independent of the data base content as possible. The query system itself should need little or no modification if user data files are added, deleted, or modified. The rationale for this requirement is that the dynamic nature of combat developments and concepts may cause the TOS data base to change and to become significantly different in content and structure from the data base used with the initial TOS.

Query System Design Driven By User Requirements

The purpose of the query system is now examined. The definition states that the query system is used "to identify and specify both the . . . data . . . and the output format." The statement implies the

existence of user requirements relating to both data and output formats for data presentation.

To identify the requirements of the G2 and G3 staff sections, U.S. Army Field Manual 101-5, Staff Officers Field Manual: Staff Organization and Procedure, is used as a starting point. It states five functions that are performed by a staff: to provide information; to make estimates; to make recommendations; to prepare plans and orders; and to supervise. These functions are performed within the overall staff mission of assisting the commander in making and implementing necessary decisions to accomplish the unit's mission.

Doctrinally, FM 101-5 also states the specific functions of each staff section which supports the five general staff functions. While this formal doctrine standardizes the specific functions of the various staff sections in a division, it does not specify the actual data files that must be maintained to support these functions, leaving this responsibility to the individual division staffs. The files maintained are usually specified by a formal or informal standing operating procedures (SOP). Differences exist in the number and type of files maintained among division TOCs. However, the similarity of functions performed among TOCs tends to minimize these differences when considering the generic content of the combined data files. The differences are primarily a matter of file structure preference, specific content of individual files, and unit-peculiar data. For example: An airborne division may maintain a file containing data peculiar to its airborne capability

which non-airborne divisions would not maintain.

In accomplishing specific functions attendant to the five broad functions noted above, the G2 and the G3 staff sections become focal points for data flow. Each is a point of data input from, or a source of data output to, subordinate, higher, adjacent, and supporting units outside of the TOC. Within the TOC, data flow also occurs between the G2 and the G3 sections and between the G2 or G3 section and other staff sections of the division. Data flow also occurs within the staff sections. The data flow process is complex and dynamic. Although varying in minor detail among individual divisions, the basic data flow process, like the generic content of the data files maintained, is somewhat standardized by the similarity of G2 and G3 staff functions from division to division.

Field studies have attempted to document the data flow process in a division TOC.¹ By viewing the results of the studies from the level of staff function similarity among division G2 and G3 staffs, generalized data (i.e., data base content-independent) and output format

¹Data relating to the sources of information, the type of information sent, and the destination of the information in a division TOC may be seen in Modern Army Selected Systems Test, Evaluation and Review [MASSTER], Staff Organization and Procedures, Test Report No. FM 119 (May 1974); U.S. Army Combined Arms Center [USACAC], Division Command Post System Test Support Package [Short Title: CP Study] (December 1974); and U.S. Army Research Institute for the Behavioral and Social Sciences, Development of an Informational Taxonomy of Visual Displays for Army Tactical Data Systems, Research Memorandum 74-4 (February 1974).

requirements can be determined.² These requirements apply to the design of a query system which can be used for a tactical data system supporting any division. Analysis of the data flow process from this level provides the information about:

- The kinds of files maintained by the G2 and the G3 staff sections and the nature of data contained therein.
- The kinds of products the G2 and G3 produce from the data files they maintain (i.e., the data outputs). This is the core of the analysis. The user product is defined by two parameters--the data needed and the output format of the data. Since the query system is essentially the means provided to the user to define a product, information about the user products relates directly to required query system capabilities.

The content of the G2 section files differs from that of the G3 section files because of the dissimilarity of functions of the two sections. In turn, the content of the products each section produces from the data contained in its files differs. Yet, the files and products are similar when described, exclusive of content, in terms of the

²The data flow process is governed by the current (manual) methods of operation. The extent to which this process will be changed when TOS is implemented will depend on several factors, including how much of the process will be automated; the restructuring of data files, if any, as a result of automation; the effect on the user of greater data availability and accessibility; and the perceived need for new data flow processes or requirements as a result of experience accumulating from more and more use of automation. In any case, the abstraction of the requirements from the data flow process in general terms makes the requirements relatively independent of changes to the data flow process and minimizes the probability of their becoming invalid.

kind of files or products they are.

The data files maintained by the G2 and G3 sections are of two general kinds: permanent or historical files and temporary or working files.³ The working files are the most frequently accessed and updated; they contain data that have the most significance for minute-to-minute operations in the TOC. Displays, a subcategory of the working files, consist of data intended for viewing and are located so as to be easily visible to those who require access to the data. Displays may be entirely alphanumeric, may consist of military graphic symbols, or may be a combination of both. Common examples of displays are the friendly (G3) and enemy (G2) situation maps (alphanumeric and military graphic symbols), the G3 task organization chart (alphanumeric), and the significant G2 enemy activity chart (alphanumeric). Working files which are not displays generally contain alphanumeric data. The current log or journal maintained in each section and the G2 workbook are examples of this kind of working file.

Historical files contain data having less significance for the immediate situation. The data stored in them are of a historical nature and are accessed when reference to some past friendly or enemy activity is needed. Data from working files, when no longer of immediate significance, are often placed in historical files. Historical files can consist of alphanumeric or military graphic symbols, or both. Examples of historical files are the G3 spot report file (alphanumeric), the

³USACAC, p. 2-20 et passim.

G3 operations order file (alphanumeric and military graphic symbols), and the G2 intelligence summary file (alphanumeric).

Either or both kinds of files described above are accessed when the G2 and G3 staff sections must provide an output elicited by the tactical situation. The major reasons for file access and the kinds of products provided as a result of the access are discussed in the remainder of this section.

- Files are accessed to obtain data used in the composition of periodic products. Periodic products are recurring and are output in multiple copy at specified times (normally stated in an SOP) for distribution to subordinate and higher units, to other staff sections within the division, and to adjacent or supporting units as required. Examples of this kind of product are the intelligence summary (INTSUM) provided by the G2 and the situation report (SITREP) provided by the G3. This kind of product is provided in hard copy and may be exclusively alphanumeric or may contain both alphanumeric data and military graphic symbols (e.g., overlays). In preparing this kind of product, one or more files may be accessed to obtain the required data. For example, in preparation of the SITREP, the G3 staff requires data from the situation map, the spot report file, the command post location chart, and the current journal or log. These periodic products normally have a specified format, usually dictated by an SOP.

- Files are accessed to obtain data used in the composition of one-time situation-related products. This kind of product is similar to

the periodic product except that its output is keyed to a specific situation and time. Like the periodic product, it normally has a specified format, is produced in multiple hard copy, may be exclusively alphanumeric, may contain both alphanumeric and military graphic symbols, and may require access to more than one file in its production. Examples of this kind of product are an operation order (OPORD) and an operation plan (OPLAN).

- Files are accessed to obtain data used in a periodic oral briefing or an oral report. An SOP normally specifies the periodic reports and briefings given. The report or briefing can be face to face or it may be by radio or telephone.

- Files are accessed to obtain data to answer one-time situation-related queries. This is the most frequent reason for file access.⁴ The query may be internally triggered within the staff section (e.g., the G2 wants additional information concerning a particular enemy unit); or, the query may be from without (other staff sections in the division, the commander, higher or subordinate units, etc.). Characteristically, this type of query is ad hoc in nature; selective of the file(s) accessed, the data desired from the accessed file(s), and the format in which the data are output; keyed to a specific situation; and usually requires an immediate response. In specifying the query:

⁴One-time situation-related queries are a direct consequence of the necessity for the commander and his staff to remain abreast of the tactical situation and, at the same time, to develop the situation so that a tactical advantage will be gained over the enemy.

1. Selectivity is provided by a number of parameters, the most commonly used being:

- a. Time: "When did the patrol from the 1/17 Cav get hit?"
- b. Distance: "How far is he from Checkpoint 16?"
- c. Direction: "Which way was the vehicle moving?"
- d. Speed: "How fast was the vehicle moving?"
- e. Unit Identification: "Which of our battalions lost most tanks today?"
- f. Strength (Percentage): "Which enemy units do you estimate to be below 70 per cent in both personnel and tracked vehicles?"
- g. Topic or Subject: "Are there any minefields south of the bridge?" This parameter offers greater or lesser selectivity by the use of subtopics. For example, the topic "vehicle" may have as subtopics "wheeled" and "tracked."
- h. Location: "Where is the command post of the 1/506 Infantry?"
- i. Branch: "Is that a mechanized or armored unit at these coordinates?"
- j. Size: "What size force is the 2/311 in contact with?"
- k. Number: "How many tanks were reported?"

2. Additional specifications of the query may be called for in the form of processing of the accessed data before it is output. The processing may include one or a combination of the following:

- a. Calculation
 - (1) Percentage: "What percentage of our contacts were at night?"
 - (2) Counting: "How many of our mechanized infantry battalions meet the radiation exposure criteria?"

(3) Adding: "What is the total number of operational tanks in the two forward brigades?"

- b. Comparison and Compatibility with Established Limiting Parameters: "Give me a list of all identified enemy armor units of battalion size or larger located in Area X-RAY."

3. The final specification of the query is the output format in which the data will be presented. The G2 and G3 sections use the following alphanumeric formats:

- a. List: Successive alphanumeric entries in a single column or sequential entries in one or more successive rows.
- b. Matrix: Alphanumeric entries in two or more columns and two or more rows.
- c. Fixed Format with Free Text Entries: Self-explanatory.
- d. Free Text: Self-explanatory.

It is noteworthy that the first three reasons stated for accessing data files actually begin with ad hoc situation-related queries. In the case of the products with specific formats which require data from several files, ad hoc queries to each file involved must be first generated in order to obtain the data which will be used to compose the product. In the case of files accessed for periodic reports or briefings, one ad hoc report, or more, must be likewise generated to obtain the required data for the briefing or the report.

Query System Capabilities

Having described the general characteristics of the user data and output format requirements, it now remains to relate these require-

ments to five broad capabilities that a query system for alphanumeric data should possess.

- It is generally agreed that the requirement for rapid output in response to ad hoc queries and the random nature of these query occurrences dictate that the query system be on-line to be effective.⁵

- The query system should be general enough to apply to any G2 or G3 staff in a division TOC. It should be flexible enough to accommodate the different user requirements generated by the dissimilarity of G2 and G3 staff functions, the different potential users within each staff section, and the different requirements that each individual user is capable of generating.

