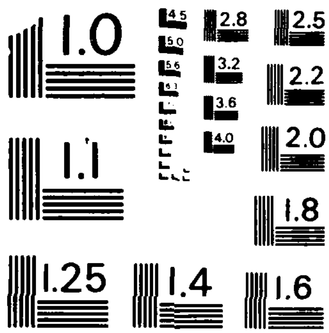
AD-A169 632 EXPERT SYSTEM DESIGN AID FOR APPLICATIONS OF HUMAN FACTORS IN ROBOTICS(U) PERSON-SYSTEM INTEGRATION INC ALEXANDRIA VA J MCGUINNESS ET AL. 12 JUN 86 1/3

UNCLASSIFIED PSI-K31-TR885 N60921-85-C-0252 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

PSI

EXPERT SYSTEM DESIGN AID FOR APPLICATIONS OF HUMAN FACTORS IN ROBOTICS

Contract Number N60921-85-C-0252

FOR PERIOD AUGUST 1985 - APRIL 1986

JUNE 1986

FINAL TECHNICAL REPORT

JAMES McGUINNESS, Ph.D., JOSEPH WAGNER, JR.,
JOHN M. NICHOLAS and CHRISTOPHER J. RHOADS

PROGRAM DESIGN SPECIFICATION

JAN E. RHOADS and JAMES McGUINNESS, Ph.D.

Prepared for

WHITE OAK LABORATORY
NAVAL SURFACE WEAPONS CENTER
ROBOTICS RESEARCH AND DEVELOPMENT LABORATORY
ATTN: SHARON HOGGE
10901 NEW HAMPSHIRE AVENUE
SILVER SPRING, MD 20903-5000

PERSON-SYSTEM INTEGRATION

Human Factors - Systems Analysis

2401 HUNTINGTON AVENUE ALEXANDRIA, VIRGINIA 22303-1531

(703) 960-5555

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Unlimited Distribution
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) PSI-K31-TR885	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Person-System Integration / 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION White Oak Laboratory, Code R402 Naval Surface Weapons Center
6c. ADDRESS (City, State, and ZIP Code) 2401 Huntington Avenue Alexandria, Virginia 22303-1531	7b. ADDRESS (City, State, and ZIP Code) 10901 New Hampshire Avenue Silver Spring, MD 20903-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION / 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Human Factors Expert System Design Aid For Robotics

12. PERSONAL AUTHOR(S)
James McGuinness, Ph.D., Joseph Wagner, Jr., John M. Nicholas, and J. Christopher Rhoads

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final Technical	FROM 850901 TO 860401	86, 06, 12	237

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Expert Systems; Robotics; Human Factors; Design Aid; Guidebook

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Research was performed to provide the groundwork for an Expert System that will function as a "design aid" for robotics applications. The Expert System is entitled Human Factors-Robotics Expert System (HF-ROBOTEX).

This effort involved a number of interrelated steps including an examination of state-of-the-art technology in the fields of Expert Systems, Human Factors, and Robotics by reviewing current literature and talking to pertinent experts. In addition, a design specification was written that incorporates the knowledge gained to guide the development of the Expert System. (over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ■ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

★U.S. Government Printing Office: 1985—639-012

0102-LF-014-6602

UNCLASSIFIED

19.

HF-ROBOTEX specification employs a modular design using a microcomputer-based system technology. It consists of three elements: a user interface, a knowledge base, and an inference driver. The knowledge base is designed to incorporate the knowledge of experts and selected sections of current Human Factors Guidebooks/Handbooks. The selection of data sources was guided by a literature review, by inputs from Human Factors Engineers, as well as by professionals involved in the application of robotics and Expert Systems. The inference driver uses rules of reasoning (i.e., heuristics) to access, as well as interpret information in the knowledge base and generate conclusions.

A-1

CONTENTS

ILLUSTRATIONS

1.0 INTRODUCTION

1.1 STATEMENT OF THE PROBLEM

The purpose of applying robotics is to automate functions which are, in a broad sense, dangerous, boring, or inefficient when performed by humans. The problem with this, however, is that in most robotics applications, human involvement is still necessary to different degrees and in a variety of roles such as a systems manager or a technician (e.g., operator or maintainer). Therefore, it is extremely crucial to define and then design robotic systems for human involvement so that implementation is safe, efficient, and effective. For example, a robotics-based manufacturing system may work around the clock without human labor, but human monitoring must ensure the automated process is initiated, continues, and meets the required quality levels or criteria.

The Department of Defense (DoD) has implemented robots and related technology to augment force effectiveness. However, the problem remains that regardless of the degree of automation, many of DoD's proposed military robotic applications involve *complex human/robot interactions*.

Thus, the success of many robotic applications in both military and commercial environments depend to a large extent on the successful integration of human factors. The chances for successful applications are greatly enhanced when human factors are considered in the early stages of system design. Ideally, this would be well before system integration and, if feasible, during the conceptual analysis stage of a project.

Putting together a human factors data base for robotics applications is not the problem. The knowledge base supporting human factors is fairly well established. There is a large amount of data available. A real problem in a high tech application such as robotics is the need for a fast, efficient, and cost-effective means to aid designers who must define problems from a "human point of view" and then "human-engineer" a robotics application. An Expert System is a part of the solution to this problem and answers such a need.

1.2 APPROACH TO THE PROBLEM

Hayes and Wheelwright (1984) postulate that only companies and countries that value and pursue manufacturing excellence will continue to thrive in the years to come. They reference Germany and Japan as two countries that value and pursue excellence and are building towards that goal. They also state that in order to rise to the challenge of international competition, American firms must rebuild their manufacturing organizations, focusing on four critical activities: developing appropriate production facilities and managing their evolution; choosing equipment and management systems appropriate to those facilities; establishing supplier relationships to provide them with parts and services; and encouraging continual improvement in their performance.

The business sector and the general public are beginning to see an influx of articles related to automation and productivity, both of which will be major influences in America in the near future. An essential theme is that computers could turn U.S. factories into world class competitors, but preparation must begin soon. To begin, the United States must exploit and integrate the latest advanced technologies including Computer Aided Design (CAD) and Computer Assisted Manufacturing (CAM); Computer Integrated Manufacturing (CIM); and Flexible Manufacturing System (FMS) applications.

Any integration of automation technology must seriously consider the interaction of Robotics and Human Factors since this interaction will have profound effects on the growth of manufacturing in the United States. Proper integration will be directly reflected in financial aspects since manufacturing automation markets are expected to climb from $25 billion in 1985 to $100 billion in 1995 (Kerr, 1985). Improper integration will indirectly continue the nation's loss of technology leadership which has occurred in many high-technology sectors over the last two decades. This is especially true in industrial hardware areas such as machine tooling and in electronic consumer products.

The United States has retained a lead in software research, development, and application. An area in which we should strive to achieve a clear lead is in "systems integration." This requires a new emphasis on truly tying man, machine, and environment together as a well-designed "whole."

Integrating humans and machines often involves very complex approaches. For example, many Research and Development (R&D) efforts in the United States have focused on the potential application of very complex technology (e.g., Artificial Intelligence) to Robotics. These efforts have often been directed toward developing very complex Robots which mimic human behavior-- a laudable R&D goal. But, to improve productivity, many other considerations must be brought into integrated manufacturing. For example, designing or redesigning an object for simple Robot assembly may be much more cost-effective than building a complex robot that mimics human behavior.

A review of the literature on the subject reveals that only limited success has been realized in integrating or applying the field of Human Factors in Robotics. In fact, some articles which use the title Human Factors, in many cases emphasize human resource utilization and social implications. For example, Chapter Four in <u>Industrial Robots</u> (SME, 1983) is titled Human Factors, but the four review articles in the chapter discuss management as well as worker views on resistance to robots in the workplace. Published in the same year, a book on assembly automation contains a section on The Human Factor which focuses on the fact that insufficient training of workers in a robot work environment will result in decreased productivity (Riley, 1983). A large number of books and articles have also been published on the subject of the human resource implications of Robotics (see Martensson, 1985 and Hunt, 1983). These references examine the unemployment, social, and economic impacts of Robots in the United States. While these references assess many critical human issues, they are largely of a social nature, and hence inappropriate in the context of this paper.

An issue we as a nation must address soon is <u>how</u> to best integrate Robots into commercial and military applications. Human Factors technology can help address this issue. To answer this question and meet the challenge of integrating a high technology area such as Robotics with the complex human factors which must support it, technology of another sort will provide a significant input. The technology which can meet the challenge is a rapidly advancing branch of Artificial Intelligence termed <u>Expert Systems.</u> Expert System technology can assist in designing more effective systems which use Robotics technology. This will result in well-integrated, safer, more efficient, more effective, and more profitable applications.

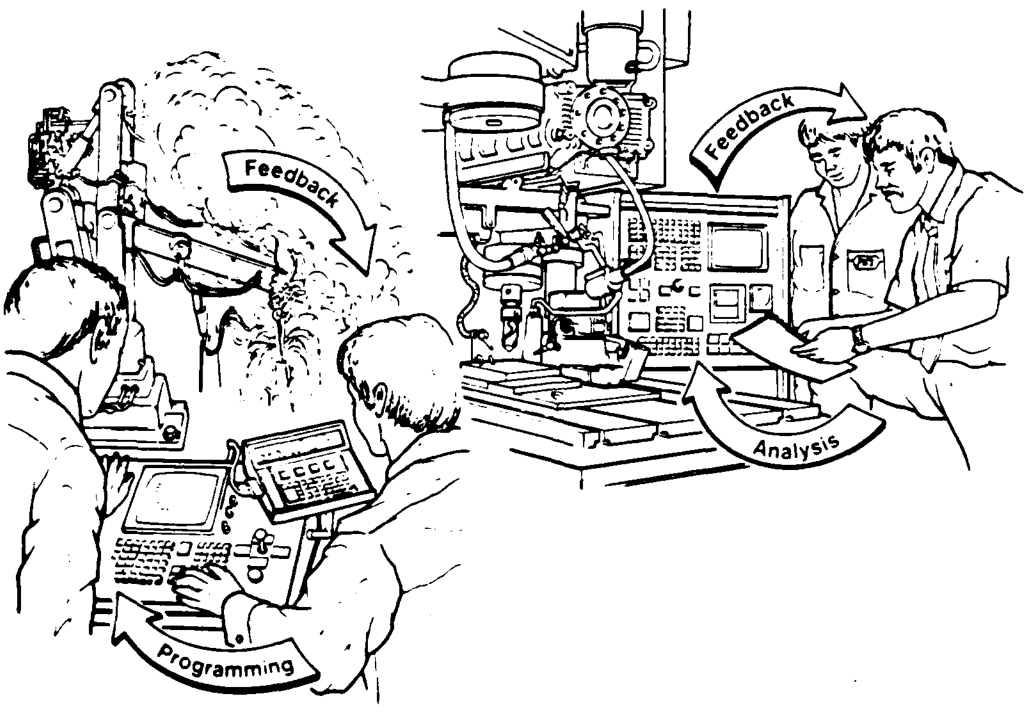
This technical report describes the analysis and design process conducted to lay groundwork for an Expert System which could aid in the application of Human Factors data and techniques within Robotics installations. PSI conducted the technical effort for the Naval Surface Weapons Center (NSWC) under Phase 1 of a Small Business Innovation Research (SBIR) contract. This technical report will cite another document produced by PSI during this Phase I contract effort, the Program Design Specification (PDS), that presents specific details of the software required to develop and apply the proposed Expert System. The PDS follows this report.

The ultimate product of this SBIR project will be a stand-alone, computer-based Expert System which a designer can use for assistance in designing and finding a "best fit" solution to interactions between humans and robots in an automated system.

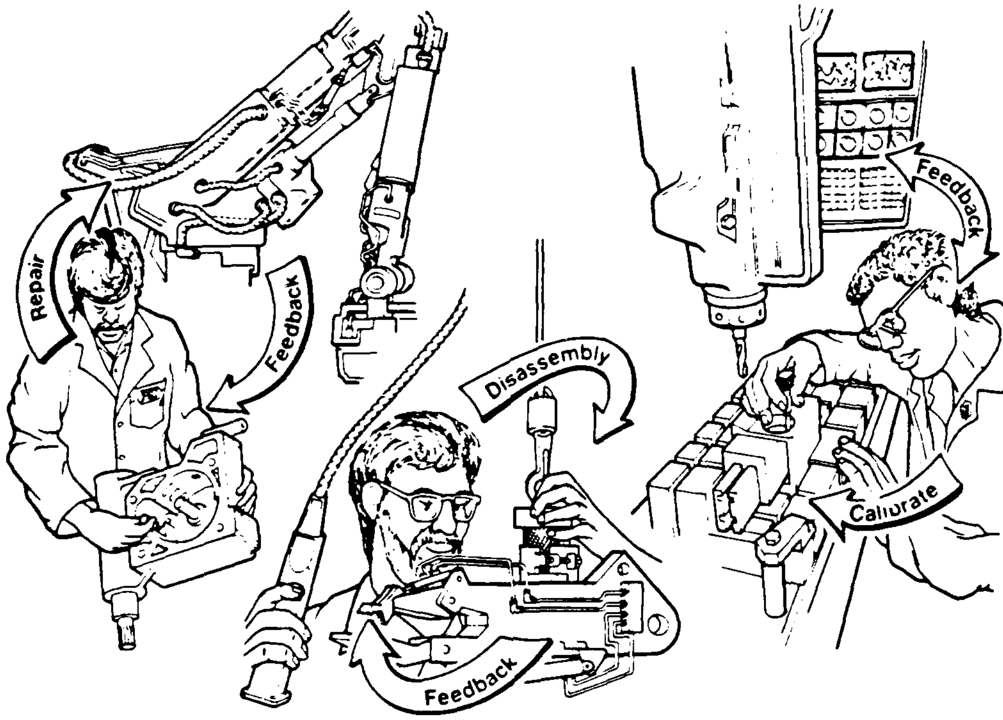
1.3 APPLICATION OF HUMAN FACTORS ENGINEERING TO ROBOTIC DESIGN TASKS

Human Factors Engineering must be considered in the design stage of robotic and automated systems to ensure effective and efficient operation and maintenance, to increase safety, and to decrease personnel training time/costs. This report describes the research performed to determine the feasibility of designing an Expert System to permit application of Human Factors principles, data, and techniques to the following:

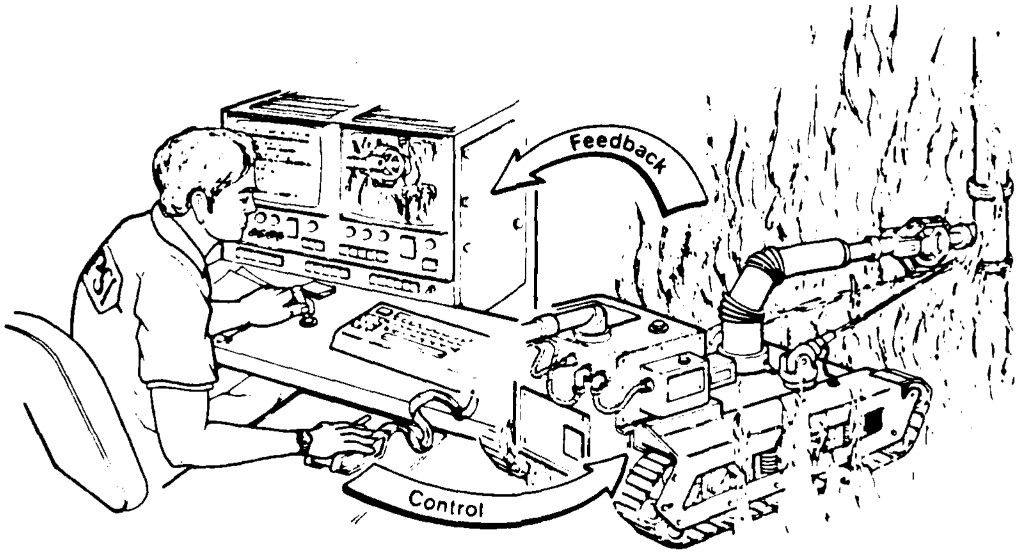
- o Direct Operations (examples: controlling a robot's movement or programming a robot for a tasking change)



o Direct Maintenance (example: hands-on maintenance, test, or adjustment of a robot)



o Remote Monitoring of Robotic Operations/Maintenance (example: access design of crucial information displayed on an operator's console)



1.4 THE OBJECTIVES OF THIS PROJECT

The first objective of this project was to review, assess, and then integrate pertinent features of the state-of-the-art in the following three technology areas:

- o Expert Systems
- o Human Factors
- o Robotics

It was believed that a proficient review and assessment would point to the most effective and efficient method to apply Human Factors in Robotics. The method selected was the use of an Expert System.

The second objective of the project was the design of an Expert System which could accurately and logically aid in the decision to design, extract, and combine the myriad of Human Factors elements to improve a robotics system.

The Expert System in this project will be designated as HF-ROBOTEX (Human Factors-Robotics Expert System).

HF-ROBOTEX is designed to assist in the application of Human Factors principles, data, and techniques to robotics systems. The goal of HF-ROBOTEX is a system that can be used by any Human Factors Engineer with limited experience in Robotics and/or any robotics-oriented engineer without extensive experience in Human Factors. This goal will be accomplished through the use of extensive expertise contained in the Knowledge Base or database of the Expert System and by means of an efficient search/access mechanism.

A note of caution is necessary and appropriate at this point. The Expert System is intended as a TOOL. It must be used by a craftsman in most cases to avoid misapplication. Used correctly by a competent, trained specialist, it will produce effective and safe system designs for Robotic applications, quickly and at low cost. Correct use will provide a critical communication link among Human Factors and other design specialities leading to more widespread utilization. The use of an Expert System will furnish a capability to quickly accomplish trade-offs during design stages. Such timing is often critical if Human Factors is to be influental in a final system design.

How then can an Expert System to apply Human Factors to Robotics be best implemented?

Figure 1 depicts the flow of activity for a hypothetical Robotics design cycle and the point at which the Expert System should be inserted in the loop to yield maximum benefit. A thorough analysis process is initiated focused on customer specifications (ALL design team members should participate in this process). The Robotics engineer/designer then starts by drafting the robot functional definition, which is next translated physically into a robot breadboard design. At this point, the functional definition and breadboard design should be reviewed by an Human Factors Engineer who will apply the Expert System to both design products. The Expert System will generate specific Human Factors guidelines. The guidelines are contrained in the Knowledge Base of HF-ROBOTEX. The Knowledge Base is delineated on three levels as shown in Figure 2.

Those guidelines that have not been fully accommodated by the current configuration are passed back as feedback to the engineer for his consideration in the design cycle, which must be repeated again for the incorporation of the pertinent Human Factors principles. Those guidelines that are finally approved with the help of the Expert System are passed on to be integrated within the robot system.

Delivery to the customer and a test and evaluation to ensure that the specification has been met, is of course, the ultimate step. This Expert System development is focused on the impact on the design and development process. Thus the Expert System stresses preventive measures although it could conceivably be adapted for a retrofit procedure.

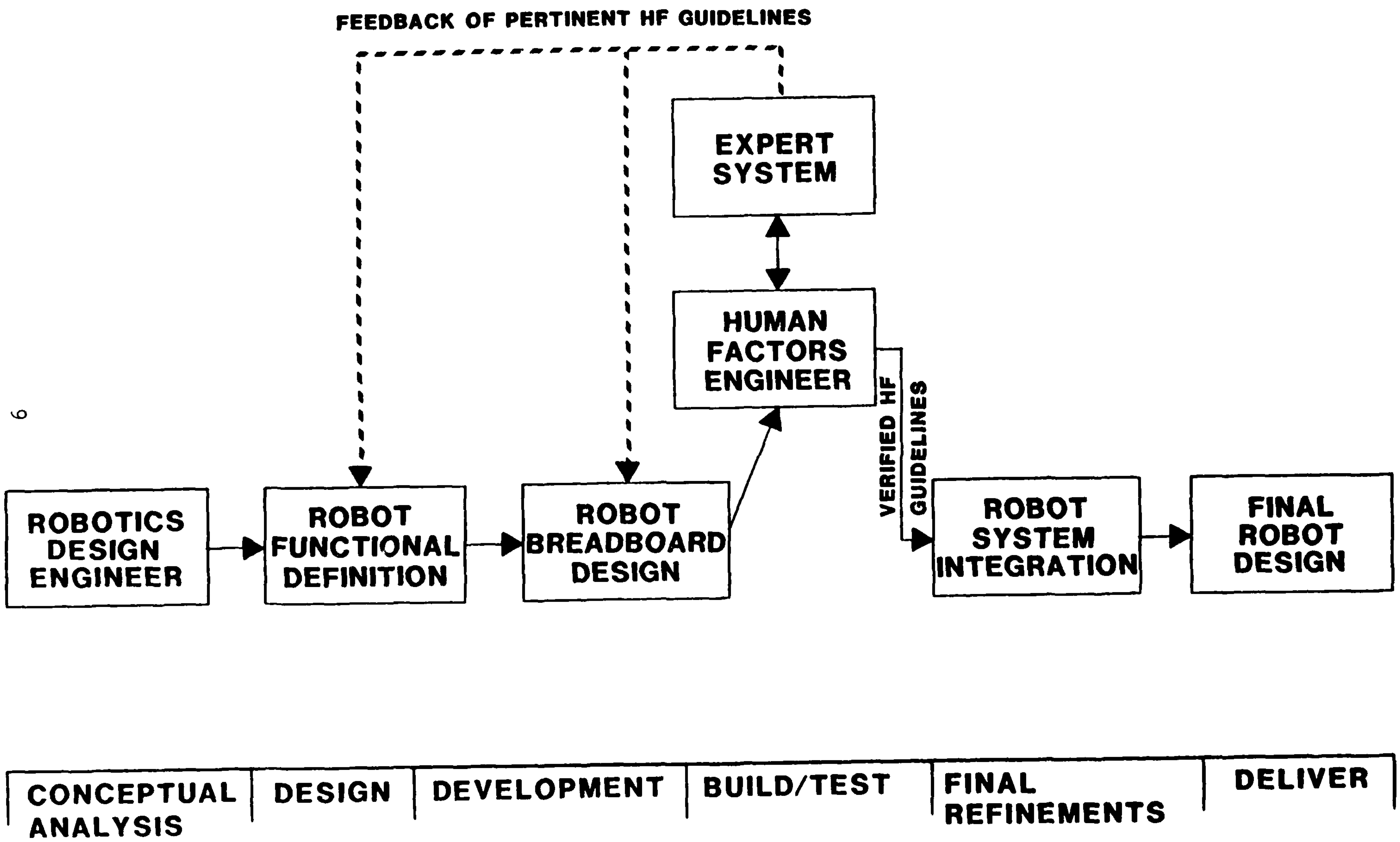


FIGURE 1. HYPOTHETICAL STAGES OF A ROBOTICS APPLICATION

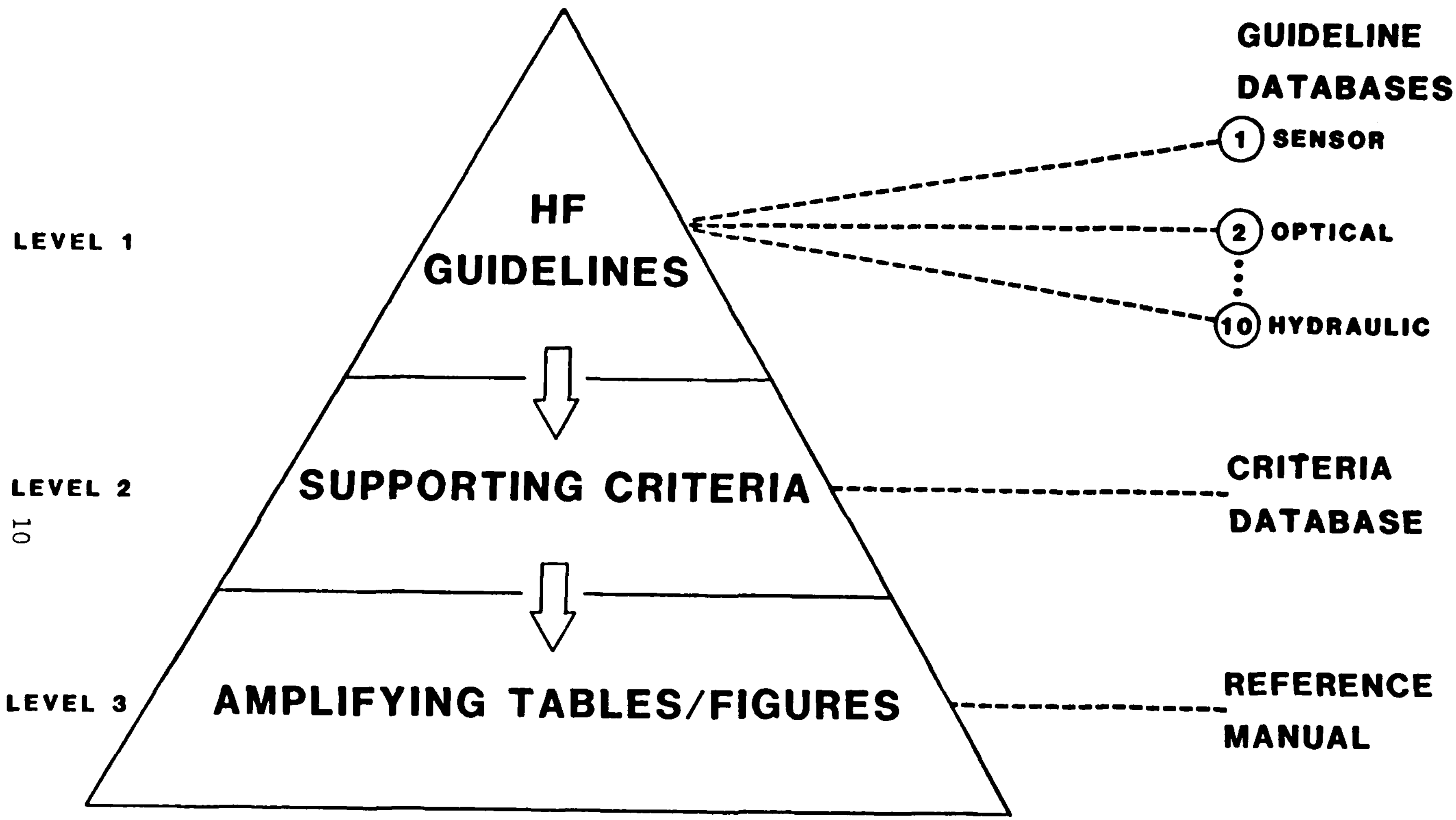


FIGURE 2. KNOWLEDGE BASE OVERVIEW

To properly analyze and select Human Factors guidelines, data, and criteria requires systematic and thorough front-end assessment of the various levels of a system. Figure 3 depicts an overview of such a system flow. There are three levels shown. Equipment and tasks must first be categorized on a general level. Assessment at the next system level (2) requires the identification of individual equipment components as well as the Human Factors associated with them. As shown in Figure 3, this comprises the "IF" portion of an "IF...THEN" algorithm.

Thus during the "IF" portion, two system levels are involved:

SYSTEM LEVEL 1 — To define a particular "object", HF-ROBOTEX employs a first set of rules at system level 1 to determine what areas of interest (equipment) and what segments of activity (tasks) the Robotex Design (RD) engineer is dealing with.

SYSTEM LEVEL 2 — To narrow the search down to only those HF guidelines which are pertinent, HF-ROBOTEX then employs a second set of rules at system level 2 to determine the object's "attributes" in terms of what equipment elements (components) are necessarily involved and what human considerations (factors) must be dealt with.

The "THEN" portion of the algorithm comprises level 3 as can be seen in Figure 3:

SYSTEM LEVEL 3 — Once these attributes are defined, HF-ROBOTEX then employs a third set of rules at system level 3 to retrieve the most pertinent "values" or HF guidelines which are values stored as frames in a knowledge base (KB).

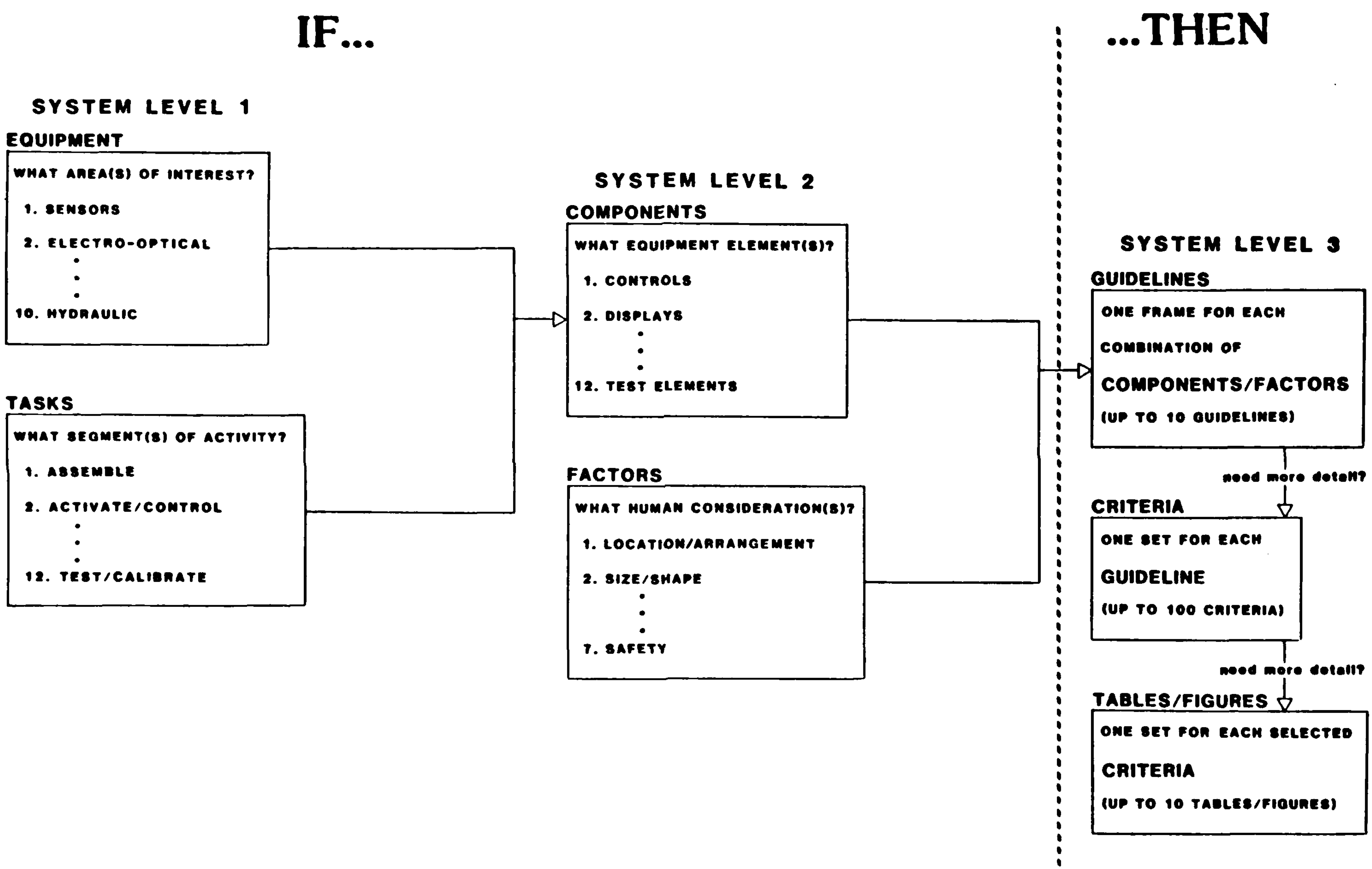


FIGURE 3. OVERVIEW OF DATA FLOW

If more detail is required, HF-ROBOTEX can further retrieve the supporting criteria for each guideline (also stored as frames). If still more detail is required, HF-ROBOTEX can, in turn, further identify specific tables/figures that amplify each criteria.

HF-ROBOTEX therefore relies on linking "IF" statements resulting from the user query process to "THEN" answers in the form of guidelines. A rule is a premise leading to a conclusion, which is commonly referred to as an IF...THEN statement. For example, IF... the RD object is "activating sensors" and the RD descriptors are "display" and "safety", ...THEN a pertinent HF guideline or conclusion would be " preferred visual areas for crucial information displayed on a panel would center around an operator's normal line of sight - approximately 10 degrees down from horizontal ").

Any given RD proposition (the " IF..." clause) may have many pertinent HF guidelines. This is also true for it's response rule, (the "...THEN" clause). Conversely, any given HF guideline may also have a great number of antecedent RD propositions. The underlying concept of HF-ROBOTEX relies heavily on both of these juxtaposed "one-into-many" logical propositions.

To exercise the IF...THEN algorithm requires a well-designed means of allowing a user to access or interface with the questioning process and with the data required. An Inference Process is activated by the user to ensure that the IF portion is thoroughly covered. An "Inference Engine" is at the heart of the Inference Process and it responds to user responses with rules or questions to define the IF conditions in greater and greater detail.

The "THEN" portion is the actual application of Human Factors data through the use of guidelines, criteria, and Tables or Figures which contain a great deal of data in graphic format. As demonstrated in Figure 4, guidelines are first employed at the most general level with increasing levels of detail being sought and accessed as needed. The guidelines are contained in the Knowledge Base in the form of individual databases with supporting criteria. Figure 4 demonstrates the structure necessary for such a frame-based Knowledge Base. In an exercise of the "...THEN" portion an operational or maintenance guideline matrix is accessed to arrive at a pertinent database structure. The Expert System will then compare and assess a matrix of equipment components and associated Human Factors to arrive at a frame which contains a number of guidelines. Pertinent guidelines are again assessed by the Expert System based upon the relevance of those contained in the record storage structure.

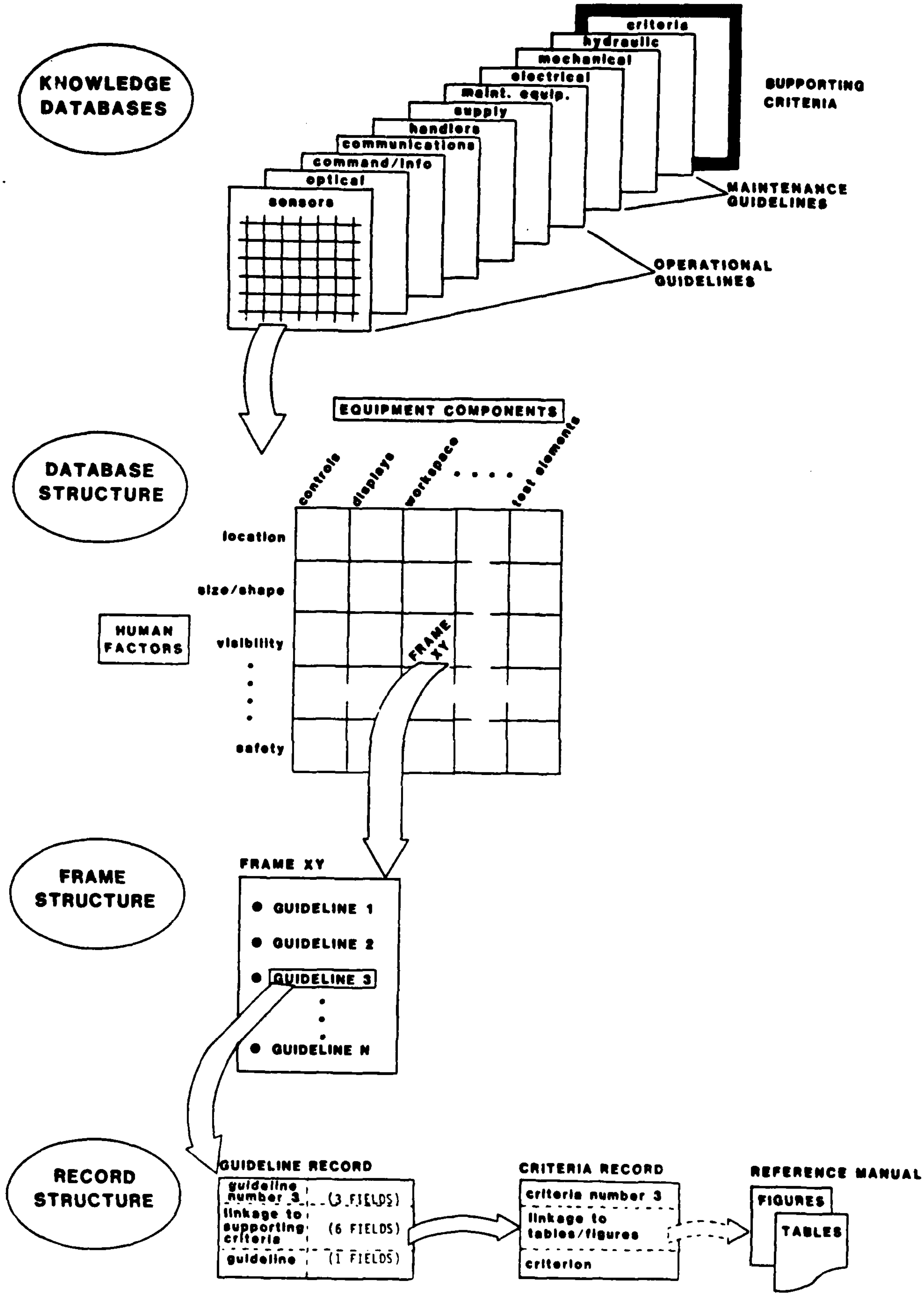


FIGURE 4. FRAME-BASED KNOWLEDGE BASE

The Expert System to be developed which will implement the flow shown in Figure 3 will have the following features and capabilities:

(1) An interactive, user-friendly interface constructed using state-of-the-art display techniques;

(2) A rule-based Inference Engine within a modularly-designed shell that will permit high speed, large capacity architecture compatible with IBM PC hardware environments; and

(3) An established knowledge base with sufficient levels of detailed data to produce guidelines in the form of design suggestions along with supporting criteria.

Further details as to how the proposed Expert System functions to derive rules and assessments based on user inputs can be found in the Program Design Specification concurrently produced under this SBIR contract.

1.5 PROJECT TASKS

Specifically, the following steps have been taken during the first phase of this SBIR program:

1. Conducted a Literature Review

2. Developed a Survey

3. Surveyed Human Factors Professionals

4. Surveyed Robotics Professionals

5. Identified Pertinent Guidebooks/Handbooks

6. Designed Algorithms to Combine Techniques that will Most Effectively Match Human Factors Data Elements to Robotics/System Elements

7. Conducted a Requirements Determination to Define the Extent of the Data Base and other Factors such as Processing Speed

8. Conducted Trade-off Analyses among Available, Pertinent Hardware/Software

9. Produced a Design Specification

The timetable for the nine (9) steps taken to design an Expert System is displayed graphically in Figure 5. A multidisciplinary team was formed by PSI to accomplish the steps necessary to design HF-ROBOTEX. The team included Engineering Personnel, Human Factors Professionals, and Systems Analysts/Programmers. This same type of interdisciplinary team structure will be required in future activities to fulfill the design specification.

PSI compiled data/information for the knowledge base of the Expert System by examining resources in two categories:

1. Literature review - from classic works through Human Factors guidebooks to current periodicals and books; and

2. Discussions with Human Factors/Robotics Professionals - individuals active in these disciplines who, because of their knowledge, experience, and training are classified as experts.

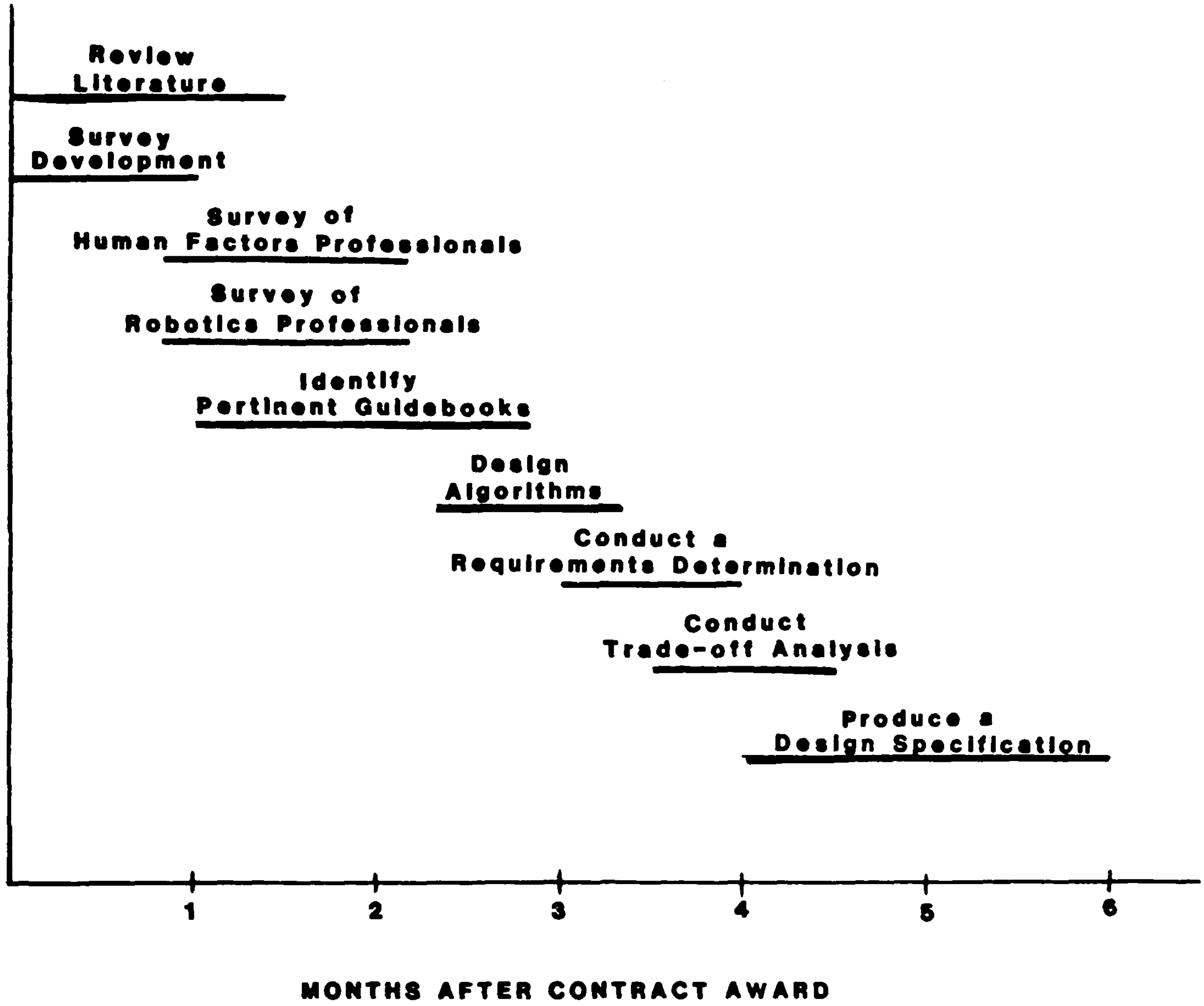


FIGURE 5. SEQUENCE OF PROPOSED TASKS

The results of the literature reviews and survey efforts are detailed in the following sections which cover the three specific technologies (Expert Systems, Human Factors, and Robotics) that were the focus of this SBIR effort. Appendix A contains a description of some of the pertinent meetings attended and individuals contacted.

PSI personnel attended ROBOTS 10 held in Chicago, Illinois (April 21 through April 24, 1986). In addition to the opportunity to view the latest state-of-the-art in robotics technology and to observe presentations by, and interact with, professionals in the field, a concentrated effort was put into obtaining the latest publications available. Significant publications obtained included a newly published (1986) catalogue from IFS (publications) LTD, 35-39 High Street, Kempston, Bedford MK42 7BT, England. This catalogue is referenced because it contains a wide variety of international works directly related to Robotics, Artificial Intelligence, and Human Factors. It also lists forthcoming international conferences including the 3rd International Conference in Human Factors in Manufacturing (HUMAN 3) to be held 4-6 November, 1986 in the United Kingdom.

Other significant documents obtained and reviewed by PSI personnel during ROBOTS 10 include the latest versions of a bibliography of Robotic Technical Papers and a compilation of Text and *Periodicals produced* by Robotics International/Society of Manufacturing Engineers (RI/SME). Personnel also received two bibliographies produced by RI/SME's Computerized Automation and Robotics Information Center (CARIC) with search terms of Expert Systems, Safety, Robotics, and Human Factors.

In summary, the latest trip to ROBOTS 10 convinced project personnel that the technical documents used to construct the technical report and specification are current and sufficient to ensure that the approach taken in this SBIR effort is valid and efficient.

Dr. McGuinness and Mr. Wagner also attended a technical meeting of the Human Factors Division of RI/SME. The outcome of this meeting is as follows. Mr. Wagner has been assigned the responsibility and authority to develop a resource guidebook to provide an up-to-date reference source for professionals in the area of Human Factors in robotics. The meeting also resulted in finalization of the responsibility and authority of Dr. McGuinness as the chairperson for a one-day symposium covering Human Factors in robotics. This will be conducted as an integral part of AUTOFACT to be held in Detroit, Michigan November 11 through November 13, 1986.

2.0 REVIEW OF SPECIFIC TECHNOLOGY

2.1 EXPERT SYSTEMS OVERVIEW

Expert Systems are computer programs designed to assist or perform like human experts in a field of expertise. In general, an Expert System must accomplish three specific goals: (1) Communication, (2) Subject Mastering, and (3) Problem Solving.

COMMUNICATION: The computer is programmed to effectively communicate with the user, a job which includes interpreting the user's information and queries and responding by posing and answering questions.

SUBJECT MASTERING: An information (or Knowledge) base is constructed by tapping experts in the field and/or by incorporating valid data and techniques.

PROBLEM SOLVING: A software program is developed on the basis of decision-tree-like logic. It is termed an Inference Engine and it accesses information in the knowledge base. It is typically constructed using an IF...THEN format to link causes with effect, such as:

> IF (your car won't start and there is adequate fuel and electricity)...THEN (check to see if the solenoid or starter is faulty).

The parts of an ideal Expert System are known as the language processor (communicator), the knowledge base (subject mastery), and the inference engine (problem solver). These three areas of human consultation along with their associated computer models are illustrated in Table 1 below.

Table 1: HUMAN ELEMENT vs MACHINE ELEMENT

HUMAN CONSULTANT		EXPERT SYSTEM COMPONENT
COMMUNICATION	-->	LANGUAGE PROCESSOR
SUBJECT MASTERY	-->	KNOWLEDGE BASE
PROBLEM SOLVING	-->	INFERENCE ENGINE

2.1.1 Expert Systems - Structure

2.1.1.1 The Language Processor. The language processor's job is to mediate information exchanges between the system and the user. Incoming questions, commands, and volunteered information are passed and interpreted by the language processor. From the system explanations, justifications of actions, data requests, and other responses are formatted for the operator to digest.

Expert Systems must have an efficient interface communicating with the user in a standardized, common language such as English. The computer must be easily able to interpret dialog from the user and formulate an intelligent response. This need for English fluency has spurred the development of commercially available natural language structures such as Q&A from Symantec and Paradox from Ansa. These two structures allow users to naturally interface with databases.

2.1.1.2 The Knowledge Base. The knowledge base is where the vital information gathered from expert sources is stored. The knowledge is organized as a web of information linked together by associations within the knowledge base. These associative links are known as inference rules. As revealed in the previous example, it was established that the automobile did not start, but neither lack of fuel nor battery power was the source of the problem. With the given information, an expert mechanic would examine the next components in the ignition system; the solenoid and the starter. With the same information, the Expert System searches its data banks for the appropriate corresponding rule. Once this rule is located, the program is instructed as to what the next step in solving the problem is. These rules then feed to the inference engine.

When considering building an Expert System, the domain of the proposed system must be studied and clearly defined. Expert Systems are best suited for areas in which the problem can be clearly defined and the variables understood by both the developer and the "expert."

Acquiring expert knowledge can be a costly, time-consuming process unless it is well-planned and structured. An entire discipline is begining to evolve to "capture" expert knowledge. This discipline is termed Knowledge Engineering (KE). A typical KE approach involves at least the following steps:

1. Careful identification and definition of the problem.

2. Designing the approach to and relationship with selected "experts."

3. Choosing effective and efficient methods for acquiring valid and reliable information (e.g., taped interviews, pencil/paper questionnaire, and unobtrusive filming-observation of work tasks/environments).

4. Structured analysis of information collected (e.g., analysis in terms of job, duties, associated tasks and then task elements).

5. Review of the information for subtleties, discontinuities, and gaps that the expert may not have communicated.

6. Refinement of the information through demonstrations by, and feedback from, the expert(s).

7. Translation of obtained knowledge into knowledge base rule structure.

2.1.1.3 The Inference Engine. The inference engine is the decision-making center of the Expert System. This is where the process of human reasoning is simulated. Here, inputted data is organized and plans of action to search the knowledge base are established. In many cases these plans will carry a projected chance of success, linked to the certainty of data and associations within the data base, and scheduled for execution on a "do the most promising thing next" basis.

The inference engine searches the knowledge base looking for similarities between the user's and computer's information. When the IF part of a rule has been adequately matched, the rule is "fired" and the THEN part is used to further investigate the problem until a final solution is attained. In the previous example, the computer might ask:

> IF (Does bypassing the solenoid enable the car to start?)...THEN (Replace solenoid).

A yes response to the question would indicate that the car would start if the solenoid was not faulty and the rule would be fired, causing a response of "Replace solenoid" and solving the problem. A no response would have told the computer to search elsewhere for more clues to diagnose and troubleshoot the situation.

2.1.2 Specific Expert System Technology

The following review is intended to serve as a backdrop to explain the choice of Expert System structure which we have selected for HF-ROBOTEX. HF-ROBOTEX is an Expert System structure to use when the most effective INTEGRATION of a PERSON within a complex SYSTEM is desired. In this project the complex system focussed upon is robotics, which is reviewed in a following section.

This report contains a review of literature pertinent to Artificial Intelligence (AI) applications in Expert System with a focus on microcomputer developments.

Attempts to develop and apply AI started in the laboratory about 1960. Scientists began to build Expert Systems for applications in Chemistry, Electronics (Troubleshooting), as well as Medicine (Diagnosis) at a cost of millions of dollars per system. Development and use was limited to organizations with access to expensive mainframe computers. Opportunities to explore AI for small organizations were limited until the arrival of recent technological advances.

With the development of powerful and affordable microcomputers during the 1980's, a number of organizations have developed AI tools for use on microcomputers. Among these are popular AI language compilers to allow the use of languages specifically developed for AI such as LISP and PROLOG. Recently very versatile skeleton or framework Expert Systems have been developed. Such systems lack a database and thus are adaptable to each user's domain of application.

This phase of the project focussed on Expert System design rather than on literature analysis and correlation. A review of some of the developmental advances in Expert Systems can be found at Appendix B. A large amount of data sources were compiled, reviewed and assessed all of which can be seen in the bibliography at Appendix C. A number of excellent sources can be sought out when a more in-depth coverage of Expert Systems is desired. For example, a classic reference source has been generated by the work of Feigenbaum and McCorduck (1983). In Section B, the authors list most of the pertinent Expert Systems (both experimental and operational) along with their domain of coverage, a concise description of the system as well as the organization which developed each system.

An Introduction to Artificial Intelligence (Feigenbaum,1983) provides a comprehensive overview of early work in the area as well as over forty pages of bibliographic listings. Famous authors and articles have been compiled in a number of sources including two very comprehensive compendia edited by Minsky (1982) and Feigenbaum and Feldman (1963). A recent treatment of the field can be found in the text "A Guide to Expert Systems" (Waterman, 1986). It is a sweeping review of a wide variety of already developed Expert Systems and also contains guidance and cautions on "how to" build expert systems. The book contains an up-to-date bibliography and a catalog of Expert Systems tools which provides a more than adequate supplement to the sources cited in this report.

This "refer to another source" procedure is being used in the current project to avoid extensive bibliographic references which have already been compiled and cited by sources such as Waterman (1986) for specific Expert Systems references and Feigenbaum (1983) in addition to others for Artificial Intelligence technology.

PSI personnel have assessed a number of recently published texts and articles related to Expert Systems in combination with computer technology. An example is a very recent book dealing with Expert Systems and Microcomputers (Simons, 1986). Citations reflecting such a combination of technology are also included in Appendix C.

2.2 HUMAN FACTORS REVIEW

This section provides details of the review performed to identify Human Factors work that has been, or is currently being, performed pertinent to robotics. The objectives of the review were to select from the extensive Human Factors literature those relevant Human Factors data/techniques that would contribute substantially to the HF-ROBOTEX data base; and to identify the scope of available data for determination of the required memory and processing capabilities of HF-ROBOTEX.

2.2.1 Scope

The review revealed numerous sources of applicable, valid Human Factors Engineering data. It encompassed a wide variety of human factors applications and literature. Literature reviewed ranged from a survey of "Scientific Management" principles developed by Frederick Taylor (1911) who was reported as the first to apply the rules of engineering to human beings, to Frank Gilbreth (1911) who improved the technique of time and motion study. The review also included a survey of the Human Factors Engineering contributions to human-computer interface design including, among others, Richard Rubinstein's and Harry Hersh's (1984) discussion of the importance of "user-centered design" for computer systems. Recent and past professional journals such as Human Factors and Ergonomics also were assessed for data, content, and application examples.

Current Robotics-oriented magazines such as Robotics Engineering and Robotics Today, that cover the fast changing state-of-the-art in robotic applications were also reviewed for pertinent content. Human Factors Engineering handbooks/guidebooks -- such as the Human Engineering Guide to Equipment Design (VanCott and Kinkade, eds., 1972), the Air Force Systems Command Human Factors Engineering Series (1972), and the Human Factors Test & Evaluation Manual (HFTEMAN) (see Malone and Shenk (1976)) -- to name but a few, meet the goals of this contract because they contain a wealth of pertinent information, are based on Military Standard 1472, and are structured such that they can be readily programmed into a Knowledge (Data) Base.

2.2.2 Books, Current Periodicals, and Technical Reports

A range of recent literature has addressed the issue of the overall impact of robotics on society in general and on the workforce in particular. This issue is an important one to social commentators, such as Asimov (1950), Naisbitt (1982), and Toffler (1970). The issue of primary concern to human factors technology is how robotics impacts the quality of the work environment and more specifically, the effectiveness of the individual worker.

Many experts believe that robotic systems are highly complex and may push humans to the limit of their ability to perform efficiently if basic Human Factors principles are not adequately applied. System developers have often taken the view that if a hardware system can be made to run, somehow human beings with the proper characteristics will be found and "fitted into" the system (Gagne, 1962). Meister and Sullivan (1968) have found that the subject of human factors is often "engineering's blind spot." McDonald (1976) points out that while most system designers are familiar with the mechanical motions that can be produced by combinations of gears, lever arms, and other components of a system, they usually have only a superficial understanding of the motions performed by the human body.

Human Factors Engineering input during design stages can ensure that complex systems do not overburden the human operater/maintainer. Inadequate human factors information can lead to overestimation of operator capabilities, human error, production inefficiencies, and safety problems. By infusing human factors data and techniques early in the design process, many of these problems can be avoided.

Rogers and Armstrong (1977) found that some Human Factors Engineering standards receive very little consideration and consequently have very little impact on design. Numerous reasons for resistance to available standards are offered by the authors as well as recommendations to improve and facilitate use of human factors standards during the design process. Not only do inconsistent standards contribute to the lack of application, but, as Salvendy (1982) points out, there is only one ergonomist (Human Factors Engineer) for every 350 engineers in the United States.

An Expert System could alleviate these problems and facilitate the correct application of Human Factors Engineering data by system designers. The system would be a "communications and design aid" providing critical Human Factors inputs during the design, development, and application stages of robotic systems. However, the data and techniques must be presented in a format which ensures user acceptance and proper interpretation. Meister (1984) states that "unfortunately, all our experience suggests that without providing the engineer with very specific design guidance, he will usually ignore the standard, if only because he will see no feasible way of incorporating it into his design." An Expert System will bring a degree of "high technology" validity to Human Factors inputs. This will increase user acceptance, and this factor will be as critical as increased ease and speed of use to proper utilization. The growth provisions inherent to the Expert System will also increase user acceptance by providing the means to update the knowledge base as robotic technology advances thereby reinforcing system validity.

The first step to correctly apply Human Factors Engineering data and techniques in the design of systems is a determination of human interaction with the system.

The allocation of functions in systems has been of concern to Human Factors Engineers for many years. Allocating functions determined by the relative strengths, weaknesses, and other attributes between man and machine were first discussed by Paul Fitts (1951). He developed principles such as a man is flexible but not a consistent performer; whereas a machine is consistent but not flexible. Kamali, Moodie, and Salvendy (1982) extended Fitts early work by comparing the abilities and limitations of combined utilization of humans, automation, conveyers, and robots to enhance productivity while increasing work satisfaction and productivity for humans. The authors develop a framework to select the appropriate robot, machine, and conveyor configuration to complement the human in the workplace. Price (1985) describes the systems approach to design and how the allocation of functions is still an integral part of it. The article by Price provides a review and synthesis of "lessons-learned" over the last 30 years pertaining to the allocation of functions in systems. Determination of what people should be doing and what robots should be doing is an imperative first step when considering input of Human Factors Engineering data into systems design. Parsons and Kearsley (1983) state that until it is clear what the human will do, it is difficult to see what equipment interfaces with workers should be engineered, what human performance should be protected, whose environment should be controlled, for whom procedures should be optimized, which workers should be involved in test and evaluation, and who should be trained to acquire what skills.

Lyman and Madni (1984) assert that the principal function of robotics in industrial, medical, and battlefield applications is to replace, augment, aid, and improve human performance in sensory, manipulative, and cognitive functions. They define operator roles under three general categories: monitor, manager, and maintainer. Parsons and Kearsley (1982) describe the functions in which humans might or should participate in roboticized operations for the U.S. Army and summarize them with the acronym SIMBIOSIS, which stands for Surveillance, Intervention, Maintenance, Backup, Input, Output, Supervision, Inspection, and Synergy.

Decisions concerning the configuration of man-robot interaction determine the requirements for equipment design, workspace layout, system flow and interaction, and environmental control. These decisions also affect personnel requirements in the form of availability, manning levels, and training. Thus, the psychological and physiological aspects of the human component within a system should be defined as early in the design phase as possible to ensure adequate consideration of his/her capabilities and limitations. An early book which provides an overview of man-machine interactions and contains a good historical bibliography was done in 1970 (de Greene, 1970). Ranta, Wahlstrom, and Westesson (1981), in their book "Guidelines for Man-Machine Interface Design" state that practical design work involves top-down planning, which proceeds through several decision-making phases from general concepts concerning the man-machine interface system to the detailed design of the various parts of the system. They discuss factors such as basic aims and goals of the automated production process, system planning and instrumentaton, and the detailed design of automation. A chapter on detailed design includes checklists, man-machine models, and human cognitive process models.

However human tasks are categorized in a robotic system, a critical component for safe and efficient operation is the feedback of information on the operational status of the robot to the human. Johnsen and Corliss (1971) state that "once control tasks have been divided between operator and machine, there remains the communication problem, which means insuring that man can command the machine efficiently and that the machine can feed back information to man with ease." Displays provide information to the human operator about the machine and controls provide information to the machine from the operator. The controls and displays are critical to the smooth functioning of robotic systems. Since most robotic systems operate under computer control, the interaction of man and machine occurs most frequently at the Video Display Terminal (VDT) and associated keyboard or control panel.

Designing computer interfaces to match human cognitive processes is becoming increasingly important as computer systems become more pervasive and sophisticated. Maguire (1982) evaluated the literature on man-computer dialogues and concludes that guidelines based on a limited field of experience are frequently offered as general purpose advice. When this occurs, contradictory recommendations arise and the information is oftentimes disregarded even though valid for some applications. Wickens and Kramer (1985) state that while numerous guidelines have been compiled for designers of human-computer interfaces, many of them appear to be based on intuition and experience as opposed to validated guidelines. The authors suggest that laboratory and operational-based validation of human-computer interfaces has been sparse and will require substantial work over the next decade. Wickens and Kramer then review and describe some of the attempts at developing and validating human performance models. The authors endorse the development of a cognitively based performance theory of the human-computer interaction which enables the derivation and empirical validation of design principles.

Edmonds (1982) proposes three levels of human-computer interface which require human factors considerations. They are the hardware ergonomics, the software ergonomics, and the cognitive ergonomics. By understanding how human capabilities and limitations affect user interaction at all three levels, designers can construct systems that facilitate productivity. Shneiderman (1980), discusses the necessary infusion of psychological principles with computer systems. To improve programmer productivity, terminal user effectiveness, and system quality, Dr. Shneiderman describes current research techniques and offers guidelines for programming and system design. The book also addresses programming management and environment, stylistic standards, language design, programmer education, database query facilities, and interactive systems.

Rubinstein and Hersh (1984) attempt to synthesize the available Human Factors data on computer systems. They present 93 guidelines for system design which cover topics such as keyboard design, conceptual models, man-machine interface, language, and internal processing. The authors propose that incongruous and illogical computer responses to incorrect user inputs can be avoided if simple human factors principles are applied early and throughout system design and development.

Michaelis, Miller, and Hendler (1982) discuss the crucial need for developing a synergism between Artificial Intelligence and Human Factors Engineering. They describe a process undertaken at Texas Instruments to develop a computer-processable, human-engineered subset of natural language to aid in system interactions. Another book of compilations, entitled Human-Computer Interaction (edited by Salvendy, 1984), gives a number of expert views on the overall interaction of humans and computers as well as a specific article on "Some Fundamental Problems of Application of Industrial Robots in Production Line." The latter article, by five Japanese authors, cites case study applications and considers them from the ergonomics point of view. The Salvendy book contains two other chapters which are germane to this SBIR project. The first deals with an application of an Expert System to problem solving in process control displays. Studies sponsored by the Nuclear Regulatory Commission as part of a Human Factors research program in man-machine interface are described. Implications of the findings for the design and evaluation of similar computer-based expert systems are presented (Jenkins, 1984). The second chapter delineates "a framework for training human expertise." The chapter discusses the process of building expert systems and retrieving the appropriate problem-solving knowledge. A framework for knowledge elicitation, analysis, and testing is shown (Boose, 1984).

Numerous organizations, including the American National Standards Institute (ANSI), the National Bureau of Standards (NBS), American Society for Testing and Materials (ASTM), Robotic Industries Association (RIA), and International Standards Organization (ISO), are involved in the formulation of pertinent standards to ensure an orderly evolution of the robotics industry. Overall, coordinated standards development promotes human safety, helps integrate automated factory systems, and encourages reliable robot performance specifications. RIA implemented a standards effort at their Annual Meeting in Dallas, Texas on February 29, 1984 by establishing an executive committee and several subcommittees. Seven subcommittees were eventually established to develop robotic standards that cover Electrical Interface, Human Interface, Mechanical Interface, Communications/Information, Performance, Safety, and Terminology. The Safety subcommittee has made the greatest progress to date. They officially introduced a draft standard at a special seminar on Thursday, April 24, 1986 in conjunction with ROBOTS 10. It is expected that this draft will be recognized as an American National Standard by the American national Standards Intitute.

Recently, the Human Factors Society has established a committee to develop technical standards for acceptable Human Factors principles and practices in the design and use of display terminals, workstations, keyboards, and their environment. The standards will be developed under ANSI rules and procedures. The committee is presenting the draft for review and comment to selected segments of the professional community.

Since the introduction of robots into the workplace is steadily increasing (see DHR report (1984), and Hunt (1985)), it logically follows that a wider variety of individuals will interact in some way with robots and hence the integration of Human Factors into robotics will become even more critical. Indeed, the Human Factors Engineer must evaluate worker aptitude, skills, and knowledge to determine factors such as trainability for robot-technology-related jobs. Maguire (1982) states that as interaction with computers by non-specialists increased, so too did criticism of poor dialogue interface increase.

Hirsch (1984) states that "until about 1970, human factors work in IBM was mostly hardware oriented." Since then, emphasis has been placed on software and user documentation because of the wider variety of users who are less "computer-sophisticated." By addressing human factors issues early in the robot development cycle, we may avoid the many roadblocks to user acceptance experienced during the early years of the computer industry.

The Human Factors community has been concentrating on technology areas other than robotics as evidenced by the lack of substantive R&D and/or applied work until quite recently. While performing research for a presentation at the International Conference on Occupational Ergonomics, Parsons (1984) found that "in terms of visible events, Human Factors Engineering has been involved in robotics for no more than 5 years." The 1985 Annual Review of Psychology contains one article by Wickens and Kramer (1985) that provides an exhaustive review of Engineering Psychology. The authors address the topic of robotics (page 334) and reference two articles that provide "general overviews" on the state of human factors in robotics, seven articles which describe work of a more applied nature (including a NASA Annual Conference on Manual Control), and one article by Birk and Kelley (1981) that provides a summary of a conference workshop on human factors in robotics.

Although the infusion of human factors into military robotic systems is comparatively extensive, the only notable industrial application found is detailed in a recent Human Factors article by Shulman and Olex (1985). The authors describe how Human Factors Engineering was applied during the design of a second generation industrial spray-painting robot manufactured by the Nordson Corporation. The robot uses microprocessor technology to increase the number of operational functions over those capable of being performed by hard-wired robotic systems. The application of human factors was limited to three specific areas during the design process of this new system including the system control panel, the training arm grip design, and the software interface design. The report also describes the interactions that occurred among Human Factors Engineers and Nordson designers including a description of design tradeoff decisions made as a result of human factors input. The author's final conclusion is that the need for Human Factors Engineering grows in direct relationship to the complexity of user-machine systems.

A group at Nottingham University in England has been working since 1967 on the problems of computer aided workplace and work task design with emphasis on ergonomic and safety principles. Errors, caused by man or by machine, hinder the manufacturing process and can be reduced in a number of ways. Human Factors Engineering data/techniques can contribute valuable guidance for an error reduction program. Bonney and Williams (1977) describe a computer program for Controls And Panel Arrangements By Logical Evaluation (CAPABLE). The program assists design engineers with control panel layout decisions by offering Human Factors principles such as limb assignment and ease of operation and viewing considerations. Correct application of the program's results can directly enhance safety by allowing the engineer to make process control design decisions based on valid ergonomic principles.

Yong, Bonney, and Taylor (1982) discuss safety aspects of industrial robot systems and how the Graphical Robot Applications Simulation Package (GRASP) can help improve the design of some of the safety features within a robot installation. The GRASP system also was developed in the Department of Production Engineering and Production Management at Nottingham University. It utilizes a data structure similar to that of SAMMIE (see Bonney, 1980, and Bonney, Case, Hughes, Kennedy, and Williams, 1974) to model and simulate industrial robot systems. The GRASP system is used by an engineer to improve his overall system and workplace design through computer aided design (CAD) techniques. Specifically, it allows the user to position (and reposition as necessary) the major components of the robot installation so that component interactions are fully considered before decisions on overall layout are made.

From here, GRASP provides the engineer with data that allows a progressively more detailed analysis of safety features including examination of robot "operating zones" and "maximum reach envelopes," guarding requirements, models of how man would interact with the robot, and the identification of potential trapping points.

A system in use at Lockheed Missile and Space Company developed to solve conceptual design problems is an example of computerized anthropometrics and provides a glimpse of how computers will be used to assist in the design of human-robot configurations. It consists of computer generated outlines of a man and woman shown on CADCAM (Computer Aided Design and Computer Assisted Manufacturing) video screens. According to Lockheed Missile and Space Company senior Human Factors Engineer Richard Davids, ADAM is the first scaled version of a human to emerge from CADCAM. ADAM gets his name from Anthropometric Design-Aid Mannequin. EVE's acronym comes from Ergonomic Value Estimator (Manufacturing Ergonomics, 1985). The figures can be called up on the CADCAM screen in top, side, and frontal views. At the touch of a light pen, mouse, or graphic tablet, body and limbs on the screen will move in working postures - bending, kneeling, reaching. Closeups can be shown, for instance, to determine the wrist or arm freedom needed to tighten a bolt in a confined work space. ADAM is used to solve conceptual design problems such as technician access to equipment during operation or maintenance. ADAM does not interfere with the engineer's prerogatives, but provides a realistic basis to show access, reach, and working postures. Mr. Davids has stated that use of such an interactive design aid has directly resulted in savings of millions of dollars in the reduction of reworks for manufacturing equipment as well as indirectly in the prevention of back injuries by the redesign of heavy equipment placement and lifting procedures.

Parsons (1985) examined robot safety issues and suggests how Human Factors "...can help prevent accidents in which robots may damage workers, equipment, or the robots themselves." He suggests several preventive techniques (transponders, visibility, "safety plug system," height of fence, safety device checking, signs, and training), defines the issue of human error, and then discusses error reduction techniques. Errors, whether caused by machine or by human, can be reduced by the prudent application of human factors data and techniques. The "Watchdog" Safety Computer developed by the National Bureau of Standards monitors robot joint velocity, acceleration, and position. (Bloom and McLean, 1983) The computer is independently capable to stop the robot if it exceeds preset limits. Kilmer, McCain, Juberts and Legowik (1984) describe the "Watchdog" Safety

Computer system in more detail including its design and implementation. Parsons (1985) citing Kinsley (1984), describes the Roboguard system developed at the General Motors Corporation. This "safety system" consists of a dedicated computer and a multi-branched antenna on the robot arm to detect persons entering the robot's work envelope.

To summarize, the literature review performed under this SBIR contract has revealed two overall "trends" related to human factors efforts in robotics. The first of which is simply that not much applied work has been done. What work that has been done is sporadic and a carryover from military and government-sponsored projects. The second overall trend is that much of the literature suggests, and indeed many of the authors specifically suggest, that there IS a need for human factors technology in robotics.

The Human Factors community must focus attention on the field of robotics to promote the appropriate application of human factors data/techniques during the design of these complex systems. Perhaps the rejuvenation of the Human Factors Division of Robotics International (of the Society of Manufactuing Engineers) will provide a stimulus and a forum for human factors to play a more significant role in robotics. Until then, we applaud accomplishments such as those conducted by the Lockheed ADAM and EVE program, computerized aids such as those realized at Nottingham University, and the type of applied human factors analyses performed at and supported by the Nordson Corporation. The results of such completed and on-going analyses in many instances, can be directly applied to robotic systems and will thus be watched closely.

The next section of this report introduces and briefly explains the available guidebooks/handbooks which contain a wealth of pertinent knowledge related to human factors data, techniques, and overall methodology.

2.2.3 Guidebooks/Handbooks

Selected sections of both classic and current Human Factors guidebooks/handbooks are directly applicable to the design of the data base for HF ROBOTEX. Most are based on data contained in the Department of Defense's Military Standard 1472 and many provide a very suitable framework for cost-effective conversion or use in an Expert System because they are highly structured, developed in a programmed, text-type of format, and hence are very conducive to programming.

Sources of valid human factors data that can be applied to this project are numerous. A sample of available resources are listed in the bibliography of this report.

2.3 ROBOTICS REVIEW

This review section focuses on selected areas of the robotics field since there is a voluminous amount of literature in the area. The goals of this project forced a review emphasizing relevant, robotics handbooks of a broad-based nature, and a concentration on sources which discussed robotics as it affects and is effected by Human Factors and safety issues. A selected bibliography of publications related to robotics, but not referenced in this technical report is included in Appendix C.

As mentioned in the preface to this report, robotics integration in US manufacturing processes is growing in size and importance. In the recent past, most robotics tasks emphasized welding or paint spraying in high volume applications such as automobile assembly lines. Robot installations in U.S. Industry are predicted to increase from approximately 8,000 in 1985 to 22,000 in 1990. Future robot applications will probably be somewhat different in scope due to R&D efforts in sensors, as well as in tactile, force, and proximity end-effectors. The National Bureau of Standards (NBS) Automated Manufacturing Research Facility (AMRF) has sponsored these types of R&D as well as being the focal point of the refinements such as the establishment of communications interface protocols and the use of Expert System technology in process planning systems.

The Department of Defense (DoD) has proposed some very fundamental as well as some exotic applications for robotics. The Air Force has established and supported efforts to automate aircraft manufacturing aspects. The U.S. Army has established five-year plans for robotics applications and has already developed robotics based ammunition handling systems as well as more theoretical systems such as a battlefield-casualty-handling robot.

The U.S. Navy also recognizes the benefits to force effectiveness that can be derived from robotics. The Naval Sea Systems Command (NAVSEA) Integrated Robotics Program was initiated to capitalize on the potential of Intelligent Machine Automation and Robotics. As stated in their 1984 Annual Report (see Naval Sea Systems Command, 1984), the goal of this program is "to ensure that the Navy of the next century uses the robotics technology that will be available to improve the quality and performance of Navy ships and weapon systems; reduce acquisition, repair, and overhaul costs; and improve readiness and endurance, while freeing human assets for higher-order functions." Everett (1985) states that "most of NAVSEA's involvement to date in robotics has been directed at the use of industrial robots for specialized tasks associated with shipbuilding and weapons manufacturing." As robotics technology advances, so too will the feasibility of expanded applications. In fact, efforts must be made to roboticize certain tasks in the Navy as a result of decreases in the available personnel resources. Hogge (1984) states that due to demographical factors, a 25 per cent decline in the national labor pool of eligible 17 to 21 year old men will result by 1992.

One very good example of a robotics application which could increase efficiency and potentially save lives and property is the automated fire-fighting vehicle work going on at the Naval Surface Weapons Center (NSWC), White Oak, Maryland. The implications of this fire-fighting system for human factors are pervasive. Envisioned as an autonomous fire-fighting vehicle on the deck of a present day aircraft carrier, the application illustrates human interactions at extreme levels of control/display use, of information requirements for monitoring, and of safety considerations.

Mavor and Parsons (1984) in their paper presented at the "Robotics and Factories of the Future" conference, include a discussion of several robotic systems under development or being proposed by the Army, the Navy, and the Air Force. The authors identified the need for Human Factors Engineering in the design of control and monitoring facilities, allocation of functions, skill level and training needs, and safety issues. The authors concluded that the lessons learned during the application of Human Factors to military robotic systems also apply to commercial robotic activities.

A DoD-wide group has recently been formed to engender information and technology. The group's charter and points of contact are contained in Appendix D.

2.3.1 Specific Technology Review-Robotics

Hazards that face robots in the industrial setting directly impact Human Factors design issues. For example, stray electrical signals, fluctuating power sources, or electrical "noise" could inadvertently activate a robot servo during maintenance activities and cause serious bodily injury or equipment damage. Likewise,the environment in which a robot is placed could be subject to corrosive chemicals, gasses, heat or other factors which normally would not be allowed near a human. In such environments, crucial switch contacts or button travel may be affected or impeded to the point of providing a serious difficulty if and when an emergency arises. Such operational or design factors are accounted for in Engelberger's classic work Robots in Practice which discusses and summarizes similar hazards, providing examples of situations which affect robot implementation (Englberger, 1980, P.76). The illustrations brought out by Engelberger would require well-thought-out human factors considerations. For example, an emergency-button operational check circuit may be required to ensure that an operator can access the integrity of the emergency subsystem. Placing switches in a control area external to the robot with television viewing is another type of design option. Engelberger stresses that a robot must be fit into the workplace in a sensible, integrated fashion and touches on a wide range of robotic technology including safety, but he does not specifically cover the human factors involved in technology applications.

A large number of books, journals, and other printed media were reviewed during this project. The Society of Manufacturing Engineers and its allied organization, Robotics International, should be a first choice for contacting professionals or for seeking information in the field of Robotics. The organization sponsors conferences and publishes proceedings covering a wide range of robotics topics. During the week of April 21-25, 1986, the ROBOTS 10 conference will be held in Chicago.

There are a number of very good reference handbooks related to robotics. For example, "The Handbook of Industrial Robotics" (Noh, 1985) contains articles by experts in the field, articles about robot installations in industrial operations, as well as many sections dealing with engineering specifications, and equipment component descriptions. It also contains an extensive bibliography and glossary of robotics terms. The Industrial Robotics Handbook by V.D. Hunt (1985) is a handbook which deals with safety considerations as well as specific robotic technology. Many other references can be found which delve into very detailed engineering aspects of robotics design. For example, a recent book by Asada and Slotine (1986) Robot Analysis and Control goes into great detail regarding kinematic and dynamic analysis of manipulator arms as well the details of techniques for trajectory and motion control.

Another example of a handbook for automated systems design can be found in "Industrial Robotics" by Stonecipher (1985). He provides many design guidelines in addition to illustrations of industrial applications. Stonecipher also provides a section on safety and gives some specific steps to undertake to ensure operator and maintainer safety. He provides a list of questions which would be helpful and necessary to adequately design for safety in a robotics application. For example, what options (such as types of warning devices or prevention subsystems) are available for intrusion control? (Stonecipher, pg. 233).

A book by Toepperwein (1983) discusses Workplace Design Conditions and poses some interesting general approaches to operator and/or maintainer safety including one example of providing a rope all around the robot work area which could activate a stop/panic button. Rathmill, MacConaill, and O'Leary, and Browne (1985) reports data on industrial deaths and accident hazards when using industrial robots. General pointers for solving observed problems include better layout, work organization, and processes.

A very good recent book detailing safety concerns and procedures has been edited by Bonney and others (1985). Individual contributors delineate issues such as problems of guarding robot work areas, application of sensor systems, and various safety interlock procedures among others. One chapter in the book edited by Bonney discusses as a trend in Manufacturing Technology the use of Computer Aided Design (CAD) as an aid to robot safety. The use of computers to aid in efficient workspace design is also discussed in other recent major publications (IEEE, 1985; Donath and Leu, 1985). The use of CAD in these applications is a direct attempt to solve robotic workstation design problems by means of the use of advanced graphics technology and artificial technology. The CAD area offers great potential for Human Factors inputs into robotics system design.

As referenced in section 2.2, Lockheed engineers have developed a system using graphic mannekins (ADAM and EVE) to assist in the application of human factors to improve Automated Manufacturing Technology (AMT). The Lockheed system has been developed on an IBM mainframe computer using IBM displays, controls, mouses, and other associated peripherals. McDonnell Douglas Aircraft Company is developing a similar mannekin design aid system in their advanced manufacturing facilities in St. Louis. They are using large VAX computers and Evans-Sutherland graphics systems and peripherals.

2.4 CONCLUSION

After a review of pertinent literature and discussions with selected professionals in the Human Factors, Expert System, and Robotics fields, a number of available guidebooks have been identified which are viable candidates for incorporation into an expert system data base. The most feasible candidate is the type of guidebook format exemplified by the Human Factors Test and Evaluation Manual (HFTEMAN) produced in versions for the Navy and the Army. The format offers the following:

- Already accepted, valid data and techniques.
- Built upon standardized data (i.e., MIL-STD-1472).
- Comprehensive in many areas of Human Factors.
- Branching format readily adaptable to expert systems programming requirements.
- Modular design readily adapts to new data which must be added to the Knowledge Base.

The format selected lends itself to the design and development of a natural-language, user-friendly interface as well as algorithms which will be built to respond to user inquiries. The selection of the above format is not without some deficiencies. Data will have to be restructured and inappropriate sections will have to be deleted. New data pertinent to robotics and man-computer interactions will have to be incorporated (e.g., pertinent ANSI standards data).

But overall, the selection permits software adaptation and allows an excellent format from which professionals can review, improve, and build upon to cost-effectively derive a working and useful knowledge base.

REFERENCES

Asada, H., and Slotine, J.J.E., Robot Analysis and Control, John Wiley and Sons, Inc., Somerset, New Jersey, 1986.

Asimov, I., I Robot, Doubleday and Company, Inc., New York, New York, 1950.

Birk, H., and Kelly, R., "Overview of the Basic Research Needed to Advance the State of Knowledge in Robotics," IEEE Trans. Syst. Man. Cyben., Vol. 11, p. 574-79, 1981.

Bloom, H.M., and McLean, C.R., Standardization Suggested by the AMRF - A Research Testbed for the Factory of the Future, Paper presented at The First International Symposium on Automated Integrated Manufacturing, San Diego, California, 5-6 Apr 1983.

Bonney, M.C., et. al., Using SAMMIE Computer Aided Design System for Workplace Design, Institute of Management Services Summer School, Cambridge, England, 1980.

Bonney, M.C., Case, K., Hughes, B.J., Kennedy, D.N., and Williams, R.W., "Using SAMMIE for Computer-Aided Workplace and Work Task Design," Society of Automotive Engineers Paper 740270, SAE Congress, Feb 1974.

Bonney, M.C., and Williams, R.W., "CAPABLE. A Computer Program to Layout Controls and Panels," Ergonomics Vol. 20, No. 3, 1977.

Boose, J., "A Framework for Transferring Human Expertise," Human-Computer Interaction, Elsevier Science Publishers B.V., Amsterdam, 1984.

de Green, K.B. (ed.), Systems Psychology, McGraw-Hill Book Company, New York, New York, 1970.

Design Handbook Series 1-0: Personnel Subsystems, Headquarters, Air Force Systems Command, 1972.

DHR Incorporated, Robotics Technology: An Assessment and Forecast, (Contract #F33615-83-D-5083 for Aerospace Industrial Modernization Office, Wright-Patterson Air Force Base, Ohio), Jul 1984.

Donath, M., and Leu, M.C., (eds.), Proceedings of Robotics and Manufacturing Automation Symposium, ASME Winter Annual Meeting, 1985.

Edmonds, E., "The Man-Computer Interface: A Note on Concepts and Design," International Journal of Man-Machine Studies, Vol. 16, p. 234, 1982.

Engelberger, J.F., Robotics in Practice: Management and Applications of Industrial Robots, American Management Association, 1980.

Everett, LCDR B., Shipboard Applications: Are the Robots Really Coming?, NAVSEA, Washington, D.C., 1985.

"Factories of the Future," Business Week-Special Report, pp. 72, 3 Mar 1986.

Feigenbaum, E.A., and Feldman, J. (eds.), Computers and Thought, McGraw-Hill Book Company, New York, New York, 1963.

Feigenbaum, E.A., and McCorduck, P., The Fifth Generation: Japan's Computer Challenge to the World, Addison-Wesley Publishing Company, Reading, Massachusetts, 1983.

Fitts, P.M., et al., Human Engineering for an Effective Air Navigation and Traffic Control System, National Research Council, Washington, D.C., 1951.

Gagne, R.M. (ed.), Psychological Principles in System Development, Holt, Rinehart, and Winston, New York, New York, 1962.

Gilbreth, F.B., Motion Study, D. Van Nostrand Co., Inc., New York, New York, 1911.

Hayes, R.H., and Wheelwright, S.C., Restoring Our Competetive Edge: Competing Through Manufacturing, John Wiley and Sons, Inc., Somerset, New Jersey, 1984.

Hirsch, R.S., "National Standards and the Practice of Human Factors," Human-Computer Interaction, Elsevier Science Publishers B.V., Amsterdam, 1984.

Hogge, Sharon M., Naval Robotics and Artificial Intelligence Systems Applications for the Future, in Proceedings of the Robots 9 Conference, 1984.

Hogge, Sharon M., Specifications and Research Issues Concerning Robotic Mobility for Naval Shipboard Applications, in Proceedings of the Robots 9 Conference, 1984.

Hunt, H.A., and Hunt, T.L., Human Resources Implications of Robotics, W.E. Upjohn Institute for Employment Research, Kalamazoo, Michigan, 1983.

Hunt, V.D., Smart Robots: A Handbook of Intelligent Robotic Systems, Chapman and Hall, New York, New York, 1985.

IEEE International Conference on Robotics and Automation (Proceedings), St. Louis, Missouri, 25-28 Mar 1985.

Jenkins, J.P., "An Application of an Expert System to Problem Solving in Process Control Displays," in Human-Computer Interaction, Elsevier Publishers B.V., Amsterdam, 1984.

Johnsen, E.G., and Corliss, W.R., Human Factors Applications in Teleoperator Design and Operation, John Wiley and Sons, Inc., 1971.

Kamali, J., Moodie, C.L., and Salvendy, G., "A Framework for Integrated Assembly Systems: Humans, Automation, and Robots," International Journal of Production Research, Vol. 20, No. 4, p. 431-48, 1982.

Kerr, A.J., "Assembly Robots Address User's Needs," Robotics World, Vol. 4, No.4, Apr 1986.

Kilmer, R.D., McCain, H.G., Juberts, M., and Legowik, S.A., Watchdog Safety Computer Design and Implementation, In Proceedings of Robots 8 Conference Dearborn, Michigan, 1984.

Kinsley, J., "Sophisticated Sensor Systems Enhance Robot Safety," Electrical Construction and Maintenance, Vol. 83, No. 2, p. 53-54, 1984.

Lyman, J., and Madni, A.M., "Operator Roles in Robotics," Robotics Age, p. 39-41, Jan 1984.

Maguire, M., "An Evaluation of Published Recommendations on the Design of Man-Computer Dialogues," International Journal of Man-Machine Studies, Vol. 16, p. 237-61, 1982.

Manufacturing Ergonomics, International Business Machines Corporation, Vol. 3, No. 1, 1985.

Martensson, N. (ed.), 14th International Symposium on Industrial Robots, in Proceedings of the 14th International Symposium on Industrial Robots Conference, 1985.

Mavor, A., and Parsons, H.M., An Overview of Human Factors and Robotics in Military Applications, Paper presented at Robots and Factories of the Future, Charlotte, North Carolina, 4-7 Dec 1984.

McDonald, D.J., "Human Factors - The Forgotten Element in Design," Machine Design, p. 108, 9 Sep 1976.

Meister, D., Human Engineering Data Base for Design and Selection of Cathode Ray Tube and other Display Systems, Navy Personnel Research and Development Center, 1984.

Meister, D., and Salvendy, G. (ed.), " New Opportunities in the Human Factors Engineering of Computer Systems," in Human-Computer Interaction, Elsevier Science Publishers B.V., Amsterdam, 1984.

Meister, D., and Sullivan, D.J., "Human Factors: Engineering's Blind Spot," Electro-Technology, pp. 39-47, Aug 1968.

Michaelis, P.R., Miller, M.L., and Hendler, J.A., "Artificial Intelligence and Human Factors Engineering: A Necessary Synergism in the Interface of the Future," in Directions in Human/Computer Interaction, Ablex Publishing Corporation, Norwood, New Jersey, 1982.

MIL-STD-1472A. Human Engineering Design Criteria for Military Systems, Equipment, and Facilities, 15 may 1970.

Minsky, M. (ed.), Semantic Information Processing, The MIT Press, Cambridge, Massachusetts, 1982.

Naisbitt, J., Megatrends: Ten New Directions Transforming Our Lives, Warner Books, New York, New York, 1982.

Naval Sea Systems Command Integrated Robotics Program, Annual Report FY84 - Navsea Technical Report No. 450-90G-TR-0002, Office of Robotics and Autonomous Systems (SEA 90G), Dec 1984.

Nof, S.Y., Ed., Handbook of Industrial Robotics, John Wiley and Sons, Inc., Somerset, New Jersey, 1986.

Ohubo, T., Aoki, Y., Horie, S., and Ueno, Y., "Some Fundamental Problems of Application of Industrial Robots in Production," Human-Computer Interaction, Elsevier Publishers B.V., Amsterdam, 1984.

Parsons, H.M., ERGONOMIE ET ROBOTIQUE (Human Factors and Robotics), Paper presented at The International Conference on Occupational Ergonomics, Toronto, Canada, May 1984.

Parsons, H.M., Human Factors in Robot Safety, Paper presented at Seminar on Robot Safety of the Robotic Industries Association," ROBOTS EAST Exposition, Boston, Massachusetts, Oct 1985.

Parsons, H.M., Robotics and the Health of Workers, Paper presented at The International Scientific Conference on Occupational Health and Safety in Automation and Robotics, Kitakyushu, Japan, 20-21 Sep 1985.