- The query system should allow selective file retrieval and, within each file selected, selective specification of the data desired from the files. The system should allow, as a minimum, retrieval using one or a combination of the retrieval parameters previously discussed: time, distance, direction, speed, unit identification, strength, topic or subject, location, branch, size, and number.

- The query system should allow the following data processing before output: (1) calculation and (2) comparison and compatibility with established limiting parameters.

- The query system should allow user designation of the output device and definition of the output format. In this regard:

⁵Richard G. Canning, "Progress in Information Retrieval," EDP Analyzer (January 1970), p. 12.

1. It is assumed that the tactical data system will have different output devices (e.g., a cathode ray tube alphanumeric display console and a printer). The query system should allow designation of the appropriate device, perhaps with function push buttons labeled "Display" or "Print" or by the use of the terms in the query formulation.

2. The following selection of alphanumeric output formats should be made available: list, matrix, fixed format with free text entries, and free text.

Figure 1 summarizes the required query system capabilities identified as a result of the foregoing analysis. A survey of the state-of-the-art interactive query language design techniques to achieve these capabilities is presented in Chapter III.

Assumed Tactical Data System Capabilities

To place the query system defined in Chapter IV in its proper context, it is necessary to describe the assumed characteristics of the tactical data system that the query system will be a part of. The characteristics are presented here as assumptions inasmuch as the TOS system concept is still undergoing evaluation and the detailed TOS design specifications remain unfinalized. A second reason for stating the characteristics as assumptions is that a general system description is sufficient for the purposes of this study. While incorporating the impact of major lessons learned thus far in Army tactical data system testing and evaluation, the more general description is less likely to

INTERACTIVE: Permits 2-way communication between user and system

ON-LINE: Permits rapid response to user queries

FLEXIBLE (GENERALIZED)

Permits use by both G2 and G3 sections

Provides for:

Specification of file(s) to be accessed

Selection of data to be retrieved from accessed file(s)

Processing of data before output

Designation of the output device

Specification of the output format

Accommodates data file structure or content modifications

EASY TO LEARN

Simple: Operation detracts little from user's staff function

Syntax: Military terminology; English-type words and phrases where possible

Procedures: Straightforward; as few as possible

EASY TO USE PROCEDURES

Permit user himself to interact

Provide user assist and error notification

Provide "shortcuts" for skilled users

RESPONSIVE

Rapid feedback: Acknowledgment of user request/command or of error notification

Rapid response to queries

Fig. 1.--Summary of Required User-Oriented
Query System Capabilities

be inconsistent with the eventual detailed TOS hardware and software specifications.

The computer-based tactical data system to be fielded for use initially by the G2 and G3 staff sections of Army divisions will be real-time,⁶ on-line, and interactive. Its purpose is simple: to provide automated assistance (for selected functions) to the G2 and G3 staffs that, when compared to current means, will allow the user to have more rapid access to needed data, will have a greater store of data available for access, and will provide more accurate and more complete responses to user queries for data. As the user becomes more experienced in the system's employment, the system can help in identifying new user requirements. In view of current developments in the state of the art, it is not unreasonable to assume that hardware technology will have developed to the point that the major equipment additions to the division will come from items the user will have had "hands-on" contact with, the system's input and output devices (i.e., peripheral devices). Miniaturization and advanced technology will lessen the impact of adding the processors and data storage devices and will be compatible with the Army's continuing efforts to make division command posts less cumbersome in terms of personnel, equipment, and mobility.

⁶The term "real time" has different meanings for different users. For this study, a real-time system is defined to be one that provides responses to user inputs within a few seconds. If the action required by the user input will take longer than 10 seconds, the user is made aware of the delay. In no case, however, should the user have to wait more than 30 seconds before beginning to receive system output in response to a given input.

The system will have militarized hardware designed to withstand the effects of operation around the clock seven days a week, environmental extremes, and the sometimes not-so-gentle treatment by users under the strain and pressures of combat. Procedures and equipment will be standardized to facilitate the interoperability of the system with similar systems supporting other echelons (e.g., other divisions or higher headquarters). Continuity of operations will be enhanced by means such as redundancy of critical system components and techniques which will minimize system down time due to scheduled and unscheduled maintenance. Finally, the equipment will incorporate the latest design measures to increase operability and survivability in the electronic warfare environment of the modern battlefield.

The system will be physically located at the division's main command post. It is assumed the main command post will be similar in configuration to that described in the U.S. Army Combined Arms Center study dated December 1974.⁷ In the configuration described, the G2 and G3 staff sections operate from separate 5-ton expansible vans. The tactical data system's peripheral devices required to support the G2 and the G3 sections would be located in these two vans. The central processors and data storage devices may also be located in the same vans, but, if space does not permit, they could be housed in a separate but smaller van (or truck) within the main command post area.

⁷USACAC, PP. I-4 & I-5.

The tactical data system will consist of a central processor (computer) with associated data storage capability and a number of peripheral (input/output) devices. While the specific number of each device needed by the G2 and G3 staffs is yet to be determined, test results indicate that one or more of the six devices whose stated capabilities are discussed below may be used in the division TOC.⁸

- Interactive graphic display console. This device is the primary user means of graphically depicting and monitoring the friendly and enemy tactical situation. The console consists of a screen for displaying the situation (friendly or enemy military unit symbols and graphic control measures) in color against a map background that is also in color. The G3 may use the console to monitor the current friendly tactical situation; the G2 uses a similar console to monitor the enemy situation. The displays that can be viewed at a console are not limited to those that are within the exclusive purview of a specific user. For example, the G3 can call up the current enemy situation and other G2 displays, while the G2 can call up the friendly tactical situation and other G3 displays. Controls appropriate to the display technology used (e.g., a light pen and keyboard would be expected with a console having a cathode ray tube screen) permit the manipulation (addition, deletion, modification) of data on individual displays. Displays on the console may be stored for later recall. Other console features include the

⁸MASSTER, IBCS [Integrated Battlefield Control System]: Automated Displays, Test Report No. FM 116 (July 1974), pp. 13-22.

ability to display data by individual category, such as radar sightings, or to "overlay" two or more categories, such as radar and air reconnaissance sightings; the ability to "point" to a symbol on the screen and automatically obtain its map coordinate location; and the ability to change the map area of coverage for the display, then automatically have displayed the appropriate data by category against the new map background. The interactive graphic display console can also be used to display and manipulate strictly alphanumeric data, but the alphanumeric display console described below is better suited for this purpose.

- Alphanumeric display console. As its name connotes, this device is the primary one for the input, retrieval, display, and manipulation of alphanumeric data. A display screen and the appropriate controls (suitable for the display technology used) enable the input of items such as unit situation reports to update the system data base. The device is also used to display alphanumeric outputs in reply to user queries. The terminal speed will be sufficient to preclude undue user wait for display of output. (An operating speed of at least 4800 bits per second is assumed. This speed will permit screen display generation in terms of a few seconds.) The query system described in Chapter IV of this study is intended for the user interacting with the tactical data system through this console.

- Group viewing device. Both the interactive graphic display console and the alphanumeric display console screens are intended for viewing by the individual console user. When numerous situations demand

that a group of users view the same display at the same time, the group viewing device may be required. Except for the display screen size (which will accommodate an audience of up to 20 persons), the displays that can be viewed on the group viewing device are identical to those that can be displayed on the interactive graphic display console. This device is most useful for briefings. It should be included as a component of the tactical data system only when anticipated usage justifies the added system space requirements, weight, and complexity.

- Map overlay input device. This device allows the tracing of free-hand graphic data, such as found on overlays, for input to the system data base. The device permits registration of the overlay with the appropriate map. Once entered into the data base, the overlay is available for display on the interactive graphic display console or on the group viewing device.

- Map overlay output device. It may be necessary to produce hard copies of a tactical situation displayed on the interactive graphic display console for use by higher, subordinate, or adjacent units or for use within the TOC itself. This device enables overlays to be produced in up to three colors and in the standard Army map scale desired. The product of the device is on transparent material suitable for placing over the appropriate map.

- Printer. Alphanumeric hard copy is provided by the printer.

The capabilities highlighted above have been stated as seen from the user viewpoint and were confined to those system components through

which the user interacts with the system. This interaction, of course, includes the querying of the system to retrieve data. Whatever the eventual design specifications of the TOS may be, it is quite certain the system capabilities will be called on through the use of peripheral devices such as described above. It is assumed that these capabilities will not significantly differ from those outlined. The query system described in Chapter IV is designed to be used with a tactical data system having the same (or similar) input/output capabilities as those just enumerated.

CHAPTER III

INTERACTIVE QUERY LANGUAGES: A SURVEY

This chapter is an overview of the state of the art in the design of interactive query languages for the automatic data processing system user. The focus is on the user who is a non-programmer, i.e., the user whose primary function is apart from the automatic data processing system operation, but who interacts with the system to obtain data needed to help perform his function. Design aspects discussed include the general alternative approaches to interactive query language design and implementation and the means by which the user formulates query statements in the language. As background, a brief discussion of the significant factors in the evolution of interactive query languages precedes the main portion of the chapter.

Two categories of documentary literature provided source material. The first, and the less abundant of the two by far, included textbooks and documents relating specifically to the data query and retrieval aspects of man-machine communication. The second consisted of descriptive documentation on existing automatic data processing systems with on-line interactive data query and retrieval capability. The documentation reviewed included descriptions of experimental systems, military systems, and systems designed for commercial applications.

Adequate published literature exists on interactive query languages described as components of operational systems. However, there are few sources which provide comprehensive and authoritative design methodology for these languages. The relative newness of significant developments in interactive query language (IQL) use, the proliferation of languages, and the absence of agreed-upon standards by users and system vendors alike contribute to the situation.

The IQLs of 15 different on-line systems were studied in detail to determine how the language enabled the user to selectively identify alphanumeric data for retrieval, processing, and output. Other user features of the language and the hardware aids associated with the language were also noted. Collectively, the systems listed in Figure 2 represent the state of the art in IQL development. The cross-section of systems from the commercial, military, and experimental sectors spans the range of the types of systems that exist.