Parsons, H.M., and Kearsley, G.P., Human Factors Engineering in Robotics, Paper presented at 13th International Symposium on Industrial Robots 7, Chicago, Illinois, 1983.

Parsons, H.M., and Kearsley, G.P., "Robotics and Human Factors: Current Status and Future Prospects," Human Factors, Vol. 24, No. 5, p. 535-52, 1982.

Price, H., "The Allocation of Functions in Systems," Human Factors, Vol. 27, No. 1, p. 33-45, 1985.

Rathmill, K., MacConaill, P., O'Leary, S., and Browne, J., Robot Technology and Applications, Springer-Verlag, IFS (publications) Ltd., United Kingdom, 1985.

Riley, Frank J., Assembly Automation: A Management Handbook, The Industrial Press, New York, New York, 1983.

Rogers, J.G., and Armstrong, R., "Use of Human Engineering Standards in Design," Human Factors, Vol. 19, No. 1, p. 15, 1977.

Rubinstein, R., and Hersh, H.M., The Human Factor: Designing Computer Systems for People, Digital Press, Burlington, Massachusetts, 1984.

Salvendy, G., "Human-Computer Communications with Special Reference to Technological Developments, Occupational Stress and Educational Needs," Ergonomics, Vol. 25, No. 6, p. 445, 1982.

Salvendy, G. (ed.), Human-Computer Interaction, Elsevier Science Publishers B.V., Amsterdam, 1984.

Shulman, H.G., and Olex, M.B., "Designing the User-Friendly Robot: A Case History," Human Factors, Vol. 27, No. 1, pp. 91-98, 1985.

Simons, G.L., Expert Systems and Micro, John Wiley and Sons, Inc., Somerset, New Jersey, 1986.

Shneiderman, B., Software Psychology: Human Factors in Computers and Information Systems, Winthrop Publishers, Inc., Cambridge, Massachusetts, 1980.

Society of Manufacturing Engineers (SME), Industrial Robots: Volume 1 - Fundamentals, SME, Detroit, Michigan, Nov 1983.

Stonecipher, K., Industrial Robotics, Hayden Book Company, Berkely, California, 1985.

Taylor, F.W., The Principles of Scientific Management, Harper and Brothers, New York, New York, 1911.

Toepperwein, L.L., Industrial Robots, Noyes Data Corporation, Park Ridge, New Jersey, 1983.

Toffler, A., Future Shock, Random House, New York, New York, 1970.

Waterman, D.A., A Guide to Expert Systems, Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

Wickens, C.D., and Kramer, A., "Engineering Psychology," Annual Review of Psychology, Vol. 36, p. 307-48, 1985.

Yong, Y.F., Bonney, M.C., and Taylor, N.K., Safety and Industrial Robot Systems - How CAD Can Help, Paper presented at The International Conference on Man-Machine Systems, University of Manchester's Institute of Science and Technology, 6-9 Jul 1982.

BIBLIOGRAPHY

Acoustical Society of America, American National Standard: Guide for the Evaluation of Human Exposure to Whole-Body Vibration (ANSI Standard S3.18-1979), New York, New York, 1979.

The Air Force Systems Command Design Handbook Series, Human Factors Engineering Series, 1972.

American Industrial Hygiene Association, Heating and Cooling for Man in Industry (2nd. ed.), Akron, Ohio, 1975.

American Industrial Hygiene Association, Industrial Noise Manual (3rd. ed.), Akron, Ohio, 1975.

American National Standard for Human Factors Engineering of Visual Display Terminal Workstations (currently in draft format)

Asada, H., and Slotine, J.J.E., Robot Analysis and Control, John Wiley and Sons, Inc., Somerset, New Jersey, 1986.

Asimov, I., I Robot, Doubleday and Company, Inc., New York, New York, 1950.

Birk, H., and Kelly, R., "Overview of the Basic Research Needed to Advance the State of Knowledge in Robotics," IEEE Trans. Syst. Man. Cyben., Vol. 11, p. 574-79, 1981.

Bloom, H.M., and McLean, C.R., Standardization Suggested by the AMRF - A Research Testbed for the Factory of the Future, Paper presented at The First International Symposium on Automated Integrated Manufacturing, San Diego, California, 5-6 Apr 1983.

Blumenthal, M.S., Brown, B.A., Smith, J.E., Dray, J., and Glassner, B., Computerized Manufacturing Automation. Employment, Education, and the Workplace, Washington, D.C.: Office of Technology Assessment, 1984.

Bonney, M.C., et. al., Using SAMMIE Computer Aided Design System for Workplace Design, Institute of Management Services Summer School, Cambridge, England, 1980.

Bonney, M.C., Case, K., Hughes, B.J., Kennedy, D.N., and Williams, R.W., "Using SAMMIE for Computer-Aided Workplace and Work Task Design," Society of Automotive Engineers Paper 740270, SAE Congress, Feb 1974.

Bonney, M.C., and Williams, R.W., "CAPABLE. A Computer Program to Layout Controls and Panels," Ergonomics, Vol. 20, No. 3, 1977.

Boose, J., "A Framework for Transferring Human Expertise," Human-Computer Interaction, Elsevier Science Publishers B.V., Amsterdam, 1984.

Boyce, P.R., Human Factors in Lighting, New York, New York, 1981.

Damon, A., Stoudt, H.W., and MacFarland, R.A., The Human Body in Equipment Design, Cambridge, Massachusetts, 1966.

de Green, K.B. (ed.), Systems Psychology, McGraw-Hill Book Company, New York, New York, 1970.

Design Handbook Series 1-0: Personnel Subsystems, Headquarters, Air Force Systems Command, 1972.

DHR Incorporated, Robotics Technology: An Assessment and Forecast, (Contract #F33615-83-D-5083 for Aerospace Industrial Modernization Office, Wright-Patterson Air Force Base, Ohio), Jul 1984.

Donath, M., and Leu, M.C., (eds.), Proceedings of Robotics and Manufacturing Automation Symposium, ASME Winter Annual Meeting, 1985.

Edmonds, E., "The Man-Computer Interface: A Note on Concepts and Design," International Journal of Man-Machine Studies, Vol. 16, p. 234, 1982.

Engelberger, J.F., Robotics in Practice: Management and Applications of Industrial Robots, American Management Association, 1980.

Everett, LCDR B., Shipboard Applications: Are the Robots Really Coming?, NAVSEA, Washington, D.C., 1985.

"Factories of the Future," Business Week-Special Report, pp. 72, 3 Mar 1986.

Feigenbaum, E.A., and Feldman, J. (eds.), Computers and Thought, McGraw-Hill Book Company, New York, New York, 1963.

Feigenbaum, E.A., and McCorduck, P., The Fifth Generation: Japan's Computer Challenge to the World, Addison-Wesley Publishing Company, Reading, Massachusetts, 1983.

Fitts, P.M., et al., Human Engineering for an Effective Air Navigation and Traffic Control System, National Research Council, Washington, D.C., 1951.

Gagne, R.M. (ed.), Psychological Principles in System Development, Holt, Rinehart, and Winston, New York, New York, 1962.

Geer, C.W., Navy Manager's Guide for the Analysis Section of MIL-H-46855 D180-19476-2, Boeing Aerospace Company, Naval Air Development Center, Jun 1976.

Gilbreth, F.B., Motion Study, D. Van Nostrand Co., Inc., New York, New York, 1911.

Hannah, L.D., and Reed, L.E., Basic Human Factors Task Data Relationships in Aerospace Systems Design and Development, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, 1965.

Hayes, R.H., and Wheelwright, S.C., Restoring Our Competetive Edge: Competing Through Manufacturing, John Wiley and Sons, Inc., Somerset, New Jersey, 1984.

Hirsch, R.S., "National Standards and the Practice of Human Factors," Human-Computer Interaction, Elsevier Science Publishers B.V., Amsterdam, 1984.

Hogge, Sharon M., Naval Robotics and Artificial Intelligence Systems Applications for the Future, in Proceedings of the Robots 9 Conference, 1984.

Hogge, Sharon M., Specifications and Research Issues Concerning Robotic Mobility for Naval Shipboard Applications, in Proceedings of the Robots 9 Conference, 1984.

Human Engineering Design Data Digest No. 544-097/20746, Washington, D.C.: U.S. Government Printing Office, 1986.

Human Factors in Computing Systems, in Proceedings of the Computer and Human Interaction Conference (CHI '83), Boston, Massachusetts, 12-15 Dec 1983.

Human Factors in Manufacturing, in Proceedings of the 1st International Conference, London, United Kingdom, 3-5 Apr 1984.

Hunt, H.A., and Hunt, T.L., Human Resources Implications of Robotics, W.E. Upjohn Institute for Employment Research, Kalamazoo, Michigan, 1983.

Hunt, V.D., Smart Robots: A Handbook of Intelligent Robotic Systems, Chapman and Hall, New York, New York, 1985.

IEEE International Conference on Robotics and Automation (Proceedings), St. Louis, Missouri, 25-28 Mar 1985.

Illuminating Engineering Society, American National Standard Practice for Industrial Lighting (ANSI Standard RP-7-1979), New York, New York, 1979.

Jenkins, J.P., "An Application of an Expert System to Problem Solving in Process Control Displays," in Human-Computer Interaction, Elsevier Publishers B.V., Amsterdam, 1984.

Johnsen, E.G., and Corliss, W.R., Human Factors Applications in Teleoperator Design and Operation, John Wiley and Sons, Inc., 1971.

Kamali, J., Moodie, C.L., and Salvendy, G., "A Framework for Integrated Assembly Systems: Humans, Automation, and Robots," International Journal of Production Research, Vol. 20, No. 4, p. 431-48, 1982.

Kerr, A.J., "Assembly Robots Address User's Needs," Robotics World, Vol. 4, No.4, Apr 1986.

Kilmer, R.D., McCain, H.G., Juberts, M., and Legowik, S.A., Watchdog Safety Computer Design and Implementation, In Proceedings of Robots 8 Conference Dearborn, Michigan, 1984.

Kinsley, J., "Sophisticated Sensor Systems Enhance Robot Safety," Electrical Construction and Maintenance, Vol. 83, No. 2, p. 53-54, 1984.

Kony, S. (ed.), Work Design: Industrial Ergonomics (2nd. ed.), Columbus, Ohio, 1983.

Kroemer, K.H.E., Ergonomics of VDT Workplaces, Akron, Ohio, 1983.

Kvalseth, T.O., Ergonomics of Workstation Design, Boston, Massachusetts, 1983.

Lyman, J., and Madni, A.M., "Operator Roles in Robotics," Robotics Age, p. 39-41, Jan 1984.

Macek, A.J., Using Human Factors to Make Automation Work, in Proceedings of the First International Robot Conference, Wheaton, Illinois, Tower Conference Management, 1983.

MACINTER I, in Proceedings of the 1st International Network Seminar on Man-Computer Interaction Research, Berlin, Germany, 16-19 Oct 1984.

Maguire, M., "An Evaluation of Published Recommendations on the Design of Man-Computer Dialogues," International Journal of Man-Machine Studies, Vol. 16, p. 237-61, 1982.

Malone, T.B., and Shenk, S.W., The Human Factors Test and Evaluation Manual (HFTEMAN), 1 Oct 1976.

Manufacturing Ergonomics, International Business Machines Corporation, Vol. 3, No. 1, 1985.

Martensson, N. (ed.), 14th International Symposium on Industrial Robots, in Proceedings of the 14th International Symposium on Industrial Robots Conference, 1985.

Mavor, A., and Parsons, H.M., An Overview of Human Factors and Robotics in Military Applications, Paper presented at Robots and Factories of the Future, Charlotte, North Carolina, 4-7 Dec 1984.

McDonald, D.J., "Human Factors - The Forgotten Element in Design," Machine Design, p. 108, 9 Sep 1976.

Meister, D., Human Engineering Data Base for Design and Selection of Cathode Ray Tube and other Display Systems, Navy Personnel Research and Development Center, 1984.

Meister, D., and Salvendy, G. (ed.), " New Opportunities in the Human Factors Engineering of Computer Systems," in Human-Computer Interaction, Elsevier Science Publishers B.V., Amsterdam, 1984.

Meister, D., and Sullivan, D.J., "Human Factors: Engineering's Blind Spot," Electro-Technology, pp. 39-47, Aug 1968.

Michaelis, P.R., Miller, M.L., and Hendler, J.A., "Artificial Intelligence and Human Factors Engineering: A Necessary Synergism in the Interface of the Future," in Directions in Human/Computer Interaction, Ablex Publishing Corporation, Norwood, New Jersey, 1982.

MIL-H-46855A, Human Engineering Requirements for Military Systems, Equipment, and Facilities, 2 May 1972.

MIL-STD-1472A. Human Engineering Design Criteria for Military Systems, Equipment, and Facilities, 15 may 1970.

Minsky, M. (ed.), Semantic Information Processing, The MIT Press, Cambridge, Massachusetts, 1982.

Naisbitt, J., Megatrends: Ten New Directions Transforming Our Lives, Warner Books, New York, New York, 1982.

National Research Council Committee on Vision, Video Displays, Work, and Vision, Washington, D.C., 1983.

Naval Sea Systems Command Integrated Robotics Program, Annual Report FY84 - Navsea Technical Report No. 450-90G-TR-0002, Office of Robotics and Autonomous Systems (SEA 90G), Dec 1984.

Nertney, R.J., et. al., Human Factors in Design, Aerojet Nuclear Company, Idaho Falls, Idaho, 1976.

Nof, S.Y., Ed., Handbook of Industrial Robotics, John Wiley and Sons, Inc., Somerset, New Jersey, 1986.

Ohubo, T., Aoki, Y., Horie, S., and Ueno, Y., "Some Fundamental Problems of Application of Industrial Robots in Production," Human-Computer Interaction, Elsevier Publishers B.V., Amsterdam, 1984.

Parsons, H.M., ERGONOMIE ET ROBOTIQUE (Human Factors and Robotics), Paper presented at The International Conference on Occupational Ergonomics, Toronto, Canada, May 1984.

Parsons, H.M., Human Factors in Robot Safety, Paper presented at Seminar on Robot Safety of the Robotic Industries Association," ROBOTS EAST Exposition, Boston, Massachusetts, Oct 1985.

Parsons, H.M., Robotics and the Health of Workers, Paper presented at The International Scientific Conference on Occupational Health and Safety in Automation and Robotics, Kitakyushu, Japan, 20-21 Sep 1985.

Parsons, H.M., and Kearsley, G.P., Human Factors Engineering in Robotics, Paper presented at 13th International Symposium on Industrial Robots 7, Chicago, Illinois, 1983.

Parsons, H.M., and Kearsley, G.P., "Robotics and Human Factors: Current Status and Future Prospects," Human Factors, Vol. 24, No. 5, p. 535-52, 1982.

Parsons, H.M., Mavor, A.S., and Shields, N., Human Factors Engineering Issues in Army Field Robotics Applications, U.S. Army Human Engineering Laboratory, Contract No. DAAK11-R-0101, Aberdeen Proving Ground, Maryland, Dec 1984.

Poulton, E.C., Environment and Human Efficiency, Springfield, Illinois, 1970.

Price, H., "The Allocation of Functions in Systems," Human Factors, Vol. 27, No. 1, p. 33-45, 1985.

Ranta, J., Wahlstrom, B., and Westesson, R., Guidelines for Man-Machine Interface Design, Espoo, Aug 1981.

Rathmill, K., MacConaill, P., O'Leary, S., and Browne, J., Robot Technology and Applications, Springer-Verlag, IFS (publications) Ltd., United Kingdom, 1985.

Riley, Frank J., Assembly Automation: A Management Handbook, The Industrial Press, New York, New York, 1983.

Rogers, J.G., and Armstrong, R., "Use of Human Engineering Standards in Design," Human Factors, Vol. 19, No. 1, p. 15, 1977.

Rubinstein, R., and Hersh, H.M., The Human Factor: Designing Computer Systems for People, Digital Press, Burlington, Massachusetts, 1984.

Salvendy, G., (Ed.), Handbook of Industrial Engineering, New York, New York, 1982.

Salvendy, G., "Human-Computer Communications with Special Reference to Technological Developments, Occupational Stress and Educational Needs," Ergonomics, Vol. 25, No. 6, p. 445, 1982.

Salvendy, G. (ed.), Human-Computer Interaction, Elsevier Science Publishers B.V., Amsterdam, 1984.

Shulman, H.G., and Olex, M.B., "Designing the User-Friendly Robot: A Case History," Human Factors, Vol. 27, No. 1, pp. 91-98, 1985.

Simons, G.L., Expert Systems and Micro, John Wiley and Sons, Inc., Somerset, New Jersey, 1986.

Shneiderman, B., Software Psychology: Human Factors in Computers and Information Systems, Winthrop Publishers, Inc., Cambridge, Massachusetts, 1980.

Society of Manufacturing Engineers (SME), Industrial Robots: Volume 1 - Fundamentals, SME, Detroit, Michigan, Nov 1983.

Stonecipher, K., Industrial Robotics, Hayden Book Company, Berkely, California, 1985.

Suggestions for Designers of Navy Electronics Equipment, Naval Electronics Laboratory Center, San Diego, California, 1975.

Tarr, T.F., Cutlip, W.E., and Hogge, S.M., Navy Fire Fighting Truck Performance Enhancement Through Remote Control, in Proceedings of Robots 10, Chicago, Illinois, 20-24 Apr 1986.

Taylor, F.W., The Principles of Scientific Management, Harper and Brothers, New York, New York, 1911.

Toepperwein, L.L., Industrial Robots, Noyes Data Corporation, Park Ridge, New Jersey, 1983.

Toffler, A., Future Shock, Random House, New York, New York, 1970.

Trends in Ergonomics/Human Factors II, in Proceedings of the Second Mid-Central Ergonomics/Human Factors Conference, Purdue University, West Lafayette, Indiana, 13-15 Jun 1985.

U.S. Department of Health, Education, and Welfare, The Industrial Environment-Its Evaluation and Control, Washington, D.C., 1973.

VanCott, H.P., and Kinkade, R.G., (eds.), Human Engineering Guide to Equipment Design (revised edition), 1972.

Video Display Terminals: Preliminary Guidelines for Selection, Installation, and Use, AT&T Bell Laboratories Circulation Department, Short Hills, New Jersey, 1983.

Waterman, D.A., A Guide to Expert Systems, Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.
Wickens, C.D., and Kramer, A., "Engineering Psychology," Annual Review of Psychology, Vol. 36, p. 307-48, 1985.

Woodson, W.E., Human Factors Design Handbook, McGraw-Hill Book Company, New York, New York, 1981.

Yong, Y.F., Bonney, M.C., and Taylor, N.K., Safety and Industrial Robot Systems - How CAD Can Help, Paper presented at The International Conference on Man-Machine Systems, University of Manchester's Institute of Science and Technology, 6-9 Jul 1982.

APPENDIX A

TECHNICAL MEETINGS ATTENDED/INDIVIDUALS CONTACTED

A. TECHNICAL MEETINGS

1. RI/SME Washington - Baltimore Chapter 303
   January 30, 1985
   Marc Carlson, GMF Robotics
   "The Unmanned Factory"

Mr. Carlson discussed Fanuc, Ltd. of Japan. Fanuc recently completed a motor assembly plant that uses 101 robots and 60 people to produce 10,000 motors per month. A detailed description of the plant's operation was followed by a general discussion of the societal impact of unmanned, automated factories.

2. Expert System Conference
   September 30 - October 1, 1985
   Washington, DC

This conference was a very significant one in terms of technical information gathered and professional contacts made. PSI personnel met with Air Force representatives and discussed technology programs on-going at the Rome Air Development Center (RADC) sponsored by the Air Force Systems Command (AFSC). Discussions with Air Force personnel also covered Expert System (ES) development work underway at Air Force Office of Scientific Research (AFOSR) and at the Air Force Human Resources Laboratory (AFHRL). The future direction of robotics in the Air Force was discussed and it is clear from the initiative to develop a graduate curricula in robotics at the Air Force Institute of Technology (AFIT) that robotics is a major consideration for the Air Force.

Army personnel contacted at the symposium included points of contact from DARCOM headquarters in Virginia to the tank automation center (Rochester, Michigan) to the Engineering Test Laboratory at Fort Belvoir, Virginia. Future references obtained included personnel located at the Human Engineering Laboratory in Maryland where a center for Robotics R&D has been established. The Army personnel also discussed the Defense Advanced Research Projects Agency (DARPA) initiatives in Artificial Intelligence and Super-Computers. Special note was made of the Artificial Intelligence Test Beds established by DARPA at Fort Leavenworth and Fort Sill. NASA officials at the conference were informative as to ES advancements made and technology gaps. A congressional mandate to NASA stipulates that 10% of the space station funding (about $800 million) is to be used for automation and robotics.

PSI personnel reviewed a wide range of issues with industry representatives at the conference. As a summary, PSI representatives met with personnel from Boeing, TRW, Logicon, Booze Allen, and Digital Equipment Corporation to discuss ES's. The type of points discussed and brief conclusions follow, but they are examples only and hardly do justice to the wealth of information obtained:

- User enters symbols as much as possible, Expert System must define and correlate;

- Success of Expert System requires deep familiarity with the technical domain and originality for data extraction and presentation;

- Rule model system developed on a WICAT 68000 in Prolog had to be translated later into PASCAL for efficiency;

- One Expert System developed by DEC used seven different languages (i.e., user interface, linking software, inference software, traditional data base management system software, special report generator software, and display and peripheral drivers);

- An Intellimac representative postulated that ADA would be a language of choice for DoD Expert System work in the future. He discussed a benchmark conversion of LISP to ADA. ADA increased Coding required by over 100%, but expanded ADA code still processed seven times faster.

3. Naval Sea Systems Command
October 18,1985
Hobart R. Everett, Director of Robotics and Autonomous Systems (SEA 90G)
William Butler, Assistant for Robotics and Autonomous Systems (SEA 90G)

A meeting among LCDR Hobart Everett, Mr. Bill Butler, Dr. James McGuinness, and Mr. Joseph Wagner was convened at the Naval Sea Systems Command. LCDR Everett and Mr. Butler were briefed on the current NSWC contract by the PSI representatives.

Pertinent Navy personnel/projects related to human factors and robotics were identified. LCDR Everett distributed a copy of the FY-84 Annual Report from the Office of Robotics and Autonomous Systems (SEA 90G) and then discussed its content. A number of systems which could benefit from human factors were noted in the report and were discussed among the group.

4. Naval Surface Weapons Center (NSWC)
October 24, 1985
Sharon Hogge

Mr. Wagner visited NSWC to video tape the operation of the Cincinatti Milacron HT3 robot. This served two company functions:

1) To guide the Natural Language design phase of the specification; and

2) To be a "communication tool" for PSI software personnel.

The operation of the robot was video taped from a human factors perspective. The focus was on the Job, Duty, Task, and Task Element work breakdown for personnel who operate the robot.

5. National Bureau of Standards
November 18, 1985
Public Test Run of the Automated Manufacturing Research Facility (AMRF)

The Automated Manufacturing Research Facility (AMRF) at the National Bureau of Standards is a major national laboratory for technical work in interface and standards activity to support the next generation of Computer Automated Manufacturing.

The AMRF consists of five machining or measurement workstations, each built around a major (off-the-shelf) machine tool and its tending robot or robots; a material handling system; a network; a data adminstration system; a cell control level; and higher levels of control.

The tour of this facility included a visit to the CAD facility equipped with an IBM 4341 computer and the Group Technology coding system running on an Iris workstation. The Process Planning system is developed on a Symbolics LISP machine. The main shop floor was the next stop during the tour. Here, the horizontal workstation, the vertical workstation, and the turning workstation were demonstrated. The last stops on the tour were at the two-robot coordination station and the inspection station, respectively.

6. SME Chapter 48
December 11, 1985
Dr. Steven Rattien
"The Center for Innovative Technology (CIT)"

The Center for Innovative Technology (CIT) is a recently organized, not-for-profit corporation created by the Commonwealth of Virginia. Dr. Rattien discussed how CIT research activities are organized, what CIT can do for high-tech, traditional, and start-up companies, and how to do business with CIT. The meeting was attended by about 15 people including Mr. Harvey Knowles of the David Taylor Naval R&D Shipyard. After Dr. Rattien's presentation, Dr. McGuinness had the opportunity to discuss with Mr. Knowles human factors applications in robotics including projects being performed at the David Taylor Naval R&D shipyard.

7. SME/CASA F188
December 18, 1985
John W. McInnis, Office of Naval Acquisition Support
"Manufacturing - Art or Science"

Mr. McInnis discussed why analysis of the physics and chemistry of the manufacturing process leads to productivity gains in manufacturing. This analysis is needed prior to the expenditure of capital on new high-tech equipment so that unit operations are scientifically based and hence repeatable. Oftentimes manufacturing is more of an art in that the "manufacturing recipe" is lost when workers are replaced or new demands are made on the system.

One of the many important points made during the presentation included an attack of the oft-quoted saying "if it ain't broke, don't fix it." It is critical to analyze even working systems to ensure that modifications do not upset productive systems. This is especially true for the introduction of automation to the workplace.

8. McDonnell-Douglas Aircraft Corproation (MCAIR)
January 16, 1986
St. Louis, Missouri

PSI personnel initially contacted and met with Mr. Gunn, Vice President, Washington, DC operations and through his office, initial phone discussions were held with Mr. Charles Plummer, Program Manager, for MCAIR's Industrial Modernization Improvement Program (IMIP). A visit was arranged and Dr. McGuinness traveled to St. Louis and held technical discussions with Dr. Tsegay Moges, Section Manager IMIP, and Mr. Hulas King, Manager, Manufacturing Systems Engineering Product Definition/Artifficial Intelligence. Mr. Len Baker, IMIP held discussions with Dr. McGuinness and demonstrated a graphics program developed on VAX 780 computers and Evans-Sutherland display sub-systems. MCAIR has a stick-like mannekin system that is being developed for anthropometric evaluations involving robot cells.

9. Essex Corporation Meeting
H. MacIlvaine Parsons and Ann Mavor
January 22, 1986

Dr. McGuinness and Mr. Wagner held a technical discussion with H. MacIlvaine Parsons and Ann Mavor which covered state-of-the-art human factors engineering applications to robotics. Essex Corporation is conducting technical work for the Army's Human Engineering Laboratory to assess and design for human factors and robotics integration. Mr. Parsons raised two human factors issues which affect worker productivity and motivation in the workplace that have not been addressed in the literature. They are that ... the effect of other people (coworkers) ... and the incentives and disincentives generic to an organization.

10. RI/SME - Human Factors Division
January 24,1986
Meeting

Representatives from the Society of Manufacturing Engineer's (SME) headquarters, labor, education, and industry attended this meeting which marked the rejuvination of the human factors division of Robotics International (of the Society of Manufacturing Engineers). The purpose of the meeting was to rebuild the division and covered topics such as Division status review, planned Division activities, one/five year plans, and membership recruitment. This division will work to emphasize the importance of human factors in robotics and will provide the opportunity for human factors professionals to discuss strategies for the infusion of human facotrs considerations in robot design and use.

11. ASME RI-DC
February 4, 1986
James Albus, National Bureau of Standards (NBS)

Mr. Albus discussed the work in robotics applications being performed at the NBS Automated Manufacturing Research Facility. An overview of the current capabilities of robots developed in the United States and the future trends in robot development was followed by a review of Japanese robotic applications and trends.

12. Software Psychology Society
Potomoc Chapter
February 14, 1986
Rose Oldfield Hayes, US Postal Service
Frederick Glickman, U.S. Postal Service
Key Dismukes, Air Force Office of Scientific Research
"American National Standards for Human Factors Engineering of Visual Display Workstations"

This seminar presented a walkthrough of the proposed standards for video display workstations being developed by the Human factors Society. The Human Factors Society has established a committee of 11 industry representatives and 6 representatives of academe to develop technical standards for acceptable human factors principles and practices in the design and use of display terminals, workstations, keyboards, and their environment.

Each of the three individuals listed above discussed the draft standard. The key points were that the committee used very few reference documents in the generation of the standard and made numerous statements with no reference sources offered. After numerous points brought out by the audience, the three member panel concurred that the standard may be better received if offered as a "guideline" and not a standard. The application of the information and data is the critical variable to successful use of the document whether offered as a standard or guideline.

13. RI/SME Human Factors Division
February 19, 1986
Meeting

This meeting was attended by individuals representing a wide variety of both government and commercial groups. Representatives from IBM, Chrysler, NASA, the Jet Propulsion Laboratory, the Navy, PSI, ESSEX, CMU, IBEW, the Army, and the Department of Education attended. To increase the Division's visibility, it was decided to develop a human factors "Resource Directory" and fully support an SME/RI symposium related to human factors in robotics. The symposium is scheduled to coincide with the AUTOFACT Conference in Detroit, Michigan during the week of 10-14 November, 1986. Dr. McGuinness will be the conference chairman responsible for planning and assessing speakers and program implementation. Mr. Wagner is coordinating the construction of the Resource Directory.

14. Association for Science, Technology, and Innovation
February 27, 1986
Collin Turner, President, LASR Robotics, Inc.
"Trends in Robotics"

Mr. Turner's speech started with a discussion of the first robot installation in 1961. Topics covered during the speech included the impact of microprocessor technology on robot development, the difference between U.S. and Japanese robot definitions and applications, and the social and economic impact of robot aplications in industry.

15. Naval Training Systems Center (NTSC)
Orlando, Florida
February 28, 1986 and March 27, 1986

Dr. McGuinness met with a number of managers and technical project personnel. Dr. Joseph Funaro, Director of the Human Factors Group, and Dr. Robert Evans were briefed on PSI efforts. Dr. Evans was very interested in applications within their technology R&D. NTSC has initiated a major effort and has been designated as tri-service coordinator for a major multi-million contract to design Expert Systems for training applications (NTSC Contact-Dr. Robert Ahlers). Dr. Art Blaiwes and Dr. Michael Lillienthal of NTSC were also contacted and technical discussions held with Dr. McGuinness.

16. Human Factors Society
April 16, 1986
Dr. Eugene Silverman
"The Role of Human Factors Engineering
in Robotic Technology of the Future

Dr. Silverman (founder and President of ARDC Corporation) discussed the various areas within the field of Robotics that require human factors input. Operator and maintainer task structure and training and technical documentation were discussed as were their effect on the efficiency of the work place. The human factors problems associated with remotely controlled vehicles were also addressed and discussed among the meeting attendees.

17. ROBOTS 10
Robotics International/Society of
Manufacturing Engineers
Chicago, Illinois
April 21 through April 24, 1986

Dr. McGuinness and Mr. Wagner attended technical presentations, participated in formal RI/SME technical group planning and review meetings, and engaged in informal discussions with robotics professionals. PSI personnel also obtained a wide variety of state-of-the-art literature and data as well as reviewed robotics equipment and peripherals in operation and on display.

APPENDIX B

EXPERT SYSTEM OVERVIEW

DEVELOPMENTAL ADVANCES

Some examples of successful Expert Systems developed to date are MYCIN, Dendral, XCON (formerly R1), and CATS-1. MYCIN is a rule-based medical consultant that diagnoses blood-borne bacterial infections such as meningitis. Dendral is a system which automatically performs mass spectroscopy with consistent accuracy and no human error or tedium complaints. XCON, developed by Digital Equipment Corporation (DEC) matches customer computer systems needs with their needs for DEC VAX equipment. XCON is reported as having saved DEC millions of dollars annually. CATS-1 is a General Electric Company Expert System for diagnosing malfunctions in diesel locomotives. These systems have all been developed or are still being developed on mainframe computers.

With the advent of new more powerful microprocessors available on such machines as the Apple Macintosh, the IBM PC AT and the NCR TOWER, useful and powerful Expert Systems are evolving on the microcomputer. Also surfacing are helpful tools with which these systems can be implemented. The choice of language for Expert Systems has primarily been between LISP and PROLOG, or one of their offshoots. LISP was developed during the late '50s and early '60s, about the same time as Fortran. It has become the standard for Artificial Intelligence (AI) applications in the United States. PROLOG, on the other hand, was developed in France in the early '70s. It has become the AI choice in Europe and has recently gained support in Japan. The LISP language lends itself to applications in AI because of its structure. Expert Systems created in LISP can communicate well with the user and offer multiple screen windows which enable the user to develop a cognitive set compatible with the computer's operating mode. Since LISP focuses on symbol manipulation rather than numbers, it lends itself to processing information on a natural language basis. This makes it easier for the inference engine to make associations between the symbols, or words, in the knowledge base and the information provided by the user. Other languages exist, but have been largely developed for system-specific uses.

The emergence of specialized LISP machines and the newer, more powerful microcomputers, has enabled the application of Expert Systems technology in a wide variety aoof tasks. Powerful LISP machines are available from Xerox, Symbolics, and LISP Machine. AI language compilers are also available on microcomputers. ExperTelligence has introduced a LISP Compiler for the Apple Macintosh and other vendors have introduced natural language compilers for the IBM PC and machines compatible with it.

MICROCOMPUTER APPLICATIONS

Using todays powerful microcomputers, it is possible to develop useful expert systems for personal computers. In this section, we will take a look at some of the packages available on popular personal computers. First, a look at developmental tools.

EXSYS

Billed as an affordable advisor, the EXSYS Expert System Development Package is compatible with IBM PC or computer users with 256K RAM. The cost is $395. The package contains an editor for creating the rules and a "run-time" program which can efficiently execute the applications programs without the memory-consuming editor.

The user creates the rules, conditions, and alternatives with the editing module. Because of this, all Expert Systems created using EXSYS look alike. The knowledge base uses straight text presentations that pose multiple choice questions according to the information required of the user. The programmer, however, can add in any comments or messages to be sent to the user during the program, providing helpful, pertinent information during the problem-solving process.

The program is written in "C"; it is fast and also relatively powerful. After loading the program, the knowledge base can make use of 192K of the 256K of RAM, at a rate of about 700 rules per 64K of memory. The extensive availability of more RAM has created a great interest in programs such as EXSYS. The programs themselves are expanded in depth, memory usage, and versatility. The program's (correctness) certainty factor can also be varied, and can be combined along the way. Since the best solution is the one with the highest rate of success, it is beneficial that EXSYS, after finding one solution to the given problem, continues to ask questions of its knowledge base and user to determine if there are more solutions. If multiple solutions are reported, then the program evaluates the probabilities of success and selects the most probable one. Other positive aspects are the program's on-line help facility and its use of color. These both make the creation of rules more manageable. EXSYS is a flexible and powerful program with modest hardware demands. It can process 5,000 rules in a PC based microcomputer.

Expert-Ease

Some experts believe that arranging their thought and ideas into rows and columns, much like a spreadsheet, helps structure their thinking and leads to new insights. This is the idea behind Expert-Ease. Conceived in the United Kingdom, Expert Ease is a very popular Expert System, and has even been called the benchmark program for microcomputer-based Expert System work.

The program's ease of use is evident in many areas. Aside from the fact that the non-expert can draw on others' expertise, it can give experts new insight on their own problems. From a programmer's standpoint, the rules that a program uses do not have to be written; the programmer need only structure the data so that Expert-Ease can infer a logic table from the data associations. This structure produces an inductive Expert System which can link observed effects with potentially unidentified causes. This is especially helpful for professionals such as medical doctors, archeologists, and scientists.

Expert-Ease has limited capability in large applications because it is only able to address 128K of memory, enough for 255 examples with 31 attributes and 31 decisions each. Improvements to the Expert-Ease program continue to provide more addressable memory. A way of getting by memory limitations in a system such as Expert Ease is to create linked modules by dividing the problem into logical sections. Conclusions can be made at the end of each section with a set of directions for each succeeding section. By creating each section separately and linking them together, large applications can be addressed.

Expert-Ease also demands that the programmer be consistent in his examples. There is no room for two identical examples leading to two separate outcomes. But since this is currently one of the first criteria for building a successful working expert system, it should not affect development significantly.

Expert Ease is easy to use with help screens available at nearly all levels, and documentation is well illustrated, including a tutorial with complete examples. Expert Ease is available for the IBM PC at $595.

ES/P ADVISOR

ES/P Advisor is currently a knowledge-based software development system available for the IBM PC that can guide a user through a complete process while furnishing information at every step, a quality found in mainframe Expert Systems. ES/P Advisor is a PROLOG-based Expert-System shell developed by Expert Systems International. The company also developed PROLOG-1, the PROLOG version for the IBM PC in which ES/P Advisor and Technowledge's M.1 systems were developed. ES/P Advisor has adopted features of the PROLOG language and can be modified by a qualified programmer to add the custom features each individual system might require.

The system uses Knowledge Representation Language (KRL), which is one of the more versatile and sophisticated languages available for the PC. KRL supports multiple variable types such as facts, numbers, categories and phrases, the key variables in clear communication of concepts.

ES/P Advisor's PROLOG contains a full set of logical operators to be used in creating a knowledge base. One example is the operator "OR." Both the inclusive and the exclusive versions of the operator are available. The inclusive "OR" allows for multiple fact parameters to be included in a rule. For example, the rule,

> IF (thunder)...OR...(lightning)...OR...(dark clouds gathering quickly) - THEN (it is going to rain, cf = 0.9,0.9,0.75),

provides for any or all of the conditions to affect the rule with the appropriate Certainty Factor (CF). Without the inclusive "OR" the rule would have to be represented as three seperate rules. With the exclusive "OR" only one of a list of fact parameters can be true. Another feature of KRL is "text animation," which allows text to be inserted at any juncture of the consultation. Since most microcomputer expert systems can only relate comments at the end of each consultation, this feature places ES/P Advisor ahead of other similar systems.

Once the knowledge base is constructed, it must be compiled into PROLOG before activated. The strict structure of the language makes it necessary to debug the material before it can run properly, with on-line debugging help available. Exceptions to this include structural changes the programmer makes to the system. Such changes will not receive debugging help from the PROLOG compiler. After all of the bugs have been corrected, one of the best environments for running expert systems on the personal computer is ready to use.

TIMM-PC

TIMM-PC, from General Research, is the first personal computer Expert System created that is capable of finding a solution when the data is incomplete. When presented with a problem that has inadequate data to completely solve it, the program uses what information it is given and formulates the most probable solution. TIMM-PC is unlike the all-or-nothing reasoners in that it finds a partial match when a concrete match is not possible.

As is the case with most Expert Systems available today, TIMM-PC uses a knowledge base composed of IF...THEN rules. The knowledge base begins, however, with a section declaring specific information about utilized attributes of individual knowledge base files. TIMM-PC is also capable of accessing separate knowledge systems via direct branching or referencing through one of its rules.

The program has larger hardware demands than most microcomputer Expert Systems. For current applications, an IBM-PC with 640K RAM, an 8087 math co-processor chip, and a hard disk are required. One of the benefits of this memory requirement is that the Expert System is almost entirely prompt driven, making documentation requirements minimal. The system consists of ten programs on five floppy disks which allow the user to build and edit a knowledge base, exercise the system in problem solving, and make queries of the system, all of which are menu driven. The drawback is that after the programming of the knowledge base is complete, the run time is still inhibited by the presence of the development tools.

TIMM-PC is best suited for problems in which many factors are used to determine a decision, but is limited to applications in which there are 25 or fewer possible outcomes of the problem. It is touted by its developers as being best suited for applications in the areas of "manufacturing, customer service, quality control, engineering, marketing, finance, personnel, research, and development." TIMM-PC's unique quality of reasoning on the basis of similarities rather than exact matching provides for powerful problem solving capabilities for the microcomputer.

M.1

From Teknowledge, an industry leader in Expert Systems, an Expert System development tool is available for the IBM-PC. Available for $10,000, the PROLOG-1 system requires 128K RAM. With two disk drives and with a color monitor, M.1 will distinguish system outputs from user responses using different colors.

According to its developers, M.1 is best suited for "structured selection" applications which are defined as those problems which a human expert can solve in thirty minutes or less, do not involve extensive calculations, can be solved through a telephone conversation with an expert, and have only a few dozen conclusions to choose from. The package includes demonstration systems such as a Wine choice Advisor, a Bank Services Advisor, a Photography Advisor, and a version of the famous Sacon system, a structural analysis engineering package.

The M.1 system consists of two major components: the knowledge base and the inference engine. The knowledge base is constructed with a series of IF... THEN rules and the inference engine checks user inputs against rules in the knowledge base to find matching information. The distinction between the two components is crystal clear. The inference engine is PROLOG-1 based and is the mechanism by which commands are carried out. The knowledge base is created using a standard text editor such as WordStar. This separation allows the user to create a knowledge base as complex as needed within a familiar environment and then access it from M.1. The M.1 system works in compiler fashion by checking the syntax and statement options. Some of the options available are text printing, use of variables, math functions, and list processing. List processing could be used to report a list of values used during various times of a consultation.

M.1 employs a certainty factor system to help in the sifting of the information during a solution search. This makes sure that M.1 will only pay attention to the most relevant rules in the knowledge base. When M.1 is working on a solution, only the questions and answers made during the consultation appear on the screen. The "thought" process is saved in the central holding area known as the cache. Using the trace command, the user can follow this process if he so desires.

The documentation that comes with M.1 is good but is designed to be used in conjunction with Teknowledge's one-week training course. Teknowledge is also using client feedback as a basis for revising their system. It is expected that a future version of M.1 will allow assembly language programmers to develop software that will allow interface between M.1 and many popular databases.

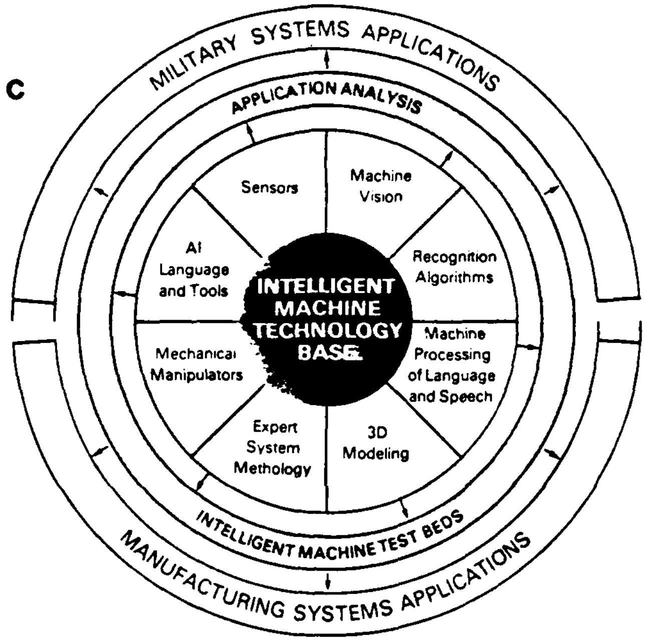
To summarize, Expert System shells and database programs with natural language interfaces are increasing in number and sophistication. Current shells other than those previously introduced include ExperOPS5 from ExperTelligence, Santa Barbara, California and MacKIT from Knowledge Systems Environment, Dilburg, Pennsylvania.

A number of companies are attempting to build natural language interfaces into their database programs. This includes Q&A from Symantec (Cupertino, California) and Paradox from ANSA (Belmont, California). "Q&A" integrates word processing and file management with a full macro facility and an effective natural-language interface. (Byte, January 1986, pp. 120) The databases can be addressed quickly by entering ordinary English phrases and sentences. Data merge, comprehensive report capabilities, and context-sensitive help are included in the word processing and database modules.

Robotics International of SME

APPENDIX C

THE MILITARY SYSTEMS COMMITTEE



The success of the DoD Forums at Robots 7 and 8 in 1983 and 1984 and the response to those planned for 1985 demonstrates the robotics community's interest in improving communication between the nation's manufacturing sector and the Department of Defense. The Military Systems Committee was established to recognize the great technological opportunities that can help improve the productivity and efficiency of both the defense and civilian industrial sectors through the assimilation of robotics technology.

The need for long-term improvements in our manufacturing technology base is broadly recognized. This need is particularly important in terms of defense production and industrial preparedness. The Department of Defense spends billions of dollars annually for a wide range of U.S. manufactured products. Increasing inflation escalates costs at a time when there are mounting pressures to limit government spending. So, it becomes important that DoD suppliers use the most cost effective manufacturing methods to improve quality and reduce costs. Of equal importance is the fact that in the future the DoD will rely more and more upon the economic strength of the U.S. manufacturing sector to keep fielded material up-to-date. A strong modern manufacturing technology base is essential if DoD is to acquire upgradable modules for continuing improvement of its equipment.

To a considerable extent the nation's future Robotics/AI technology base will evolve from the development of intelligent machines for DoD needs. Defense requirements drive technological developments in terms of achieving new performance levels and also in terms of the timeframe within which new developments occur. DoD is also the major source of risk capital for new technology in the country.

The mission of the Military Systems Committee is to stimulate interactive communications between all sectors of DoD and the nation's defense and civilian manufacturing sectors.

Organization

The committee will be organized during the coming year, with meetings on November 29, 1984 (Robots West) and June 3, 1985 (Robots 9).

The committee will consist of a Chair, several Vice-Chairs, and the membership. The head of the committee should have corporate support in order to actively participate in planning and executing committee activities. Candidates for Vice-Chair will be selected on a geographic basis, depending on the level of active interest and participation in various regions. The Chair will be elected by the membership from the group of Vice-Chairs.

Local RI/SME Chapters may establish special interest groups for defense applications of robotics. These groups will form the membership base and produce the leadership of the Military Systems Committee.

Committee meetings and special local events (one-day seminars or workshops) will be organized by the committee, approved by the RI/SME Technical Council, and sponsored by interested RI/SME Chapters. These local events should produce chapter revenues, stimulate interest, and promote membership growth. National events such as the DoD Forums at Robots 8 and Robots 9 will be organized by the committee, approved by the RI/SME Technical Council, and sponsored by RI/SME.

Chapters (or individuals) wishing to participate on the Military Systems Committee should assess the level of interest of their Chapter members and plan to have chapter representatives attend one or both of the scheduled committee meetings. Send suggestions and expressions of interest to:

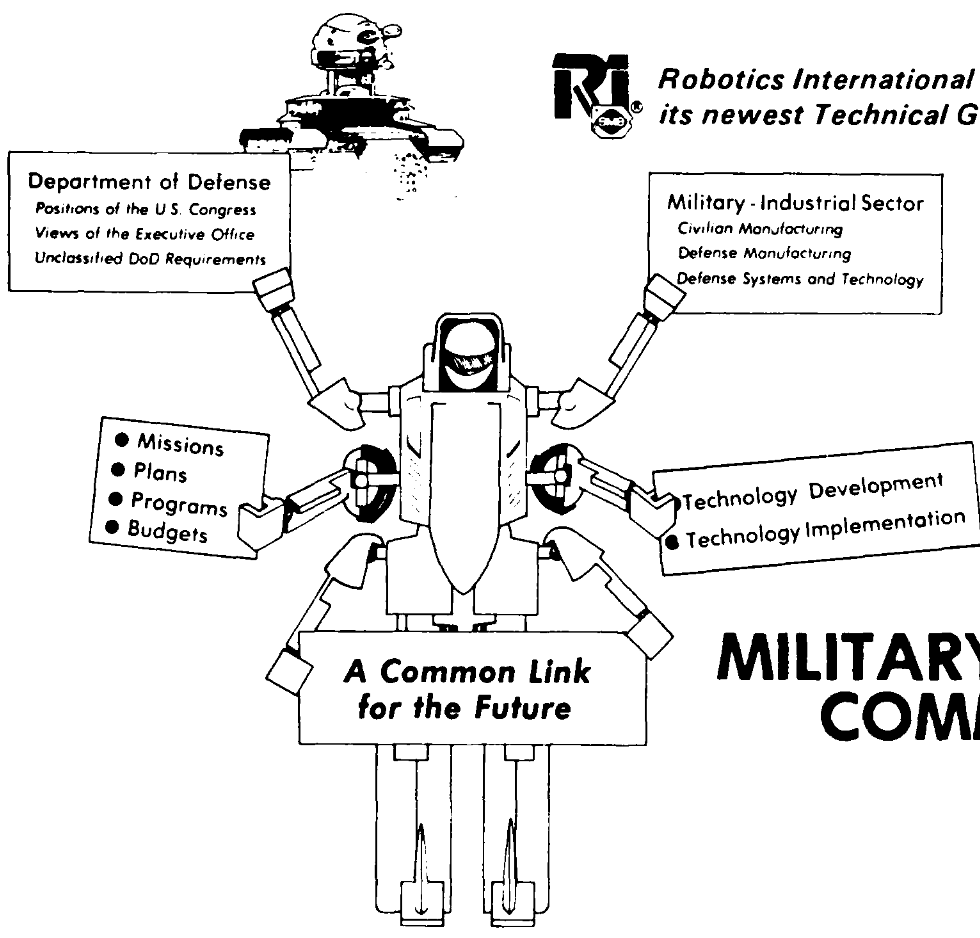
Dave Visscher
RI/SME
One SME Drive
P.O. Box 930
Dearborn, MI 48121
(313) 271-1500

Typical Technical Activities of the Military Systems Committee

- Small Business Opportunites
- Short Term DoD Applications
- Long Term DoD Applications
- DoD Requirements
- DoD Plans and Programs
- DoD Congressional Budgets
- Defense Manufacturing Issues
- Application in Logistics
- Soldier Support Functions
- Material Handling
- Artificial Intelligence
- Sensor Technology
- Hybrid Systems
- Locomotion / Autonomous Vehicles
- Simulation and Modeling
- Automatic Control
- Robotic Data Bases
- DoD Software Plans and Programs
- Intelligent Machine Systems
- Super Computer Architecture

Membership in the Military Systems Committee is open only to U.S. citizens. All members will be required to sign a citizenship form.

Robotics International of SME announces its newest Technical Group



MILITARY SYSTEMS COMMITTEE

Society of Manufacturing Engineers
One SME Drive, P.O. Box 930
Dearborn, Michigan 48121

☐ ***Yes, I am interested in the Military Systems Committee. Please send me more information!***

Name ____________________

Title ____________________

Company ____________________

Division ____________________

Address ____________________

____

____

City ____________________

State ____________________

Zip ____________________

Telephone ____________________

TWX # ____________________

I am currently a member of:
☐ RI SME ☐ CASA SME

Return to: David Visscher
RI SME Aerospace Division
Technical Activities Department
Society of Manufacturing Engineers
One SME Drive, P.O. Box 930
Dearborn, MI 48121
(313) 271-1500, ext. 354
TELEX 297742 SME UR (VIA RCA)

NON PROFIT ORG.
U.S. POSTAGE
PAID
DEARBORN, MICH.
Permit No. 542

APPENDIX D

HUMAN FACTORS DATA AND GUIDELINES AVAILABLE IN HFTEMAN

TABLE D-1. RULES TO GUIDE HUMAN FACTORS DATA USE

SUPPORT, SUPPLY, SERVICE — MATERIEL PRODUCTION & ENVIRONMENT CONTROL

OBJECTIVE: Evaluate the effectiveness and safety of the design of materiel production and environmental control systems and equipment to enable the operator to assemble and set up the item, prepare it for use, and actually use it. Consideration should be given to including conditions representative of those expected in actual use, such as: 1. User Conditions - body size and clothing; 2. Environmental Conditions - weather, climate, light levels, sea state; 3. Operational Condition - use condition (duration, throughput rates, types of materiel or environments).

USER ACTIVITIES			
ASSEMBLE/SET UP		PREPARE FOR USE	
ASSEMBLE/DISASSEMBLE	EMPLACE	ALIGN/CALIBRATE/ADJUST	SERVICE
PURPOSE: Evaluate the design of the test item for ease and safety of assembling component parts and the procedural guidance provided.	PURPOSE: Evaluate the design of the test item for ease of situating the item in its use area and connecting vent pipes, power cables, liquid hoses and pipes.	PURPOSE: Evaluate the design of the test item for ease and reliability of performing alignment and calibration of operating components.	PURPOSE: Evaluate the design of the item for ease of loading raw materials, adding liquids, storing materiels, and servicing.
USER TASKS			
Unstow components. Read/interpret instructions/technical manuals. Identify parts. Connect components. Mate components to chassis.	Install subassemblies. Make connections. Position for use. Connect lines and hoses. Connect structures. Connect mechanical drives.	Read labels/instructions. Tighten/loosen fasteners. Set/adjust controls. Read displays. Identify set points. Read calibration tables. Verify readiness.	Determine status of expendables Open/close access covers. Remove/replace filler caps. Tighten connections. Clean components. Lubricate-oil. Fill-drain.
EQUIPMENT COMPONENTS			
Labels, Manuals, Markings Fasteners, Connectors Handles	Lines, Cables, Hoses Fasteners, Connectors Handles	Displays Labels, Manuals, Markings Fasteners, connectors	Accesses, Covers, Caps

TABLE D-2. HUMAN FACTORS DATA MATRIX

EQUIPMENT COMPONENTS / HUMAN FACTORS CONSIDERATIONS	5 CONTROLS	6 DISPLAYS	8 WORKSPACE
	Components used to activate, deactivate and modify the equipment power source and operating elements. Knobs, switches, wheels, pedals, triggers, levers, cranks, etc.	Components that provide visual and auditory information to the operator concerning the status of operation. Provide positive indication of malfunctions.	The area within which the user operates the equipment. Space for controls, displays, optics, electronic, devices, windows, weapons, storage, etc. Includes seats and consoles.
A LOCATION/ARRANGEMENT The location and orientation of a component as it affects the ability of the operator to reach, operate, or manipulate it, including location of openings and doors, location of components as well as relationships to other components.	• Control relationship to its display is apparent. (1.1-3) • Functionally related controls grouped together. (1.4,8,2.3) • Groups of controls provide for left to right and/or top to bottom order of use. (1.5,2.2) • Controls in functional groups located according to operational sequence, function. (1.9-10) • Oriented to operator. (2.1) • Lifting equipment controls: in easy reach; load visible. (4.1)	• Display is related to its control, system. (1.1,1.6,1.10,3.1) • Functionally related displays grouped together. (1.2,5,7) • Groups of displays provide for left to right and/or top to bottom order of use. (1.3,8) • Positions of related controls, displays on different panels correspond to each other. (1.9) • Displays located so they can be read to the required degree of accuracy by users. (2.)	• Displays placed above standing [seated] surface: normal, 41-74" [6-48"]; precisely, frequently read, 50-69" [14-37"]; ± 22" from user centerline. (1.3-4) • Controls on vertical surface above floor [seat]: normal, 34-74" [8-35"]; precisely, frequently used, 34-57" [8-30"]; ± 22" from user centerline. (1.5-6) • Critical displays on vision over the top (sit) consoles are at least 22.5" above seat. (1.8)
B SIZE/SHAPE The maximum and/or minimum dimensions of components that are required for adequate use, including anthropometric and special clothing considerations, the shape and contour of components as well as the necessary operating clearances.	• Control spacing min: TAB 5.B.1; blind operation 5". (1.1,4) • Controls operable by suitably clothed users: body dimensions from 5% to 95%. (1.3,4.0,6.1) • Rotary control size, shape: FIG 5.B.1-6. (2.1) • Linear control size, shape: FIG 5.B.7-13. (3.1) • Controls have nonslip surfaces. (2.2-6, 3.2-3) - Range of control action, reach angle taken into account.	• Display viewing distance: 28" max; 13" min. (1.) • Pointer does not obscure, exceed width of index marks. (2.1) • Pointers close to dials to eliminate parallax. (2.2) • CRT target visual angle >20 minutes, exceeds 10 lines; distance up to 16" (10" min). (3.) • Counters, flags mounted close to panel surface. (4.)	• Vehicle sizing accommodates at least 90% of users. (1.1-5) • Lateral work [writing] space: 30x16" [24x16"] wxd. (2.4,5) • Back, arm rests and knee, foot room sizes adequate. (2.6-8) • Console design conforms to that is FIG 8.B.2-4. (2.11-15) • Vehicle workspace accommodates suitably clothed personnel. (3.1)
C DIRECTION/FORCE The movement and/or force required to operate or generally manipulate a component, with emphasis on the direction of motion corresponding to the display, component, total item reaction or standard practice as well as the strength required.	• Adequate control feedback. (1.1) • Control motion CW, forward, up, right produces corresponding display motion on fixed scale (or reverse motion with moving scale, fixed pointer) with increasing reading magnitude. (1.2-13,2.1-3) • Rotary valves open CCW. (2.5) • Control forces, displacements: FIG 5.B.1-13. (3.0,4.0) • High force: FIG 5.C.1-2. (5.0) • Pedal operation (7.0)		• Vertical seat adjustment: 16-21" in 1" increments. (1.1) - Seat adjusts fore, aft: 4" min. - Operator does not have to lift self to adjust seat. - Seat adjustment overhead clearance from seat pan: 40" min.
D INFORMATION The information available to the operator in the form of color, size, shape, place or auditory coding of components, including marking and labeling as well as procedures found in design handbooks, job aids and repair manuals.	• Min decoding required. (1.0) • Valve operation labeled. (2.1) • Size code: 3 sizes max. (2.7-3) • Color relates to display. (2.4-7) • Labeling is concise, functional, well located, visible. (5.0) • Operating instructions provided except where obvious. (6.1-4) • Control movements shown parallel to actual control motion. (6.5) • Lifting controls labeled as to function, direction. (6.6) • Main power switch labeled. (7.1)	• Min decoding required. (1.) • Trademarks, irrelevant info do not appear on panel face. (2.1) • Coding techniques uniform; facilitate discrimination, identification, relationship. (2.2-4) • Audio warnings use standard signals, don't interfere with critical functions, signals. (6.) • Verbal warnings: intelligible, apt, concise. (7.2) • Labels: functional, basic; well located, properly sized. (8.)	• Adequate storage space for manuals, worksheets, etc. (1.1) • Standees have work surfaces to support manuals. (1.2) • Hydraulic work platforms have max load signs. (1.3) • Placards mounted next to equipment which presents a hazard to personnel. (2.1) • Areas of operation requiring special clothing, equipment are identified. (2.2) - Instructions kept simple.
E VISIBILITY Those aspects of a component that contribute to the operator's ability to clearly see it, including location, orientation, shape, size, contrast, color viewing distance, field of view and illumination.	• Control shape visually/tactually identifiable. (1.1) • Control color contrasts with background. (1.2) • Ambient light color determines useable control colors. (1.3) • Rotary switch has contrasting reference line, min pointer parallax. (2.1-2) • Thumbwheel control digits are visible to operator. (2.3-6) • Legend switch is legible. (3.1)	• Illumination uniform. (1.1-2,2.7) • Display face >45° to sight line; min parallax, reflection. (1.3-5) • Critical displays are in optimal visual zone: FIG 6.E.2. (1.6) • Indicator lights: show response; frugal use; visible. (2.1-5,9-10) • Contrast >50%. (2.6,3.1,5.7-8) • Flashing lights: 3-5/sec. (2.8) • Coding properly used. (3.2-3) • Counters visible. (5.1-3) • Indicators used during night operations are lighted. (7.)	• Instrument reflection nil. (1.1) • Console view angle <190°. (2.1) • Illumination: levels, TAB 8.E.1; no glare; dimmers used. (3.1) • Remote viewing system provides spatial info (x,y,x). (4.1) • Direct view if possible. (4.2) • Distortion avoided. (4.3) • Remote lighting adequate. (4.4) • Vehicle operator forward field of view: 180 min. (5.1) • Reduce external glare: visors, no tinted windshield. (5.4)
F USE CONDITIONS Those aspects of the component that pertain to its operational status before, during and after use, as well as the maintenance of an acceptable work environment, including temperature, humidity, ventilation, noise, and vibration.	• Control manipulation precision is consistent with system. (1.1) • Controls selected, distributed so that none of operator's limbs are overburdened. (2.1) • Movement oriented to user when several stations are used. (2.3) • Control motion minimized; not cycled On/Off unnecessarily. (2.4) • Control useable despite inadvertent operation protection. (2.6) • Linear control actuation: positive, appropriate. (3.0)	• Display precision, response consistent with system. (1.) • Information displayed: clear, specific, precise, useable, not degraded by vibration. (2.2) • Lights show function. (3.) • Scales: linear; use whole numbers oriented upright. (4.4,5,7) • Audio signals: TAB 6.F.3. (6.) • Audio, verbal warnings: 20dB above background noise. (7.0,8.0) • Audible warning when lift exceeds allowable load. (9.1)	• Heating, A/C: 50-85°F; does not discharge on crew. (2.1-3,6-7) • Ventilation: 30 ft^3/min/man; velocity, <100'/min. (2.4-5) • Airborne [structure] noise in ship equipment, compartments: TAB 8.F.5-6 [FIG 8.F.2]. (3.4-6) • Steady-state, workspace noise limits: TAL 8.F.7-10. (3.7-8) TAB 8.F.10 +5dB, max. (3.9) • Equipment vibration permits safe operation: FIG 8.F.3. (4.0)
G SAFETY Those aspects of a component that could cause personnel injury, including electrical, chemical, mechanical, structural, radiation, and pressurization hazards.	• Shape coded controls free of sharp edges. (1.1) • Critical controls cannot be moved accidentally. (1.2) • "Dead man" control used where incapacity produces a critical condition. (1.3) • Controls initiating hazardous operations require prior operation of locking control. (4.1) • Main power "On-Off" switch cuts all power to equipment. (4.2)	• Display failure apparent. (1.) • Absence of signal does not indicate "go" condition. (2.1,5.1) • Master lights set apart. (2.2) • Yellow, red, flashing red lights used where appropriate. (2.3-5) • Audio warnings directed to earphones and work area. (6.1) • Action segment of audio signal gives nature of problem. (6.2-5) • Prohibited, persistent signals are not used; critical signals	• Personnel not exposed to excess toxic substances. (1.0,5.7) • Hearing Conservation Program observed: TAB 8.G.1. (2.1) • Whole body vibration within twice that of FIG 8.F.3. (3.1) • Windshields, windows are shatter proof, transparent. (4.2) • Hazard warning provided. (5.1) • Adequate illumination. (5.3) • Equipment guarded if temp over 140°F (120°F if handled). (5.4)

PSI

EXPERT SYSTEM DESIGN AID FOR APPLICATIONS OF HUMAN FACTORS IN ROBOTICS

JAN E. RHOADS and JAMES McGUINNESS, Ph.D.

JUNE 1986

PROGRAM DESIGN SPECIFICATION

Contract Number N60921-85-C-0252
FOR PERIOD AUGUST 1985 - APRIL 1986

Prepared for

WHITE OAK LABORATORY
NAVAL SURFACE WEAPONS CENTER
ROBOTICS RESEARCH AND DEVELOPMENT LABORATORY
ATTN: SHARON HOGGE
10901 NEW HAMPSHIRE AVENUE
SIVER SPRING, MD 20903-5000

PERSON-SYSTEM INTEGRATION
Human Factors - Systems Analysis

2401 HUNTINGTON AVENUE ALEXANDRIA, VIRGINIA 22303-1531
(703) 960-5555

CONTENTS

FIGURES

TABLES

1.0 INTRODUCTION

Human Factors (HF) must be considered in the design stages of a robotic system to ensure effective and efficient operation and maintenance, to increase safety, and to decrease personnel training time/costs. Implementation of a system in accordance with this Program Design Specification (PDS) will result in an Expert System which can apply Human Factors Technology to the following:

- Direct Operations (examples: controlling a robot's movement or applying inputs necessary to program a robot for a tasking change)
- Direct Maintenance (example: hands-on maintenance, test, or adjustment of a robot)
- Remote Monitoring of Robotic Operations/Maintenance (example: access design of crucial information displayed on an operator's console)

The PDS will guide the necessary analysis as well as the development and implementation of an Expert System to apply Human Factors within system configurations which employ robotics to accomplish tasks. Almost all such configurations require the interaction of one or more human beings to fully support operational as well as maintenance activities.

The Expert System to be discussed in this specification will be designated as <u>HF-ROBOTEX</u> (Human Factors-Robotics Expert System). HF-ROBOTEX is designed to assist in the application of Human Factors (HF) principles, data, and techniques to robotics systems. The system will be designed so that it can be used by any HF engineer with limited experience in robotics, and/or any robotics-oriented engineer without extensive experience in HF.

HF-ROBOTEX is intended as a tool. It must be preceded by well-thought out analysis; automation is not a substitute for good front-end work. Input quality still will be reflected in quality output. Such a tool must be used by a craftsman in most cases to avoid misapplication. Used correctly by a competent, trained specialist, it will produce effective, safe system designs for robotic applications quickly and at low cost. Correct use will also provide a critical communication link among Human Factors, Engineering, and other design specialities leading to more widespread utilization. Such use will furnish a capability to quickly accomplish trade-offs during design stages. Timing is often critical if Human Factors are to be influental in a final system design.

Figure 1-1 depicts the flow of activity for a hypothetical robotics design cycle and shows the point at which an Expert System should be inserted in the flow to yield maximum benefit. Initially, a thorough analysis process is conducted focused on customer specifications. The robotics engineer/designer then starts off by drafting the robot functional definition, which is next translated physically into a robot breadboard design. During this phase, the functional definition and breadboard design should be reviewed by an HF engineer who will apply HF-ROBOTEX to both design products. The Expert System will generate specific HF guidelines. Those guidelines that have not been fully accommodated by the current configuration are passed back as feedback to the engineer/designer for his consideration in the design cycle, which must be repeated again for the incorporation of pertinent HF principles. Guidelines that are finally approved are integrated within the robot system.

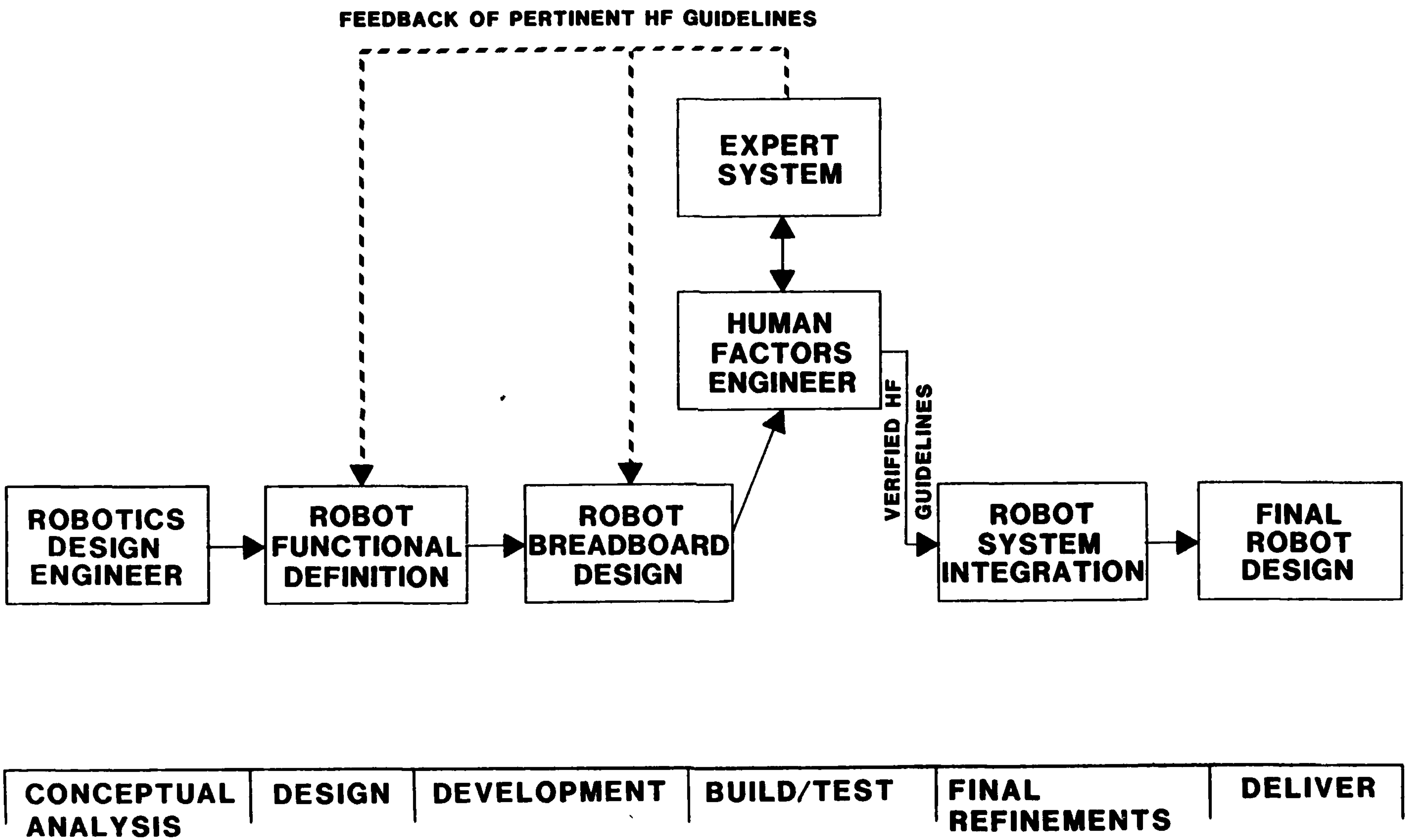


FIGURE 1-1. HYPOTHETICAL STAGES OF A ROBOTICS APPLICATION

1.1 PURPOSE

This PDS describes the program architecture and delineates interfaces crucial to the success of an Expert System to apply Human Factors in robotics. It also includes important programming guidelines to be followed in order to implement the digital processor program.

1.1.1 Scope

The Expert System to be developed and implemented in accordance with this PDS shall have the following features and capabilities:

(1) An interactive, user-friendly interface constructed using state-of-the-art display techniques;

(2) A rule-based Inference Engine within a modularly-designed shell that will permit high speed, large capacity architecture compatible with IBM PC hardware environments; and

(3) An established knowledge base with sufficient levels of detailed data to produce guidelines in the form of design suggestions along with supporting criteria.

1.1.2 Identification

The following paragraphs provide an explanation of the three basic components of an Expert System discussed above and their relationship to the major functions of the digital processor program. The three basic components are:

(1) Input Stage (User Interface),

(2) Processor Stage (Inference Engine), and

(3) Output Stage (Knowledge Base).

The input, processor, and output stages are cited here to help conceptualize the relationship between an Expert System and clearly organized, sequential concepts originally developed for system analysis.

1.2 SUMMARY

1.2.1 Overview of Expert System Structure

The three major components of an Expert System and their relationship to the major functions of the digital processor program are delineated below:

The Input Stage (User Interface) will allow software control of the process by which a user can effectively turn on the system, access it, modify it, stimulate it and receive coherent, correct responses from the system. The input procedures should be human-engineered to reduce drudgery, obviate errors, and ease user interaction with the system. The goal is to design an input module with menu-driven graphics and multiple "pop-up" windows that do not require the user to "learn" how to use the system each time it is turned on. The underlying programming design will ensure that the User Interface is efficient timewise, is effective, and allows a user to employ the Expert System with a minimum of difficulty and a limited amount of training. The interface will permit a wide variety of technical questions and responses.

The Processor Stage (Inference Engine) is the heart of the Expert System. When the user is examining multiple human task elements for "fit" within a robotics system, the Inference Engine must effectively optimize the combination of Human Factors data or techniques to make the best fit. This rule-based Inference Engine is specifically tailored to help the user identify appropriate equipment and associated tasks and sequence them for use. Subgoals are then translated into the components and, finally, Human Factors categories are searched and selected. The resulting goals point to, or identify desired frames which are contained in a knowledge base for the user to review.

The Output Stage (Knowledge Base) of the Expert System is where the Human Factors data and techniques reside. The latest validated state-of-the-art must be compiled from Human Factors experts, interviews and writings, source books, and other resources for inclusion in the Knowledge Base. The Knowledge Base will be frame-based with three levels of data. It will contain sufficient data to respond to broad-based user inquiries with guidelines in the form of design suggestions along with supporting criteria. Amplifying figures/tables will be employed where necessary for further elaboration.

The associated hardware and software for the Expert System will be designed by first examining the performance requirements, then finding flexible, comprehensive software, and finally defining the hardware necessary to support the software. This SBIR project is intent upon providing modular hardware and software. The system modules will be proven state-of-the-art, expandable, and transferable to ensure growth with future improvements in software or hardware. The digital processing program is to consist of an originally conceived and developed user interface interacting with a modified state-of-the-art shell program (i.e., Insight 2+ from Level Five Research, Inc.) integrated with a state-of-the-art database management program (i.e., dBase III from Ashton-Tate). PSI has initiated an agreement with Level Five Research to modify Insight 2+ to meet HF-ROBOTEX needs.

1.2.2 Knowledge Representation

All Expert Systems must have a way of representing knowledge or "structuring" factual information, and then a way of accessing that knowledge. The Expert System proposed in this PDS blends several Artificial Intelligence (AI) strategies for representing knowledge into one coherent package: (1) a semantic network built upon a variation of object-attribute-value (O-A-V) triplets, (2) a rule-based inference engine, and (3) a frame-based knowledge base.

A semantic network is one of the most general representational schemes in AI. It is a tree-like structure of information in which the branches consist of "nodes" that represent "objects" and "descriptors". "Links" relate the two together. The links also serve to direct the search flow through the nodes to "goals" or conclusions at the bottom of the tree.

For the Expert System described in this report, the OBJECTS are both physical objects such as "sensors" and "handlers," and abstract acts such as "activating" and "troubleshooting". The DESCRIPTORS are both physical components of the objects such as "displays" and "cables", and abstract attributes such as "location" and "visibility". The links merely show how the objects, descriptors, and goals are related. The goals are the desired HF guidelines that apply to the specific design.

However, to accomodate the complexity of object-descriptor relationships in robotics design (RD), HF-ROBOTEX has adopted an AI strategy that uses "rules" in place of "links" in the semantic network. A rule is a premise leading to a conclusion, which is commonly referred to as an IF...THEN statement. For example, IF... the RD object is "activating sensors" and the RD descriptors are "display" and "safety", ...THEN a pertinent HF guideline or conclusion would be "preferred visual areas for crucial information displayed on a panel would center around an operator's normal line of sight - approximately 10° down from horizontal").

Any given RD proposition (the " IF..." clause) may have many pertinent HF guidelines. This is also true for it's response rule, (the "...THEN" clause). Conversely, any given HF guideline may also have a great number of antecedent RD propositions. The underlying concept of HF-ROBOTEX relies heavily on both of these juxtaposed "one-in-to-many" logical propositions.

HF-ROBOTEX has adopted another AI strategy called "object-attribute-value (O-A-V) triplets". Figure 1-2 illustrates how knowledge can be represented as rules comprising O-A-V triplets, which are actually a specialized case of a semantic network. Using this representation scheme, any number of objects can be described by the same attributes, and, equally important, any number of attributes can lead to the same values. Yet, any given object acting as a "root" node at the top of the object tree will lead to its own set of values at the bottom of the tree.

Moreover, HF-ROBOTEX has adopted still another AI strategy of using "frames" in the place of "values" within the O-A-V triplets. As an AI strategy, "frames" provide modular representation of facts and relationships. Each frame is a description of an object, containing "slots" for all pertinent information associated with the object. As shown in Figure 1-3, these slots may be used to store not only the attributes of the object, but also, the desired values (HF guidelines) pertinent to the object, "pointers" to other frames where more values may be found, or even the rule itself which "links" the object to its values.

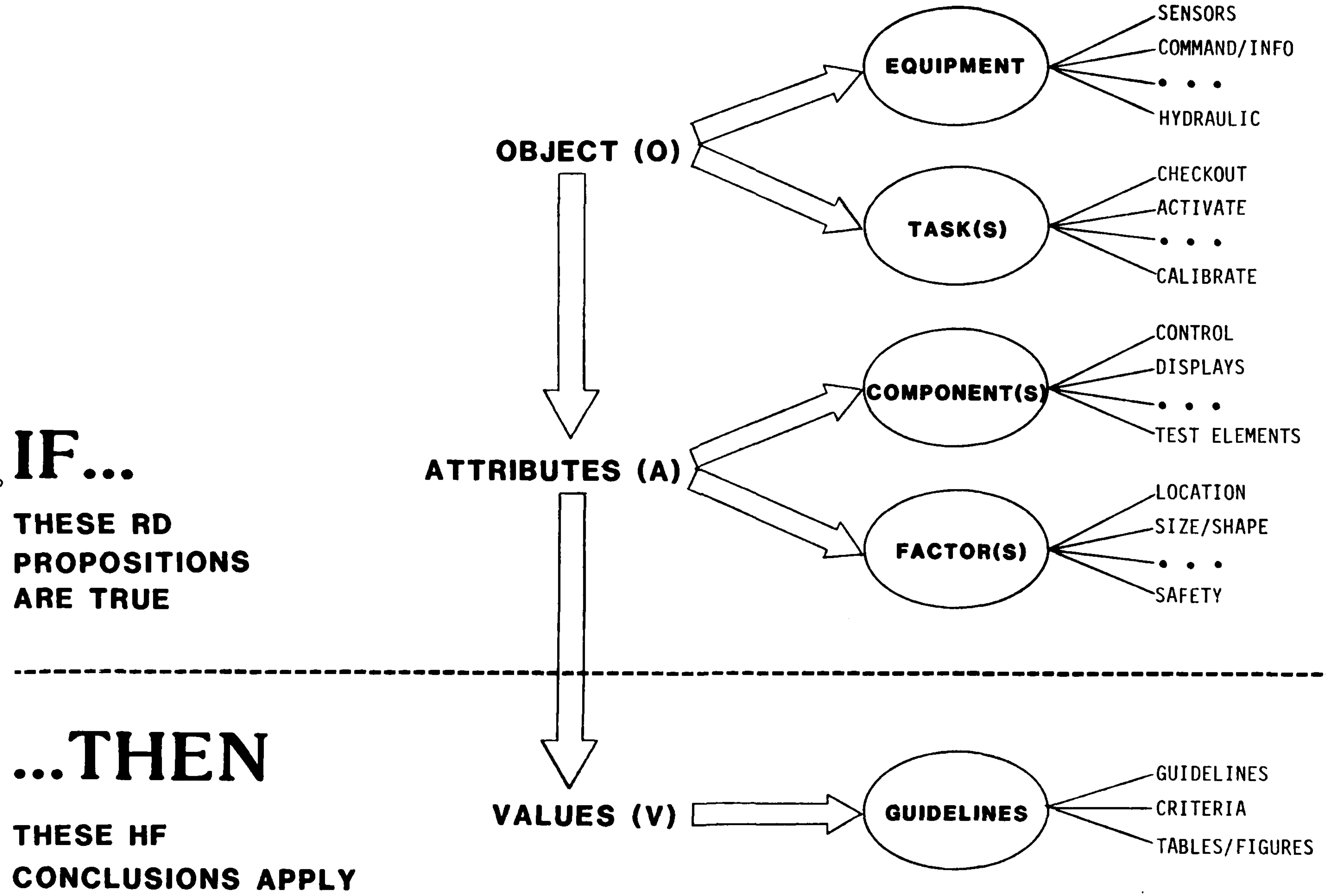


FIGURE 1-2. KNOWLEDGE REPRESENTATION AS RULES COMPRISING O-A-V TRIPLETS

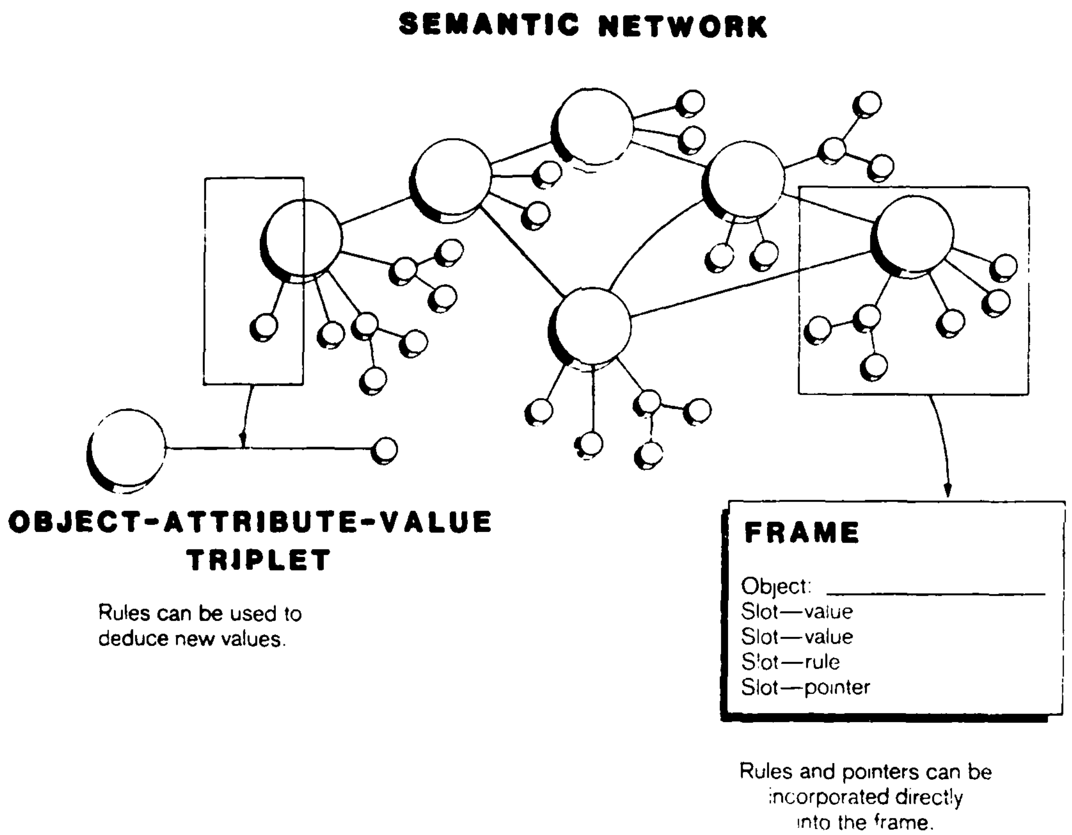


FIGURE 1-3. SEMANTIC NETWORKS, OBJECT-ATTRIBUTE-VALUE TRIPLETS, AND FRAMES

1.2.3 Overview of System Functioning

Figure 1-4 shows an overview of HF-ROBOTEX data flow and how it relates to O-A-V triplet rules. An analogy lies here between the O-A-V triplets of Figure 1-3 and the lower-level rules fired at each system level (1, 2, 3) of Figure 1-4, respectively:

SYSTEM LEVEL 1 — To define a particular "object", HF-ROBOTEX employs a first set of rules at system level 1 to determine what areas of interest (equipment) and what segments of activity (tasks) the Robotics Design (RD) engineer is dealing with.

SYSTEM LEVEL 2 — To narrow the search down to only those HF guidelines which are pertinent, HF-ROBOTEX then employs a second set of rules at system level 2 to determine the object's "attributes" in terms of what equipment elements (components) are necessarily involved and what human considerations (factors) must be dealt with.

SYSTEM LEVEL 3 — Once these attributes are defined, HF-ROBOTEX then employs a third set of rules at system level 3 to retrieve the most pertinent "values" or HF guidelines which are values stored as frames in a knowledge base (KB).

If more detail is required, HF-ROBOTEX can further retrieve the supporting criteria for each guideline (also stored as frames). If still more detail is required, HF-ROBOTEX can, in turn, further identify specific tables/figures that amplify each criteria. The HF-ROBOTEX data flow and structure will be discussed in more depth in Sections 3.1 and 3.4.

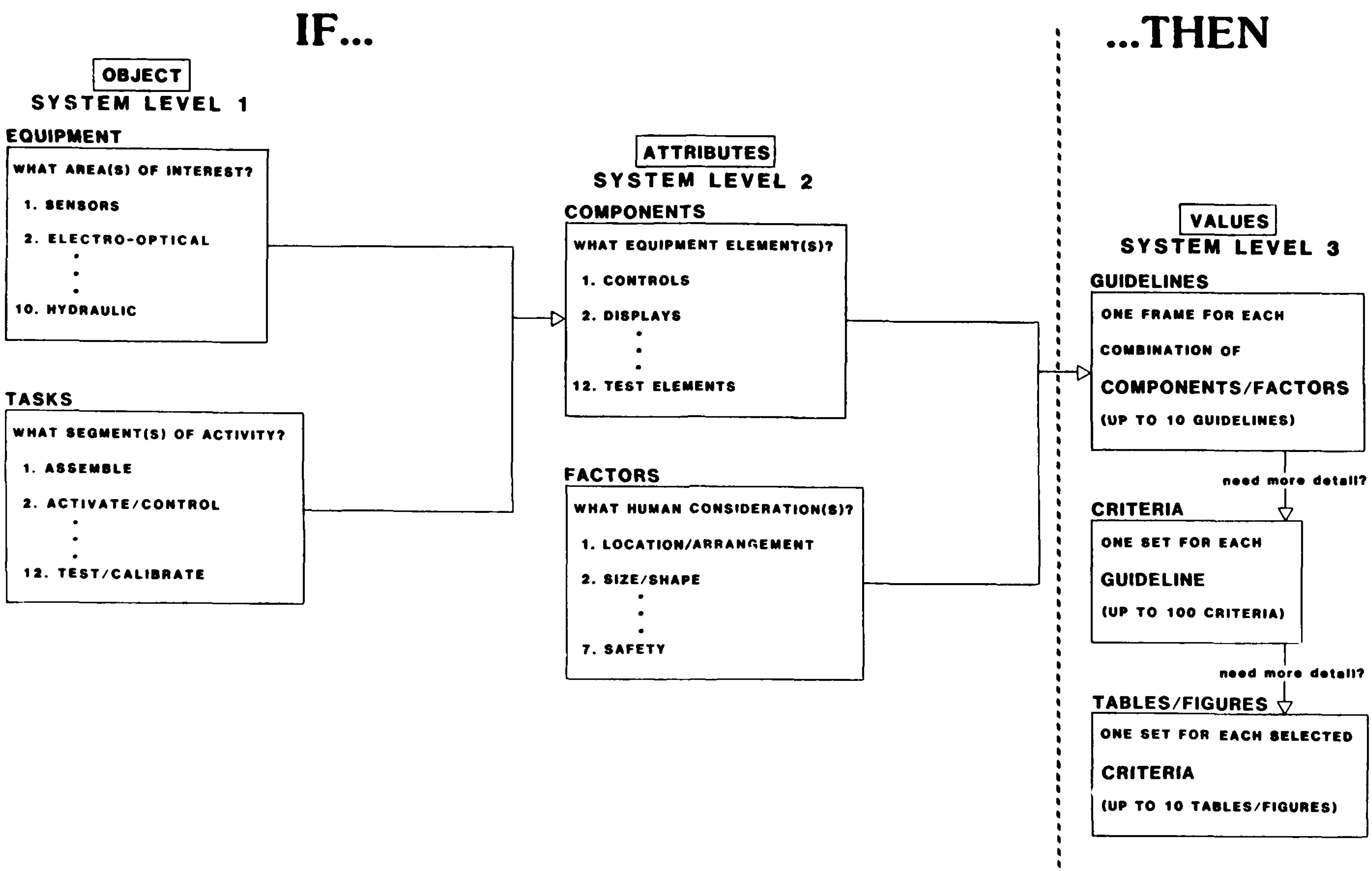
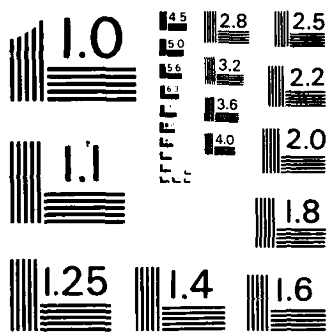


FIGURE 1-4. OVERVIEW OF DATA FLOW

AD-A169 632 EXPERT SYSTEM DESIGN AID FOR APPLICATIONS OF HUMAN FACTORS IN ROBOTICS(U) PERSON-SYSTEM INTEGRATION INC ALEXANDRIA VA J MCQUINNESS ET AL. 12 JUN 86 2/3

UNCLASSIFIED PSI-K31-TR885 N60921-85-C-0252 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

Figure 1-5 shows an overview of the two-phase HF-ROBOTEX system operation:

SEARCH PHASE first, via the Insight 2+ expert system for the Search Phase in which the user formulates his RD proposition; and,

OUTPUT PHASE secondly, via the dBASE III database management system (DBMS) for the Output Phase in which the resulting HF guidelines are displayed to the user.