In addition to the systems listed in Figure 2, the general interactive query features and capabilities of approximately 25 other on-line systems were examined.

Background

The state of the art in IQL design reflects the increased attention being given to the operational user who is a non-programmer. The linking of the teletypewriter with the computer made on-line man-machine communication a reality. At first, this interaction was exclusively between the computer and the system programmer, in a programming

<u>System Name</u>	<u>Type</u>	<u>Developer</u>
CONVERSE	Experimental	System Development Corporation
ENVIRON	Experimental	Stanford Research Institute (for Office of Naval Research)
Information Management System (IMS)	Commercial DBMS	International Business Machines
Management Data Processing System (MADAPS)	Military	System Development Corporation (for U.S. Air Force)
MARK IV	Commercial DBMS	Informatics, Inc.
Model 204 GDBMS	Commercial DBMS	Computer Corporation of America
Multi-Access Retrieval System (MARK VI)	Commercial DBMS	Control Data Corporation
On-Line Information Retrieval System	Experimental	University of Pennsylvania School of Electrical Engineering
On-Line Information System for Army Force Planners	Military	Research Analysis Corporation (for U.S. Army)
Program Assisted Console Evaluation and Review (PACER)	Military	Rome Air Development Center U.S. Air Force
REALITY	Commercial DBMS	Microdata Corporation
SYSTEM 2000	Commercial DBMS	MRI Systems, Inc.
Tactical Information Processing and Interpretation (TIPI) System	Military	System Development Corporation (for U.S. Air Force)
Texas Water Oriented Data Bank	In-House DBMS	State of Texas Water Development Board
User-Oriented On-Line Data System	Experimental	Syracuse University Research Corporation (for U.S. Air Force)

Fig. 2.--Interactive Query Languages Studied

language. When the remote operation of the teletypewriter (and later, the cathode ray tube (CRT)) terminal became possible, the concept of several remote terminals operating simultaneously through time-sharing evolved.

Early applications of interactive data processing in business, industry, and the military required the user to learn the appropriate programming language if he wanted to interact with the system; otherwise he had to relay his requirements to a system programmer who performed the actual interaction.

Widespread application of interactive data processing systems in business, industry, and the military in the 1960s required system designers to consider users who were non-programmers. Many of these had neither the time nor the inclination to learn a programming language in order to interact with the system and, at the same time, were less than willing to rely exclusively on a system programmer to obtain data. Thus, languages less complicated than programming languages for communicating data requests to interactive data systems received impetus for development. Even so, IQLs traditionally were not accorded heavy attention relative to the design of the rest of the system. The language syntax often evolved in an ad hoc manner after the rest of the system was designed.¹ User characteristics and criteria were often neglected,²

¹George H. Walther, "Is Flexibility in Query Languages 'Good' for Everyone?," Proceedings of the 4th Annual Computer Related Information Systems Symposium (U.S. Air Force Academy, January 1974), p. 8-2.

²H. Sackman, Experimental Investigation of User Performance in

and IQL design favored user-system communication in terms more convenient for the system than the user.

Sufficient user outcry (and the spur of competition) caused more and more system designers to provide interactive capabilities that were easier to use. By the late 1960s, new data management techniques which would affect the user-system interface were evolving. The most significant of these was the concept of generalized file processing and the outgrowth of the terms "data management system (DMS)," "data base management system (DBMS)," and "generalized data base management system (GDBMS)."

Military requirements provided the original thrust for generalized file processing.³ These requirements were the large volume of data that had to be processed quickly and the need for rapid response to a variety of queries for several files in a data base. The accepted procedure for querying the diversity of files was to write a program for each file. When the file was modified, this often necessitated a modification of the query routine as well. Furthermore, queries written for one file could not be executed on other files since they were applicable only to that file. Fry mentions other motivations for generalized file processing, including that of allowing the non-programmer to interface

Time-Shared Computing Systems: Retrospect, Prospect, and the Public Interest (Santa Monica, Calif.: System Development Corporation, May 1967), p. 62 et passim.

³J. Fry and J. Gosden, Survey of Management Information Systems and Their Languages, Report No. MTP-313 (Bedford, Mass.: Mitre Corporation, May 1968), p. 1.

with the data base.⁴

The trend, then, has been away from application oriented data files toward common data bases that serve many or all applications and toward generalized software for processing the data files. Generalized file processing software acceptance and use have spread in much the same way as the acceptance and use of operating systems.

Canning sees the DMS as a broader term which includes DBMS as a subordinate term, along with other aspects of data management such as data communications.⁵ A DBMS (sometimes also referred to as a "generalized" DBMS (GDBMS) to emphasize its general applicability) controls a data base consisting of logically-connected files. The DBMS (or GDBMS) has four basic functions: data base creation, data base update (maintenance), data retrieval, and report generation.

A DBMS has two significant technical features. The first of these is the concept of a file description existing as a separate entity from the file itself. This permits the writing of programs with a degree of independence from the specific file formats. The second feature is the use of a command language designed specifically for querying and updating data files. The form and intended user of the language tends to place a DBMS in one of two classes. A "host language" DBMS uses a command language that is used in conjunction with a host

⁴James P. Fry, "Managing Data Is the Key to MIS," Computer Decisions (January 1971), p. 8.

⁵Richard G. Canning, "Trends in Data Management, Part I," EDP Analyzer (May 1971), p. 1.

programming language such as COBOL (common business oriented language) and is intended for use by the expert programmer. A "self-contained" DBMS uses a high level, task-oriented language with a vocabulary intended for use by the non-programmer. Some DBMSs provide both a host language and self-contained capability.⁶

The DBMS is a relatively new concept, and its evolution continues. Its wide acceptance has resulted in a proliferation of a great number of dissimilar systems. While this provides the potential user with a wide range of initial choices, the lack of standardization among systems often "locks in" a user to a specific vendor once the vendor's system has been selected, because of the potential difficulty of software conversion when switching to a different or more sophisticated system. Nonetheless, it is clear that the DBMS represents the state of the art for file processing techniques in general and for interactive data query in particular.

It is difficult to predict where the continuing evolution of DBMS design will lead, but Martin summed up the case for even greater attention to be paid to the user. He wrote:

Increasingly in the next decade, man must become the prime focus of system design. The computer is there to serve him, to obtain information for him, and to help him do his job. The ease with which he communicates with it will determine the extent to which he uses it. Whether or not he uses it powerfully will depend upon the

⁶For example, the U.S. Air Force Military Personnel System (HQ, USAF, System 2.5). The system is described in U.S. Air Force Academy, Generalized Data Base Management Systems and Selected Air Force Applications (April 1973), pp. 64ff.

man-machine language available to him and how well he is able to understand it. To be effective, systems will have to be designed from the outside, in. The terminal or console operator, instead of being a peripheral consideration, will become the tail that wags the whole dog.⁷

Design Approaches to Interactive Query Languages

The systems listed in Figure 2 (page 35) are representative of the different approaches that have been taken in the design of an IQL for alphanumeric data. Martin listed a number of approaches which encompass graphic as well as alphanumeric data base queries.⁸ However, a convenient way of classifying the general design approaches is whether the language allows the query to be formulated as a sentence of the language in one step or whether it requires multiple steps with assistance from the system. Using this classification, the types of approach are discussed below.

- The query can be formulated as a well-formed sentence of the IQL. The alternatives for this approach are the types of syntax used for the language.

1. Programming languages. Concise, precise, and flexible queries are possible with a syntax using a programming language (or programming-like statements). Host-language DBMSs typify this approach. The disadvantage, of course, is that it is unsuitable for the vast number of users who neither learned to program nor want to.

⁷James T. Martin, Design of Man-Computer Dialogues (Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1973), p. 3.

⁸Ibid., pp. 12-13.

2. Natural language. While this may be the most natural way of communicating data queries to the system, and indeed that it is possible, as the CONVERSE and other experimental systems demonstrate, its implementation is fraught with difficulties. The syntactic and semantic ambiguities of the English language raise huge problems in precisely interpreting a query. In fact, even the "natural language" systems such as CONVERSE use only an English subset comprehensive enough to cover the range of expression of the query language. Moreover, queries with this syntax tend to be wordy (therefore more lengthy) and take longer to compose and type. Until breakthroughs are made in voice recognition (so the user can talk to formulate his query) and natural language interpretation by the system, this approach will not be a realistic one for commercial or military applications.

3. Limited English. The majority of languages studied use a subset of English. Some, such as the IQL on the TIPI System, have very cryptic queries, while others, such as the ENGLISH language on the REALITY DBMS, allow fairly conversational queries to be formulated. The advantages of this approach are that the interpretive problem is reduced to manageable proportions, yet the user is provided with a vocabulary of words familiar to him. A possible disadvantage is that some users may credit the system with more intelligence than the interactive query language permits and may overstep the language rules of syntax (or data base knowledge) for query formulation.

4. Mnemonics. Some systems have a language relying entirely

on mnemonic commands, the ENVIRON system being a case in point. Most systems which use mnemonics, however, include these abbreviations with a more extensive English subset. For example, the SYSTEM 2000 DBMS has a list of commands which can be used in complete or abbreviated (mnemonic) form, depending on the experience (or whim) of the user. While the exclusive use of mnemonics allows the formulation of concise and precise queries, its disadvantage is that the user must know them. A large number of mnemonics would take time to learn and remember. Some users may not use the language often enough to justify the training time required.

- The query is formulated with assistance from the system.

This approach seems to be geared toward the user who is untrained or who uses the system infrequently enough to require system assistance. The approach may also be useful if the queries to the system tend to be long and/or complex. Common means of applying this design approach are:

1. Question and Answer. In this technique the dialogue is usually system-prompted, that is, the user responds to questions or directions from the system. Since the system, in effect, tells the user what to do at each step of the query formulation, little user training is required. A disadvantage of this technique is that the dialogue can be lengthy and slow. It may even be frustrating to the experienced or skilled user if he has no "shortcut" option and must doggedly follow the step-by-step process. It is also possible that the question and answer routine may satisfy only a limited range of user queries and would

require modification if other query requirements were identified.