Essentially, Insight 2+ provides the Inference Engine (IE) and an associated search mechanism to establish the "IF..." portion of the O-A-V triplet shown in Figure 1-4, while dBASE III provides the KB structure and associated access mechanism to establish the "THEN..." portion of Figure 1-4.

Initially, the user, who could be either the RD or HF engineer shown in Figure 1-1, formulates a search query via the user interface. This is done by several levels of interactive questions/responses. This interactive process ultimately yields well-defined search goals which are passed by Insight 2+ as KB access parameters to dBASE III.

Upon receipt of these parameters, the dBASE knowledge base interface uses them to access the designated records (frames) within the KB which contain the pertinent HF guidelines. Once the designated frames are retrieved, the user can selectively display any one or all of the retrieved HF guidelines, and/or their supporting criteria, via the user interface on the right of Figure 1-5. This is done once again by several levels of user-controlled interaction with the KB. HF-ROBOTEX operation will be discussed in more detail in Sections 3.2 and 3.4.

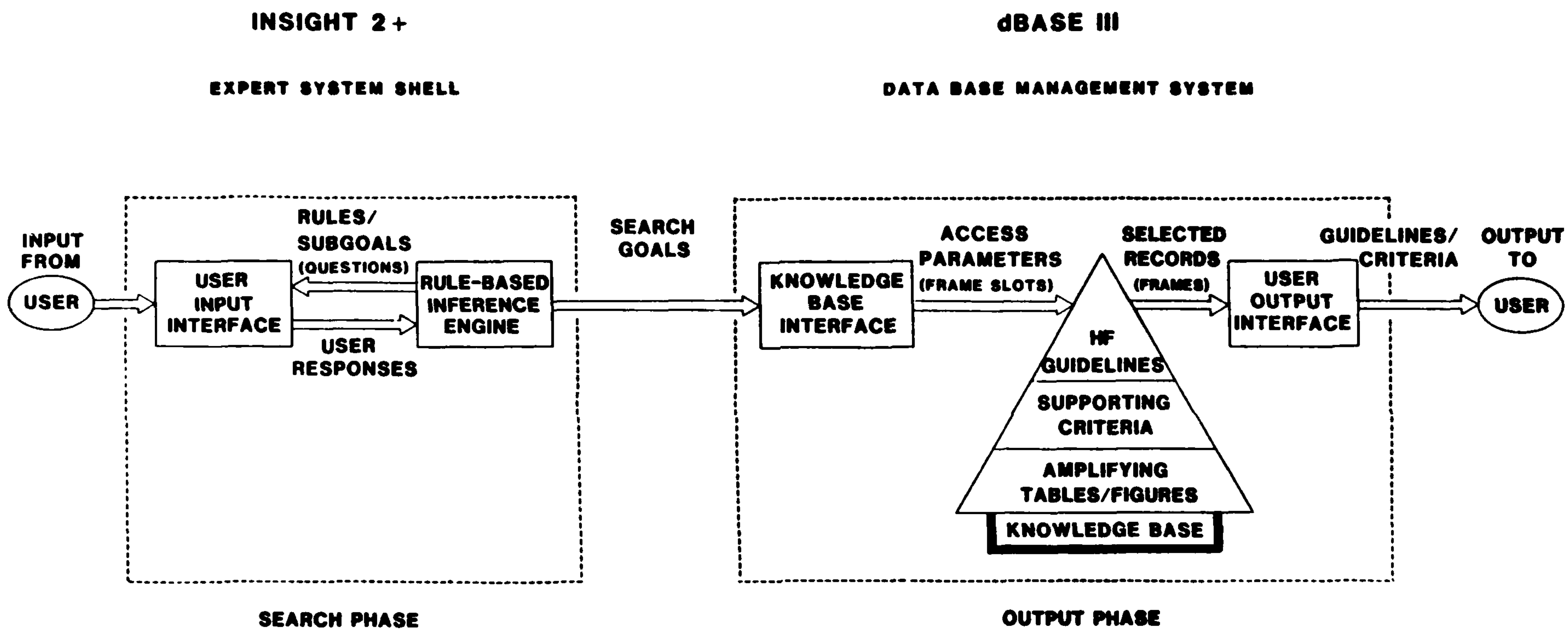


FIGURE 1-5. OVERVIEW OF SYSTEM OPERATION

2.0 APPLICABLE DOCUMENTS

The following documents of the issue in effect on date of completion form part of this specification to the extent referenced herein.

2.1 MILITARY STANDARDS

MIL-STD-847B	Format Requirements for Scientific and Technical Reports prepared by or for the Department of Defense
MIL-STD-1679A (NAVY)	Weapon System Software Development

2.2 DATA ITEM DESCRIPTION

DI-E-2138A	Program Design Specification

2.3 PROGRAMMING REFERENCE MANUALS

ADVANCED Programmers Guide:
Featuring dBASE III and dBASE II
Castro, L., Hanson, J. and Rettig, T.
Published by Ashton Tate, 1985

Insight 2 - Reference Manual
Level Five Research, Inc., November, 1985

Insight 2+ - Addendum to Reference Manual
Level Five Research, Inc., March, 1986

2.4 PUBLICATIONS GUIDES

NSWC MP82-2
Naval Surface Weapons Center 1 April, 1982

3.0 REQUIREMENTS

The following sections contain a comprehensive description of the program design structure and processing necessary to build a complete Expert System for applying Human Factors to Robotics Design (referenced in this PDS as HF-ROBOTEX). The present project requires the development of a program design specification (PDS) generated from a review of currently available robotics applications, needs, relevant computer hardware/software, and pertinent Human Factors data sources.

No formal program performance specification was available at the time of project initiation, nor was one required. Instead a set of performance requirements was derived from the survey and assessment conducted during early Phase I Small Business Innovative Research (SBIR) initiatives. The resulting performance requirements have been documented in this section for convenient review.

3.1 FUNCTION ALLOCATION

This section will first define the performance requirements which must be met by an expert system to apply Human Factors to robotics design. Following this will be the identification of the functions and tasks which must be allocated to meet the requirements. Finally, the specific design structure of the individual digital processor modules, which activate and accomplish the expert system module functions and tasks, will be described.

3.1.1 Performance Requirements

As a primary overall system requirement, HF-ROBOTEX must be designed to allow a user (RD engineer, HF engineer, etc.) to formulate a reasonably narrow query as to his specific RD problem. In response to this query, the system must, within a reasonable time, extract and display a set of pertinent HF guidelines, supported by generic HF criteria, and, in turn, by HF tables/figures (where applicable). These two propositions as to query formulation and data extraction can be logically separated into two distinct system processing segments: a Search Phase and an Output Phase (see Table 3-1).

As a secondary overall system requirement, HF-ROBOTEX must be designed to allow an Update Phase in which a knowledge engineer (KE) can enter data into the system as new rules/goals for the SAearch Phase and as new guidelines/criteria for the Output Phase.

The two overall system requirements can be broken down into more specific performance requirements. Table 3-1 lists the operational performance requirements related to the Search Phase (S1...S10) and the Output Phase (O1...O10), that, to a great extent consist of parallel functions. The remainder of this section breaks each requirement (S1...S10) and (O1...O10) down into specific functions, just as they would be contained in a fully developed Expert System.

TABLE 3-1. PERFORMANCE REQUIREMENTS	
SEARCH PHASE:	formulate query as to a specific RD problem (controlled by user)
OUTPUT PHASE:	extract and display pertinent HF guidelines/criteria (selectively paced and sequenced by user)
UPDATE PHASE:	enter data into system as new rules/goals and new guidelines/criteria (controlled by KE)
	SEARCH PHASE
S1	display search overview (including procedures)
S2	monitor function inputs (for mode changes at any time)
S3	update inference engine (enter new rules/goals, if requested by KE)
S4	access explanation subsystem (if requested)
S5	display definitions, explanations, help messages (if requested)
S6	monitor keyboard/cursor inputs (for user response to each rule)
S7	access inference engine (for next rule, if any)
S8	display current subgoal/next rule (if any)
S9	encode final goals (output as KB access parameters)
S10	maximum allowable response time
	OUTPUT PHASE
O1	display output overview (including procedures)
O2	decode final goals (input as KB access parameters)
O3	monitor function inputs (for mode changes at any time)
O4	update knowledge base (enter new guidelines/criteria, if requested by KE)
O5	monitor keyboard/cursor inputs (for user advance to next element)
O6	advance to next/last data element (sequential or nonsequential) step to next/last data element (guideline or criteria) step to next/last frame (guideline or criteria) step to next/last database (guideline only)
O7	access knowledge base (for next data element, if any)
O8	display next data element (guideline/criteria, if any)
O9	display output summary
O10	maximum allowable response time

3.1.1.1 Search Phase Requirements

S1 display search overview: one or more overview screens will be presented upon startup of the search phase to explain the purpose of the system, the operating modes and controls available to the user (QUERY, EXPLANATION, RULE ENTRY, etc.), and the procedures for formulating a search query.

S2 monitor function inputs: the function keys must be monitored at all times during the search phase for mode changes requested by the user, upon which the system will present a screen indicating the appropriate procedure for each user option.

S3 update inference engine: if selected by the user, the system must provide a mechanism for accessing the IE, selectively inserting new rules/definitions into the IE structure, and modifying or deleting existing rules/definitions therefrom. Since this procedure is not a part of the operational search strategy, it should therefore be performed via an independent offline interface. Moreover, the IE structure must be modular, must be flexible, and must provide a 25% overhead margin to allow for growth in any dimension vertically or horizontally, or in any structural capacity (e.g., storage capacity, operating speed, maximum rules allowed, etc.).

S4 access explanation subsystem: if selected by the user, the system must provide a definition for the current rule (if any), an explanation of the current search status (if any), or a help message on how to perform a desired function.

S5 display definitions, explanations, help messages: upon retrieval, the system will display the appropriate response (if any) to the user request.

S6 monitor keyboard/cursor inputs: the keyboard and cursor must be monitored at all times during the search phase (excpet during an IE access) for the user response to each rule presented by the IE.

S7 access inference engine: upon user keyboard response, the IE must interpret the response and fire the next rule appropriate to that response (if any), yielding the next set of subgoals and/or final goals. Wherever possible to predict the volume of output being requested by the user, the IE should announce that the requested output may exceed 20 frames of guidelines. In any event, to avoid overwhelming the user the IE must not allow the output to exceed 40 frames.

S8 display current subgoal/next rule: upon completion of IE access, the system will display the current subgoal and/or the next rule (if any). For user convenience, the screen format should permit at least five related choices within a rule (or subgoal) to be displayed at once, and, thereafter, any remaining choices should be scrolled by one at a time.

S9 encode final goals: upon reaching the final goals for a given query, the system must encode those goals into parameters suitable for KB access, and pass those parameters to the output phase.

S10 maximum response time for search phase:

10 seconds to generate overview search display
1 second to fire next rule
3 seconds to display next subgoal
5 seconds to display final goals
10 seconds to encode KB access parameters

3.1.1.2 Output Phase

O1 display output overview: one or more overview screens will be presented upon startup of output phase to explain the purpose of the output displays, the operating modes and controls available to the user (GUIDELINE, CRITERIA, DATA ENTRY, etc.), and the procedures for formulating a query.

O2 decode final goals: upon receiving the final goal(s) from the search phase, the system must decode those goals into parameters suitable for KB access and establish the most efficient sequence of access as an output list of pertinent HF guidelines.

O3 monitor function inputs: the function keys must be monitored at all times during the output phase (except during a KB access) for mode changes requested by the user, upon which the system will present for each user option a screen indicating the appropriate.

O4 update knowledge base: if selected by the user, the system must provide a mechanism for accessing the KB structure and modifying or deleting existing guidelines/criteria therefrom. Since this procedure is not a part of the operational search strategy, it should therefore be performed via an independent offline interface. Moreover, the KB structure must be modular, must be flexible, and must provide a 25% overhead margin to allow for growth in any dimension vertically or horizontally, or in any structural capacity (e.g., storage capacity, accessing speed, maximum frames allowed, etc.).

O5 monitor keyboard/cursor inputs: the keyboard and cursor must be monitored at all times during the search phase (except during a KB access) for the user response to each data element presented by the KB.

O6 advance to next/last data element
(sequential or nonsequential option)
step to next/last data element (guideline or criteria),
step to next/last frame (guideline or criteria),
step to next/last database (guideline only):

the system will allow the user to selectively advance the output display to the next/last data element, next/last frame, next/last database (if any), allowing the user to follow the ordinary output sequence or to selectively skip forward or back with a single button-push on the keyboard.

O7 access knowledge base: upon user keyboard response, the KB must interpret the response and access the next data element appropriate to that response (if any). Wherever possible to predict the volume of output being requested by the user, the KB should announce that the requested output may exceed 20 frames of criteria. In any event, in order to avoid overwhelming the user, the KB must not allow the output to exceed 40 frames.

O8 display next data element: upon completion of each KB access, the system will display the next data element available (if any). For user convenience, the screen format should permit at least five related guidelines (or criteria) to be displayed at once, with any remaining to be scrolled by one at a time.

O9 display output summary: upon exhausting all data elements on the output list, the system will display a summary of all guidelines on the output list, regardless of whether accessed or not.

O10 maximum response times for output phase:

10 seconds to decode KB access parameters
10 seconds to generate overview output display
1 second to display next guideline/criteria
3 seconds to access next frame
5 seconds to access next database

3.1.2 HF-ROBOTEX Design Structure

After much deliberation, two State-of-the-Art software subsystems were selected to fully and efficiently meet the Expert System requirements delineated in Section 3.1.1. The subsystems provide sufficient memory capacity and processing speed to implement an Expert System to Apply Human Factors to Robotics. Two are established, expandable in all dimensions, modular in construction, and readily link to one another. PSI software specialists will modify all interface software to refine an appropriate query system, to adapt specific aspects of Insight 2+, and to ensure efficient interlinking. PSI developed software will ensure these subsystems provide a complete, cost-effective microcomputer-based Expert System. The two programs which form the subsystem structure are:

- INSIGHT 2+ (used as an Inference Engine (IE) for the Search Phase)

- dBASE III (used as a Knowledge Base (KB) for the Output Phase)

Figure 3-1 is a system block diagram showing a comprehensive overview of the basic HF-ROBOTEX architecture as a system block diagram. The diagram also shows how HF-ROBOTEX can be updated by inputs from the RD engineer and HF expert via the the knowledge engineer (KE).

The two major portions of the system, the Search Phase (using Insight 2+) and the Output Phase (using dBASE III), are demarcated by dotted lines. Each of these phases comprises four major blocks which represent the individual subprograms, or modules, required to perform the search and output functions. Each of these modules has large and/or small arrows going into and out of it, which represent the main flow of data during a search (large arrows) or during data entry (small arrows). The type of data flowing between the modules is identified by the label on each arrow.

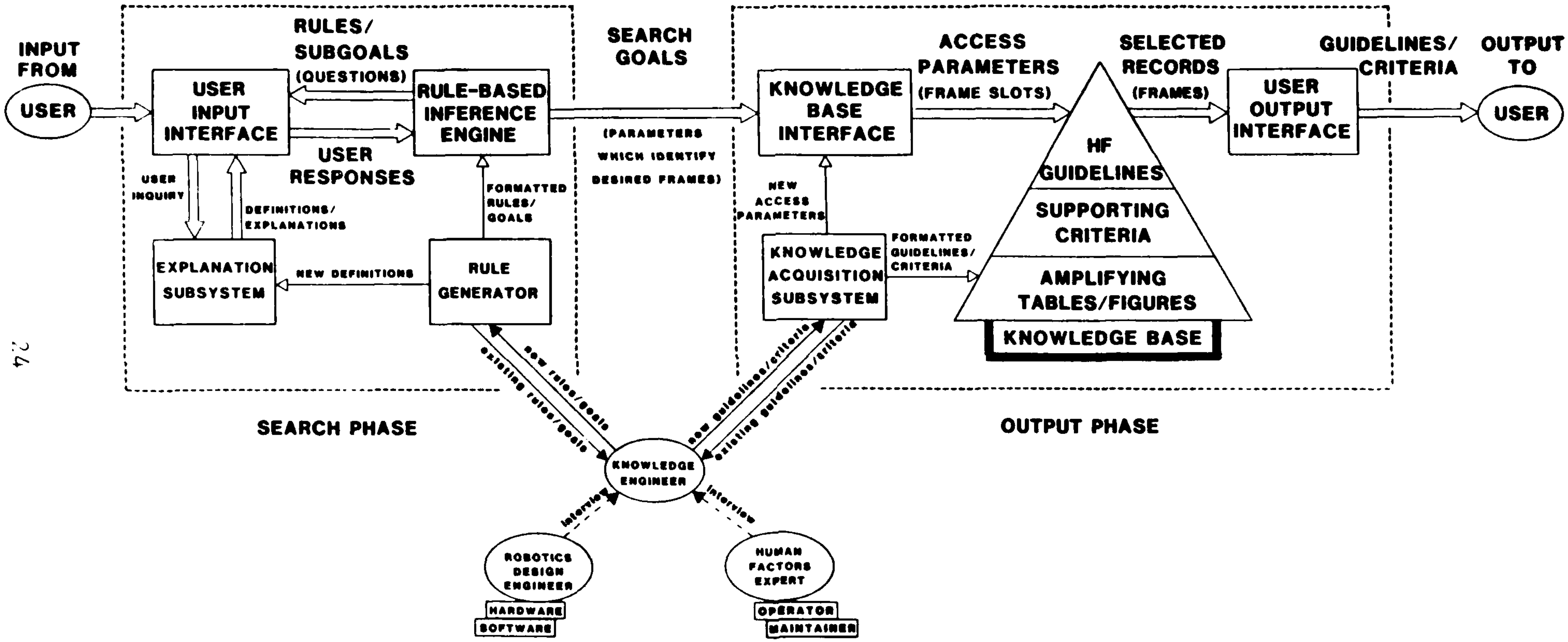


FIGURE 3-1. SYSTEM BLOCK DIAGRAM

The primary function of the system is to allow the user to conduct an expert search of the HF knowledge base. To do this, the user must:

> first, under a QUERY operating mode, formulate a query as to his specific RD problem, by interacting with the inference engine (IE) of the Expert System; and
>
> second, under a GUIDELINE operating mode, display the resulting guidelines/criteria by interacting with the knowledge base (KB) in which they are stored.

The QUERY mode is under control, with the sequence of "rules" fired being dictated by the IE in response to the user. The GUIDELINE mode is also under user control, with the sequence of "frames" displayed being dictated by the KB also in response to the user.

This primary function is enabled online by the six major modules shown generally in the upper portion of Figure 3-1:

SEARCH PHASE	User Input Interface (UII) Rule-Based Inference Engine (IE) Explanation Subsystem (ES)
OUTPUT PHASE	Knowledge Base Interface (KBI) Frame-Based Knowledge Base (KB) User Output Interface (UOI)

The secondary function of the system is to allow the KE, under a "data entry" operating mode, to enter data into the system as new rules/goals or new guidelines/criteria. For this function, the main flow of data is generally vertical from bottom (input from KE) to top (insertion into the IE or KB) along the small arrows. This secondary function is enabled offline by the remaining two major modules shown in the center of Figure 3-1:

SEARCH PHASE	Rule Generator (RG) (in conjunction with the IE and ES)
OUTPUT PHASE	Knowledge Acquisition Subsystem (KAS) (in conjunction with the KB and KBI)

3.1.2.1 Inference Engine Structure. The Inference Engine (IE) is the first of two central modules in the RES architecture. As discussed in the AI description of paragraph 1.3, the IE comprises a strategically structured series of "rules" in the place of "links" in the semantic network of Figure 1-3. These rules are structured as sets of rules (conditions) on several levels, each having its own set of corresponding "goals" (conclusions).

By virtue of this multilevel rule-goal structure, the IE essentially embodies the expert knowledge of robotics design (RD) in such a systematic manner that it can be accessed by an RD or HF engineer. By carefully articulating the "object" (equipment and task) and its most salient "attributes" (components and factors), the user can help the IE pinpoint the most pertinent "values" (HF guidelines) that can be applied to his problem (see Figure 1-4). As a general rule, the more specifically the object and attributes are articulated to the IE, the more focused and relevant the resulting values will be.

Figure 3-2 is an overview of the general structure of the IE. The IE comprises four major object/attribute classes (equipment, tasks, components, factors), each of which in turn, comprises up to 12 *categories* of RD-related considerations. The rule-based IE is structured such that there is a set of rules for each major class that enables the IE to narrow the user's search query down to the fewest possible categories in each class. This reduces the volume of goals (frames) that result from the search and, at the same time, serve to increase the pertinence of the resulting HF guidelines (contained in the frames) to the original RD problem. For convenience, Figure 3-2 shows the range of equipment/task categories dedicated to "operate" and "maintain"; beyond this simple arrangement, there is no particular significance to the order of the categories in any of the classes.

	EQUIPMENT		TASKS	COMPONENTS	FACTORS
1	SENSORS		ASSEMBLE/UNPACK	CONTROLS	LOCATION
2	ELECTRO-OPTICAL		CONFIGURE	DISPLAYS	SIZE/SHAPE
3	COMMAND/INFO		ENTER/EXIT	LABELS/MARKINGS	DIRECTION/FORCE
4	COMMUNICATIONS	OPERATE	CHECKOUT/VERIFY	WORKSPACE	INFORMATION
5	MATERIAL HANDLERS		ACTIVATE/CONTROL	CABLES/HOSES	VISIBILITY
6	SUPPLY/STORAGE		MONITOR/ACQUIRE	CONNECTORS	USE CONDITIONS
7	MAINT. EQUIPMENT		INSPECT/CHECKOUT	OPTICS	SAFETY
8	ELECTRICAL		SERVICE	COMM. EQUIPMENT	
9	MECHANICAL		ADJUST/ALIGN	ACCESSES	
10	HYDRAULIC	MAINTAIN	TROUBLESHOOT	EXTERNAL	
11			REPAIR/REPLACE	EXPENDABLES	
12			TEST/CALIBRATE	TEST ELEMENTS	

FIGURE 3-2. INFERENCE ENGINE OVERVIEW

Figure 3-3 illustrates how the rule-based IE is structured. There are three system levels which correspond to object-attribute-value (O-A-V) triplets:

OBJECT — On system level 1, the equipment and tasks desired by the user are used to define the object of the search.

ATTRIBUTES — On system level 2, the components pertinent to the defined object and human factors desired by the user are then used to define the attributes of the given object.

VALUES — And finally, on system level 3, the given object-attribute configuration(s) are used, in turn, to derive resulting values representing the desired set of guideline frames to be accessed in the KB.

Each of the major areas (e.g., equipment) has its own hierarchy of rules (1, ..., n) which the IE fires, based on successive responses from the user, until the area cannot be narrowed down any further. The IE then steps to the next area (e.g., tasks) and proceeds in a similar fashion through its hierarchy of rules. Once all of the areas at the same level have been addressed (e.g., level 1), the IE then moves to the next level and repeats the same process.

This IE strategy shown in Figure 3-3 is regarded as "breadth-first" search because the IE exhausts all possibilities from left to right at each level (and, within that, at each sublevel) before proceeding on to the next level. This search is also regarded as "forward-chained" in that the IE proceeds forward from hypotheses (the IF part of the rule) to conclusions (the THEN part of the rule), rather than going backwards through the hierarchy from each goal seeking to find a match on all of its conditions.

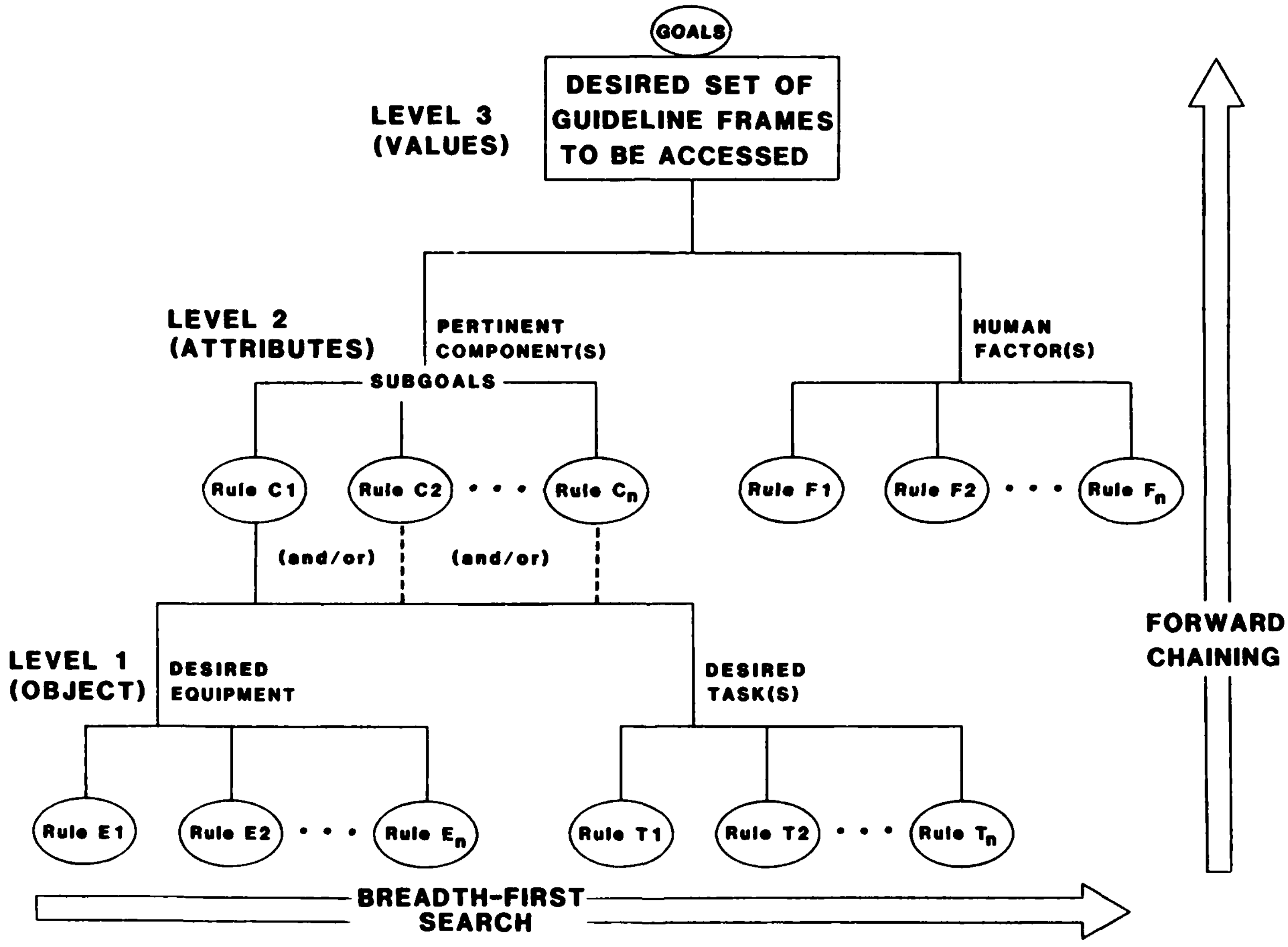


FIGURE 3-3. RULE-BASED INFERENCE ENGINE

3.1.2.2 Knowledge Base Structure. The Knowledge Base (KB) is the second of two central modules that dominate the HF-ROBOTEX architecture. The KB comprises a strategically structured hierarchy of "frames" (guidelines) in the place of "values" in the O-A-V triplets of Figure 1-2. These frames have "slots" which point, in turn, to lower levels of more generic frames (criteria and tables/figures).

By virtue of this structured hierarchy, the KB essentially embodies the expert knowledge of Human Factors such that it can be accessed by an RD or HF engineer who is not an expert in either field. By requesting more detailed data from the KB, the user can trace any given HF guideline to its supporting criteria and, in turn, to the tables/figures from which the criteria was synthesized.

Figure 3-4 is an overview of the general structure of the KB. The KB comprises three major classes of HF knowledge (guidelines, criteria, and tables/figures) which are associated with respective data levels 1, 2, 3, on the left of Figure 3-4.

GUIDELINES At data level 1, the HF guidelines have been broken down into 10 databases (sensor, optical, ..., hydraulic) to help pinpoint the appropriate HF guidelines for the exact RD equipment originally specified.

CRITERIA As just stated, these guidelines have been synthesized from a wealth of generic supporting criteria which appears at data level 2.

TABLES/ FIGURES These criteria have been synthesized, in turn, from amplifying tables/figures containing specific human measurements and data points, which have been compiled and offloaded into an external reference manual at data level 3.

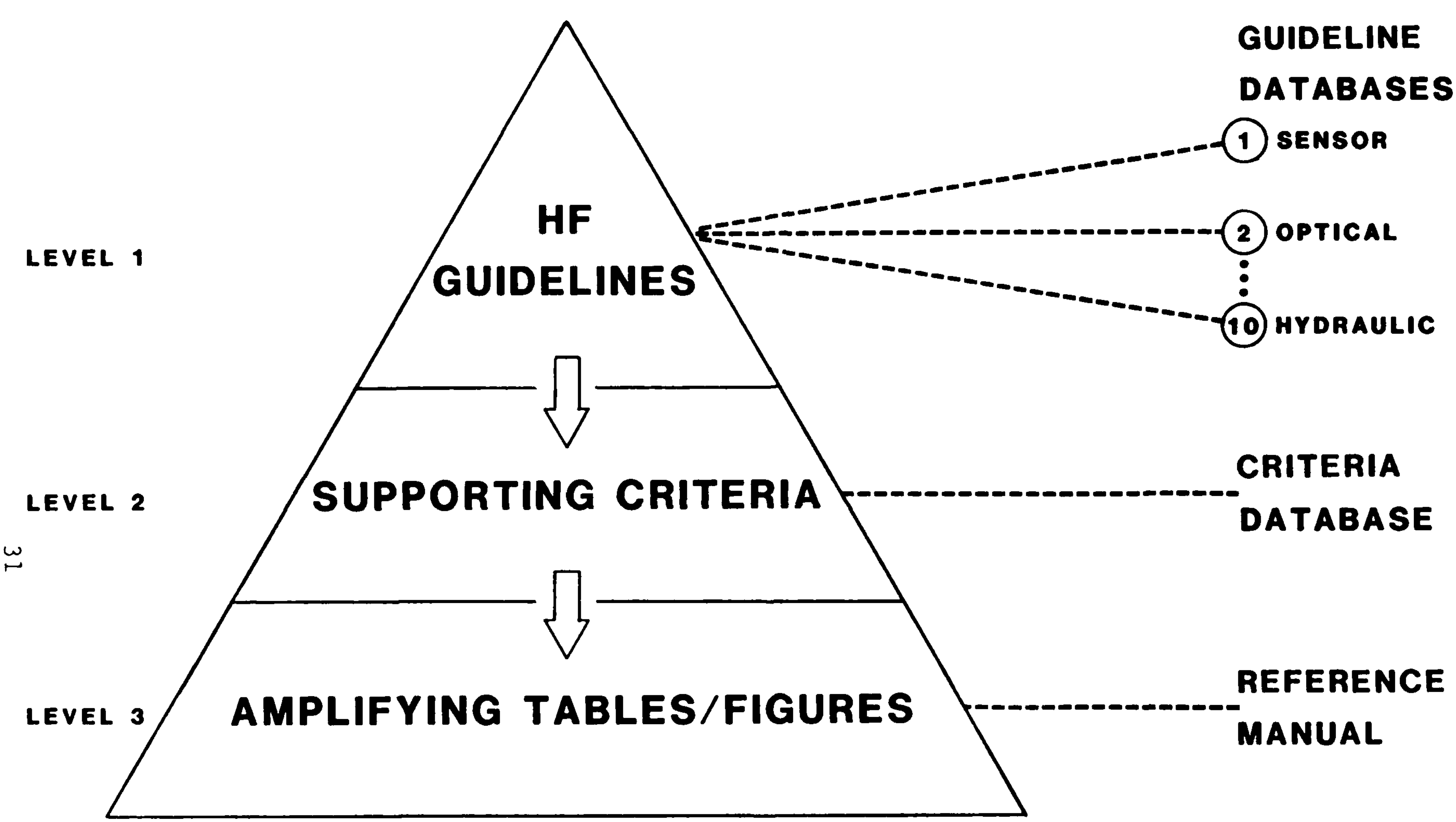


FIGURE 3-4. KNOWLEDGE BASE OVERVIEW

Figure 3-5 illustrates how the frame-based KB is structured, starting with overall database structure, proceeding within that to generalized frame structure, and, within that, to individual record structure. The knowledge databases shown at the top comprise HF guidelines which represent all of the "values" that can possibly be referenced as goals produced by the IE, as explained in the preceeding section. There are 10 guideline databases (sensors, optical, ..., hydraulic) of which 7 are operational-oriented and 3 are maintenance-oriented, followed by a single criteria database. These guideline databases will use presently structured and validated data for the initial KB construction, but will be adopted for each particular application. The criteria database has been included at the top of this chart simply because it observes the same frame and record structure as the guideline databases.

As shown in Figure 3-5, the typical KB database comprises a frame for each XY combination of equipment components along the X axis (controls, displays, ..., test elements) and human factors along the Y axis (location, size/shape, ..., safety). The typical frame "XY", which is the elementary KB unit referenced by the IE, comprises from 1 to 10 guidelines, with an average of 6 per frame. The typical guideline, which is the elementary KB record for the entire system, comprises 3 fields for the unique guideline number, 6 fields for linkage to its supporting criteria, and 1 field for the guideline itself. For moe detail on the record format, see Table 3-8 of paragraph 3.3.

As shown at the bottom of Figure 3-5, the guideline linkage serves as a "pointer" to specific criteria records (or even several record sequences) within the criteria databases shown above. Similarly, the criteria linkage points to specific tables/figures (or sequences of either one) which can be found in an external reference manual.

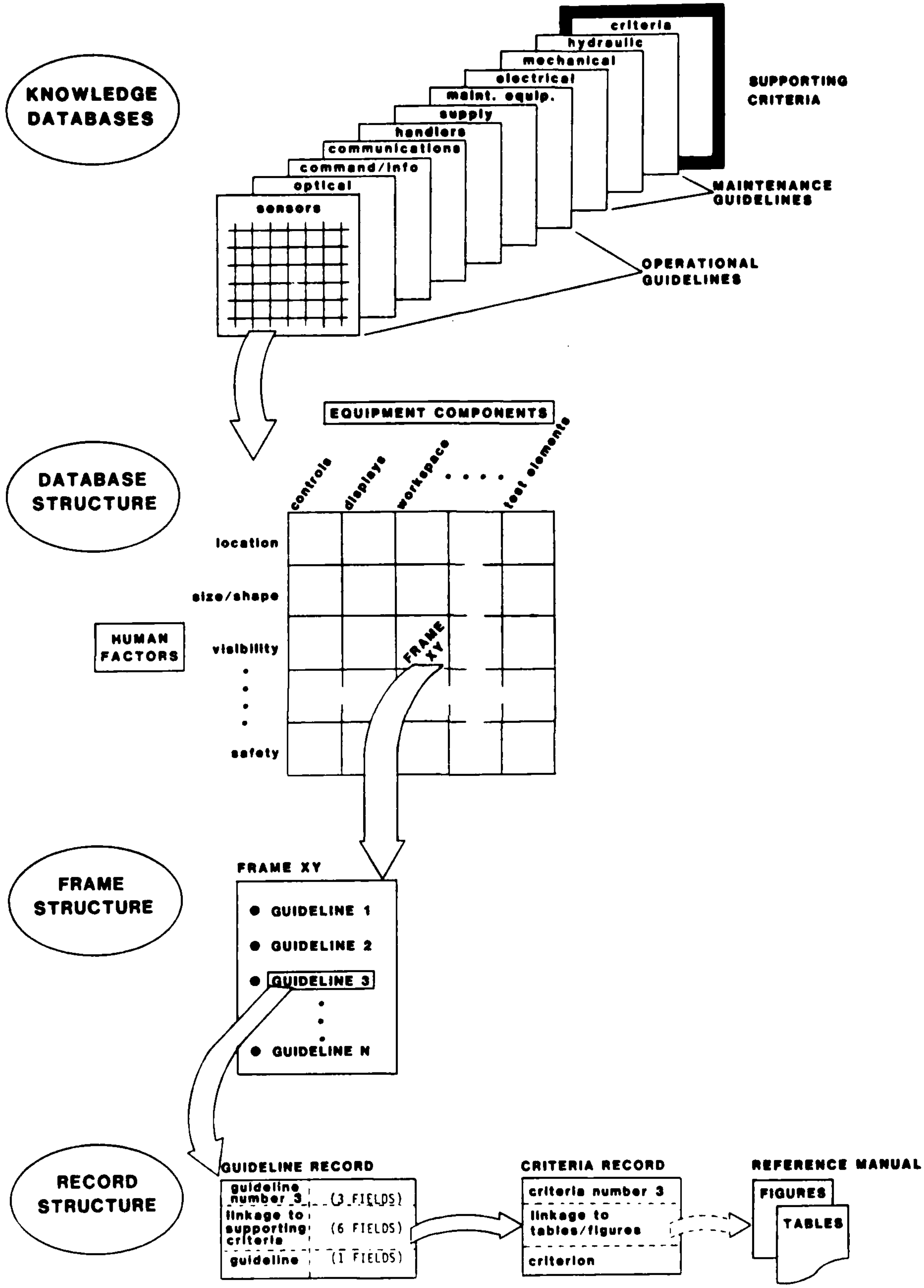


FIGURE 3-5. FRAME-BASED KNOWLEDGE BASE

3.1.3 Function Allocation to System Modules

To enable these primary and secondary system functions, the eight major modules of Figure 3-1 must perform a number of I/O and processing functions. Table 3-2 summarizes the allocation of generic functions to these modules, such as "monitor" and "display"; the specific functions will be discussed in detail in the next paragraph 3.2. as a functional description of the entire system. To avoid redundant description there, the generic scope and content of these functions across the system are briefly outlined below:

DISPLAY: Includes registering newly-provided data on the display screen in the appropriate screen format for the current system mode, ranging from a single new line (e.g., showing the next question immediately beneath the last answer) to a completely fresh screen (e.g., the next set of HF guidelines responsive to a request for the next output frame).

MONITOR: Includes polling inputs from the user via the keyboard, particularly the following:

- Function Keys indicating change to a new mode
- Alphanumeric Keys indicating text response to a question
- Cursor Position indicating choice of several options on the display

ACCESS: Includes activating another module, passing parameters to control its operation (such as a "yes/no/don't know" response to the last question posed by that module), and accepting the output of that operation in return (such as the next question dictated by the last response).

SEARCH: Includes scanning through a data index, directory, or other access control mechanism to find all matches in a database to a given request for data (e.g., by matching the access parameters passed to the module with the attributes or "conditions" attached to the stored rules, frames, etc.).

TABLE 3-2. ALLOCATION OF GENERIC FUNCTIONS

A. Insight 2+ — NUMBER OF FUNCTIONS

1. User Interface
   - display (4)
   - monitor (2)
   - access (3)
   - encode (1)

2. Rule Generator
   - display (2)
   - monitor (2)
   - access (1)
   - store (2)
   - retrieve (1)

3. Explanation Subsystem — retrieve (3)

4. Inference Engine
   - search (2)
   - encode (1)

B. dBASE III

1. User Interface
   - display (4)
   - monitor (2)
   - access (3)
   - decode (1)

2. Knowledge Acquisition
   - display (2)
   - monitor (2)
   - access (1)
   - store (2)
   - retrieve (1)

3. Knowledge Base Interface
   - decode (1)
   - store (1)
   - retrieve (1)
   - access (1)

4. Knowledge Base — search (2)

SUMMARY:

TYPE OF FUNCTION	Insight	dBase	Total
display	6	6	12
monitor	4	4	8
access	4	4	8
search	2	2	4
store	2	3	5
retrieve	4	2	6
encode	2	0	2
decode	0	2	2

system total 24 + 23 = 47 functions

STORE/RETRIEVE: includes the storage or retrieval of specific data within a currently active data base in memory (for KB requests, the data is stored/retrieved as frames, and the database may be overlayed between memory and disk).

ENCODE: includes the process of translating text-type responses from the user (e.g., a search query formulated as "sensors" for equipment, "labels" for component, and "location" for factor) into a coded set of symbols that can be used to access a database (e.g., two bits set "on" within two different KB access parameters uniquely indicating the frame for "labels" and "location" within the "sensors" database).

DECODE: includes the process of translating encoded symbols which request access to a database (such as the KB access parameters mentioned under ENCODE) into internal memory variables which can be used to retrieve specific data responsive to that request (e.g., the "labels/location" frame of HF guidelines within the "sensors" database).

Table 3-3 delineates the specific I/O and processing functions required of the eight major modules of Figure 3-1 to enable the primary and secondary system functions. This table summarizes the allocation of specific functions to each module, such as "monitor keyboard inputs" and "display equipment rules", and cross-references the functions to the performance requirements delineated in paragraph 3.1.1. The nature and scope of the functions in Table 3-3 will be discussed in depth, along with their specific inputs, outputs, and intrinsic functions, in the next paragraph 3.2.

TABLE 3-3. ALLOCATION OF SPECIFIC FUNCTIONS

CROSS-REFERENCE TO SEARCH PERFORMANCE REQUIREMENTS	A. Search Phase Modules (Insight II Plus)
	1. User Input Interface (UII)
S1/S10	a. display search overview (including procedures)
S2	b. monitor function inputs (for mode changes)
S3	c. access rule generator (if requested)
S4	d. access explanation subsystem (if requested)
S5	e. display definitions/explanatons/messages
S6	f. monitor keyboard inputs (for query formulation)
S7	g. access inference engine
S8/S10	h. display inference rules
S8	i. display resulting goals
S9/S10	j. encode goals (for output phase)
S3	2. Rule Generator (RG)
	a. display rule format
	b. monitor function inputs (for mode changes)
	c. access inference engine (optional inquiry)
	d. display rules/goals
	e. store/retrieve definitions
	f. monitor keyboard inputs (for rule entry)
	g. store rule/goal
S4	3. Explanation Subsystem (ES)
	a. retrieve definitions
	b. retrieve explanations
	c. retrieve help messages
S7/S10	4. Inference Engine (IE)
	a. search equipment/task rules
	b. search component/factor rules
S9/S10	c. encode goals (as access parameters)

CROSS-REFERENCE TO OUTPUT PERFORMANCE REQUIREMENTS	B. Output Phase Modules (dBase III)
	1. User Output Interface (UOI)
O1/O10	a. display output overview (including procedures)
O2/O10	b. decode goals (from search phase)
O3	c. monitor function inputs (for mode changes)
O4	d. access knowledge acquisition subsystem (if requested)
O5	e. monitor keyboard inputs (for sequence control)
O6	f. access knowledge base interface
O7	g. access knowledge base
O8/O10	h. display guidelines/criteria
O8	i. display table/figure references (if requested)
O9	j. display output summary
O4	2. Knowledge Acquisition Subsystem (KAS)
	a. display record format
	b. monitor function inputs (for mode changes)
	c. access knowledge base (optional inquiry)
	d. display guideline/criteria frames
	e. monitor keyboard inputs (for data entry)
	f. store/retrieve guideline/criteria records
	g. store access parameters
O7	3. Knowledge Base Interface (KBI)
O2	a. decode goals (into access parameters)
	b. store/retrieve access parameters
	c. access knowledge base
O7/O10	4. Knowledge Base (KB)
	a. store/retrieve guideline/criteria frames
	b. search guideline/criteria frames

3.2 FUNCTION DESCRIPTION

This section contains a summary of the specific inputs, outputs, and processing functions associated with each major function to be performed by the HF-ROBOTEX module. Table 3-3 summarized these major functions, and correlated each one with its associated HF-ROBOTEX module and its antecedent performance requirement (S1,...,S10) and (O1,...,O10). This section also identifies the specific data required both as inputs to each function and their sources, and as outputs from that function and their destinations For convenient correlation with Table 3-3, this section has been organized such that it is symmetrically divided between the Search Phase (Insight 2+ modules) under paragraph 3.2.1 and the Output Phase (dBASE III modules) under paragraph 3.2.2.