2. Menu Selection. A technique often combined with the question and answer technique is for the system to present the user with a "menu" from which he selects one or more items to respond to a system question or direction. Depending upon hardware implementation, the user may make his selection by typing, that is, depressing appropriate function buttons, or by pointing to the items with devices such as a light pen, a movable cursor, or even his finger. The example below, from the Texas Water Oriented Data Bank,⁹ is typical of the menu selection technique in which responses are typed by the user.

System direction: WHICH SUBCATEGORY ARE YOU
INTERESTED IN (ENTER NUMBER)

- 1) SURFACE
- 2) SUBSURFACE
- 3) MAN'S ACTIVITIES

User reply: 3

Where the number of possible user responses to a system prompting is known and limited, the menu selection technique is advantageous in that it simplifies the procedure for the user.

3. Format Filling. This technique displays a "format" which the user fills out. While its use, once learned, is straightforward, the disadvantages of procedure slowness and inflexibility described in Chapter I of this study restrict the use of the technique to those

⁹Texas Water Development Board, Texas Water Oriented Data Bank: General Information On and Introduction To the Data Bank Monitor (Austin, Tex., February 1973), p. 8.

applications which have relatively standard queries (i.e., not ad hoc). Formats may be most efficient for specialized, non-changing queries and when the number of formats is small.¹⁰

Given the alternative language approaches just described, the selection of the approach (or combination of approaches, as appropriate) depends on three major factors: the purpose of the interactive query, the characteristics of hardware terminal at which the user exercises the capabilities of the interactive query language, and the characteristics of the user. The purpose of the query and the hardware terminal characteristics are readily definable; the characteristics of the user are less objectively stated.

Martin¹¹ has stated several major considerations for establishing user criteria. Those applicable to IQLs are:

- The type of user. A dedicated user spends his entire time working at the terminal. A casual user spends most of his day doing something entirely different from using the terminal and approaches the terminal only occasionally to query for data needed to help perform his job.
- Programming skill of the user.
- User intelligence. The IQL degree of difficulty in learning and use should be commensurate with the capability of the prospective user.

¹⁰U.S. Air Force Academy, p. 38.

¹¹Martin, pp. 25-30.

- Length of training. Good training is desirable, but some users will be unable to undergo detailed, lengthy training. Length of training can also be affected by the expected availability of the system for actual user training.
- Intermediary users. The users may not actually interact with the system; they may depend upon an intermediary to bridge the gap between the user and the system.
- User harassment. Working conditions of the user could range from quiet seclusion to turmoil and occasional emergencies. Time, other users, and external pressures will also affect him.
- User tolerance. Some users will be too busy for trial and error. Others, in spite of errors, will have the persistence and the time to proceed with the dialogue until successful.
- Multiple applications. Some users may perform a single application at a terminal. Others may carry out a variety of applications, each with its peculiar dialogue structure.

Formulation of Queries Using Interactive Languages

The functions a query formulated in the syntax of an IQL performs are (1) to selectively identify that (alphanumeric) data which the user desires to retrieve from a data base for output (or for further processing before output) and (2) to specify the output device and output format. This section, which describes the syntactic features of state-of-the-art IQLs by which these two functions are accomplished, is

based upon research of the IQLs listed in Figure 2 (page 35). The discussion focuses on IQLs which are not programming languages.

Before proceeding with the discussion, however, it is useful to establish the following definitions. A record is a collection of related values treated as a logical unit. Related values are called fields. Each field is a set of characters containing a unit of data. A field normally has one or more values in each record. A collection of related records makes up a file, and a data base is the collection of all data files (and file descriptions). The data base structure is the manner in which data values in a file are logically organized and visualized so that all legal relationships among values, fields, records, and files can be expressed or implied.¹²

A question affecting IQL design is: "How much should the operational user need to know about the data structure?" Ideally, he should simply state what data he wants retrieved, not how the data are to be retrieved.¹³ Theoretically, this is possible with a data base structure in precise alinement with the user's data needs and characteristics. In practice, the rules of syntax of the IQL are governed to a significant degree by the way in which the data are structured. To formulate syntactically valid queries, the user is required to know what data are

¹²U.S. Air Force Systems Command, Electronic Systems Division, Operational Considerations in the Use of Data Management Systems, Report ESD-TR-72-265 (May 1973), "Glossary," pp. xii-xxxii.

¹³Richard G. Canning, "Problem Areas in Data Management," EDP Analyzer, March 1974, p. 10.

contained in the data base and to be aware of the relationships among fields, records, and files. However, he should not need to be aware of the storage structure (i.e., the mapping of logical records onto physical storage).¹⁴

Selective Data Identification

The IQLs are quite similar in that they have a more or less basic set of capabilities for selective data identification and they use similar techniques in providing these capabilities. The capabilities and techniques are as described below. The examples provided to amplify the description are not associated with any one language; they are typical of what is available.

- Identification of the file to be searched. Each file is normally assigned a name by the user when it is defined and made a part of the data base. The most common way of identifying the file to be searched is to state its user-assigned name as part of the query, e.g.: "LIST (file name) WHERE" The majority of the languages permit only one file to be associated with a query. With some languages, "file" is synonymous with "data base," and the data base name must be given in an initial command, e.g.: "DATA BASE NAME is (data base name)."

- Selection of records

1. Some DBMS have a unique identifier associated and stored

¹⁴Canning, "Trends in Data Management, Part I," p. 9.

with each record. This permits records to be selected by specifying the unique identifier, or a range, if the identifier is numeric, e.g.:

"PRINT (file name) ("n₁ - n₂") . . ." could be used to identify records with unique identifiers between n₁ and n₂, inclusive. Records selected by unique identifier preclude a search of all records in the file.

These records can be further qualified as described below.

2. The overwhelming method of record selection is by qualification through user-provided criteria. The criteria are defined by the values of one or a combination of fields in the query statement. This method of record selection requires that every record in the file (or those records previously selected by unique identifier) be tested to determine whether they qualify, i.e., meet the user-provided criteria. In the query statement, a reserved word of the language (e.g.: WHERE (qualifications), WITH (qualifications)) commonly precedes the user specified criteria.

- Qualification of records

1. Essentially, every IQL allows comparison of the actual value of a user specified field against a user specified value. The results of individual comparisons on the fields are then logically combined using Boolean operators to establish a "qualified" or "not qualified" result for the record.

2. Individual comparisons require a user specified field, a relational operator, and a user specified value for the field. The user specified field is normally entered as its legal field name. (The use

of codes for field names in the query statement is optional with some languages.) Available relational operators vary from one language to another, but a somewhat standard set is common: greater than, GT; less than, LT; equal to, EQ; greater than or equal to, GE; less than or equal to, LE; not equal to, NE. Other available relational operators may include those for existence tests (e.g.: EXISTS, FAILS) and for range tests (e.g.: IS BETWEEN). The languages usually provide more than one way to specify the field value against which the actual field value is compared. Some possible ways are numeric constant, an alpha literal, a computed value (usually limited to the use of the four arithmetic operators), a range, and the value of another field in the same record. The general form of the individual comparison in a typical query statement is: . . . (field name) (relational operator) (value)

3. The results of individual comparisons are combined through the use of Boolean operators. The use of AND and OR is almost universal; a majority of the languages also provide NOT. To eliminate ambiguity when a number of connections are made, common techniques include the establishment of a precedence among operators, the use of parentheses, or both. Other languages simply connect the clauses from left to right. The usual form of the combination of individual comparisons in the query statement is

. . . (individual comparison) (Boolean operator) (continuation) . . .

where (continuation = (individual comparison)

= (individual comparison) (Boolean operator) (continuation)

- Processing of data in qualified records. The IQLs provide for calling upon a number of similar data processes with the query statement. The four most commonly available in the DBMS languages are adding the numeric values of a set of fields (e.g.: SUM); counting the occurrences of a given item, such as the number of records that meet a set of user-specified criteria (e.g.: COUNT); computing an average (e.g.: AVG); and finding a maximum or minimum value of a field common to a set of records (e.g.: MAX, MIN).

Output Specification

Three specifications are stated in this portion of the query: the designation of the output device, the data to be output (based upon the user criteria to be or previously provided), and the format of the output data.

Most of the languages studied support systems having two kinds of output devices: one for hard copy (a printer or a typewriter terminal) and one for display of the output (a CRT terminal). Therefore the designation of the kind of output desired is used to connote which output device will be used. The command PRINT is used by virtually all the languages to specify hard copy, and in systems having both a printer and a CRT console its absence from the query statement results, by default, in a display of the output.

Designation of the data to be output is usually a sequential listing of the desired field(s) of the records which qualify, using either the legal field name or its code. In some of the IQL query

statements, commas are used to delimit the fields, while in others a blank between the fields serves the purpose. The entire record is usually output by default if specific fields are not stated.

Some IQLs for specialized applications automatically format the output data and provide no user control over the format. Other languages, such as those available on commercial DBMSs, provide options that allow total user control of the output format. It is not uncommon for the optional format routine to be a module separate from the basic IQL itself; the routine would be called upon by the IQL in the query statement. In those instances the data retrieved as a result of the query is normally placed in a system "scratch pad" and is acted upon when the user defines the desired format for the data. The format routine usually allows, as a minimum, title and column heading definition, establishment of the field size for output values, left and right character justification, horizontal and vertical spacing, and paging.

When the optional format routine is not called on, a default format is used for the output and is usually a single column of sequential data. However, there are variations. The ENGLISH language of the REALITY DBMS automatically uses a multi-column format with suppressible headings (a single column is used only if the number of columns exceeds the width of the page), while the language of the SYSTEM 2000 DBMS can provide a single or multi-column format depending upon the command word used in the query statement.

An output specification capability of most IQLs is the sorting of the output data (alpha and numeric) in ascending or descending order according to some user-provided key, usually the name or code of a field (e.g.: ". . . SORT BY (field name) DOWN . . ."). A number of languages also provide a command to limit the volume of the output, usually by a maximum number of output values (e.g.: ". . . LIMIT n . . ."; or, ". . . STOP IF LINES EXCEED n . . .").

Ease of Learning and Use

Every self-contained commercial DBMS brochure extols the ease with which the non-programmer can learn and use the system's query language. Much emphasis is given to the "naturalness" of interaction with the system through the use of English-like query statements. Unfortunately, the claims fall somewhat short. Without concentrated training (an estimate, based upon the study of a number of DBMS user manuals, would be a minimum of two concentrated 8-hour days), the typical casual user can never hope to do more than formulate only the simplest queries, a trivial use of the system. Much of the training time must necessarily be spent learning the data structure and the rules governing relationships between fields and records in the files.

With training, the casual user can use most of the basic capabilities of a DBMS query language. Even so, the syntax of the typical DBMS language provides capabilities beyond the basic ones for the formulation of sophisticated and complex queries that would overwhelm all but the IQL designer, the system analyst, or very experienced (and most

likely, dedicated) user. The casual user may know only one way to formulate a query for certain data; the expert would probably know several.

The rules of syntax for IQLs which allow queries to be formulated as a sentence of the language are as varied as the names of the systems they support. Most languages do attempt to let the query approximate an English sentence. For example, the ENGLISH language of the REALITY DBMS defines a basic query as a sentence consisting of a verb (e.g.: PRINT) followed by one or more nouns (data to be output; records to be searched), then by one or more attributes (qualifications of the records), and linked by appropriate connectives. As another example, the query sentence of the Model 204 DBMS uses several lines in a query sentence, but attempts to approximate the natural way a query would be stated in English by allowing all queries to start with "FIND ALL RECORDS FOR WHICH . . ." followed by the records qualification statement and the output statement, also in English-like clauses, on succeeding lines. The naturalness of the clause is enhanced by the use of filler words, a technique used in some IQLs. In the clause "FIND ALL RECORDS FOR WHICH," the only word the system recognizes is the key word FIND. All the other words are non-key word fillers and serve only to make a more English-like query statement, but the system ignores them when the query statement is processed.

With all of the languages, the basic sentence building blocks are not difficult to learn in themselves. The difficulty is in learning

the various rules that govern the legal combination of the individual building blocks to make up valid queries.

Except for the interactive query languages which support computer-prompted interactions, direct assistance in formulating queries is not normally provided the user. All of the languages provide diagnostic error messages when the user makes an illegal query. The utility of the messages varies from IQL to IQL. Some of the messages are descriptive enough to enable the user with sufficient training to recover from the error. Other IQLs do not provide very helpful diagnostics and place the burden on the user to determine the exact nature of the error. None of the languages studied states explicitly whether an error in query formulation requires that the user start anew and reformulate the query; or, if that part of the query prior to the error is saved and the formulation continues from that point after the error is corrected.

Immediate feedback as the user formulates his query is a function of the terminal at which the query is formulated. Available terminals let the user view as a display or as a line of print the progressive formulation of the query as he types each character.

A number of languages permit the user to define synonyms for certain system words and, once defined, their use in sentence structure is as legal as the system words. In addition, the languages allow for the definition and use of abbreviations for words which are frequently used.

The use of default values or conditions is common in virtually

all the IQLs to help simplify the procedure for choosing between options.

Most languages allow valid query statements to be saved and recalled for execution on demand. Some languages also allow for the saving of logical parts of a query statement. The parts can be recalled and used in the makeup of a new query.

Hardware

The vast majority of existing systems using an interactive language to query an alphanumeric data base employ user terminals having a standard alphanumeric keyboard with a printing device or a CRT display screen, and with terminal speeds ranging from 300 to 9600 bits per second. On both types of terminals, the user types on the keyboard to formulate his query or responses in the man-machine dialogue.

Building all or part of the interactive language dialogue into the terminal hardware is a design possibility; however, it suffers from the possible disadvantages of inflexibility and higher cost.¹⁵ The necessity to keep costs as low as possible in order to successfully compete for computer system applications is one constraint a system developer faces; another is that the applications themselves are subject to a high rate of change. Designing a dialogue around a custom-built keyboard containing many words of an interactive language may be building resistance to change in the system. However, where the

¹⁵Martin, p. 143.

application is precisely defined and relatively unchanging, the use of the special terminal has been an acceptable design approach.¹⁶

If part of an interactive language were built into the hardware, its most likely manifestation would be in a special keyboard containing function push buttons that would represent words, commands, or other parts of the language vocabulary. The function push buttons, when activated, would elicit the appropriate system response. When the special keyboard is used for more than one application, an overlay or template can be used for each application. The overlay, placed over the keyboard, relabels the keys for the application.

Computer-prompted dialogues employing a CRT screen can employ, in addition to the alphanumeric keyboard, a light pen device for pointing to one or more items on the screen. The light pen is a convenient way of selecting items from a displayed menu. Future systems may not need a light pen. Systems with screens that detect the touch of a finger are being developed.¹⁷

¹⁶Martin, p. 152.

¹⁷Martin, p. 161.

CHAPTER IV

A PROPOSED QUERY SYSTEM

This chapter describes a query system employing an interactive query language and associated hardware aids for on-line query of alphanumeric data from an Army tactical data system. The tactical data system, whose assumed characteristics were described in Chapter II, supports users from the G2 and G3 sections of a division. The query system is designed to provide the user-required capabilities also described in Chapter II.

The query system is presented as a proposed solution or function to meet user needs. In this light, the definition of the IQL syntax and the description of the interactive procedure are not concerned with the invisible task of system interpretation of query language statements to machine language instructions for operation on the stored data.

The major element of the query system is the interactive query language. A detailed definition of the language syntax required to formulate queries and an equally detailed description of the interactive query procedure are provided. Other capabilities and characteristics of the system are less germane to the interactive query problem and are described only as is necessary to provide an understanding of the capability or characteristic.

Selection of General Design Approach

As noted in Chapter III, two distinct design approaches to interactive query dialogue are a dialogue in which the query is stated as a legal sentence of an IQL (interactive query language) and a dialogue in which the query is formulated in a series of computer-prompted steps. The former approach is the overwhelming preference, particularly in systems marketed commercially. The main reason appears to be economic. Computer-prompted dialogues, while undoubtedly easier for the untrained and casual users, are more costly to implement than having the user type the query in one step. The computer-prompted dialogue may not be feasible for very complex applications with a vast range of possible query structures.

In selecting the design approach for the proposed query system, cost is not as great a constraint as in commercial systems, where the pressure to minimize costs (for both the system designer and the user) in order to survive in the competitive environment is a major determinant of the capabilities that will be designed into the system. This is not meant to imply that cost will not be a consideration in the selection; only that the main consideration should be on choosing the hardware and software design alternatives that would be most advantageous for the user. It was attempted to incorporate existing IQL features which would satisfy user requirements in as simple a manner as possible. The vocabulary of the IQL was constrained to the minimum essential to meet the majority of the user's potential queries.

Neither of the two general design approaches was relied on exclusively. Instead, the proposed query system combines both approaches. The user can formulate a query either by typing it in one step or by interacting with the system in several computer-prompted steps to create the same query, using the same language syntax. The five advantages below are complementary and to the user's benefit.

- Either query option allows the user himself to interact with the system.

- Users with different levels of skill are easily accommodated. The computer-prompted option has two further options: one in which the instructions are complete but do not contain explanatory detail; and another in which the instructions are very detailed and tutorial in nature (for the new user and for training in the use of the language). For the experienced user, the option to type the query may be preferable.

- The availability of the computer-prompted option minimizes the requirement for casual users to be thoroughly familiar with the IQL syntax.

- The use of the same language syntax for both query options facilitates the learning of the language. Once familiar with either option, the user can use the option most advantageous for the specific query.

- Typing the query may be faster, minimizes the length of the dialogue, and is preferable for short, simple queries. The computer-

prompted option may be easier for more complex queries and may produce fewer errors.

The IQL of the proposed system uses a limited English syntax. A programming language was out of the question since the language users would not be programmers. Natural English is not yet state of the art, and the use of mnemonics requires unnecessary memorization in addition to producing very cryptic query structures. The limited English syntax makes the user-to-system communication more natural than either mnemonics or programming statements.

The computer-prompted query system uses menu selection and user-entered responses to formulate queries in the syntax of the IQL. The selection from a displayed menu is by the most convenient means possible: direct pointing by the user to the item(s) on the screen.

Hardware Aids

The alphanumeric display console through which the IQL will be used will have, in addition to the keyboard and a display screen, a pointer for selecting from menus displayed on the screen. If the display technology allows, pointing will be by the user's finger or by an instrument external to the console, such as a pencil. The console will also have a visible cursor for locating the position of a character on the screen. The cursor will be capable of being both automatically positioned by the system and manually controlled (e.g., backspacing to permit correction of detected typing errors).

The following function push buttons will be part of the keyboard

or located on the console within easy reach of the user: QUERY, ABORT, HELP, SAVE, PRINT, and TRANSMIT. Their use is explained in subsequent parts of this chapter.

The hard copy device, a high speed printer, is controlled through the alphanumeric console. In all cases, output from a query is first displayed on the console screen before transmittal to the printer. Depressing the PRINT push button will cause the printer to output hard copy data identical to that displayed on the console screen at the time the PRINT push button was depressed. The exception to this is when the output exceeds one page, in which case the PRINT push button activation provides the entire multiple-page output rather than individually by page as displayed on the console screen.

Language Syntax

The syntax of the IQL is described through the use of metalinguistic symbols. These symbols and their meanings are:

\$ \$ Left and right dollar signs are used to contain one or more characters representing a metalinguistic variable in a metalinguistic formula. A right dollar sign enclosing a variable immediately followed by a left dollar sign enclosing a second variable means that the second variable must follow the first.

::= means "is defined as"; it separates the metalinguistic variable on the left of the formula from its definition on the right.

/ means "or"; it separates multiple definitions of a metalinguistic variable.

; means that any constructs which follow it are optional.

: denotes that the query construct is complete.

() Parentheses are used to enclose metalinguistic variables defined by the meaning of the English-language expression contained within the parentheses. This formulation is used when it is impractical or impossible to use a metalinguistic formula.

Any symbol in a metalinguistic formula which is not one of the above symbols denotes itself.

The above symbols are used in forming a metalinguistic formula.

A metalinguistic formula is a rule which will produce an allowable sequence of characters and/or symbols. The entire set of such formulas defines the constructs of the interactive query language.