3.2.1 Search Phase

The search phase is devoted to formulating a user query out of the constructs of the user's RD problem, and conducting a search to find the HF guidelines most pertinent to that problem. The following is a detailed descripton of each search function, with its associated performance requirements (S1, ..., S10) indicated in the left margin.

3.2.1.1 User Input Interface (UII).

MODULE INPUTS:
System activate signal from DOS
Function keys (F1-F7) to change modes
Keyboard/cursor keys to control display
Current goals/next rule from the IE
Definitions/explanations/messages from the ES

MODULE OUTPUTS:
Module activate signals to the RG/ES
Series of display screens for search overview
Display screens for current goals/next rule
User response to each rule to the IE
Display screen for final goals of search phase

MODULE FUNCTIONS: The user input interface (UII) is activated when HF-ROBOTEX is initially booted up. It provides a main menu from which the user can, among other things, learn about the search process from a search overview, enter a RULE ENTRY mode to enter new rules into the IE, or simply proceed directly into a SEARCH QUERY mode where he can formulate his RD query to the system.

In dealing directly with the user and being totally responsive to user command, the UII essentially becomes the executive control routine for all of the other modules in the search phase -- the RG, the ES, and the all-important IE. As such, the UII serves to activate each of these modules (generally in response to user request), to supply them with the necessary operating parameters (such as the current user response to the IE), and to await their subsequent responses (such as the next rule from the IE to the user).

The following is detailed description of the specific inputs, outputs, and intrinsic functions for each of the functions allocated to the UII in Table 3-3 of paragraph 3.1.3:

S1/S10	display search overview (including procedures)
S2	monitor function inputs (for mode changes)
S3	access rule generator (if requested)
S4	access explanation subsystem (if requested)
S5	display definitions/explanations/messages
S6	monitor keyboard inputs (for query formulation)
S7	access inference engine
S8/S10	display inference rules
S8	display resulting goals
S9/S10	encode goals (for output phase)

S1 - DISPLAY SEARCH OVERVIEW

INPUTS: System activate signal from the DOS
Keyboard keys to control display
Function keys (F1-F7) to exercise options

OUTPUTS: Series of overview display screens
Signal to enter RULE ENTRY mode
Signal to enter SEARCH QUERY mode

FUNCTION: Upon activation, the user input interface (UII) will display a series of screens to identify the purpose of the search phase, delineate its features, and explain its options in a completely user-friendly manner. These initial screens will associate the available system features and options with the various user-controlled modes of operation that enable them, including query/explanation modes for the search phase, and the rule entry mode for the update phase (see Section 3.4 for more detail on system modes). The overview will also explain how the user can change modes and invoke the various system options via the function keys and/or the alphanumeric keyboard. Finally, the overview will delineate the procedures for operating the system within each mode. The overview screens will be user-oriented, designed with optimum sequencing, clear-cut language, and comprehensive screen content that is not overwhelming (7 points plus/minus 2).

S10-System startup time must be accomplished within 10 seconds to first overview screen, which shall be the main menu. If the IE becomes too expansive to load within the allotted ten seconds, then it may become necessary to resort to subdividing the IE into segmented overlays. Every effort will be employed to ensure a user-friendly system; for example, color-coded diskettes will be used for program and database backups. Next screen selection will normally be advanced by hitting function F1, with alternate paths available via the remaining function keys (F2-F7) to provide more detail about the various options. The user will be allowed to "escape" the ordained sequence of overview screens at any time by hitting the "ESC" key, which will force the system to revert back to the main menu. The first two choices on the menu will be devoted to entering the QUERY mode to conduct a search, or the RULE ENTRY mode to update the IE, respectively (see description of the MENU function (F5) later in this paragraph).

S2 - MONITOR FUNCTION INPUTS

INPUTS: Function keys to activate functions (F1-F7)
Any keyboard key to reset functions

OUTPUTS: Earlier display screen in QUERY mode (F1,F2)
New display screen in EXPLANATION mode (F3,F4,F6)
Overview display screen with menu (F5,F7)
System exit to pass control back to DOS (F7)

FUNCTION: Once the overview screens have been displayed, the UII will continuously monitor the function keys to detect any mode changes requested by the user. The specific functions enabled will permit the user to have the greatest flexibility in controlling the system with the least amount of prior explanation and/or training. The function keys will be designed on a human factors basis to ensure that functions are clearly and unambiguously marked. The functions themselves will be identified on the bottom of the user screen to promote user recognition of what options the system provides and how to activate them (see the exemplary screen format for the function DISPLAY INFERENCE RULES later in this paragraph).

The following are examples of specific functions that should be made available to the user by simply hitting the function keys (F1, BACKUP; F2, RESTART; F3, STATUS; F4, EXPAND; F5, MENU; F6, HELP; F7, EXIT) at any time during the search phase.

F1-BACKUP: upon user request to backup to the preceeding screen, the system backs up to the last screen presented along whatever path the user is currently pursuing, abandoning whatever interactions have taken place on the current screen. For example, a user realizes he has made a mistake in specifying his query on the current screen, so he backs up to the preceeding screen and picks up from there. Upon reaching the first screen in the first path, any further requests to backup will be ignored.

F2-RESTART: upon user request to restart the query, the system suspends the current screen and generates a new screen asking if, in fact, the user does want to abandon his query before reaching its conclusion (final goals). If so, the user must hit F2 again to confirm his desire to quit the search and start anew. Upon user confirmation, the system aborts the current search and reverts back to the first screen of the QUERY mode. Otherwise, the user may hit any other key (including the other function keys) to revert back to the current screen without disturbing the status of the current mode. For example, a user realizes he is pursuing a non-productive path through the IE semantic "tree" network because he was too specific, so he hits F2 twice to RESTART the entire query (alternatively, he could have BACKed UP to the point in the query where his response was too limited). This safeguard has been inserted to protect the user from pressing the F2 RESTART key by accident, thus wasting the valuable time and effort spent in pursuing the immediate query.

F3-STATUS: upon user request for the status of the current query, the system generates a new screen which designates the current level of user query and delineates the path along which the query has thus far advanced. For example, since RES fires its own rules to determine the most pertinent equipment components as subgoals, the user may reach a set of components which he did not anticipate in his query; hence, the user must be able to request the cumulative search status to recall whether or not he specified the most appropriate equipment/tasks at the outset (e.g., "monitoring" in conjunction with "activating" the robot system). To revert back to the current rule being displayed, the user may hit any alphanumeric key on the keyboard including "ESC" and "RETURN", but excluding the function keys (since they will activate another function instead). If the query has not advanced past the first search level, the request for status will be ignored.

F4-EXPAND: upon user request for expansion of the current rule being displayed, the system access the Explanation Subsystem for the goal structure and definition associated with the current rule (if any). Since rule definitions are only provided initially at the discretion of the knowledge engineer during the update phase, every rule may not have a definition, in which case the system returns a message stating "no definition available". If the definition should overflow the first screen, the user may hit any key (other than the function keys) to advance to the second screen; otherwise, hitting any key will force the system to revert back to the current rule.

F5-MENU: upon user request to see what other system features and options are available to him, the system will revert back to the main menu, which shows all systems features and options provided during the search phase. This request acts to suspend the current user query so that the user can consider other alternatives during the search phase (such as updating the "definition" file) without disturbing the status of the current query. For this screen, the user must use the cursor to step through the various alternatives being displayed. Upon reaching an alternative he wishes to pursue, the user must hit "RETURN" to activate it; otherwise, the user must hit "ESC" to revert back to the current query. The system will ignore any attempt by the user to activate functions F1-F4 while the menu is being displayed, since these functions pertain to the current query which has been suspended; however, the system will honor a request for HELP (F6) or EXIT (F7).

F6-HELP: upon user request for help in understanding the current mode of operation, the system generates an appropriate help message. This message comprises user-friendly narrative describing what mode of operation is in effect (such as update, search, explanation, etc.), what system features and options are related to the current mode (such as provisions for looking at rules/goals previously stored in the IE), what procedures must be followed to operate in the current mode (such as answering a question with "yes/no/don't care" by hitting the "Y/N/RETURN" keys, respectively), and what procedures are required to switch to another mode (such as by hitting function key F4 to expand the current rule into its associated IF...THEN structure and supporting definition). As with function F4, if the help message should overflow the current screen, the user may hit any key (other than function keys) to advance to the next screen; otherwise, hitting any key will force the system to revert back to the screen from which the user requested help. With function F5, the system will ignore any attempt by the user to activate any function other than F7 (EXIT) while the help message is being displayed.

F7-EXIT: upon user request to exit the query, the system acts in the exact same manner as with the RESTART function (F2) above, requiring the user to hit F7 a second time to confirm his desire to quit the current search. In this case, however, the system generates the main menu with an additional message at the bottom to the effect that "...if you wish to exit the program, hit F7 again--otherwise, hit any key". Upon user confirmation of the EXIT request, the system aborts the current search and passes control back to the disk operating system (DOS). As with the RESTART function, the user may hit any other key (including function keys F1-F6) to revert back to the current screen without disturbing the status of the current mode. This safeguard has been inserted to protect the user against striking the F7 EXIT key by accident.

S3 - ACCESS RULE GENERATOR (RG)

INPUTS: Activate signal from the UII
Function keys to activate functions (F1-F7) in RG
Keyboard keys to enter rules/definitions within RG

OUTPUTS: Signal to activate RG in RULE ENTRY mode
New/modified rules for the IE out of RG
New/modified definitions for the ES out of RG

FUNCTION: The RG is activated via user selection on the system's main menu. The RG is essentially an independent, offline interface for the KE to review, insert, modify, or delete rules in the IE. Since it is offline from the normal search phase, its time responses are not critical. Furthermore, its constituent functions (display, monitor, access, etc.) are identical to, or at least very similar to, the functions described above for the UII (for more detail, see paragraph 3.2.1.2). Moreover, growth provisions for the IE are discussed at Paragraph 3.3 in conjunction with the configuration and limits of the present system.

S4 - ACCESS EXPLANATION SUBSYSTEM (ES)

INPUTS: Function keys (F3,F4,F6)

OUTPUTS: Three (3) signals to activate ES to retrieve:
Search Explanation (for F3)
Rule Definition (for F4)
Help Message (for F6)

FUNCTION: The ES is activated via the user input interface (UII) by function keys F3 (STATUS), F4 (EXPAND), and F6 (HELP). These keys respectively represent the functions of each search status (including an explanation of the search path taken to the current level), rule expansion (including a definition of the rule and its background), and help message (including operating procedures for the current mode of operation). Thus, the ES module must have three entry points to differentiate these three types of user requests. Once activated, the ES retrieves the pertinent search explanation, rule definition, or help message corresponding to the current search status, query rule, or operating mode, respectively. The ES then sends its output back to the user interface to be formatted for display. For more detail as to specific ES functions, see paragraph 3.2.1.3).

S5 - DISPLAY DEFINITIONS/EXPLANATIONS/MESSAGES

INPUTS: Definition/explanation/message from the ES
Keyboard keys to control display

OUTPUTS: Three (3) series of display screens formatted for:
Rule Definition (responsive to function F4)
Search Explanation (responsive to function F3)
Help Messages (responsive to function F6)

FUNCTION: The UII receives the current rule definition, search explanation, or help message requested by the user from the ES, and formats it appropriately for display to the user:

The Rule Definition generally comprises one or more paragraphs of text describing the origin of the rule and/or the reason for its position within the semantic tree network of rules. The UII displays the structure of the rule first (the compound IF...THEN clauses) followed by the definition text.

The Search Explanation generally comprises a "trace" of the rules fired by successive user responses, which represents the unique path of the user query through the IE semantic network. The UII displays the path as a series of goals (decisions by the user or the system), starting with the first goal at the top and proceeding to the most recent goal at the bottom.

The Help Message generally comprises one or more paragraphs of text describing the current operating mode, what features and options it incorporates, and what operating procedures are required of the user. The UII displays the help message in the same order, correlating specific operating procedures with each of the system features and options available under the current mode.

Upon activation by one of the function keys, the UII suspends the current screen and displays the retrieved definition, explanation, or message. If the narrative text overflows the first screen, the UII continues the text on the next screen upon the user hitting any key (other than the function keys). In any event, upon reaching the last screen, the user can hit any key to force the UII to revert back to the suspended, current screen. If no definition/explanation/message is available or otherwise warranted (e.g., no definition was ever entered by the KE or no rule has yet been accessed by the user), the UII will display an appropriate default response, rather than ignore the user request.

S6 - MONITOR KEYBOARD/CURSOR INPUTS

INPUTS: Keyboard keys to facilitate user response
Cursor position to facilitate user choice

OUTPUTS: User response to current rule

FUNCTION: The UII polls the keyboard for user responses at all times during the search phase, except during an IE access (This is because each next IE access is set in motion by the last user response and cannot be recalled or altered while in progress). If for any reason the user wants to change his last response (i.e., leading to a different IE access), he can hit function F1 (BACKUP) after the current IE access is completed and proceed forward again through the search path from the last "tree" node. This monitoring strategy further protects against accidental keystrokes by the user while a legitimate IE access is in progess. The specific format for user responses depends on the scope of the response:

For a simple YES/NO-type response, the user should be forced to hit the "Y" or "N" key to guarantee a positive, explicit answer. Likewise, a "D" may be used for any possible "don't know" or "don't care" responses.

Where the user must choose from a set of propositions or statements, the response should be tied to cursor position, allowing the user to manipulate the cursor up and down until arriving at the most appropriate choice and then hitting "RETURN". Provision must be made among the displayed choices for "don't know", "don't care", and "none of the above", as appropriate.

Where the user must provide one or more statements as his response (e.g., during an update to the IE), the response will be regarded as all of the text preceeding the user's RETURN. Provision must be made to test and reject non-responses where a user response is required (such as when entering a new rule into the IE).

Moreover, the cursor and/or numeric keys can be coupled with the function keys to permit the user to choose among multiple choices. After making a function selection such as F5 (MENU) or F6 (HELP), the user can manipulate the cursor and hit "RETURN" upon arriving at his desired choice. Alternatively, the user can hit a numeric key associated with the number of his choice among the multiple choices being displayed.

S7 - ACCESS INFERENCE ENGINE (IE

INPUTS: Activate signal from the UII
User response to control IE search

OUTPUTS: Signal to activate IE in SEARCH mode
Current goals resulting from last IE search
Next rule (if any) resulting from last IE search
Final goals as products of entire IE search

FUNCTION: Upon completion of the user response with "RETURN" or other expected response, the UII accesses the IE. The parameter passed to the IE is the horizontal value of the cursor position, representing the user's selection among the multiple choices being displayed. This value specifies a unique path out of the current node in the semantic "tree" network to the next rule.

To find this path, the IE attempts to "match" the value as another condition precedent among the rules leading out of the current mode. If all conditions are met, the IE "fires" the matched rule to obtain its goals, which are then returned for display to the user. Otherwise, the IE must return the next set of conditions for the user to choose among. If the user's response is a "don't know" or "don't care" choice, then the IE must examine all possible paths out of the current mode, sequentially returning to the user the next set of conditions down each path.

In any event, if there are no further conditions to examine, then the IE has reached the subgoals for the current search level, and must proceed to the next level. If there are no further levels, then the IE has reached the final goals, and must encode them. For more detail about the IE, refer to paragraph 3.2.1.4.

S8 - DISPLAY INFERENCE RULES

INPUTS: Current goals resulting from last IE search
Next rule to be fired (if any) from last IE search
Keyboard/cursor keys for display control

OUTPUTS: Display screen formatting next rule as:
Conditions met thus far in the search
Conditions to be queried at next search level
Cursor positioned at first condition to be queried

FUNCTION: Upon receiving the next set of conditions from the IE, the UII formats the "conditions met" thus far in the search at the top of the screen, followed sequentially by the set of conditions to be queried from the user at the bottom (see Figure 3-6). The system must reserve sufficient display area at the bottom for at least five conditions for the user to choose among; beyond this, the system will display as many "conditions met" as possible in the space that remains. If there are more than five conditions and the screen becomes saturated, the system must display a "rule continued" message at the bottom, requesting the user to "scroll up" each subsequent condition one at a time by hitting RETURN. The message is discontinued with display of the last condition, and further attempts to hit RETURN are ignored.

S10-This display function represents the logical conclusion of the 4-second IE cycle for firing a given inference rule. Since the UII has only three seconds to generate this display, it may become necessary to store the conditions met with each successive cycle in a cumulative memory array for iterative display. This takes advantage of the time already spent in each prior 4-second display cycle to identify these conditions. After generating the display, the UII merely "idles" until the user responds via the keyboard (see the provisions above for the system to MONITOR KEYBOARD/CURSOR INPUTS to meet requirement S6).

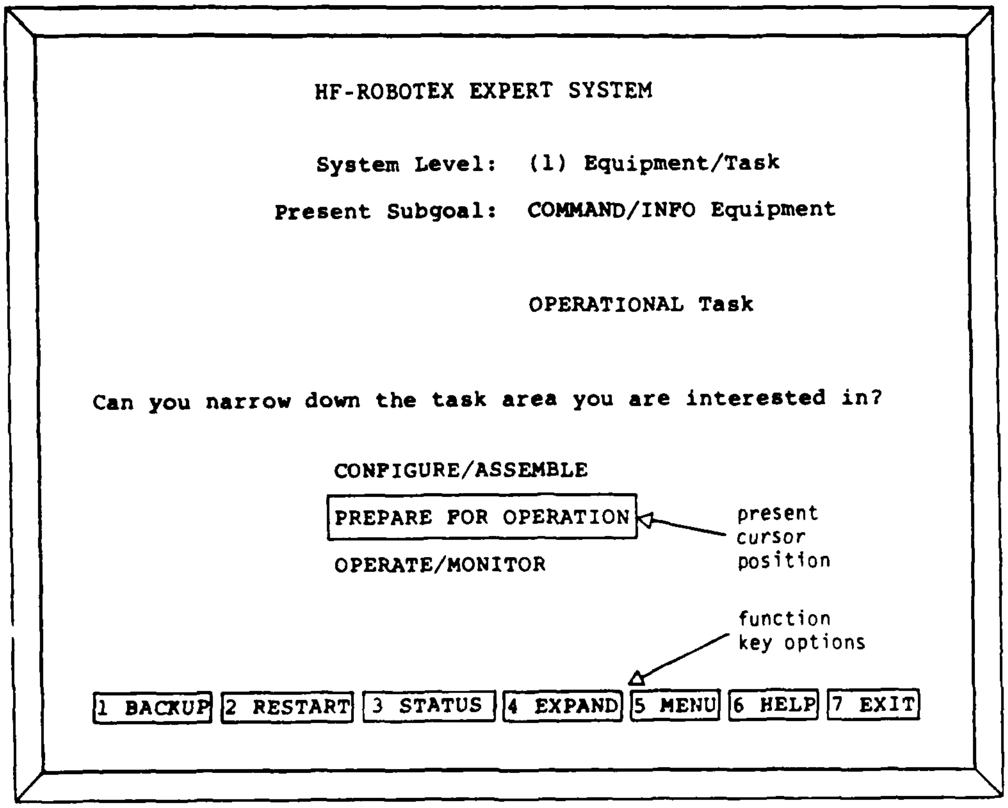


FIGURE 3-6. EXAMPLE OF A USER INPUT SCREEN

S8 - DISPLAY RESULTING GOALS
(special case for requirement S8)

INPUTS: Final goals as products of entire IE search
Keyboard/cursor keys for display control

OUTPUTS: Display screen formatting final goals as:
Cumulative conditions met during IE search
Resulting goals as access parameters to KB

FUNCTION: Upon receiving the final set of goals from the IE, the UII formats the "conditions met" for each level throughout the search at the top of the screen, culminating in the final goals for the entire search at the bottom. These goals indicate that all rules at all levels have been exhausted, and that one or more paths have been successfully established through the semantic "tree" network. The display concludes with an instruction to the user on how to activate the output phase, if desired.

This display is similar in format and operation to the DISPLAY INFERENCE RULES function, except that the goals are indicated to be the final products of the search phase. These O-A-V goals are identified by their object name (e.g., COMMAND/INFO equipment), specific attributes (e.g., LABELS as the component and SIZE/SHAPE as the human factor), and associated values (e.g., the pertinent HF guidelines in the KB). However, to retrieve these pertinent HF guidelines from the massive KB, the goals must next be encoded by the IE as KB access parameters (see the next following ENCODE GOALS function for requirement S9).

S9 - ENCODE GOALS

INPUTS: Final goals as products of entire IE search

OUTPUTS: Resulting goals encoded as access parameters to KB
Signal to activate output phase

FUNCTION: This is the last operatonal function in the search phase, performed independently by the inference engine (IE) upon exhausting all levels of rules in the semantic "tree" network. The final goals of the IE search, which are actually pertinent frames of HF guidelines within the KB, are encoded into parameters to access those frames in the KB. The need for this function arises from several functional constraints imposed on RES by Insight II Plus (e.g., the number/depth of rules allowed, number/size of parameters to be passed, etc.), which force a systematic encode/decode to communicate between Insight II and dBASE III.

S10-This function (S9) must be initiated just as soon as the final goals are reached by the IE. However, in the limited time available (10 seconds) to encode as many as 40 access parameters (upper system limit), it may become necessary to "multiplex" the encoding across the many 3-second subgoal time slots that IE will have at its disposal in a complex "multi-path" search. This takes advantage of the fact that IE will know when a given "subgoal" is actually the "final goal" along one of the many paths to the bottom of the "tree" network.

The specific encoding algorithms and techniques used for this function are strictly matters of program design choice. However, for illustration here, a viable example is set forth in detail in the description of the IE at paragraph 3.2.1.4.

3.2.1.2 Rule Generator (RG).

MODULE INPUTS: Signal from UII to enter RULE ENTRY mode
Function keys (F1-F7) to change mode
Keyboard/cursor keys to enter rules/definitions
Previously-entered rules/definitions

MODULE OUTPUTS: Signal to activate IE in the INSERTION mode
Request for previously-entered rules/definitions
Display of previously-entered rules/definitions
New/modified rules for the IE
New/modified definitions for the ES

MODULE FUNCTIONS: As one of the four principal search modules, the rule generator (RG) completely accomodates performance requirement S3. The RG module is relatively autonomous since it is essentially an offline interface for the KE to selectively update the IE, in the same way that the UII is an online interface for the user to selectively search the IE. Figure 3-7 (referred to hereafter as the "IE screen") shows an example of a screen format for IE rules that might be presented to the KE for reviewing or modifying old rules already stored in the IE, or for entering entirely new rules therein.

Although it is initially activated by the user via the main menu in the UII (see the DISPLAY SEARCH OVERVIEW function (S1) in paragraph 3.2.1.1), the RG performs its updating tasks totally independent of the UII and without any further communication between the two. Thus, as an independent interface, the RG operates function-by-function in much the same way as the UII, without the stringent operational time constraints that have been imposed on the UII to accommodate the day-to-day needs of the user (see the UII functions with maximum time requirements (S10) in paragraph 3.2.1.1).

The following is a description of each of the functions performed by the RG, cross-referenced back to the analogous, parallel functions in the UII. To avoid redundancy, the description here will address only those portions of each RG function that are different from, or in addition to, the parallel UII functions:

S3 display rule format
monitor function inputs (for mode changes)
access inference engine (optional inqiury)
display rules/goals
store/retrieve definitions
monitor keyboard/cursor inputs (for rule entry)
store rule/goal

```
TITLE   HF-ROBOTEX EXPERT SYSTEM

THRESHOLD = 100 (certainty factor)

GOALS (partial list of two goals fired by two exemplary rules)

  1. The COMPONENTS are WORKSPACE and ACCESSES
  2. The COMPONENTS are CONTROLS, DISPLAYS, and LABELS
  •
  •
  •
RULE  For COMPONENTS are WORKSPACE and ACCESSES
  IF  The EQUIPMENT is ELECTRO-OPTICAL
 AND  The TASK is OPERATIONAL
 AND  The TASK is  PREPARE FOR OPERATION
 AND  The TASK is ENTER/EXIT STATION
THEN  The COMPONENT is WORKSPACE
 AND  The COMPONENT is ACCESSES
  •
  •
  •
RULE  For COMPONENTS are CONTROLS, DISPLAY, and LABELS
  IF  The EQUIPMENT is ELECTRO-OPTICAL
 AND  The TASK is OPERATIONAL
 AND  The TASK is PREPARE FOR OPERATION
 AND  The TASK is CHECKOUT/VERIFY READINESS
THEN  The COMPONENT is CONTROLS
 AND  The COMPONENT is DISPLAYS
 AND  The COMPONENT is LABELS
  •
  •
  •
  •
 END (statement following last rule entered)
  !   RULE DEFINITION: to prepare ELECTRO-OPTICAL equipment
  !   for operation requires CONTROLS with proper LABELS.
  !   To check out and/or verify equipment readiness requires
  !   DISPLAYS with PROPER labels.
  •
  •
  •
  !   (defintion may continue with "!" comment lines, as needed)


[F1 TOP] [F2 BOTTOM] [F3 SAVE] [F4 COMPILE] [F5 COPY] [F6 HELP] [F7 EXIT]
```

FIGURE 3-7. EXAMPLE OF AN IE SCREEN FOR RULE ENTRY

S3 - DISPLAY RULE FORMAT
(no parallel function in UII)

INPUTS: Function keys (F1,F2)

OUTPUTS: Blank rule format for data entry by KE
Cursor positioned at data slot requested by user

FUNCTION: Upon activation by the UII, the RG enters the RULE DATA mode and presents a blank rule format, or "template", for the KE to start entering data, if he wishes. The IE screen of Figure 3-7 shows an exemplary screen format for rule entry, but at this initial point, only the first rule in the IE data file would appear. The cursor is initially positioned by the RG at the first line of the IE data file (e.g., at "TITLE" on the IE screen) to allow the KE to declare his intentions.

At this juncture, the KE has total discretion to manipulate the RG to suit his updating needs. He must decide whether he wants to establish a new rule for the IE, review the existing IE data sequentially, rule-by-rule, or modify a specific rule of his choosing:

if the KE wishes to review the existing IE rules, he merely hits function key F1 to go to the top of the IE data file, and then advances the cursor sequentially to scroll through the successive rules stored therein.

if the KE wishes to modify a specific rule in the IE he merely stops "scrolling" the cursor at the desired IE rule, and then types in the modified data at the appropriate line; or,

if the KE wishes to establish a new rule for the IE, he must hit function key F2 to go to the bottom of the data file, and then begin ENTRY.

In any event, once the KE has declared his intention with the first rule on display, the cursor is repositioned by the RG at the first data slot (i.e., at the line labeled "RULE" on the IE screen). From this point on, the KE can selectively enter new rules in the slot, modify old rules already there, or simply skip to the next rule slot, as he wishes. At any time, he can advance to the top or bottom of the data file via F1/F2.

S3 - MONITOR FUNCTION INPUTS
(similar to UII function S2)

This function is identical to the parallel UII function at function keys F6-F7 which permit the user to "escape" to help messages or back to the main menu. The remaining functions F1-F5 are devoted to system-level data file operations, as indicated to the KE at the bottom of the IE screen (see Figure 3-7):

F1 TOP	(return to top of file)
F2 BOTTOM	(advance to bottom of file)
F3 SAVE	(save file on disk)
F4 COMPILE	(compile file saved on disk)
F5 COPY	(copy a block of rules to new area)
F6 HELP	(same as UII function)
F7 EXIT	(same as UII funciton)

The following is a brief description of each of the data file operations initiated by F1-F5, with a cross-reference wherever applicable to the equivalent control function in WORDSTAR on which the RG is based (see DISPLAY RULES/GOALS function later in this paragraph):

F1-TOP: At any point in the progression through the IE rules, the KE can immediately return to the TOP of the file via F1. The cursor will return to the same initial start position as it did for the preceding function DISPLAY RULE FORMAT. (F1 is equivalent to (CTL-QR) in WORDSTAR).

F2-BOTTOM: At any point in the progression through the IE rules, the KE can immediately advance to the BOTTOM of the file via F2. The cursor will be positioned at the last line of the file to allow the KE, for example, to enter new rules. (F2 is equivalent to (CTL-QC) in WORDSTAR.)

F3-SAVE: At any point in the process of entering new or modified rules, the KE can SAVE the currently updated file on disk as a ".PRL" file. This function should be exercised periodically to avoid loss of data due to unforseen hazards such as catastrophic power failure. The compiler seeks out this ".PRL" file when the next COMPILE function is exercised, and the ES seeks out this uncompiled ".PRL" file upon each user request for a rule definition (see ES description at paragraph 3.2.1.3). (F3 is equivalent to (CTL-KS) in WORDSTAR.)

F4-COMPILE: Upon completion of entering all new or modified data, the KE must COMPILE the currently updated ".PRL" file on disk to create a PASCAL-compiled data file for subsequent IE execution. Compiling the updated file of IE rules with F4 permits the IE to run 20-50 times faster than if the IE were forced to "interpret" each rule in its uncompiled, text format.

F5-COPY: At any point in the process of updating the file, the KE can COPY a whole block of data from one area in the file to another. This function is extremely useful where many rules are virtually identical except for a few parameters, permitting the KE to COPY the first-entered rules, for example, at the BOTTOM of the file where he can next modify the few parameters which were difficult. (F5 is equivalent to (CTL-KC) in WORDSTAR, preceded by (CTL-KB) and (CTL-KK) appropriately positioned to mark the beginning and end of block respectively.)

F6-HELP: At any time the KE can request HELP for the current mode of operation via F6. This is a particularly usefull if he needs help with the operating procedures for special features, such as scrolling backwards through the file or placing "block" markers to COPY one area to another (for more description, see the parallel UII function at paragraph 3.2.1.1).

F7-EXIT: At any time, the KE can EXIT from the UPDATE phase via F7. As a user safeguard, he will first return to the main menu, which allows him the chance to re-establish the UPDATE phase if F7 key was hit accidentally (for more description, see the parallel UII function at paragraph 3.2.1.1).

S3 - ACCESS INFERENCE ENGINE
(identical to UII function S7)

This RG function is provided to allow the KE to selectively review data already in the IE, at his total discretion.

S3 - DISPLAY RULES/GOALS
(similar to UII function S8)

This RG function is similar functionally to the parallel UII function, except for the fact that the RG displays each rule sequentially (until directed otherwise) in the full-screen format of the IE screen (including comments and definitions) as shown in Figure 3-7. Since the RG of Insight 2+ is based, as much as possible on WORDSTAR editing commands, the display of NEXT RULE is simplified down to a choice of highly-efficient WORDSTAR cursor control commands that involve use of the control key (CTL) on the keyboard, such as (CTL-QZ) to scroll text up continuously, line-by-line, and (CTL-QQC) to scroll text up continuously, screen-by-screen.

S3 - STORE/RETRIEVE DEFINITIONS
(similar to UII function S4)

This RG function is similar to the parallel UII function S4 which activates the ES via the EXPAND function key F4. Upon activation, the ES retrieves the requested rule definition from the IE rule file stored on disk and displays it in the format of the IE screen (see description of the ES at paragraph 3.2.1.3). With this format, the user or KE can quickly and conveniently discriminate the rule definition from the body of the rule. Thus, rule definitions are entered into the RG via the keyboard immediately following each rule, and are subsequently stored therewith by the RG SAVE function F3. Thereafter, the KE can retrieve and review them via the preceding DISPLAY RULES/GOALS function.

S3 - MONITOR KEYBOARD/CURSOR INPUTS
(similar to UII function S6)

This RG function is provided tro allow the KE to selectively control the cursor position up and down the file as desired, and, otherwise, to enter text-type data as new or modified rules/goals (see exemplary rules/goals shown on the IE screen). To enter a new rule, the KE must first enter the goals for the rule at the end of the GOALS list at the front of the file (shown at the otp of the IE screen). The KE then skips to the botton of the file via function F2 and enters the rule, as shown in the center of the IE screen. The KE must then enter a definition for the rule, if appropriate, immediately following the rule, as shown at the bottom of the IE screen. As just indicated for the DISPLAY RULES/GOALS function above, the modification of existing rules is simplified down to a choice of highly efficient WORDSTAR cursor control commands, such as (CTL-QD) to move cursor to end of line, and (CTL-QB) to move cursor back to beginning of block.

S3 - STORE RULE/GOAL
(no parallel function in UII)

This RG function is accomplished by simply hitting the SAVE function key F3, as described above under MONITOR FUNCTION INPUTS. Individual rules/goals may be entered anywhere in the data file by simply inserting text at the desired point, as just described above under MONITOR KEYBOARD/CURSOR INPUTS.

3.2.1.3 Explanation Subsystem (ES).

MODULE INPUTS:	Three (3) signals from the UII to enter EXPLANATION mode for: Search explanation (from function key F3) Rule definition (from function key F4) Help message (from function key F6)
MODULE OUTPUTS:	Search explanation (F3) Rule definition (F4) Help message (F6)

MODULE FUNCTIONS: The ES is activated upon user request by the UII via function keys (F3, F4, F6). The ES has three entry points differentiate these three types of user requests, as follows:

ENTRY POINT	FUNCTION CATEGORY	EXPLANATION SUBSYSTEM INTERACTIVE RESPONSE
A	STATUS F3	retrieves an explanation of the status, including a trace of the search path taken to reach the current search level and/or to reach the available subgoals from the current level
B	EXPAND F4	retrieves an expansion of the current rule being displayed, including its complete structure, its surrounding definition, and its historical background (if available)
C	HELP F6	retrieves a help message for the current operating mode, including specific operating procedures for each system feature available under the current mode

The ES operates interactively with the user by first displaying a "summary" screen of the requested explanation, definition, or message, which will usually by sufficient to satisfy the user. At this point, the user has total discretion to manipulate the ES further in any fashion he pleases. He may choose to continue with the succeeding screens in each category (if any) by simply hitting RETURN, or to enter one of the other categories via the function keys (F3, F4, F6), or to just return to the current search display via the EXIT function key (F7).

The remainder of this paragraph is a detailed description of the specific functions of the ES that are required to enable the associated performance requirement (S4) of paragraph 3.1.3:

S4 retrieve search explanation
S4 retrieve rule definition
S4 retrieve help message

S4 - RETRIEVE SEARCH EXPLANATION

INPUTS: Request for search explanation (entry point A)

OUTPUTS: Explanation summary of current search status
Trace of "backward-chaining" toward root node
Trace of "forward-chaining" to available subgoals

FUNCTION: Conceptually, this function answers the fundamental user inquiry "WHY..?" with a summary of what is "known" thus far (and its source) and what remains "unknown" to the IE. Upon activation at entry point A, the ES generates an explanation summary of the current search status, showing all categories of EQUIPMENT, TASK, COMPONENT, and/or FACTORS that have been formulated thus far in the search query.

At this juncture, the user has the option to EXIT via function key F7, or to continue the explanation by simply hitting RETURN to see a "trace" of the current search path. The ES responds by generating a trace "backward-chained" through the IE rules to the root node at which the search started; and thereafter, a trace of the search path "forward-chained" *through the IE rules* to the available subgoals toward which the search can proceed.

For these optional traces, the ES must access the non-compiled version of IE rules (stored as a ".PRL" file on disk) so that it can present the rules in text format to the user. Neither this ".PRL" file nor the compiled rule file can in any way be modified by such accessing for display purposes only (see paragraph 3.1.2.2 for RG procedures to modify both files).

S4 - RETRIEVE RULE DEFINITION

INPUTS: Request for rule definition (entry point B)

OUTPUTS: Expansion of current rule being displayed

FUNCTION: Upon activation at entry point B, the ES generates an expansion of the current rule on display into its complete"IF...THEN" structure. As noted in paragraph 3.2.1.2 for RG procedures to enter new rule definitions, the KE can insert important notes as "comments" on a separate comment line (beginning with a "!") anywhere desired in the rule structure. This feature is particularly useful for aligning the historical background and underlying reasoning (if any) behind each "IF" premise or "THEN" conclusion, right at the point where it pertains to the rule structure itself.

In any event, as a matter of format convention, the rule definition must always follow the END statement for the rule, as illustrated at the end of the rule format on the IE screen (Figure 3-7). Thus, the ES must access the non-compiled version of IE rules stored as a ".PRL" file on disk, so that it can present the rule definition and any background comments as text to the user (i.e., because all comments are otherwise stripped from the file at compile time).

S4 - RETRIEVE HELP MESSAGE

INPUTS: Request for help message (entry point C)

OUTPUTS: Help message summary for current operating mode
Operating procedures for available system features

FUNCTION: Upon activation at entry point C, the ES generates a help message summary which describes the current operating mode in detail and shows what system features are available.

At this juncture, the user has the option to EXIT via function key F7, or to continue the message by simply hitting RETURN to see the operating procedures for the indicated system features. The ES responds by generating a set of procedures for each system feature, page by page.

For these optional features, the ES must access the ".PRL" data file supporting the initial system function DISPLAY SEARCH OVERVIEW (S1), which provides all of the pertinent operating procedures as a part of its "nested" overview sequence. Such "nesting" of procedures helps at the outset to explain system operation in "graduated" degrees of complexity to the user, and does it here again via the ES on a demand basis during the search.

3.2.1.4 Inference Engine (IE).

MODULE INPUTS: Activate signal from the UII
User response to control IE search

MODULE OUTPUTS: Current goals resulting from IE search at each level
Next rule (if any) resultintg from IE search
Component subgoals encoded at system level 2
Final goals encoded at system level 3
Warning to user that output will exceed cutoff limits

FUNCTIONS: Upon activation by the UII, the IE returns to the user a set of initial propositions to begin formulating his search query. Upon each user response (which is generally a selection of one of the propositions), the IE attempts to match the proposition selected with the conditions of the rules on the next sublevel. Upon making such a match, the IE returns the conditions remaining for the matched rules as the "next rule" proposition to be selected by the user.

This cyclic "question/answer" process continues until all the rules at each level are exhausted, or, in the case of a "don't know" or "don't care" response from the user, until the IE can go no further. At this point the IE returns to the user the current goals pertaining to the current system level, along with the first rule leading to the next system level.

This cyclic "rule/goal" process continues until the rules on all three system levels have been searched and exhausted. Finally, if the current level just completed is the final system level 3, then the current goals become the "final goals" for the entire search. In addition to returning these goals to the user as a search summary, the IE must also encode them as KB access parameters and then pass them to the Output Phase.

At the outset here, it must be understood that the IE is structured into three major system levels (1,2,3) and, within those levels, into several subordinate sublevels of inference rules. The general structure of the IE was described earlier under paragraph 3.1.2.1 with respect to how the specific "object" and "attribute" categories are structured within the IE (Figure 3-2), and how the IE rules are structured into 3 system levels, with several sublevels at each level (Figure 3-3).

Figure 3-8 shows the general structure of the IE in greater detail, integrating the underlying concepts of Figures 3-2 and 3-3 into one drawing:

SYSTEM LEVEL 1 — To formulate the object of the search, system level 1 comprises 2 sets of "equipment" and "task" rules organized into 3 sublevels.

SYSTEM LEVEL 2 — To formulate the attributes of the object, system level 2 comprises 2 sets of "components" and "factors" rules organized into 3-4 sublevels.

SYSTEM LEVEL 3 — To identify the values of the search, system level 3 comprises sets of "access" rules for translating the final search goals into KB access parameters.

The remainder of this paragraph is a detailed description of the specific functions of the IE that are required to enable its associated performance requirements (S7, S9, and S10):

Level 1	S7	search equipment/task rules
Level 2	S10	seqarch component/factor rules
Level 3	S9	encode goals (as access parameters)

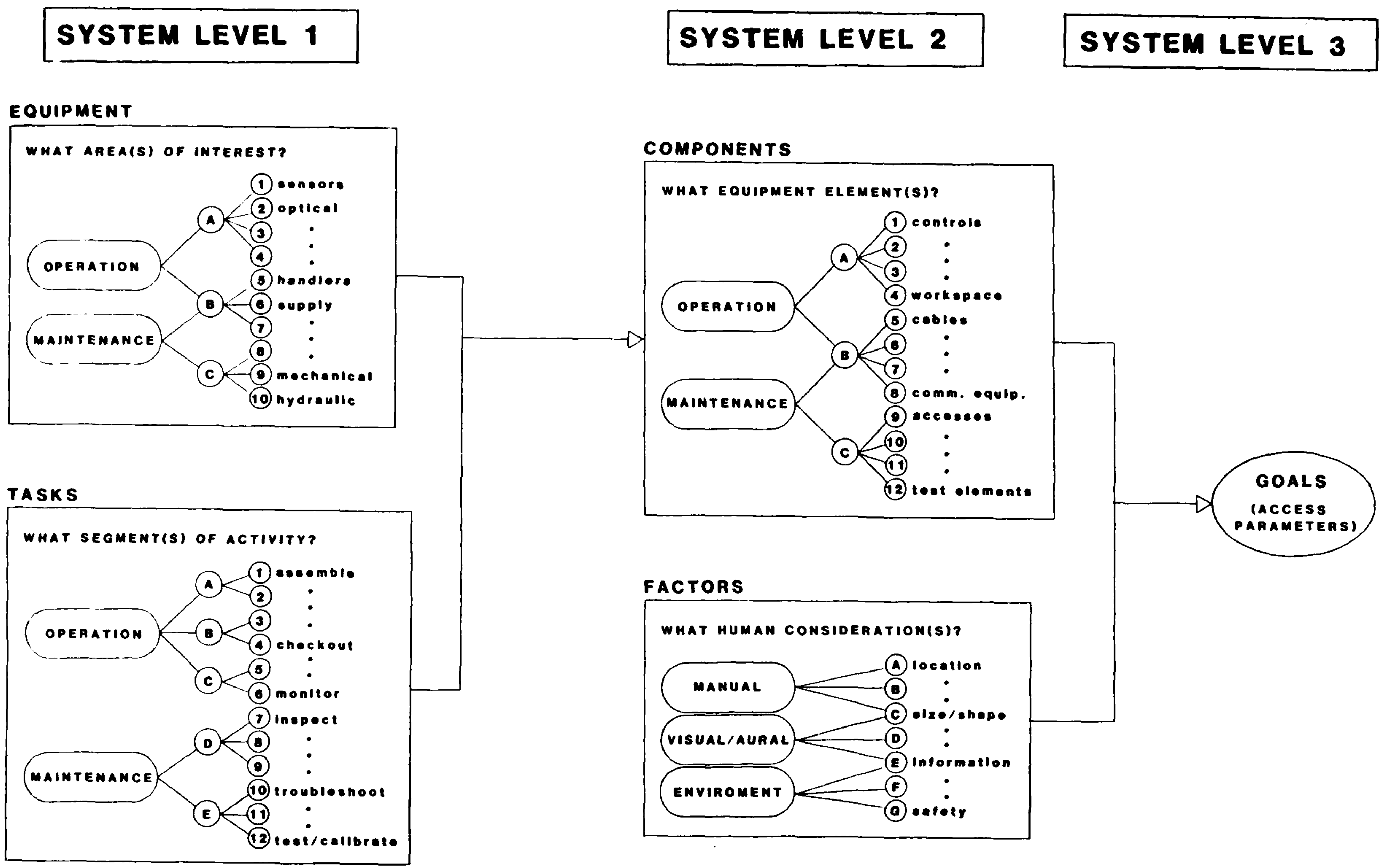


FIGURE 3-8. STRUCTURE OF INFERENCE ENGINE

S7 - SEARCH EQUIPMENT/TASK RULES (system level 1)

INPUTS: User response to control IE search at system level 1

OUTPUTS: Current goals/next rule (if any) from IE search

FUNCTION: Upon each user response (or choice of propositions) at system level 1, the IE search for a "match" among the next sublevel of EQUIPMENT/TASK rules, as just described above. For example, in Figure 3-8, the EQUIPMENT rules are organized into 3 sublevels to determine, first, whether "operations" or "maintenance" equipment is more applicable to the user's RD problem; second, what type of equipment (operational, support, servicing) is most applicable; and third, on what specific equipment category (sensors, optical, ..., hydraulic), if any, the RD problem can be focused.