The syntax for a valid query is as follows:

\$valid query\$::= \$data output command\$ \$output specification\$; \$selection criteria\$:

\$data output command\$::= COUNT / GET SUM OF / GET MAXIMUM VALUE OF /
GET MINIMUM VALUE OF / FIND PERCENTAGE OF /
LIST

\$data output command\$ specifies one of several types of output:

COUNT results in an integer number output of the number of records meeting criteria specified in the remaining part of the query.

GET SUM OF results in a numeric output value. It is the algebraic sum of the values of a specified field belonging to the records

meeting criteria specified in the remaining part of the query.

The abbreviation of this command is SUM. GET and OF are filler words.

GET MAXIMUM VALUE OF results in a numeric output value. It is the highest algebraic value of a specified field belonging to records meeting criteria specified in the remaining part of the query. The abbreviations of this command are MAX or MAXIMUM.

GET MINIMUM VALUE OF is similar to GET MAXIMUM VALUE OF except that the lowest algebraic value of the field is output. The abbreviations of this command are MIN or MINIMUM.

FIND PERCENTAGE OF results in a numeric output value. The values of two specified fields belonging to records meeting criteria specified in the remaining part of the query are algebraically summed, then the sum of the first field is divided by the sum of the second field to arrive at the percentage. The abbreviations of this command are PCT, PERCENT, or PERCENTAGE.

LIST results in a sequential listing of the values of specified fields (up to and including all fields of a record) belonging to records meeting criteria specified in the remaining part of the query.

\$output specification\$::= \$report\$ / \$file name\$ /

\$file name\$ \$field names\$ /

\$file name\$; SORT BY \$sort key\$ /

\$file name\$; SORT BY \$sort key\$ DOWN /

\$file name\$ \$field names\$; SORT BY \$sort
key\$ /

\$file name\$ \$field names\$; SORT BY \$sort
key\$ DOWN

\$report\$::= (a formatted or free text alphanumeric entity of one or more pages, identified by a name or a code, and stored in the tactical data system for recall as required; an example of a report might be the current intelligence summary, identified by the name INTSUM)

\$file name\$::= (any valid name or abbreviation of the file being queried; for example, ENEMY ORDER OF BATTLE may be the name of a file, abbreviated ENOB)

\$field names\$::= (one or more legal names or abbreviations of fields belonging to records in the file being queried; the fields are listed sequentially, separated by commas; for example, field names of records belonging to the ENOB file might include IDENTIFICATION, TYPE, and LOCATION)

\$sort key\$::= (the legal name or abbreviation of the field upon which the output will be ordered; the value of the field may be alpha, numeric, or alphanumeric, but the field itself must have only one value)

The following constraints apply to \$output specification\$:

- The presence of SORT BY (or its abbreviation, SORT) in the

\$output specification\$ results in output ordered on the \$sort key\$. The ordering will be ascending unless DOWN follows the \$sort key\$, in which case the output will be in descending order. SORT BY is legal only when the \$data output command\$ is LIST.

- No \$field names\$ may be used when the \$data output command\$ is COUNT, since this command counts records, not fields.

- Only one legal name can be used in \$field names\$ when the \$data output command\$ is GET SUM OF, GET MAXIMUM VALUE OF, or GET MINIMUM VALUE OF. The field must be single valued, and the value must be numeric.

- When the \$data output command\$ is FIND PERCENTAGE OF, only two single valued numeric fields, separated by a comma, can be used in \$field names\$.

- Fields with multiple values are legal in \$field names\$ when the \$data output command\$ is LIST. All values of a multiple value field contained in \$field names\$ will be output.

- Output from the LIST command will be automatically formatted in multiple columns. Each column will be headed by the name of a field listed in \$field names\$, followed by a listing of the values of the field. If \$field names\$ was not used, the name and values of all fields in the qualifying records will be listed.

- All output resulting from a query will include the query statement as part of the output.

- A limit of 6 pages of output will be in effect, to avoid

burdening the user with excessive data volume. When the limit is reached, a message will be displayed on the user's console screen alerting him to this fact and providing him with the option of overriding the limit if he desires.

\$selection criteria\$::= WHERE \$individual comparison\$ / WHERE \$individual comparison\$ \$comparison clause\$

\$comparison clause\$::= \$Boolean connector\$ \$individual comparison\$ /
\$Boolean connector\$ \$individual comparison\$ \$comparison clause\$

\$Boolean connector\$::= AND / OR

\$individual comparison\$::= \$field name\$ \$relation\$ \$comparand\$

\$field name\$::= (a legal field name or abbreviation of a field belonging to the records in the file being searched)

\$relation\$::= EQ / = / NE / LT / < / GT / > / LE / GE / BETWEEN /
WITHIN

\$comparand\$::= \$f\$ / '\$n'\$ / '\$a'\$ / '\$an'\$ / '\$n1, n2'\$ / '\$n', f\$ /
\$f, 'n'\$ / \$f1, f2\$ / \$named area\$

\$f\$::= (a legal name or abbreviation of a field belonging to the records in the file being searched)

'\$n'\$::= (a numeric value enclosed in single quotes)

`'a'$` ::= (an alpha value enclosed in single quotes)

`'an'$` ::= (an alphanumeric value enclosed in single quotes)

`'n1, n2'$` ::= (two numeric values separated by a comma and enclosed in single quotes)

`f1, f2$` ::= (the legal names or abbreviations of two fields belonging to the records being searched, and separated by a comma)

`$named area$` ::= (a map area bounded and defined by specific map coordinates and given a name or code)

The following constraints apply to `$selection criteria$`:

- `WHERE` begins the `$selection criteria$` and separates it from the foregoing portion of the query statement.

- The Boolean connectors `AND` and `OR` are used to combine `$individual comparison$` results to obtain one result for the combination.

The implicit processing order of the connectors is `AND`, then `OR`. Parentheses may be used to show the explicit grouping of the individual comparisons, and they are recommended when complex groupings are used. For example, the explicit grouping `((V) OR (W AND X)) OR (Y AND Z)` is equivalent to `V OR W AND X OR Y AND Z`. The `AND` of `(W AND X)` and the `AND` of `(Y AND Z)` are processed first, followed by the `OR` of `((V) OR (W AND X))`, then by the final `OR` to obtain the result for the expression.

- When the `$relation$` is `LT` or `<` (is less than), `GT` or `>` (is greater than), `LE` (is less than or equal to), or `GE` (is greater than or

equal to), \$field name\$ and \$comparand\$ must have numeric values unless the value is one for which hierarchies have been defined for the system. For example, BATTALION may be defined to be "greater than" COMPANY, COMPANY "greater than" PLATOON, etc.

- Alpha, alphanumeric, and numeric values of the \$field name\$ and the \$comparand\$ are valid with EQ or = (is equal to) and NE (is not equal to).

- The \$relation\$ BETWEEN specifies a range comparison. The \$comparand\$ must contain two values: the first being the lower value of the range; the second the upper value. Thus \$f1, f2\$, \$f, 'n'\$, \$n1, n2\$, and '\$n', f\$ are the only valid \$comparand\$ structures for BETWEEN. In addition, the values in \$field name\$ and \$comparand\$ must be numeric.

- When the \$relation\$ is WITHIN, the only applicable \$comparand\$ is \$named area\$. The value of \$field name\$ must be a map coordinate. This value will be checked to determine if it lies within the boundaries of the specified \$named area\$.

Using a hypothetical data base called ENSIT (ENEMY SITUATION) having files named ENEMY ORDER OF BATTLE (ENOB) and ENEMY ACTIVITY (ENACT), the following examples of query statements and corresponding outputs illustrate typical constructs in the IQL:

Query: "How many contacts of company size have we had in AREA X-RAY in the past 72 hours?"

Query statement in the IQL:

COUNT ENACT; WHERE TYPE EQ 'CONTACT' AND SIZE EQ 'COMPANY' AND LOCATION
WITHIN X-RAY AND TIME BETWEEN '041600, 071600':

(Note: If the query statement is being typed, a carriage return continues the query statement on the next line. A colon completes the query statement. Depression of the TRANSMIT push button instructs the system to process the query.

Output:

QUERY:

COUNT ENACT; WHERE TYPE EQ 'CONTACT' AND SIZE EQ 'COMPANY' AND LOCATION
WITHIN X-RAY AND TIME BETWEEN '041600, 071600':

DTG: 071626 MAR

COUNT: 8

Query: "What do we know about the location and strength of the
312th Tank Regiment?"

Query statement in the IQL:

LIST ENOB IDENTIFICATION, STRENGTH, LOCATION; WHERE IDENTIFICATION EQ
'312 TANK REGIMENT':

Output:

QUERY:

LIST ENOB IDENTIFICATION, STRENGTH, LOCATION; WHERE IDENTIFICATION EQ
'312 TANK REGIMENT':

DTG: 112108 JUL

IDENTIFICATION	STRENGTH (PCT)	LOCATION
312 TK RGT	90	021630 PL 24101970
		050730 PK 99311620
		101500 PL 07301400

Query: "Get me the last Division INTSUM."

Query statement in the IQL:

LIST INTSUM;:

Output:

QUERY:

LIST INTSUM;:

DTG: 211432 APR

(The output would then be the INTSUM as stored in the tactical data system)

Query: "I need a list, by time, of enemy movement reported since this morning. Give me the location, direction, what was moving, and the size of the movement."

Query statement in the IQL:

LIST ENACT TIME, LOCATION, DIRECTION, SUBJECT, SIZE; SORT BY TIME; WHERE
TYPE EQ 'MOVEMENT' AND TIME BETWEEN '160001, 161900':

Output:

QUERY:

LIST ENACT TIME, LOCATION, DIRECTION, SUBJECT, SIZE; SORT BY TIME; WHERE
TYPE EQ 'MOVEMENT' AND TIME BETWEEN '160001, 161900':

DTG: 171948 SEP

TIME	LOCATION	DIRECTION	SUBJECT	SIZE
160315	NB 31706250	N	PATROL	SQUAD
160321	NB 32506130	NE	PATROL	5 PERS
160418	NB 14205710	S	TRUCKS	2
160628	NB 18106000	N	PATROL	PLATOON
....
....