A similar structure of multiple sublevels is shown in Figure 3-8 for the TASK rules, as well as the COMPONENTS and FACTORS rules on system level 2. The "circles" drawn around each item are essentially "nodes" in the tree-like semantic network on which the IE is structured. There is at least one rule for every node, but many nodes will require more than one rule, particularly if they have more than one "link" coming into them.

Furthermore, to stay within the upper limits of 40 output frames imposed by requirement S7, several more rules must be inserted after the lowest sublevel reached in each set, to predict whether the current level of output is likely to exceed 20 frames (warning threshold) or 40 frames (cutoff threshold). Upon such a warning, it will be up to the user to BACKUP or completely RESTART the search to narrow down his query (for more detail, see functions F1/F2 in paragraph 3.2.1.1). Failing this, the IE will simply abort the search after 40 output frames have been identified, issue a final warning to the user and proceed to encode the access parameters for the 40 frames identified.

S10 - SEARCH COMPONENT/FACTOR RULES (system level 2)

INPUTS: Subgoals of IE search at system level 1
User response to control IE search at system level 2

OUTPUTS: Current goals/next rule (if any) from IE search

FUNCTION: Upon reaching the subgoals at system level 1, the IE performs its own search of the COMPONENT rules, independent of any user input. The COMPONENT subgoals reached here are returned to the user, together with the the rule for the first sublevel of FACTORS. Upon each user response at system level 2, the IE searches for a "match" among the next sublevel of FACTORS rules, exactly as was done with the EQUIPMENT/TASK rules. The subgoals reached by this process at system level 2 will ultimately become the final goals for the Search Phase, upon encoding at system level 3.

The IE searches the COMPONENT rules autonomously here at level 2, simply because it is "smart enough" to deduce the appropriate components from the specific equipment and tasks identified at level 1. Table 3-4 shows an exemplary logic table that the IE might use to draw such conclusions as to COMPONENTS:

the Y-axis (top margin) is the specific EQUIPMENT identified at level 1;

the X-axis (left margin) are the specific operational/maintenance TASKS from level 1;

the X-Y matrix slots contain the appropriate COMPONENTS (controls, displays, ..., test elements) for each combination of EQUIPMENT and TASKS; and,

the numbers along the X- and Y- axes and in the matrix slots correspond to the items listed earlier in Figure 3-2.

TABLE 3-4. LOGIC TABLE FOR COMPONENT SUBGOALS

EQUIPMENT CATEGORY

OPERATIONAL ACTIVITY/TASK	1 SENSORS	2 ELECTRO-OPTICAL	3 COMMAND/INFO	4 COMMU-NICATIONS	5 MATERIAL HANDLERS	6 SUPPLY/STORAGE	7 MAINT. EQUIPMENT
CONFIGURE/ASSEMBLE							
1 unpack/assemble components	2/6/9	2/6	6/9	6/9	6/9	6/9	6/7/9
2 configure stations	2/9	2/8	2/8/9	2/5/11	2/7	7/9/12	2/5/9
PREPARE FOR OPERATION/USE							
3 enter/exit station/position	5/6	2/6	5/6/9	5/6	5/6/9	5/6	5/6
4 checkout/verify readiness	4/6	4/8	2/4/6	4/7/9	2/4/10	2/4/6	2/4/6
OPERATE/MONITOR							
5 activate/control/adjust	2/4	2/8	2/11	2/11	2/7/10	2/5/12	2/5/12
6 acquire/monitor output/feedback	2/4	4/8	4/11	4/11	4/6	4/6/9	4/5/6

MAINTENANCE ACTIVITY/TASK	ELECTRICAL	MECHANICAL	HYDRAULIC
	EQUIPMENT CATEGORIES 1 2 3 4 only	EQUIPMENT CATEGORIES 5 6 7 only	
PREVENTIVE MAINTENANCE			
7 inspect/checkout	4/5/6	4/6	4/5/6
8 service	9/11	4/11/12	9/11/12
9 adjust/align	6/12	5/6/12	5/6/12
CORRECTIVE MAINTENANCE			
10 troubleshoot	2/4	4/6	2/4/6
11 repair/replace	7/9/11	5/7/11	5/9/11
12 test/calibrate	4/5/12	5/12	4/5/12

Each of the matrix slots designates at least two COMPONENTS, one general in nature (such as "workspace" or "test elements") and the other more specific (such as labels" or "connectors"). This guarantees that, regardless of how narrowly the user formulates his RD query, he will see at least two output frames from the search, one with a "macro-scopic" and the other with a "micro-scopic" view of the problem.

To meet performance requirement S10 (which requires the IE to encode the final goals within 10 seconds), it may become necessary to "multiplex" some of the encoding at level 3 within the interactive user response cycle here at level 2. For example, upon submission of the next rule to the user, the IE could constructively encode the COMPONENTS portion of the final goals during the user's half of the cycle. And, in the event of multiple FACTORS subgoals, the IE could encode in the same "multiplex" fashion each FACTORS subgoal as it emerges from the search. The net result would be that, upon reaching the final FACTORS subgoal, only one rule would have to be fired per each partially-encoded parameter to finish the encoding, thereby effectively reducing the user wait for encoding at the end of the Search Phase to a minimum.

S9 - ENCODE GOALS AS ACCESS PARAMETERS (system level 3)

INPUTS: Subgoals of IE search at system level 2

OUTPUTS: Final goals encoded as access parameters
Signal to present user with Searech Phase menu

FUNCTION: Upon reaching the subgoals at system level 2, the IE performs a search of its own internal ENCODE rules, independent of any user input. The final goals are presented to the user as they emerge from level 2, so that there is no need for further user interaction at this point. Once the goals are encoded here at level 3, the IE signals the UII to display the menu for the Search Phase, from which the user can activate the Output Phase, as he wishes.

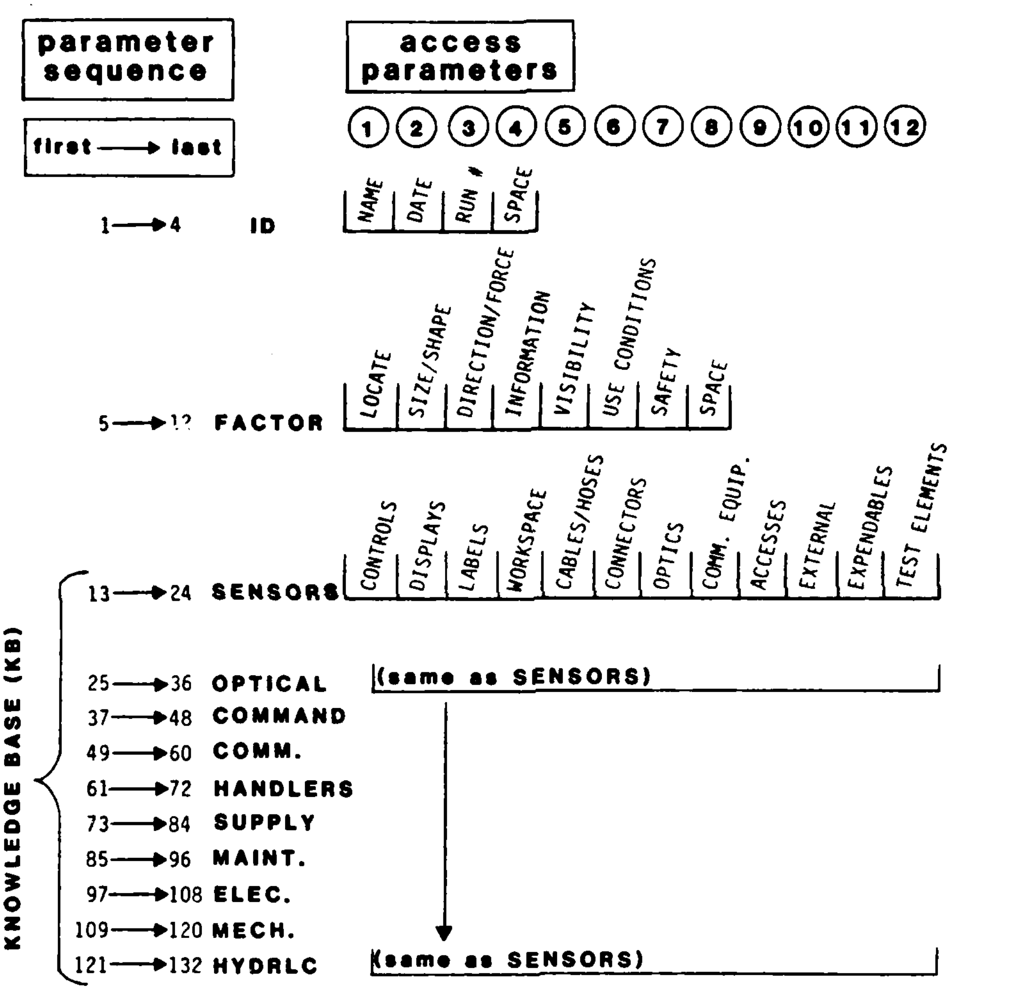
The specific encoding technique to be used here at system level 3 is strictly a matter of program design choice. The only obvious constraints are that the format and quantity of the parameters, as well as the mechanism for transferring them, must be compatible with dBASE III which must accept them and store them internally. Here are two recommended ways of enabling such a transfer mechanism:

STORED FILE OPTION (refer to Figure 3-9)	The final goals are encoded as a series of ID and FACTOR parameters, followed by 10 sets of COMPONENT parameters, which are stored in a data file expected by dBASE III (e.g., under a file name "KBFRAMES" with a dBASE III suffix ".PRG").
DIRECT TRANSFER OPTION (refer to Figure 3-10)	The final goals are encoded as a series of positional bits within 11 parameters (P0,...,P10) of which the first is a FACTOR parameter and the following 10 are COMPONENT parameters which are transfered directly to dBASE III (e.g., as parameters passed via an ACTIVATE command which activates a dBASE III command file programmed to store them).

INSIGHT 2+ ⟹ dBASE III

ACCESS PARAMETERS

(stored file option)



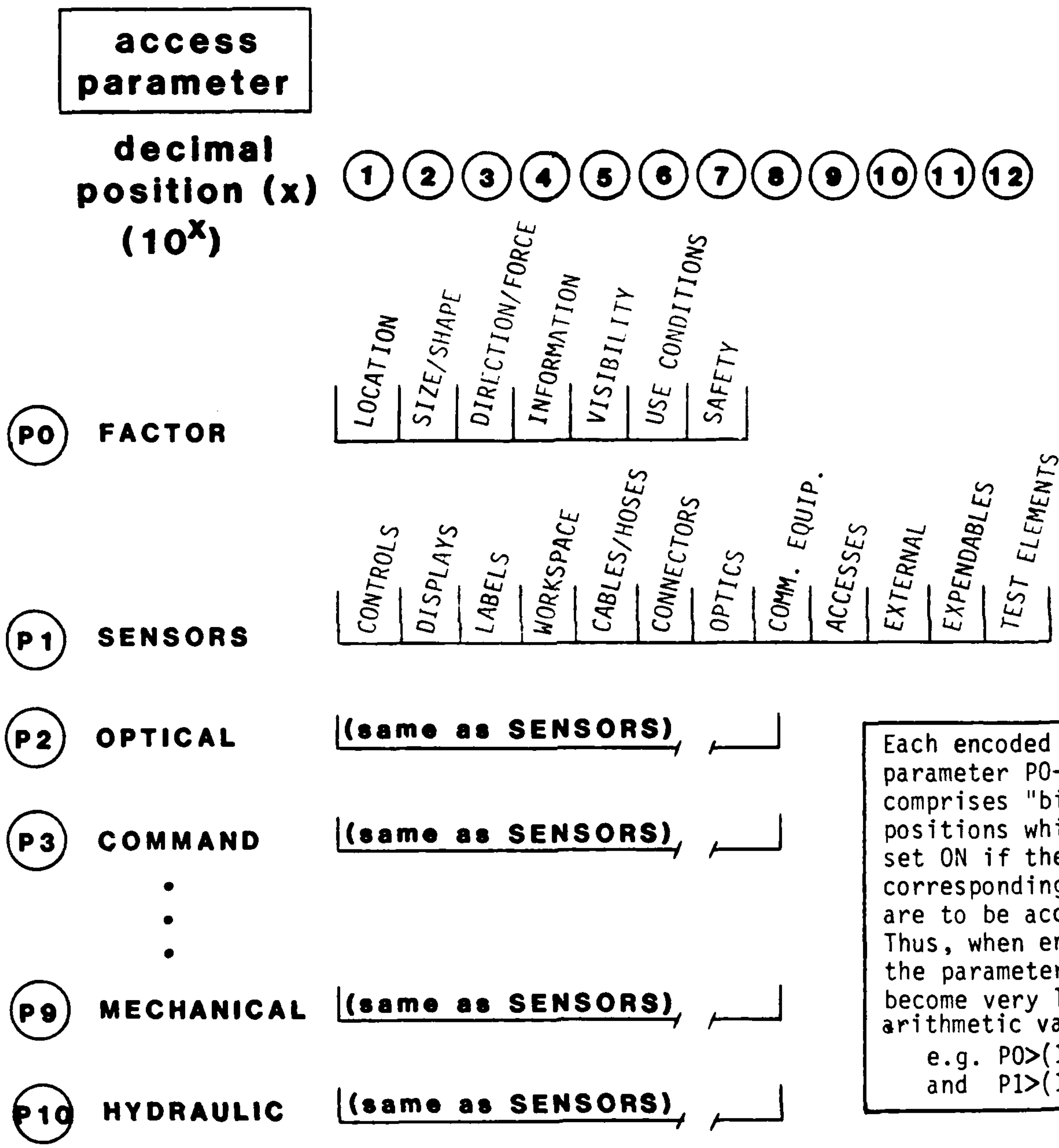
The first four ID parameters are character strings that may contain up to 12 characters each and the remaining parameters 5-132 are merely logical (ON/OFF) variables. Access parameters 5-12 represent individual FACTORS that must be accessed (if set ON) within any subsequent database (SENSOR → HYDRAULIC) that has one or more parameters also set ON. There will always be at least one FACTOR and two parameters in one database set ON to control KB access.

FIGURE 3-9. STORED FILE OPTION

INSIGHT 2+ ⟹ dBASE III

ACCESS PARAMETERS

(direct transfer option)



Each encoded access parameter P0→P10 comprises "bit" positions which are set ON if the corresponding fields are to be accessed. Thus, when encoded, the parameters may become very large in arithmetic value:

e.g. $P0>(10)^6$
and $P1>(10)^{11}$

FIGURE 3-10. DIRECT TRANSFER OPTION

A tradeoff analysis should be conducted to determine whether both methods can operate within Insight 2+ and dBASE III constraints, and which method is the most performance-effective without sacrificing user-friendliness. For example, it is obvious that the DIRECT TRANSFER option stands to be the most effective time-wise, since there is no intrinsic delay for storing and retrieving a ".PRG" parameter file. However, this must be traded off against the extra time required to ENCODE here and DECODE at the receiving end. As another tradeoff example, the STORED FILE option requires an additional step for the user to activate the Output Phase from the search menu. This extra step may be desirable if the user would rather RESTART his search anew at this juncture, but it may prove too confusing and/or time-consuming to be considered user-friendly.

Figure 3-11 shows an exemplary procedure for encoding parameters under the latter DIRECT TRANSFER option. Using rules here at level 3 that test each possible goal from the Search Phase independently, each parameters can be bit-encoded with a bit "ON" for each goal that emerged from the given search. As a minimum, following the IE structure described above, there will always be at least one bit set ON in FACTOR parameter (P0) and at least two bits set ON among the COMPONENT parameters (P1, ..., P10). For purpose of illustration here, the exemplary Insight 2+ ENCODE procedure shown in Figure 3.2.1.4F has been based on decimal "place" shifts rather than binary "bit" shifts, but this is a matter of program design choice.

example: Assume that FACTORS (1)LOCATION and (3)DIRECTION are desired for the following:

EQUIPMENT: SENSOR and TASK: CHECKOUT

Upon determining this combination, the system will fire a subgoal yielding:

COMPONENTS: (2)DISPLAY, (4)LABELS

```
INSIGHT 2+ ENCODE procedure

IF FACTOR is LOCATION
THEN P0=: P0+1
IF FACTOR is DIRECTION
THEN P0=: P0+100
  •(this routine encodes appropriate
  • "bits" in parameter P0)
  •
IF EQUIPMENT is SENSOR
AND TASK is CHECKOUT
THEN P1=: P1+10   (for DISPLAYS)
THEN P1=: P1+1000 (for LABELS)
  • (same routine is used
  •  for parameters P2→P10)
  •
```

FIGURE 3-11. PROCEDURES FOR PARAMETER ENCODE

Moreover, there is another more costly alternative to the above two ENCODE options that is within the scope of Insight 2+; namely, the ENCODE function here could be programmed as an independent PASCAL module. Both of the above options take full advantage of the speed and processing capability of Insight 2+ by forcing the ENCODE upon reaching each final goal. Alternatively, the complete set of final goals could be passed as "raw" unmodified parameters to an independent PASCAL routine for subsequent encoding and storage. This would be done via an Insight 2+ CALL command to a PASCAL-compiled program (i.e., with a suffix of ".PCO"), followed by a string of SEND commands for the parameters, as follows:

```
IF... (all conditions for final goals are met)
THEN  (final goals are displayed to user)
AND   CALL ENCODE (programstored on disk as "ENCODE.PCO")
SEND  Parameter 1
SEND  Parameter 2
  :
  :
SEND  Parameter N
AND   (return to main menu)
AND   STOP
```

In addition to the extra PASCAL programming involved for this ENCODE technique, it would also add the extra time burden of activating an intermediate program and encoding all parameters at once, rather than "multiplexing" them as suggested above. As a further handicap, invoking any such PASCAL program would raise the overhead memory required for Insight 2+ by 35K for DBPAS (an extended PASCAL interpreter) and 45K for the PASCAL program itself (including its I/O buffer areas). As a nominal offset, however, DBPAS does allow up to 4 DB files to be "open" concurrently, which may be used judiciously to help accelerate the ENCODE process.

3.2.2 Output Phase

The output phase is devoted to accessing and displaying the HF guidelines resulting from the final goals of the search phase, and any supporting criteria or tables/figures spontaneously requested by the user. As a parallel system, the output phase comprises many functions which are virtually identical to functions in the search phase (Paragraph 3.2.1), such as DISPLAY OUTPUT OVERVIEW and MONITOR FUNCTION INPUTS. The following is a detailed description of each output function with its associated performance requirements (O1,...,O10) indicated in the left margin.

3.2.2.1 User Output Interface (UOI).

MODULE INPUTS:
System activate signal from Search Phase
Function keys (F1-F7) to change modes
Keyboard/cursor keys to control display
Output string of decoded KB access parameters
Output pointer positioned at next frame for display
Next GUIDELINE database transferred to memory
Next CRITERIA segment overlayed in memory

MODULE OUTPUTS:
Module activate signals to the KAS/KBI
Series of display screens for output overview
Display screen for current GUIDELINE/CRITERIA frame
CRITERIA screen updated with TABLE/FIGURE references
User request for next/last frame or next database
Display screen for output summary of Output Phase

MODULE FUNCTIONS: The User Output Interface (UOI) is activated when the user selects the Output Phase on the main menu. The UOI, in turn, provides an output menu from which the user can, among other things, learn about the output process from an output overview, enter a DATA ENTRY mode to enter new guidelines/criteria into the KB, or simply proceed directly into an OUTPUT GUIDELINE mode where he can review the results of his RD query to the system.

In dealing directly with the user and being totally responsive to user command, the UOI essentially becomes the executive control routine for all of the other modules in the output phase -- the KAS, the KBI, and the all-important KB. As such, the UOI serves to activate each of these modules (generally in response to user request), to supply them with the necessary operating parameters (such as the current user request to the KB), and to await their subsequent responses (such as the next frame from the KB to display to the user).

The following is a detailed description of the specific inputs, outputs, and intrinsic functions for each of the functions allocated to the UOI in Table 3-3 of paragraph 3.1.3:

O1/O10	display output overview (including procedures)
O2/O10	decode goals (from search phase)
O3	monitor function inputs (for mode changes)
O4	access knowledge acquisition subsystem (if requested)
O5	monitor keyboard inputs (for sequence control)
O6	access knowledge base interface
O7	access knowledge base
O8/O10	display guidelines/criteria
O8	display table/figure references (if requested)
O9	display output summary

O1 - DISPLAY OUTPUT OVERVIEW

INPUTS: System activate signal from the Search Phase
Keyboard keys to control display
Function keys (F1-F7) to exercise options

OUTPUTS: Series of overview display screens
Signal to enter DATA ENTRY mode
Signal to enter OUTPUT GUIDELINES mode

FUNCTION: Upon activation, the user output interface (UOI) will display a series of screens to identify the purpose of the output phase, delineate its features, and explain its options in a completely user-friendly manner. These initial screens will associate the available system features and options with the various user-controlled modes of operation that enable them, including guideline/criteria modes for the output phase, and the KB data entry mode for the update phase (see Section 3.4 for more detail on system modes). The overview will also explain how the user can change modes and invoke the various system options via the function keys and/or the alphanumeric keyboard. Finally, the overview will delineate the procedures for operating the system within each mode.

O10-System startup time must be accomplished within 10 seconds to first overview screen which shall be the output menu. If the KB becomes too expansive to load within the allotted 10 seconds, it may become necessary to resort to subdividing the criteria database into segmented overlays. Next screen selection will normally be advanced by hitting function F1, with alternate paths available via the remaining function keys (F2-F7) to provide more detail about the various options. The user will be allowed to "escape" the ordained sequence of overview screens at any time by hitting the "ESC" key, which will force the system to revert back to the output menu. The first two choices on the menu will be devoted to entering the GUIDELINE mode to display the output frames, or the DATA ENTRY mode to update the KB, respectively (see the description of the MENU function (F5) in the preceding paragraph).

O2 - DECODE GOALS

INPUTS: Encoded KB access parameters from search phase

OUTPUTS: Signal to activate KBI in OUTPUT mode
Output string of decoded KB access parameters

FUNCTION: This is the first operational function in the output phase, performed independently by the knowledge base interface (KBI) upon activation. For this function, access parameters, which have been encoded by the IE at the conclusion of the user query, are decoded by the KBI into a string of parameters for accessing the KB. This output string dictates what guideline frames will be retrieved for display to the user, and in what sequence.

O10-This function (O2) must be initiated just as soon as the user activates the output phase via the preceding function, DISPLAY OUTPUT OVERVIEW. However, in the limited time available (10 seconds) to decode as many as 40 access parameters (upper system limit), it may become necessary to initiate this function (O2) as an independent task concurrently with the preceding function (O1). This takes advantage of the preliminary 10-second time frame allocated to output startup. In any event, for less than 40 parameters, this technique will proportionately reduce the 10-second period allocated to decoding, thereby reducing the time the user must wait to proceed with the output display.

The specific decoding algorithms and techniques used for this function are strictly matters of program design choice. However, for illustration here, a viable example is set forth in detail in the description of the KBI at paragraph 3.2.2.3.

03 - MONITOR FUNCTION INPUTS

INPUTS: Function keys to activate functions (F1-F7)
Any keyboard key to reset functions

OUTPUTS: Earlier display screen in GUIDELINE mode (F1,F2,F5)
New display screen in GUIDELINE mode (F3,F4)
New display screen in CRITERIA mode (F6)
Overview display screen with output summary (F5,F7)
System exit to pass control back to DOS (F7)

FUNCTION: Once the overview screens have been displayed, the UOI will continuously monitor the function keys to detect any mode changes requested by the user. The specific functions enabled will permit the user to have the greatest flexibility in controlling the system with the least amount of prior explanation and/or training. The functions themselves will be identified on the bottom of the user screen to promote user recognition of what options the system provides and how to activate them (see the exemplary screen format for the function DISPLAY GUIDELINES/CRITERIA later in this paragraph).

The following are examples of specific functions that should be made available to the user by simply hitting the function keys (F1, BACKUP; F2, RESTART; F3, NEXT DATABASE; F4, NEXT FRAME; F5, GUIDELINE MODE; F6, CRITERIA MODE; F7, EXIT) at any time during the search phase.

F1-BACKUP: This output function is identical to the parallel search function F1 described above, except that it deals with backing up through the output frames resulting from the search, rather than "nodes" in a semantic network. For example, should a user become inspired by a guideline in the current frame, he can repeatedly back up through the preceeding frames to find an earlier related guideline. This sequence reversal requires that the UOI acts as an output "pointer" that can be moved forward with NEXT FRAME (F4) and backward with BACKUP (F1), not only within a database, but also between databases in either direction. For this reason, the UOI must maintain a separate set of memory variables to track the database to which each frame belongs.

F2-RESTART: As with BACKUP (F1), this output function is identical to the parallel search function F2 described above, except that it addresses RESTART of the sequence of output frames resulting from the search phase, rather than the query itself. As before, this function requires the user to hit F2 a second time to confirm that he wishes to restart the entire sequence of output frames. Once again, this safeguard prevents loss of user output time and energy due to striking F2 by accident.

F3-NEXT DATABASE: Upon user request to advance to the next database, the UOI shifts the output "pointer" forward to the next database and displays the first frame of guidlines scheduled for output therefrom. This NEXT DATABASE function has been provided to allow the user to accelerate out of a database he is not interested in (e.g., when he has seen the guidelines before in another search). This shift forward in the output stream may be accomplished in a number of effective ways, including overlaying the potentially large KB databases on top of each other in memory via the knowledge base interface (KBI). Memory overlays should prove quite efficient for this purpose since on line access is needed for but one database at a time. Using this method, disk transfers of up to 100KB can be performed well within the 5-second allowable time. If the current database is the last database, the UOI issues a "no data available" message and awaits the next user command.

F4-NEXT FRAME: Upon user request to advance to the next frame, the UOI shifts the output "pointer" to the next frame scheduled for output and displays the guideline/criteria contained therein. This NEXT FRAME function has been provided to allow the user to accelerate out of a frame he is not interested in (e.g., where he has seen the guidelines before in a related database). This avoids the normal "next element" advance mechanism where the user must step the cursor successively down the screen through all guidelines/criteria on the current display. If the current frame is the last frame scheduled for output, the UOI generates the closing search summary for user review, which implies that no further data is available.

F5-GUIDELINE MODE: This guideline function F5 is essentially a "toggle" mechanism for returning back to GUIDELINE mode once the user has gone to CRITERIA mode. Upon user request to shift to the GUIDELINE mode, the UOI takes one of two courses of action, depending on the current operating mode. If the CRITERIA mode is in effect as is normally the case (i.e., the criteria for the current guideline are being displayed), then the UOI regenerates the suspended GUIDELINE display from which the user shifted out to seek its supporting criteria. If the GUIDELINE mode is already in effect (i.e., guidelines are currently being displayed), then the UOI displays a "guidelines on display" message and awaits the next user command.

F6-CRITERIA MODE: This CRITERIA function F6 is essentially a "toggle" mechanism for switching from GUIDELINE mode to CRITERIA mode or, otherwise, to obtain table/figure references. This mode switching is in response to the user's desire for more detail about the current guideline/criteria on display at the cursor. Upon user request to shift to the CRITERIA mode, the UOI takes one of two courses of action, depending on the current operating mode. If the GUIDELINE mode is in effect, as is normally the case, the UOI suspends the current guideline frame from being displayed and generates the frame containing the supporting criteria for the guideline at the current cursor position. If the CRITERIA mode is already in effect (i.e., criteria are currently being displayed), the UOI displays a "reference" message at the bottom of the screen showing specific references to amplifying tables and/or figures for the criterion at the current cursor position. Where sequential tables and/or figures are referenced, they will be displayed as inclusive tables/figures (e.g., Tables 1A1-1A9 and Figures 12F9-12G3).

F7-EXIT: As with BACKUP (F1) and RESTART (F2), this output function is identical to the parallel search function F7 described above, except that it exits from the output phase instead of the search phase. As with the search EXIT function, the user must hit F7 twice to confirm that he does, in fact, want to abandon the remaining guideline frames scheduled for display. The system does this by suspending the current screen and generating a new screen which asks whether the user wants to quit the program. However, for this EXIT function, the new screen also contains a summary of output available versus output displayed up to and including the current, suspended screen. Hence, this screen acts as the output summary that would have otherwise appeared at the normal exit from the output phase after all frames had been displayed.

04 - ACCESS KNOWLEDGE ACQUISITION SUBSYSTEM (KAS)

INPUTS: Activate signal from UOI
Function keys to activate functions (F1-F7) in KAS
Keyboard keys to enter guideline/criteria in KAS

OUTPUTS: Signal to activate KAS in DATA ENTRY mode
New/modified KB guidlines/criteria out of KAS
New/modified KBI access parameters out of KAS

FUNCTION: The KAS is activated via user selection on the system's main menu. Just as with the parallel RG for the search phase, the KAS is essentially an independent, offline interface for the KE to review, insert, modify, or delete data elements in the KB. Similarly, its time responses are not critical, and its constituent functions are identical to, or at least very similar to, the functions described herein for the UOI (for more detail, see paragraph 3.2.2.2). Moreover, growth provisions for the IE are discussed at paragraph 3.3 in conjunction with the configuration and limits of the present system.

05 - MONITOR KEYBOARD/CURSOR INPUTS

INPUTS: Cursor position to step to next data element
Keyboard keys to facilitate user data entry

OUTPUTS: User request for next guideline/criteria frame
User response during DATA ENTRY mode

FUNCTION: The UOI polls the keyboard for user responses at all times during the output phase, except during a KB access (this is because each next KB access is set in motion by the last user response and cannot be recalled or altered while in progress). If for any reason the user wants to change his last response (i.e., leading to a different KB access), he can hit function F1 (BACKUP) after the current KB access is completed and proceed forward again from the last guideline/criteria frame. This monitoring strategy further protects against accidental keystrokes by the user while a legitimate KB access is in progess. Just as with the parallel MONITOR KEYBOARD function (S6) in the search phase, the specific format for user responses depends on the scope of the response:

For a simple YES/NO-type response, the user should be forced to hit the "Y" or "N" key to guarantee a positive, explicit answer. Likewise, a "D" may be used for any "don't know" or "don't care" responses.

Where the user must choose from a set of propositions or statements, the response should be tied to cursor position, allowing the user to manipulate the cursor up and down until arriving at the most appropriate choice and then hitting "RETURN". In the GUIDELINE or CRITERIA mode, the last choice on the display screen should always be "next data element", so that the cursor can be used to request the next-scheduled frame by merely stepping it to the bottom of the page.

Where the user must provide one or more statements as his response (e.g., during an update to the KB), the response will be regarded as all of the text preceeding the user's RETURN. Provision must be made to test and reject non-responses where a user response is required (such as when entering a new guideline into the KB).

Moreover, the cursor and/or numeric keys can be coupled with the function keys to permit the user to choose among multiple choices. After making a function selection such as F7 (EXIT) leading to the output menu, the user can manipulate the cursor and hit "RETURN" upon arriving at his desired choice. Alternatively, the user can hit a numeric key associated with the number of his choice among the multiple choices being displayed.

O6 - ADVANCE TO NEXT/LAST DATA ELEMENT

INPUTS: Output string of decoded KB access parameters
Keyboard/cursor keys for sequential advance/backup
Function keys for non-sequential advance/backup

OUTPUTS: Output pointer positioned at next/last frame
Request KBI to access next database (function F5)
Next frame moved into current display (function F4)
Last frame moved into current display (function F1)

FUNCTION: This function is largely accommodated by the combination of output functions DECODE FINAL GOALS (O2), MONITOR FUNCTION KEYS (O3), and MONITOR KEYBOARD/CURSOR KEYS (O5) (see the detailed description of these functions earlier in this paragraph). Upon initial activation, the UOI establishes an output string of KB frames scheduled for display to the user, and sets an output "pointer" to the first frame in the string. Thereafter, the UOI continuously polls the keyboard and function keys in the GUIDELINE MODE for user requests to advance to the next frame (normal, sequential operation), to backup to the last frame (non-sequential option with function F1), or to skip the current frame and accelerate ahead to the next frame or even the next database (non-sequential option with functions F4/F5).

In response to these requests, the UOI steps the output pointer to the appropriate next/last KB frame, and requests access to that frame. If the desired frame is within the current database in memory, then the frame is immediately moved to the display area (see the following function (O8) to DISPLAY NEXT DATA ELEMENT). If the desired frame is not currently in memory, then the UOI shifts the output pointer to the first frame of the next database scheduled for output (if any), and requests access to the KB for that frame via the KBI (see the next following function (O7) to ACCESS KNOWLEDGE BASE). Once the database is loaded, the desired next frame is moved to the display area, as before for more detail, see paragraph 3.2.2.3 and 3.2.2.4).

If the output string has been exhausted and there is no "next frame", the UOI immediately display the output summary showing that all scheduled frames have been displayed (see the following function (O9) to DISPLAY SYSTEM SUMMARY). If the output string does not have any frames in the "next database", the UOI promptly displays a "no data available" message on the bottom of the current screen, thereby preserving the user's option to continue reviewing the frames in the current database. In any event, the user can hit function F7 (EXIT) at any time if he wishes to escape the ordained output sequence and see what remains in the output string, or if he simply wants to quit for the day (see the description for function F7 earlier in this paragraph).

O7 - ACCESS KNOWLEDGE BASE (KB)

INPUTS: Activate signal from the UOI
Access parameters for next GUIDELINE database
Access parameters for next CRITERIA segment

OUTPUTS: Signal to activate KB in output made
Next GUIDELINE database transferred to memory
Next CRITERIA segment overlayed in memory

FUNCTION: Upon user request for a "next database" or a "next frame" that is not currently loaded in memory, the UOI accesses the KB to obtain the database (DB) containing the desired frame. Upon such a request, the preceding function, ADVANCE TO NEXT/LAST DATA ELEMENT, first determines that the DB with the desired frame is not present in memory and, if so, issue a request to the KBI for an access parameters to the next GUIDELINE database. The KBI then scans its output string of decoded KB access parameters and send the UOI the pertinent DB parameter, if available, or otherwise, a unique symbol indicating a "null" response. The UOI then requests the KB to transfer the desired DB to memory, or otherwise, displays a "no data available" message.

A similar procedure may be used for the CRITERIA database, should that DB prove to be too large for the memory still available after the largest guideline DB has been loaded. In this case, the criteria DB would first have to be divided into a series of 2-12 physical segments (7 is preferred), and each segment "tagged" with a unique descriptor. The KBI would then have to maintain a second "output string" for these CRITERIA segments, just as it does for the guideline frames. This function (O7) would then be expanded to accessing the KB to overlay the desired criteria DB segments in memory. For more detail about the IE functions, see paragraph 3.2.2.4.

O8 - DISPLAY GUIDELINES/CRITERIA

INPUTS: Output pointer positioned at next/last frame
Next GUIDELINE/CRITERIA frame to be displayed
Keyboard/cursor keys to control display

OUTPUTS: Display screen formatting output data elements as:
O-A-V descriptors for last/current/next frame
Individual guidlines/criteria for current frame
Cursor positioned at first data element

FUNCTION: Upon repositioning the output pointer to the next frame to be displayed, the UOI formats the O-A-V descriptors at the top of the screen, and the individual guidlines/criteria at the bottom (see Figure 3-12). The O-A part of the triplets include the equipment, components, and human factors determined during the search phase, while the "values" part of the triplets are the products of the search, which are the HF guidelines/criteria themselves. The references to "criteria" here are used interchangeably with "guidelines" only because they follow the same display format shown in Figure 3-12.

The UOI must reserve sufficient display area at the bottom for at least five guideline/criteria data elements, separated by at least one space (two is preferred). If there are no more than five data elements and the screen becomes saturated, the system must display a "frame continued" message at the bottom, requesting the user to scroll up each subsequent condition one at a time by hitting RETURN. The message is discontinued with display of the last condition, and further attempts to hit RETURN are ignored.

O10-This display function represents the logical conclusion of the 4-second KB cycle for accessing a given display frame. Since the UOI has only one second to generate this display, it may become necesary to concurrently store the O-A-V descriptors with each successive database access in a cumulative memory array for iterative display. This takes advantage of the 5-second timeframe allocated to DB access during which the memory array can be concurrently updated prior to the 4-second display cycle. After generating the display, the UII merely "idles" until the user responds via the keyboard (see the provisions above for the system to MONITOR KEYBOARD/CURSOR INPUTS to meet requirement O5).

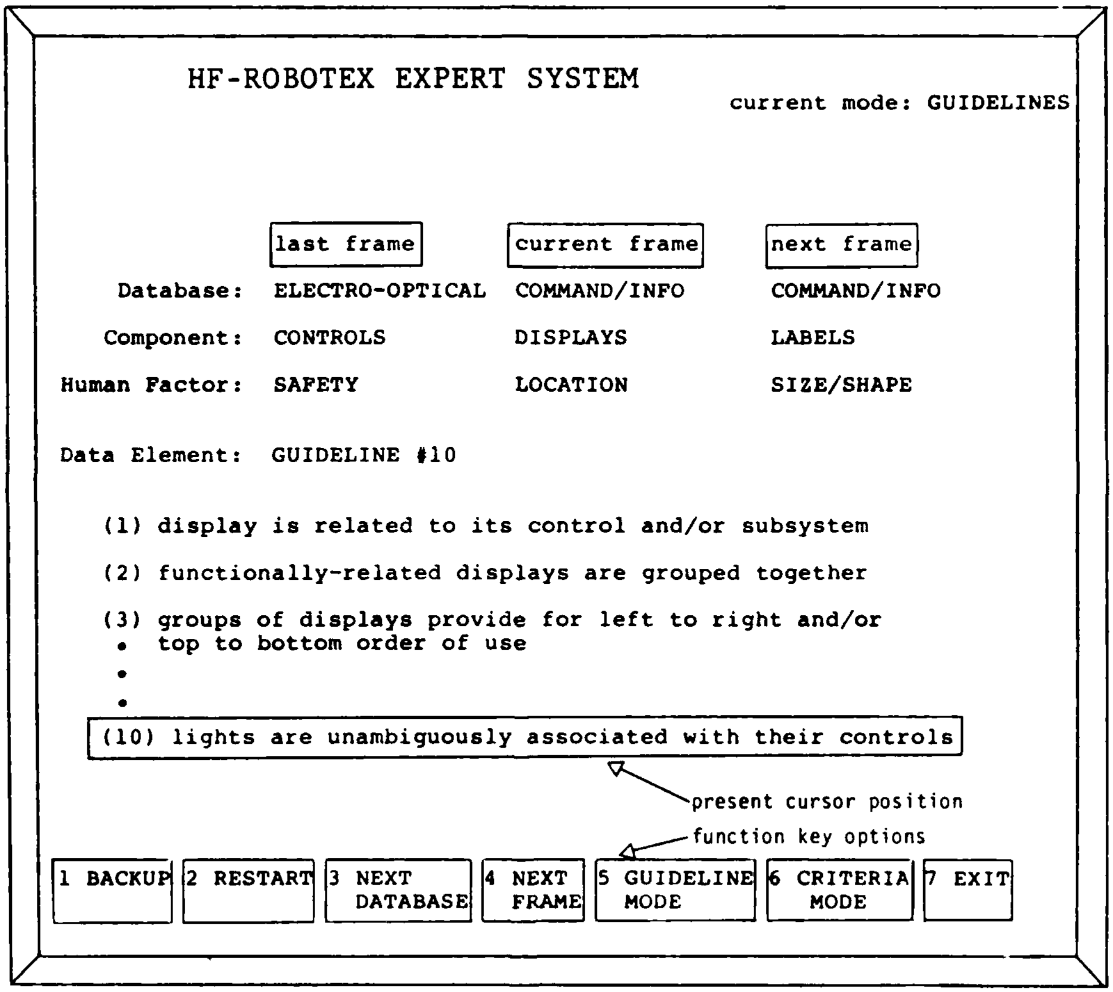


FIGURE 3-12. EXAMPLE OF A USER OUTPUT SCREEN

O8 - DISPLAY TABLE/FIGURE REFERENCES

(special case for requirement O8)

INPUTS: Current CRITERIA frame being displayed
Function key (F6) torequest references

OUTPUTS: Table/figure references on CRITERIA display

FUNCTION: Upon user request to display references to amplifying tables and/or figures for the criteria at the current cursor position, the UOI displays the references (if any) in the space which immediately follows the current criterion on display (see the display format of Figure 3-12). This function may be initiated only by hitting the CRITERIA mode key (F6) when the system is already in CRITERIA mode (see the description for function F6 earlier in this paragraph).

The UOI will obtain the reference information from the data record for the given criterion loaded in memory. Successive tables and/or figures will be shown as "A-B", where the dash ("-") indicates "A through B inclusive". If there are no tables or figures associated with the current criterion, then a "no references available" message will be displayed.

O9 - DISPLAY OUTPUT SUMMARY

INPUTS: Function keys (F3,F4,F7)
Output string of decoded KB access parameters
Output pointer positioned at current frame
O-A-V descriptor for current/past frames

OUTPUTS: Display screen formatting an output summary as:
Output frames scheduled for display
Output frames currently displayed
O-A-V descriptor summary (optional)

FUNCTION: Upon exhausting all frames on the output string, or upon user request to quit the program (via EXIT function F7) or to access a "next frame" which is not available (via NEXT FRAME function F3 or NEXT DATABASE function F4), the UOI generates and displays an output summary to permit the user to review the status of the output being displayed. The output summary includes at least an identification of all frames scheduled for display in the output string, plus the total number of frames already viewed versus the number remaining.

Ideally, the O-A-V descriptors related to each frame can be displayed alongside the frames to which they belong. However, for user convenience, the output summary should be confined to a single display screen. This screen must conclude with a statement to the effect that, if the user does, in fact, want to quit the program, he must hit the F7 key (see the description of the EXIT function F7 earlier in this paragraph). Therefore, a compressed O-A-V descriptor format should be used to maintain the single-screen requirement. Optionally, an expanded O-A-V descriptor list should be attached as "continuation" screens to the output summary, but this becomes a matter of program design choice.

3.2.2.2 Knowledge Acquisition Subsystem (KAS).

MODULE INPUTS:
Signal from UOI to enter DATA ENTRY mode
Function keys (F1-F7) to change mode
Keyboard/cursor keys to enter data/parameters
Previously-entered data/parameters

MODULE OUTPUTS:
Signal to activate KB in the STORAGE mode
Request for previously-entered data/parameters
Display of previously-entered data/parameters
New/modified data for the KB
New/modified parameters for the KBI

MODULE FUNCTIONS: As one of the four principal output modules, the knowledge acquisition subsystem (KAS) completely accomodates performance requirement O4. Just as with the RG module in the search phase, the KAS is relatively autonomous since it essentially an offline interface for the KE to selectively update the KB, in the same way that the UOI is an online interface for the user to selectively output from the KB. Figure 3-13 (referred to hereafter as the "KB screen") shows an example of a screen format for KB data that might be presented to the KE for reviewing or modifying old data already stored in the KB, or for entering entirely new data therein.

Although it is initially activated by the user via the output menu in the UOI (see the DISPLAY OUTPUT OVERVIEW function (O1) in paragraph 3.2.2.1), the KAS performs its updating tasks totally independent of the UOI and without any further communication between the two. Thus, as an independent interface, the KAS operates function-by-function in much the same way as the UOI, without the stringent operational time constraints that have been imposed on the UOI to accommodate the day-to-day needs of the user (see the UOI functions with maximum time requirements (O10) in paragraph 3.2.2.1).