Description of Query Procedure

A user desiring to formulate a query statement may use the proposed query system in one of three ways, depending on his skill, experience, and preference. First, he may elect to type the entire query. Second, he may formulate the query with moderate prompting by the system; he asks for help only if he needs it. Third, he can formulate the query with maximum system prompting and explanation at each step of the procedure. The user selects the means he will use immediately after activating the QUERY function push button on the console.

The first procedure requires no explanation; the examples provided in the language syntax definition illustrate what the well-formed query constructs are. The second procedure is described below. The third way of using the system is similar to the second except for the much greater amount of explanatory assistance provided by the system. (The detailed guidance is not actually provided here, but it would need

to be a part of the query system if implemented.)

To illustrate the steps of the procedure given below, the example of a query provided earlier is used again: "I need a list, by time, of enemy movement reported since this morning. Give me the location, direction, what was moving, and the size of the movement." To formulate a valid query statement for the above, the user performs the steps below.

Step 1: Activate the QUERY function push button on the alphanumeric display console.

Step 2: The system instructs the user to select the type of query. (Note: "Select" as used in this procedure means the user points at one or more items on the screen to indicate his choice(s).)

QUERY:

SELECT THE TYPE OF QUERY

- NEW QUERY
- USE QUERY PREVIOUSLY SAVED

The user points to the top choice and continues with the procedure below. The use of previously saved queries is described in another part of this chapter.

Step 3: The system displays this menu:

QUERY:

SELECT QUERY FORMULATION METHOD

- TYPE THE QUERY IN ONE STEP
- FORMULATE IN STEPS WITH SYSTEM ASSISTANCE

The user selects the second choice.

Step 4: The query formulation begins. The system displays:

QUERY:

SELECT ONE OF THE FOLLOWING COMMANDS

- LIST
- COUNT
- GET SUM OF
- GET MINIMUM VALUE OF
- GET MAXIMUM VALUE OF
- FIND PERCENTAGE OF

The user selects LIST.

Step 5: If LIST was selected, the system displays Instruction (b). For all other commands, Instruction (a) is displayed. A cursor is automatically positioned where the first character will be typed.

QUERY:
LIST

(a) ENTER FILE NAME

(b) ENTER FILE NAME OR REPORT

The user types ENACT, then depresses the TRANSMIT push button, which takes him to the next step. (Note: If the user types a report name, the query is completed, and the report is retrieved.)

Step 6: The system displays Instruction (a), (b), or (c) below, depending upon the command selected in Step 4. (Note: If COUNT was selected in Step 4, Steps 6, 7, and 8 are bypassed, and the system goes

to Step 9.)

QUERY:
LIST ENACT

- (a) ENTER ONE FIELD NAME FOR WHICH SUM, MAX OR MIN
VALUE WILL BE FOUND
- (b) ENTER TWO FIELD NAMES F1, F2 SEPARATED BY A COMMA.
PERCENTAGE = $F1/F2$
- (c) ENTER ONE OR MORE FIELD NAMES SEPARATED BY COMMAS

For the query example, Instruction (c) would be displayed, and the user types TIME, LOCATION, DIRECTION, SUBJECT, SIZE and depresses the TRANSMIT push button. If there was insufficient space on one line, a carriage return would position the cursor on the next line for continuing.

Step 7: The system displays the following:

QUERY:
LIST ENACT TIME, LOCATION, DIRECTION, SUBJECT, SIZE;

WILL OUTPUT BE SORTED?

- YES, ASCENDING
- YES, DESCENDING
- NO

The user points to YES, ASCENDING and the system takes him to the next step. If he had selected NO, Step 8 would have been bypassed.

Step 8: The system instructs the user to enter the sort key.

QUERY:

LIST ENACT TIME, LOCATION, DIRECTION, SUBJECT, SIZE; SORT BY
ENTER SORT KEY FIELD NAME

The user types TIME and depresses the TRANSMIT push button.

Step 9: If the user has no further search criteria, the query formulation ends with this step. The following message is displayed:

QUERY:

LIST ENACT TIME, LOCATION, DIRECTION, SUBJECT, SIZE; SORT BY TIME;

FURTHER SEARCH CRITERIA?

- YES
- NO. QUERY IS COMPLETE

The user points to YES.

Step 10: The system instructs the user to complete the query by entering the search criteria:

QUERY:

LIST ENACT TIME, LOCATION, DIRECTION, SUBJECT, SIZE; SORT BY TIME;

WHERE

COMPLETE QUERY BY ENTERING ONE OR MORE INDIVIDUAL COMPARISONS

CONNECT INDIVIDUAL COMPARISONS WITH AND/OR

END QUERY WITH A COLON (:)

The user types TYPE EQ 'MOVEMENT' AND TIME BETWEEN '160001, 161900':

Step 11: The procedure is completed. The user may now save the query for later use by depressing the SAVE function push button, or he can depress the TRANSMIT push button to obtain output.

The ABORT, QUERY, and HELP function push buttons may be used at any time during the procedure. When the ABORT push button is activated, the system reverts to the mode it was in before the user depressed the

QUERY push button. If the user wants to re-formulate his query, he can re-depress the QUERY push button, which would return him to Step 2 of the procedure.

If the user is unclear as to the response he should make at any step in the procedure, activation of the HELP push button provides him with detailed instructions for that step of the procedure. When the procedure step has been completed, the system returns to the less detailed prompting mode until the user again depresses the HELP push button during any subsequent steps. Detailed instructions for every step of the procedure from start to finish are provided when the HELP push button is activated immediately after activating the QUERY push button. Besides being available to more experienced users to review as needed, the detailed instructions can provide the new user with self-training in the use of the query system.

Saving of Queries

Queries which are frequently used may be stored by the system for later recall and use. To save a query, the user depresses the SAVE function push button immediately after formulating the query statement. The system automatically assigns a code (e.g., a mnemonic, number, or other identification means) to the query and adds it to the others previously saved and stored. The user can call for a display of stored queries exactly as formulated to review them for selection of one to use or to delete those no longer needed.

An option to use a previously formulated query is provided the user when he initiates a query through depression of the QUERY function push button. He may then enter the code of the query he wishes to use or, not knowing the code, he can display the list of queries and select the one he wants.

A previously formulated query, when displayed, may be modified. Any or all elements comprising the \$data output command\$, \$output specification\$, and \$selection criteria\$ can be changed. The modified query can be stored for later use as described above.

A query no longer required to be stored can be deleted by first displaying the list of queries, selecting the DELETE option from a concurrently displayed menu, and then pointing to the query to be deleted.

When the user takes the option to use previously stored queries, the system concurrently displays a menu at the edge of the console screen containing the commands DISPLAY QUERY LIST, MODIFY QUERY, and DELETE QUERY. Selection from these commands is made to accomplish stored query manipulation as outlined above.

Other User Assistance

Diagnostic error messages will be displayed whenever the user makes an error. The error message will be as explicit as possible and, if appropriate, will state what corrective action the user should take. For example: FIELD NAME F1 MUST HAVE AN ALPHANUMERIC VALUE.

RE-ENTER F1.

Reformulation of queries will not be necessary when errors occur, whether the error happens when the query is being formulated (typed) in a single step or when in a step of the system-prompted procedure. Manual repositioning of the cursor to the incorrect character(s) and entry of the correct data followed by depression of the TRANSMIT push button will permit recovery from the error.

Abbreviations for commonly used words of the language are a convenient way to help shorten the dialogue. In addition to abbreviations, it is anticipated that the syntax of the command language for data base creation will permit the user to define SYNONYMS for certain words. The use of legal synonyms would then be acceptable in the interactive query language.

CHAPTER V

SUMMARY AND RECOMMENDATIONS

The query system described in Chapter IV is a proposed alternative to the use of prestructured formats for querying an alphanumeric data base. The system embodies state-of-the-art interactive query language (IQL) capabilities oriented toward the user requirements described in Chapter II. This writer believes the system is superior to formats mainly in the ability to express ad hoc queries and in ease of use. Still, the query system is only a proposed solution, not the solution. First, it requires evaluation and refinement by users through actual implementation and experimentation. Second, it is a subset (though a major one) of a broader command language. The query language syntax may need modification to insure compatibility with the command language syntax used for data base creation and update.

The study of state-of-the-art IQLs denoted the great progress made in data management software in the past decade, particularly in the latter half. The technology that a data base management system (DBMS) represents provides powerful capabilities. Generalized file processing routines independent of the data base content and a user-oriented language to call on those routines are the two features of DBMSs which are contributing to their widespread acceptability and popularity.

Alphanumeric console display terminals currently rely primarily on keyboard entry of queries. Even display terminals employing menu selection generally require the typing of the appropriate selection. Future terminals will likely see increased use of function push buttons and devices for pointing directly at the screen. Light pen selection from cathode ray tube screens has long been popular on graphic display systems. As graphic systems increase in popularity and as technology improves, pointing at the screen, perhaps with the finger, will be the preferred way to "pick" items in menu-selection alphanumeric dialogues.

During the definition of the IQL syntax for the system proposed in this study, related areas impacting upon the tactical data system and requiring coordination with the query system effort were identified.

These areas were:

- Data base structure. The data content of the tactical operations system (TOS) data base is being determined through continuing field experimentation and studies. Assuming that DBMS technology will be used to develop a command language for the TOS, it is just as important to begin determination of the structure of the data base. Various storage structure alternatives (e.g., inverted, hierarchical, network, relational) exist, and the relationship among data elements defined in the data storage structure has direct impact on the command language syntax, especially in creating and updating the data base.

- Data terms and abbreviations. The names and abbreviations of data items to be used in Army tactical data systems, not just the TOS,

require standardization if interoperability among systems is to be realized. For example, it would not do for one system to use the term COORDINATES while another system uses LOCATION to mean the same thing; nor for one system to use the abbreviation PLT for PLATOON while another uses PLAT.

- Command language design. The command language syntax for data base creation and update should undergo development concurrent with the query language syntax.

This study concludes with four recommendations. The first would continue the effort toward developing an operational interactive query language for users of an Army tactical data system in a division tactical operations center (TOC). The other three would study related areas that impact on the implementation of an interactive query language.

- A query system employing the interactive query language and the query formulation options described in this study should be implemented on an experimental basis, perhaps at the Combined Arms Combat Developments Activity (CACDA) or at the Modern Army Selected System Test, Evaluation and Review (MASSTER). Users with extensive experience in division TOCs as well as users with limited or no experience should be called upon to evaluate the usefulness of having the option to type a query without system assistance or to receive system prompting at each step. The suitability of the language syntax should also be evaluated. The experimentation would include the following tasks:

1. Definition and creation of a test data base.

2. Developing the software for processing interactive query language statements into machine code for execution.
 3. Developing the software programs for system-prompted query formulation.
 4. Developing the error detection and notification programs.
 5. Implementation of the interactive query language on an existing computer system having the appropriate hardware.
 6. User tests and evaluation of the interactive query language.
 7. Continuing modification of the interactive query language through user feedback and experience gained from experimentation results.
 8. Finalization of the interactive query language definition for implementation with Army tactical data systems.
- Concurrent with the experimentation proposed above, a preliminary study effort to determine the command language syntax for data base creation and update should be undertaken. The syntax should be fully compatible with the query language syntax. This study should draw upon the results of the experimentation proposed in the first recommendation.
 - A study should be conducted to determine the most appropriate type of data storage structure for the data base that will support an Army tactical data system in a division TOC. The study should examine the types of storage structures in use and their applicability to the characteristics of data used in a division TOC.

- Continuing effort should be made to standardize data base terms to facilitate the operational compatibility of Army tactical data systems and to standardize the vocabulary used by the command language for these systems.

BIBLIOGRAPHY

BIBLIOGRAPHY

Government Documents

Department of the Army. Staff Officers Field Manual: Staff Organization and Procedure. FM 101-5. July 1972.

Modern Army Selected Systems Test, Evaluation and Review. IBCS [Integrated Battlefield Control System]: Automated Displays. Test Report No. FM 116.

_____. Staff Organization and Procedures. Test Report No. FM 119.

_____. Tactical ADP [Automatic Data Processing] Test Experience Review. June 1973.

_____. Staff Organization and Procedures. Test Report No. 113. September 1972.

_____. IBCS IIA: Experiment 3 Results. Test Report No. 108. January 1972.

Proceedings of the 4th Annual Computer Related Information Systems Symposium. U.S. Air Force Academy, January 1974.

Proceedings of the Second Annual World-Wide Data Management Symposium. U.S. Air Force Academy, December 1971.

U.S. Air Force Academy. Generalized Data Base Management Systems and Selected Air Force Applications. April 1973.

U.S. Air Force Systems Command. Electronic Systems Division. Operational Considerations in the Use of Data Management Systems. Report ESD-TR-72-265. May 1973.

_____. Rome Air Development Center. Optimization of Retrieval Techniques and File Structures for the CIRCOL (Central Information Reference and Control On-Line) System. Report RADC-TR-73-420. February 1974.

- _____. Program Assisted Console Evaluation and Review (PACER) System Description. Vol. I. Report RADC-TR-72-89. April 1972.
- _____. Modifications to On-Line File Manipulation Techniques (MADAPS). Report RADC-TR-71-263. November 1971.
- _____. On-Line File Manipulation Techniques for the Management Data Processing System (MADAPS). Report RADC-TR-69-362. November 1969.
- _____. Study for the Development of Computer Augmented Management Techniques. Report RADC-TR-69-98. March 1969.
- _____. On-Line File Manipulation Techniques. Report RADC-TR-67-600. February 1968.
- _____. A User-Oriented On-Line Data System. Report RADC-TR-66-837. Vol. I. March 1967.
- _____. On-Line Intelligence Processing System. Vol. I. Report RADC-TR-66-2. July 1966.
- U.S. Army Combined Arms Center. Division Command Post System Test Support Package [Short Title: CP Study] December 1974.
- U.S. Army Computer Systems Command. Letter CSCS-TSP-C, Subject: Narrative History of DEVTOS. 23 March 1974.
- _____. TOS Fact Sheet--Revised. October 1971.
- _____. TOS Fact Sheet. 1970.
- U.S. Army Missile Command. Redstone Scientific Information Center. Analysis of Automated Control, Retrieval and Transmission System for STINFO. Report RSIC-978. March 1970.
- U.S. Army Research Institute for the Behavioral and Social Sciences. Development of an Informational Taxonomy of Visual Displays for Army Tactical Data Systems. Research Memorandum 74-4. February 1974.

Books

- Koehr, G., and others. Data Management Systems Catalog (MTP-139). Bedford, Mass.: Mitre Corporation, January 1973.

Krauss, Leonard I. Computer-Based Management Information Systems. New York: American Management Association, Inc., 1970.

Martin, James T. Design of Man-Computer Dialogues. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1973.

Meadow, Charles T. Man-Machine Communication. New York: John Wiley & Sons, Inc., 1970.

Walker, Donald E. (ed.). Interactive Bibliographic Search: The User/Computer Interface. Montvale, N. J.: AFIPS Press, 1971.

Reports

Baies, Stephen J. "User Behavior on an Interactive Computer System." Yorktown Heights, N. Y.: IBM Watson Research Center, August 1972.

Braddock, Dunn, and McDonald, Inc. BDM Information Management System (BIMS). Document No. BDM/W/74-076-BR. Vienna, Va., November 1974.

Cautin, H., and others. An Experimental On-Line Information Retrieval System. Philadelphia: University of Pennsylvania, Moore School of Electrical Engineering, May 1967.

Codd, E. F. "Seven Steps to Rendezvous with the Casual User." Proceedings of the IFIP [International Federation of Information Processing] Working Conference on Data Base Management. New York: North-Holland Publishing Company, 1974.

Computer Corporation of America. User Language Reference Manual, Model 204 Database Management Software System. Cambridge, Mass., September 1974.

Control Data Corporation. Reference Manual, Version 2-1, Multi-Access Retrieval System (MARS VI) for Partial Inversion. Sunnyvale, Calif., 1972.

Evans, David A. Environ: An On-Line Doodlepad for Planners. Final Report, SRI Project 7344. Menlo Park, Calif.: Stanford Research Institute, May 1969.

Fry, J., and J. Gosden. Survey of Management Information Systems and Their Languages. Report No. MTP-313. Bedford, Mass.: Mitre Corporation, May 1968.

GUIDE International Corporation. Requirements for a User Language.
Chicago, June 1973.

IBM [International Business Machines] Corporation. Information Management System Virtual Storage (IMS/VS) General Information Manual.
Document No. GH20-1260-0. Palo Alto, Calif., January 1973.

_____. ALERT I-II (Automated Law Enforcement Response Team): Kansas City, Missouri, Police Department. IBM Application Brief, Document GK20-0332-1. White Plains, N. Y., August 1972.

Informatics, Inc. MARK IV Systems (Technical System Description).
Document No. SM-7406-T76A. Canoga Park, Calif., 1974.

Kellogg, Charles H. Data Management in Ordinary English: Examples.
TM-3319/000/00. Santa Monica, Calif.: System Development Corporation, May 1968.

_____. On-Line Translation of Natural Language Questions Into Artificial Language Queries. Report SP-2827/000/00. Santa Monica, Calif.: System Development Corporation, April 1967.

_____. An Approach to the On-Line Interrogation of Structured Files of Facts Using Natural Language. Santa Monica, Calif.: System Development Corporation, April 1966.

Microdata Corporation. REALITY Computer System Reference Manual.
Irvine, Calif., August 1974.

MRI Systems Corporation. System 2000 Reference Manual. Austin, Tex., July 1973.

Muller, K. G. Present Development and Future Trends in Data Management Systems. NATO [North Atlantic Treaty Organization] Unclassified Professional Paper STCPP-64. Hague, The Netherlands: SHAPE [Supreme Headquarters Allied Powers Europe] Technical Centre, April 1972.

North Atlantic Treaty Organization. Advisory Group for Aerospace Research and Development. Storage and Retrieval of Information: A User-Supplier Dialogue. Paris, June 1968.

Research Analysis Corporation. An On-Line Information System for Army Force Planners. McLean, Va., June 1969.

Rubinoff, Morris. Man-Machine Communication Through a Teletypewriter. Philadelphia: University of Pennsylvania, Moore School of Electrical Engineering, May 1973.

Sackman, H. Experimental Investigation of User Performance in Time-Shared Computing Systems: Retrospect, Prospect, and the Public Interest. Report No. SP-2846. Santa Monica, Calif.: System Development Corporation, May 1967.

Science Applications, Inc. Division Validation Standard SOP's [Standing Operating Procedures]. McLean, Va., 1974.

_____. G-2 Workshop Planning Guide. McLean, Va., 1974.

_____. G-3 Pre-Workshop Planning Guide. McLean, Va., 1974.

Stefferd, Einar A. New Software for Data Management. Report No. SP-3221/000/01. Santa Monica, Calif.: System Development Corporation, September 1968.

System Development Corporation. DC/SR Segment Users Manual. Vol. 3: File Utilization for the Tactical Information Processing and Interpretation System (WS-428A). Hampton, Va., August 1974.

_____. Semi-Annual Report to the Director, Advanced Research Projects Agency, for the Period 1 Jan 1967 to 30 June 1967. Technical Memorandum 687/008/00. Santa Monica, Calif., June 1967.

Texas Water Development Board. Texas Water Oriented Data Bank: General Information On and Introduction To the Data Bank Monitor. Austin, Tex., February 1973.

Articles and Periodicals

Bockelman, Melvin F. "Police and Courts Get Law Enforcement Data Fast With On-Line Net." Data Communications User.

Burch, John G., and Felix R. Strater. "Tailoring the Information System." Journal of Systems Management, February 1973, pp. 34-38.

Canning, Richard G. "Problem Areas in Data Management." EDP Analyzer, March 1974, entire issue (13 pp.).

_____. "Trends in Data Management, Part II." EDP Analyzer, June 1971, entire issue (12 pp.).

_____. "Trends in Data Management, Part I." EDP Analyzer, May 1971, entire issue (14 pp.).

_____. "Progress in Information Retrieval." EDP Analyzer, January 1970, entire issue (14 pp.).

Fry, James P. "Managing Data Is the Key to MIS." Computer Decisions, January 1971, pp. 6-10.