The following is a description of the functions performed by the KAS to enable performance requirement O4, cross-referenced back to the analogous, parallel functions in the UOI. To avoid redundancy, the description here will address only those portions of each KAS function that are different from, or in addition to, the parallel UOI functions:

O4
- display record format
- monitor function inputs (for mode changes)
- access knowledge base (optional inquiry)
- display guideline/criteria frames
- monitor keyboard inputs (for data entry)
- store/retrieve guideline/criteria records
- store access parameters

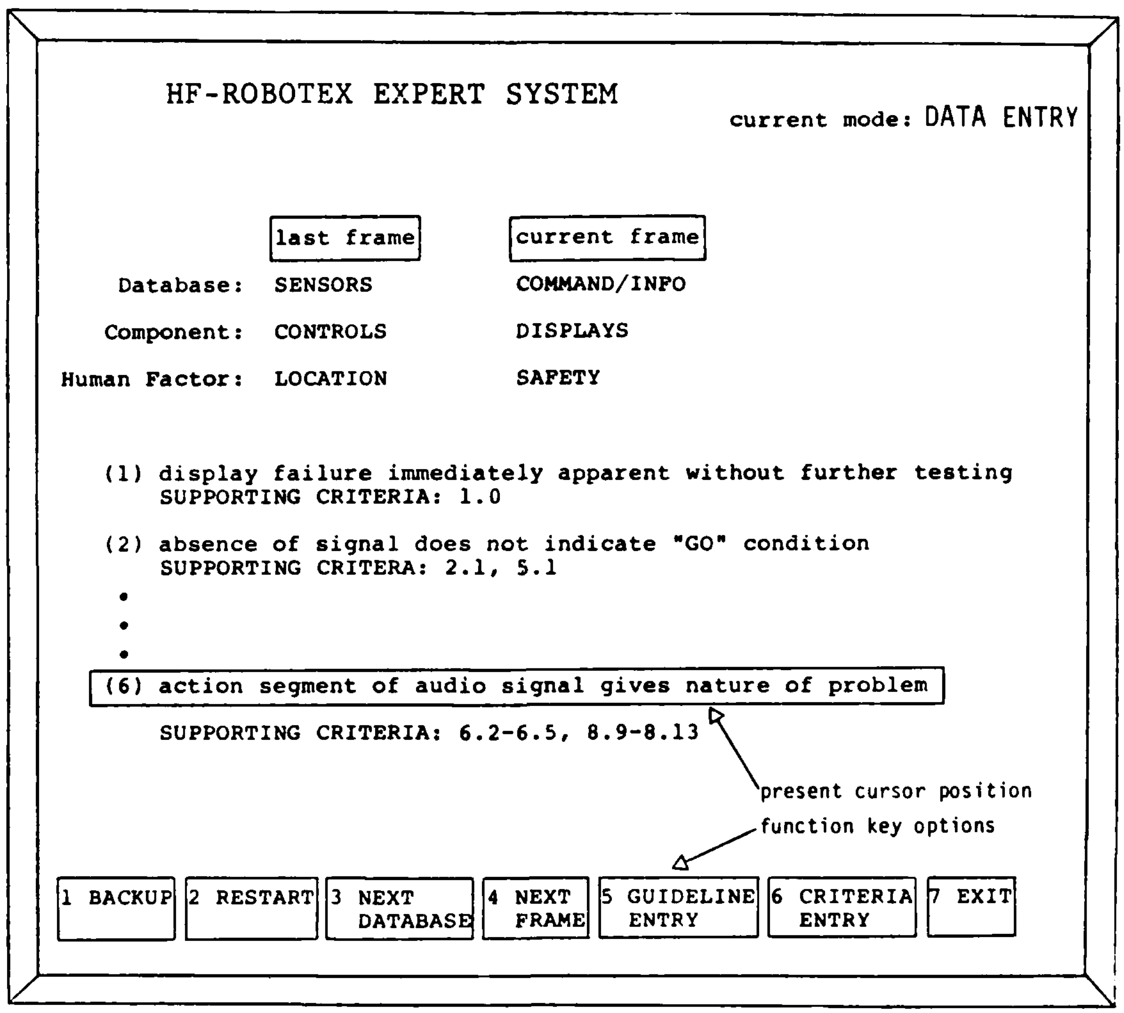


FIGURE 3-13. EXAMPLE OF A KB SCREEN FOR DATA ENTRY

O4 - DISPLAY RECORD FORMAT
(no parallel function in UOI)

INPUTS: Function keys (F3-F5)

OUTPUTS: Blank record format for data entry by KE
Cursor positioned at current frame slot
Cursor positioned at first data slot

FUNCTION: Upon activation by the UOI, the KAS enters the DATA ENTRY mode and presents a blank record format, or "template", for the KE to start entering data, if he wishes. Figure 3-13 shows an exemplary screen format for data entry, but, at this initial point, no frame descriptors or guideline data would appear. The cursor is initially positioned by the KAS at the first descriptor for the "current frame" slot (i.e., at "command/info" on the KB screen) to allow the KE to declare his intentions.

At this juncture, the KE has total discretion to manipulate the KAS to suit his updating needs. He must decide whether he wants to establish a new frame for the KB, review the existing KB data sequentially frame-by-frame, or go directly to a specific KB frame of his choosing:

if the KE wishes to review the exisiting KB data, he merely hits function key F3 to advance sequentially to the NEXT DATABASE, and function key F4 to advance sequentially to the NEXT FRAME in the current database;

if the KE wishes to go to a specific frame in the KB, he must type in the exact "database/component/factor" descriptors of the desired KB frame, and then hit F4 to advance to that frame as the NEXT FRAME; or,

if the KE wishes to establish a new frame for the KB, he must type in the exact descriptors of the new frame, as above, and then hit function key F5 to begin GUIDELINE ENTRY.

In any event, once the KE has declared his intention with the first frame on display, the cursor is repositioned by the KAS at the first data slot (i.e., at the guideline labeled "(1)" on the KB screen). From this point on, the KE can selectively enter new data in the slot, modify old data already there, or simply skip to the next data slot, as he wishes. At any time, he can advance to the next frame or next database via F3/F4.

04 - MONITOR FUNCTION INPUTS
(similar to UOI function 03)

This function is virtually identical to the parallel UOI function, except for two important distinctions at function keys F5/F6 which permit data entry:

F1 BACKUP (same as UOI function)

F2 RESTART (same as UOI function)

F3 NEXT DATABASE (same as UOI function plus initial access)

F4 NEXT FRAME (same as UOI function plus initial access)

F5 GUIDELINE ENTRY (data stored, rather than retrieved)

F6 CRITERIA ENTRY (data stored, rather than retrieved)

F7 EXIT (same as UOI function).

Functions F3/F4 above augment the parallel UOI functions by simply allowing the user to access the first guideline database and/or the first frame within the current DB, with the initial keystroke of F3 and/or F4. Functions F5/F6 above differ from the parallel UOI functions by the direction in which the KB data is traveling. That is, F5/F6 here are intended to initially store the data in the KB, while corresponding functions F5/F6 in the UOI are intended to subsequently retrieve the data, thus stored, from the KB.

04 - ACCESS KNOWLEDGE BASE
(identical to UOI function 07)

This function is provided to allow the KE to selectively review data already in the KB.

04 - DISPLAY GUIDELINE/CRITERIA FRAMES
(similar to UOI function 08)

This KAS function is virtually identical to the parallel UOI function in function and format, except for the fact that any supporting CRITERIA must be entered and/or displayed in the space immediately beneath the associated GUIDELINE (see exemplary CRITERIA references on the KB screen).

04 - MONITOR KEYBOARD INPUTS
(similar to UOI function 05)

This KAS function is provided to allow the KE to selectively control the cursor position, as desired, and, otherwise, to enter text-type data as new or modified GUIDELINES/CRITERIA (see exemplary guidelines shown on the KB screen). The limits placed on entry of individual data elements are 200 characters for any GUIDELINE and 400 characters for any CRITIERIA (see the data record formats of Table 3-5, page 112). The KE must also encode references to the supporting criteria for each guideline (if any), and to the tables/figures for each criteria (if any). This must be done immediately beneath the guideline/criteria being entered, with no more than 4 individual references or 2 sets of inclusive references (compare Table 3-5, page 112 to the KB screen).

O4 - STORE/RETRIEVE GUIDELINE/CRITERIA RECORDS
(no parallel function in UOI)

INPUTS: Keyboard/cursor inputs for data entry
Function keys (F5,F6) for storing records
Function keys (F1-F4) for retrieving records

OUTPUTS: Signal to activate KB in STORAGE mode
Request for previously-entered data from KB
Cursor repositioned at each succeeding data line
New/modified data to be stored in KB

After initially activating the KB in STORAGE mode, this KAS function allows the KE to selectively store or retrieve any individual GUIDELINE or CRITERIA record in the KB. These records observe the format delineated in Table 3-5 (page 112) which includes "linkage" to the next lower data level that has been encoded by the KE at data entry time and verified by the KB at data storage time. This KAS function is closely integrated with the O4 functions to DISPLAY RECORD FORMAT and to MONITOR FUNCTION INPUTS. With respect to retrieving records, a KE request for a different frame via function keys F1-F4 ia a constructive request to retrieve the individual data records within that frame. Hence, the records thus retrieved are displayed in the DATA ENTRY format of the KB screen, rather than the storage format of Table 3-5 (page 112). With respect to storing records, a KE request to enter data via function keys F5/F6 prompts the KAS to physically store the data at the current cursor position as an individual record in the KB in the storage format of Table 3-5 (page 112).

To enter new GUIDELINES, the KE must position the cursor at the desired data slot (e.g., at one of slots (1), (2),..., (6) on the KB screen) and type in up to 200 characters of GUIDELINE text followed by RETURN. The KAS immediately positions the cursor at the succeeding data line, which allows the KE to enter the supporting CRITERIA (if any) as up to 4 individual numbers or 2 inclusive sets of numbers, followed again by RETURN. This time, if the KE is satisfied with the entire GUIDELINE, he hits function key F5 for GUIDELINE ENTRY into the KB. The KAS immediately submits the new data to the KB for storage (see paragraph 3.2.2.4) and positions the cursor at the next sequential data slot for entry of the next guideline.

This entry/storage cycle is repeated until the KE hits one of the function keys F1-F4 to "escape" from the current frame, or hits F7 to simply EXIT back to the main menu. Upon reaching the bottom of the KB screen, the KAS scrolls the screen up, one guideline at a time, to allow entry of the next element. Upon reaching a total of 10 new/modified guidelines (which is the maximum permitted per frame), the KAS simply reverts back to the main menu if the KE attempts to enter any further data.

To modify old GUIDELINES, the KE must again position the cursor at the desired data slot and type over and/or insert the modifications, followed by RETURN. Thereafter, the KAS will behave exactly as if it is addressing a new GUIDELINE, as just described. The process for entering/modifying CRITERIA follows the exact same pattern delineated for GUIDELINES, except that the KE can enter up to 400 characters per CRITERIA via the function key F6 for CRITERIA ENTRY.

04 - STORE ACCESS PARAMETERS
(no parallel function in UOI)

INPUTS: Function keys (F5,F6) for storing new/modified records

OUTPUTS: New/modified access parameters passed to KBI

FUNCTION: This KAS function is closely integrated with the preceding function to STORE GUIDELINE/CRITERIA RECORDS. As just described, the KAS responds to user requests via function keys F5/F6 by storing GUIDELINES or CRITERIA, respectively, in the KB. At the same time, the KAS formulates an access parameter for each KB frame that is added or modified by each user request and passes it dirextly to the KBI (i.e., only one parameter is generated for each new/modified KB frame). These parameters are formulated in the same manner as the KBI decode procedures to simplify KBI processing (see exemplary DECODE procedure shown in Figure 3-14, page 106). The KBI uses these parameters to update its own internal table of database access limits.

3.2.2.3 Knowledge Base Interface (KBI).

MODULE INPUTS:
Activate signal from UOI
Encoded access parameters from Search Phase
User request to access next database (DB) from UOI

MODULE OUTPUTS:
Output string of decoded KB access parameters
KB access parameter identifying next DB
Output pointer positioned at first frame in next DB
Signal indicating "access parameter beyond limits"
Signal indicating "no data available"

FUNCTIONS: The Knowledge Base Interface (KBI) is activated by the UOI when the user selects the Output Phase on the main menu. Alternatively, the KBI may be activated automatically along with the Output Phase when the Search Phase passes its encoded KB access parameters. The exact method of KBI activation is entirely dependent on whether the Search Phase passes its encoded parameters via a stored file or via a direct transfer, which is strictly a matter of program design choice (see the exemplary encoding options described in paragraph 3.2.1.4).

Upon initial activation, the KBI decodes the access parameters and stores them in a common memory array as an "output string" which dictates the sequence of KB frames to be accessed. Thereafter, upon user request via the UOI to access the next GUIDELINE database, the KBI scans the output string for the next sequential DB and sends the resulting access parameter to the KB to initiate the appropriate KB access. In addition, to accelerate UOI processing, the KBI positions the output pointer at the first frame to be accessed in the next DB.

If the CRITERIA database is deemed too large to reside permanently in memory, it may become necessary to subdivide the DB into 2-12 segments. If such is the case, then all of the above KBI functions must be expanded to accommodate the CRITERIA segments in the same manner as, and parallel to, the GUIDELINE databases. For example, whether the user requests access to the next DB or next segment will depend upon whether the system is in GUIDELINE mode or CRITERIA mode, respectively (see the operating modes of Section 3.4). The KBI must also establish and maintain an independent CRITERIA output string and output pointer, similar in structure and operation to their GUIDELINE counterpart.

The remainder of this paragraph is a detailed description of the specific functions of the KBI that are required to enable the associated performance requirements (O2 and O7) of paragraph 3.1.3:

O2 decod goals into access parameters
O7 store/retrieve access parameters
O10 access knowledge base

O2 - DECODE GOALS INTO ACCESS PARAMETERS

INPUTS: Encoded KB access parameters from Search Phase

OUPTUTS: Output string of decoded KB access parameters

FUNCTION: Upon initial activation, the KBI retrieves the encoded access parameters from a data file ending in ".PRG" stored on disk. Alternatively, the KBI accepts them as input parameters transferred directly via an "active" statement from the Search Phase (see exemplary encoding options in paragraph 3.2.1.4). The KBI proceeds to decode them into individual parameters suitable for accessing the KB and then stores them in a common memory array. Before storing them, the KBI must determine what storage sequence is most efficient for accessing not only the DB's on disk, but also the frames within each DB in memory.

Just as with the ENCODE GOALS functions (S9) in the Search Phase, the exact method and manner of implementing the DECODE GOALS function (O2) here is strictly a matter of program design choice. Although dBASE III has been deemed the best candidate vehicle for the Output Phase, certain dBASE III constraints give rise to some strategic program considerations that have a significant impact on program efficiency. The following are some simple examples that should be considered:

> The "stored .PRG file" mentioned above is actually a dBASE III command file that can also serve to initially activate the UOI. It may prove more efficient to allow this command file to DECODE the parameters ahead of, and totally independent of, the UOI. However, if this option is pursued, then the dBASE III memory variables (or "memvars") containing the DECODED parameters must be declared PUBLIC for subsequent global access by the UOI and KB.
>
> As a data handling tool, dBASE III does not accommodate memory arrays conveniently at all, requiring extensive manipulation of internal "memvars" to do so. If the above "stored .PRG file" option is adopted to pass the parameters, then the ENCODE/DECODE technique should attach a unique prefix to the names of the parameters (such as lowercase "m-"). This would allow the program to SAVE, STORE, and RESTORE them as a group with a single dBASE III command. Such a provision will become quite useful if the number of active memvars should ever exceed 232 (which is the upper limit for dBASE III).

Also, the names of these parameter "memvars" can be designated in an ordained sequence by simply attaching a numeric suffix (such as the numeral "01"). This would allow the memvar names themselves to dictate the parameter sequence (e.g., "m-factor 01, ..., m-factor 97"). Equally important, the suffix could also serve as a sort of "loop control" for processing a psuedo-dBASE III array via the dBASE III command "STORE SUBSTR (suffix) TO (counter)".

Furthermore, the parameters should be ordered in the same sequence as their corresponding frames appear in the KB. This would allow the UOI and KB to take advantage of the dBASE III SEEK command, which accesses all frames with the same "index" as a group. Such a provision would vastly simplifiy dBASE III programming where, for example, all frames with the same FACTOR in the current DB could be accessed with a single SEEK.

On the other hand, even if the other "direct transfer" option is adopted for passing the parameters, a similar .PRG command file should be set up to initially accept them as input parameters. Likewise, the internal "memvars" should be named and employed in a similar fashion for SAVE/RESTORE as a composite group, and for SEEK as indexed subgroups.

Moreover, if the "direct transfer" option of paragraph 3.2.1.4 is adopted, then the DECODE process becomes much more involved, since the ENCODE was performed at the decimal "bit" level (see the exemplary ENCODE algorithm shown in Figure 3-11). Hence, the dBASE III DECODE scheme here must resort to some sort of decimal "shifting" algorithm as a loop control while processing each parameter. Figure 3-14 shows an exemplary dBASE III procedure for such a bit-by-bit DECODE. This procedure uses DO WHILE and DO CASE commands as array loop controls, in place of the much slower STORE SUBSTR technique mentioned above. Note that Figure 3-14 only addresses the first parameter P0; similar DO WHILE and DO CASE loops must be set up for each of the remaining parameters (P1,..., P10).

```
dBASE III DECODE procedures

DO WHILE P0>0  (this routine initializes
                program control variables
                m_factor 01,...,m_factor 07)

DO CASE
   CASE P0>111111
     STORE 1 TO m_factor 07
     STORE P0-10**7 TO P0
   CASE P0>11111
     STORE 1 TO m_factor 06
     STORE P0-10**6 TO P0

   CASE P0>0
     STORE 1 TO m_factor 01
     STORE 0 TO P0
     •
     •
     •
ENDCASE  (similar routine is used to store
          parameters P1→P10 into e.g.,
          m_var 001,...,m_var 107)

ENDDO
```

FIGURE 3-14. EXEMPLARY PROCEDURE FOR PARAMETER DECODE

07 - STORE/RETRIEVE ACCESS PARAMETERS

INPUTS: Output string of decoded access parameters
New/modified access parameters from KAS
User request to access next database from UOI

OUTPUTS: Output string of sequenced access parameters
Updated table of access limits for each database
Access parameter for next database
Signal to UOI indicating "parameter beyond limits"
Signal to UOI indicating "no data available"

FUNCTION: This KBI function is closely integrated with the other two KBI functions of DECODE GOALS and ACCESS KB. Upon receiving the decoded acces parameters, the KBI quickly validates the parameters on a high, generic level by comparing each parameter to an internal table of upper/lower DB limits; if it fails, then the KBI issues a "parameter beyond limits" message to the UOI. The KBI then arranges the output string into the optimal sequence for accessing the DB's on disk, and the frames within each DB. It should be noted that this output string sequencing can be accomplished to a great extent by strategically correlating the sequence of rules in the IE as much as possible with the sequence of data in the KB. Special attention should be given to the program considerations for "parameter sequencing" under dBASE III just mentioned in the preceding paragraph under DECODE GOALS.

Upon receiving any new or modified access parameter from the KAS, the KBI updates its own internal table of access limits for the GUIDELINE/CRITERIA databases. This table is maintained by the KBI to reject any spurious attempt to access data outside the upper/lower limits of the current DB's for whatever reason (faulty transmission of encoded access parameters, KB not updated at same time as IE, etc.). This safeguard is intended to help protect against data inconsistencies introduced offline by the independent KAS and RG modules in the UPDATE mode.

Upon user request to access the next database, the KBI scans forward through the output string to find the first parameter in the next DB. Assuming that a SEEK has already been issued, one exemplary way to do this in dBASE III is to issue repeated SKIP commands until a test on the parameter memvar name fails to match. This means the output pointer has finally reached the parameter for the first frame past the frames in the current DB (which were sequentially grouped together). Achieving this, the KBI uses the parameter to access the KB via the next function; failing this, dBASE III will return an "end-of-file" (EOF) indication which the KBI must translate and return to the user as a "no data available" message. This EOF routine is actually the normal KBI end-of-program exit which stimulates the UOI, in turn, to display the output summary to the user.

010 - ACCESS KNOWLEDGE BASE (KB)

INPUTS: Access paramaters for next database

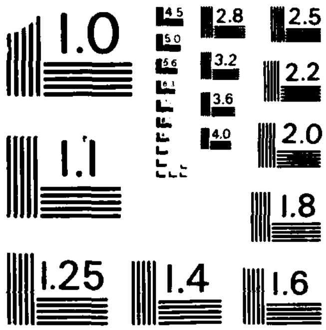
OUTPUTS: Next database transferred to memory

FUNCTION: This KBI function is closely integrated with the preceding RETRIEVE ACCESS PARAMETERS function. Upon receiving the pertinent access parameters for the next database requested by the user, the KBI accesses the knowledge base (KB) which, in turn, transfers the designated database to memory via routine CLOSE and USE database commands.

It should be noted that the composite set of access parameters for frames within each DB represent an "index" for the DB. Maintaining these parameters as a separate file permits an INDEX to be specified with the USE command, and permits the SEEK command to search that index for a match, as was just described. Such random accesses performed in this manner can be accomplished in less than 2 seconds, which is well within the 5 seconds allotted by performance requirement 010.

It should also be noted that the CLOSE command closes all open dBASE III databases and their associated index files, regardless of their work area location. Thus, any CRITERIA database segment that might have been summoned by the user for review with the current GUIDELINE DB, would also be closed. This dBASE III constraint implies that the greater number of segments the CRITERIA DB is divided into, the less delay will be experienced by the user in shifting to CRITERIA mode for the first time within any given GUIDELINE DB. However, it appears that if the CRITERIA DB were divided logically into 7 segments (one for each human factor), none of the 7 CRITERIA segments would be so large as to exceed the 5-second access time allotted by performance requirement 010.

AD-A169 632 EXPERT SYSTEM DESIGN AID FOR APPLICATIONS OF HUMAN 3/3
FACTORS IN ROBOTICS(U) PERSON-SYSTEM INTEGRATION INC
ALEXANDRIA VA J MCGUINNESS ET AL. 12 JUN 86
UNCLASSIFIED PSI-K31-TR885 N60921-85-C-0252 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

3.2.2.4 Knowledge Base (KB).

MODULE INPUTS:
- Activate signal from the UOI
- Access parameters for next GUIDELINE database
- Access parameters for next CRITERIA segment

MODULE OUTPUTS:
- Next GUIDELINE database transferred to memeory
- Next CRITERIA segment overlayed in memory
- Warning that CRITERIA linkage is improper/excessive

FUNCTIONS: Upon activation by the UOI, the KB searches for the first GUIDELINE database requested and transfers it from disk to memory (with the output pointer positioned at the desired guideline frame), permitting the user to begin displaying his output guidelines. Upon user request for "next frame", the UOI checks to see if the frame lies within the database currently in memory: if so, the UOI advances the output pointer to the next frame (without any need for accessing the KB on disk); if not, the UOI requests access to the desired database (via an access parameter from the KBI). Upon such a UOI request for "next database" (or upon a separate user request via UOI function F3), the KB matches the pertinent access parameter from the KBI with the desired database and transfer it to memory. This cyclic request/transfer process continues until all databases in the output string have been exhausted.

If the CRITERIA database is determined to be too large to remain resident in memory as a single entity, the database can be segmented into 2-12 physical segments (preferably divided into 7 segments, indexed by the 7 FACTORS in the parallel GUIDELINE database). Regardless of whichever parameters is chosen as an index, the CRITERIA segments can then be overlayed at the same fixed baseline address in memory in the same manner as the GUIDELINE databases are overlayed upon demand. There is a strategic reason for maintaining separate and independent memory partitions for the current GUIDELINE database and CRITERIA segment: namely, that any CRITERIA segment can be loaded upon user demand while viewing a GUIDELINE database without "swapping out" that database to enter CRITERIA mode.

At the outset here, it must be understood that the KB is structured into three major data levels (1, 2, 3) and, within these levels, into the subordinate GUIDELINE databases and CRITERIA segments just described. The specific structure of the KB was described earlier under paragraph 3.1.2.2 with respect to Figures 3-4 and 3-5. For reference, paragraph 3.1.2.2 describes how the HF knowledge has been structured into a KB comprising 3 data levels (Figure 3-4); and how each database in the KB is divided into "frames" where each frame is, in turn, subdivided into "records" (Figure 3-5).

Table 3-5 shows the specific structure of the elementary KB record in Figure 3-5 in greater detail, showing how the three data levels of Figures 3-4 - 3-5 are linked together at the lowest KB element:

to discriminate an individual GUIDELINE at data level 1, there is one record for each criteria , likewise having a unique "record number" (fields 1-3) and "linkages" to one or more supporting criteria A-B and C-D (field 4-9) at data level 2;

to discriminate an individual CRITERIA at data level 2, there is one record for each criteria, likewise having a unique "record number" (fields 1-3) and "linkages" to one or more tables A-B and/or figures C-D (fields 4-9) at data level 3;

to discriminate an individual TABLES/FIGURE at data level 3, there is a separate entry for each table and figure in an off-loaded reference manual, likewise indexed by a unique "table/figure" number for convenient user reference.

The remainder of this paragraph is a detailed description of the specific functions of the KB that are required to enable its associated performance requirements (O7 and O10) at the above three data levels:

O7 store/retrieve guideline/criteria records
O10 search guideline/criteria frames

TABLE 3-5. dBASE III STRUCTURE

RECORD FORMAT FOR GUIDELINES

ONE RECORD FOR EACH GUIDELINE

	FIELD	NAME	TYPE	WIDTH	RANGE
record number	1)	Component	C	2	1-12
	2)	Factor	C	2	1-7
	3)	Guideline #	C	2*	1-10
linkage to criteria	4)	Criteria A	C	4	1.0-99.9
	5)	Inclusive A+B	C	1**	"-"
	6)	Criteria B	C	4	1.0-99.9
	7)	Criteria C	C	4	1.0-99.9
	8)	Inclusive C+D	C	1**	"-"
	9)	Criteria D	C	4	1.0-99.9
data	10)	Guideline	C	100*	(1-3 lines of 33 chars each)

total=124 chars.

RECORD FORMAT FOR CRITERIA

ONE RECORD FOR EACH CRITERION

	FIELD	NAME	TYPE	WIDTH	RANGE
record number	1)	Component	C	2	1-12
	2)	Factor	C	2	1-7
	3)	Criteria #	C	2*	1-60
linkage to tables/figures	4)	Table A	C	4	1A1-12G9
	5)	Incl Table A-B	C	1**	"-"
	6)	Table B	C	4	1A1-12G9
	7)	Figure C	C	4	1A1-12G9
	8)	Incl Figure C-D	C	1**	"-"
	9)	Figure D	C	4	1A1-12G9
data	10)	Criteria	C	200*	(1-3 lines of 67 chars each)

total=224 chars.

*if guideline (or criteria) is longer than 100 (or 200) chars, then use the next record for the overflow (up to 100 (or 200) additional chars) by adding "A" to the guideline # (or criteria #) (e.g., record "2" followed by "2A")

**if supporting criteria (or tables/figures) are inclusive (e.g., criteria 1-2 or criteria 3-4), then this field contains a dash ("-"); otherwise, the field is blank.

07 - STORE/RETRIEVE GUIDELINE/CRITERIA RECORDS

INPUTS: User request via KAS to store/retrieve records
New/modified guideline/criteria record from KAS

OUTPUTS: Requested guideline/criteria record
Warning to user that linked CRITERIA do not exist
Warning to user that linked CRITERIA exceed limits

FUNCTION: Upon user request via KAS to store/retrieve a guideline or criteria record, the KB accesses the disk for the pertinent GUIDELINE database or CRITERIA segment that should contain the desired record (the usual storage or retrieval follows). As a safeguard to preclude future references to non-existent data, the KB must verify that all criteria identified in any guideline's "linkage" are, in fact, already present and accounted for in the CRITERIA database. If present, the guideline record may be stored; if not, it must be rejected with a warning to the user that the "linked" CRITERIA has not yet been entered. The user must then either correct the linkage and resubmit, or enter the missing criteria to make the linkage consistent.

To perform this latter verification, the KB must decode the "inclusive" feature (fields 5 and 8) of the linkage. Namely, if the dash ("-") is present, then the references to A and B are inclusive (e.g., all criteria A through B, including A and B); otherwise, only A and B are being referenced. The KB should set up an "inclusive string" similar to the "output string" constructed by the KBI, and check off each criteria by an attempted access. A warning must be issued to the user if the linkage references criteria that does not exist.

As an additional safeguard to preclude excessive output being presented to the user, the KB must also verify that the cumulative number of criteria frames being referenced in the linkage does not exceed the limits of performance requirement (07). If the number exceeds 20 frames, a warning must be issued to the user; if it exceeds 40 frames, the record must be rejected from storage until reduced by the user.

O10 - SEARCH GUIDELINE/CRITERIA FRAMES

INPUTS: User request via KBI for next database
Access parameters for next GUIDELINE database
Access parameters for next CRITERIA segment

OUTPUTS: Next GUIDELINE database transferred to memory
Next CRITERIA segment overlayed in memory

FUNCTION: Upon user request for the next GUIDELINE database or CRITERIA via the KBI, the KB accesses the disk for the desired database or segment with the access parameter provided (the routine transfer to memory follows).

To meet the performance requirment O10 (which requires the KB to access the next database within 5 seconds), it may become necessary to further subdivide each of the 10 GUIDELINE databases into 2-12 segments (preferably 7), as was recommended if the CRITERIA database was too large. This further subdivision would allow the KB to stay within the alotted 5 seconds since the smaller segment would take far less time to load. However, before resorting to more than 2 segments per database, a tradeoff analysis should be conducted to determine the worst-case cumulative delay imposed on the user by repeated "swap-outs" of successive database segments. An optimum level of segmentation can be derived by limiting up to 10 successive "swap-outs" to no more that 20 seconds, which is not an unreasonable delay.

3.3 STORAGE AND PROCESSING ALLOCATION

This paragraph describes the allocation of memory storage space and processing time to the HF-ROBOTEX modules, which encompass the executive and interrupt routines, common subroutines, and the common database. In addition, this paragraph discusses the minimal timing, sequencing, and equipment constraints that arise from the reasonable memory space and processing time allocations for HF-ROBOTEX.

At the outset, it should be noted that the RES architecture has endeavored to use the common technique of program and data "overlays" at strategic points in the design. This technique minimizes the otherwise enormous requirements for memory space by imposing only a token increase in processing time to "overlay" one program/data segment on another already in memory. For example, by using this simple technique on the knowledge base (KB), memory requirements can be reduced by a factor of 5 at a token cost of only a couple hundred milliseconds per "overlay" transfer.

Table 3-6 shows estimates of how much memory and processing time should be allocated to the RES modules. The memory estimates have been split into "AVG" and "MAX" categories to illustrate the difference between the nominal configuration anticipated with this specification and the maximum configuration possible within the constraints of Insight 2+ (2000 rules) and available memory (1 MB). This table also illustrates, in terms of available memory, RES has made provisions for future growth of up to 317% (search phase) and 30% (output phase) for a combined potential of 77%. However, by resorting to an "overlay" of dBASE III ove Insight 2+, the output phase could be expanded to 250% of its nominal estimate here of 420K.

Of particular note in Table 3-6 are the module overlays (RG/ES in the search phase and KAS/KBI in the output phase) and the database overlays (10 guideline databases and 7 criteria segments). The module overlays permit Insight 2+ to operate within a 64K ceiling and dBASE III within a 320K ceiling; while the database overlays permit the KB to operate within a 110K ceiling. As another program consideration, the KB could be expanded from 110K (48K guidelines + 62K criteria) to over 600K, by resorting to an overlay of dBASE III over Insight 2+. This means that either the entire guideline DB (480K) or the entire criteria DB (432K) could be kept resident in memory while the other continued as overlays. However, these memory considerations are all matters of program design choice.

TABLE 3-6. ESTIMATED ALLOCATION OF MEMORY AND PROCESSING TIME

module	memory AVG	memory MAX	processing time	
SEARCH PHASE				
INSIGHT 2+	64K (plus IE rule structure)	64K*	10 seconds	(display overview)
UII (resident executive)	24K	24K	3 seconds**	(display subgoals)
			.5 seconds**	(display final goals)
RG (overlaid editor)	20K	20K*	1 minute	(compile 400 rules)
			5 minutes	(compile 2000 rules)
ES (overlaid report return)	20K	20K*	1 second	(display definition)
			3 seconds	(display explanation)
IE (resident rule structure)	76K (400 rules)	380K (2000 rules)	1 second**	(fire next rule)
			10 seconds	(encode parameters)
TOTAL	140K	444K	(317% growth potential)	
OUTPUT PHASE				
dBASE III	320K (plus KB databases)	320K*	10 seconds	(display overview)
UOI (resident executive)	30K	30K	1 second**	(display next element)
			3 seconds**	(display next frame)
KAS (overlaid editor)	50K	50K*	3 seconds	(display data element)
			5 seconds	(store/retrieve element)
KBI (overlaid command file)	5K	5K*	10 seconds	(decode parameters)
KB (resident overlaid database/segment)			5 seconds**	(access next database)
Guidelines (each of 10 databases)	48K	112K		
Criteria (each of 7 segments)	62K	114K		
TOTAL	420K	546K	(30% growth potential)	
COMBINED TOTAL	560K	990K	(77% growth potential)	

* The sums of these overlays at any given time never exceeds a ceiling of 64K for Insight 2+ (or 320K for dBASE III).

** For most cases, the system response time is well under 1 second for a complete IE (or KB) access/display cycle.

The processing time estimates in Table 3-6. are based on the maximum time allotted under the performance requirements delineated in paragraph 3.1.1. The "compile time" was listed for the RG module to illustrate the worst-case delay the user can expect to see anywhere in the system; otherwise, the RG access and display cycles have the same response time as the UII module. Of particular note in Table 3-6 is the second footnote which references the combined access and display cycle of the IE and the KB. Although the performance requirements allow as much as 6 seconds for this cycle, the typical system response time is well under 1 second. Since this is the most common user/system interaction, every attempt must be made to minimize this combined cycle time (e.g., by optimizing the size and frequency of KB data overlays).

3.3.1 Inference Engine Estimates

Since the user-controlled UII/RG/ES modules of Insight 2+ operate within a 64K ceiling, the only remaining consideration for the Search Phase is the system-controlled IE. Table 3-7 summarizes the IE parameters associated with each system level (1, 2, and 3). These parameters include the type of subgoals achieved by firing the rules at each level, whom they are fired by, how many are required, and the minimum depth required within each set of rules.

The resulting totals show that the IE comprises 6 sets of subgoals at 3 system levels which contain 362 total rules. It also shows that the IE spans a hierarchy of rules 17 levels deep. This means that, as a "worst-case" scenario in which the user descends to the most specific rule at each system level (i.e., with a "don't know" response), the IE would have to search through all 362 rules across the 17 levels.

The IE of HF-ROBOTEX has been designed and structured to accommodate this highly impossible "worst case" within the time constraints imposed on the system. The unit of reference for any IE is a "nominal" rule; for Insight 2+, a nominal rule comprises 3 antecedent conditions ("IF A and B and C..") and 2 conclusions ("...THEN D and E") (see the rule format shown for the RG in paragraph 3.2.1.2). As upper operating limits, the IE for Insight 2+ can accommodate up to 2000 such nominal rules, all of which it can search through in about 5 seconds. This means that, for less than 400 rules estimated for RES, the IE could search the entire set of rules in less than 1 second. Hence, as a worst-case scenario, the IE could meet the stringent 1-second performance requirement shown in Table 3-7 to fire the next rule, even if it had to search the entire rule base to find a "match".

It is possible to shift the burden of parameter ENCODE from the IE to an independent PASCAL program, as was described under paragraph 3.2.1.4. However, apart from the increased accessing/processing time burden added to the Search Phase, such an ENCODE technique would also increase the present 64K memory ceiling for Insight 2+ by an additional 80K. This is because Insight 2+ requires an additional 35K for its PASCAL interpreter, DBPAS, and 45K for the program itself (including all work areas). Additional memory space would be required for any external DB files used by the program.

TABLE 3-7. INFERENCE ENGINE PARAMETERS

THE INFERENCE ENGINE COMPRISES 3 SYSTEM LEVELS AS FOLLOWS:

KNOWLEDGE REPRESENTATION	SYSTEM LEVEL	TYPE OF SUBGOALS	FIRED BY	ESTIMATED NUMBER OF RULES	ESTIMATED DEPTH OF RULES
OBJECT	1. (A)	EQUIPMENT	USER	2 X 3 X 4 = 24	3
	(B)	TASKS	USER	2 X 3 X 3 = 18	3
ATTRIBUTES	2. (A)	COMPONENT(S)	SYSTEM	2 X 3 X 4 X 7 = 168	4
	(B)	FACTOR(S)	USER	2 X 3 X 3 = 18	3
VALUES	3. (A)	KB ACCESS	SYSTEM	2 X 7 = 14	2
	(B)	KB PARAMETERS	SYSTEM	10 X 12 = 120	2
TOTALS:	3 SYSTEM LEVELS	6 SETS OF SUBGOALS		362 RULES	17 LEVELS DEEP

3.3.2 Knowledge Base Estimates

Since the user-controlled UOI/KAS/KBI modules of dBASE III operate within a 320K ceiling, the only remaining consideration for the Output Phase is the system-controlled KB. Table 3-8 summarizes the KB parameters associated with each data level (1, 2, and 3). These parameters include the number of factors, components, frames, elements, and even characters that can appear in any given database as a maximum limit, and that otherwise appear as an average across all of the databases. The resulting totals show that the KB comprises 11 databases stored on disk plus 1 database offloaded as a reference manual. Each database has 7 factors and as many as 12 components. However, there are only 8 components on the average (since not all component/factor combinations give rise to meaningful guidelines).

This translates into the fact that, out of 924 MAX possible frames, only 560 frames are ultimately required across all 11 databases. This, in turn, translates into a significant reduction in the number of elements (4K) and characters (.5M) that have to be stored in the system. This means that, as a "worst-case" scenario in which a database comprises 7 factors and 8 components yielding 56 frames (each containing 10 MAX guidelines), the KB would have to load roughly 112 KB from disk to memory.

The KB of HF-ROBOTEX has been designed and structured to accommodate this unlikely "worst case" within the time and memory constraints imposed on the system. The unit of reference for any KB is its "nominal" record; for the dBASE III files, there are two nominal records, one for guidlines and one for criteria, which only differ by the length of the data element (see the record formats shown for the KAS in paragraph 3.2.2.2). Table 3-9 shows the calculations for such a "nominal" record across the entire spectrum HF databases scheduled to become a part of the KB. These calculations are based on the "average" level of data that is distributed across the guideline/criteria databases in an effort to arrive at reasonable memory requirements for HF-ROBOTEX (i.e., at least 48K required for guidelines and at least 62K required for criteria). Whether the guideline and/or criteria databases are segmented and brought into memory as overlays remains a matter of program design choice.

TABLE 3-8. KNOWLEDGE BASE PARAMETERS

THE KNOWLEDGE BASE COMPRISES 3 SYSTEM LEVELS AS FOLLOWS:

DATA TYPE	DATA LEVEL	DATA BASES	FACTORS per database	COMPONENTS (MAX) (AVG) per database	FRAMES (MAX) (AVG) per database	ELEMENTS (MAX) (AVG) per frame	CHARACTERS (MAX) (AVG) per element
GUIDELINES	1	10	7	(12) (7)	(84) (49)	(10) (6)	(200) (100)
(each GUIDELINE may reference up to 2 sets of supporting criteria)							
CRITERIA	2	1	7	(12) (10)	(84) (70)	(60) (15)	(400) (200)
(each CRITERIA may reference up to 2 sets of supporting TABLES/FIGURES)							
TABLES/FIGURES	3	1	7	(12) (10)	(84) (70)	(10) (4)	- -
(these TABLES/FIGURES may be off-loaded as a frame-oriented reference manual)							
TOTALS: (stored data only)	DATA LEVELS 3	DATABASES 11 (stored)	FACTORS 7	COMPONENTS 12 MAX/8 AVG	FRAMES 924 MAX/560 AVG	ELEMENTS 12K MAX/4K AVG	CHARACTERS 1.6M MAX/.5M AVG

TABLE 3-9. ESTIMATES OF KB MEMORY REQUIREMENTS

GUIDELINES DATABASES

Structure of Level 1

database	1	2	3	4	5	6	7	8	9	10	TOTAL	AVG
number of components	6	4	8	8	8	7	8	7	6	8	70	7

The average database estimates for guidelines on level 1 of the KB are:

each database = 7 components (avg)

each frame = 6 guidelines (avg)

each guideline = 80 characters (avg)

Therefore, typically:

each frame = 6 guidelines x 80 chars. = 480 characters (avg)

each database = 7 components x 7 factors = 49 frames (avg)

Guideline Database Totals (estimated)

49 frames x 10 databases = 490 frames

490 frames x 6 guidelines = 2940 guidelines

3K guidelines x 80 characters = 240K characters

240K characters x 2 bytes/char = 480K bytes (grand total)

Guideline Database Overlays (optional)

480K bytes/10 equipment classes = 48K bytes per database

CRITERIA DATABASE

Structure of Level 2

The supporting criteria data sources on Level 2 of the KB contain generic criteria to support the application of individual Human Factors guidelines. There is a frame for each combination of factors and components (i.e., up to a max of 7 factors x 12 components = 84 frames (max)). However, these sets of frames range dramatically in size from min to max:

for each frame, min = 3 criteria = 600 chars (200 chars/criteria)

max = 60 criteria = 12K chars

avg = 15 criteria = 3K chars

However, since no supporting criteria are needed for combinations of factors and components that are not present in the GUIDELINE DATABASE, about 12 of the maximum possible 84 frames are "empty" frames -- leaving 72 frames total. Therefore, the more accurate database estimates for the average criteria are:

each criteria = 200 characters (avg)

each set frame = 15 criteria (max) x 200 chars/criteria

= 3K chars (max)

Criteria Database Totals (estimated)

72 frames x 15 criteria = 1080 criteria

1080 criteria x 200 chars = 216K chars

216K chars x 2 bytes/char = 432K bytes (grand total)

Criteria Database Overlays (optional)

432K bytes/7 human factors = 62K bytes (per segment)

3.4 PROGRAM FUNCTIONAL FLOW

This section describes the system-level of both flow program data and execution control among the HF-ROBOTEX modules. The section breaks down the requisite execution control necessary to operate the program coherently and efficiently into three major aspects:

(1) system-level operating modes which defines where the HF-ROBOTEX modules perform their functions;

(2) functional flow diagrams which show why and how control must be sequenced among the modules; and,

(3) module-level program interrupts which reveal when and how control must be passed between modules.

This paragraph 3.4 begins the description with an overview of what the various operating modes are and how they are interrelated. The next paragraph 3.4.1 describes the functional flow of the system with respect to the operating modes, including functional flow diagrams which cross-reference each module with its respective mode and, within the modules, each function with the original performance requirement (S1,..., S9) and (O1,..., O9) of paragraph 3.1.1 from which it has arisen. The next paragraph 3.4.2 carries the description to an even lower level by explaining the requisite program interrupts in terms of the modules and operating modes to which they relate. The final paragraph 3.4.3 concludes with a delineation of pertinent timing constraints, cycle times, and priority assignments that attach to the interrupts of paragraph 3.4.2. Beyond this, there are no special control features contemplated by this PDS which lie outside of the normal operating procedures otherwise covered in this section.

At the outset, it should be noted that the comprehensive flow of data across the system has been generally covered in paragraph 3.1.2 with respect to Figure 3-1, page 24. Each of the arrows in Figure 3-1 are labeled with the specific type of data that is flowing between the system modules. The large arrows reflect data flowing during the Search Phase (on the left) and the Output Phase (on the right side), while the small arrows reflect data flowing during the Update Phase. The composition and format of each type of data appearing in Figure 3-1 has already been described and shown earlier in section 3.2, as indicated in Table 3-10.

TABLE 3-10. HF-ROBOTEX DATA TYPES AND FORMATS (cross references)

DATA			FORMAT		
TYPE	SOURCE	DESTINATION	DESCRIPTION	PARAGRAPH	DRAWING
user response	UII	IE	MONITOR functions S2/S6	3.2.1.1	——
user inquiries	UII	ES	MONITOR functions S2/S4	3.2.2.1	3-12
definitions/explanations	ES	UII	IE screen for rule entry S3	3.2.1.3	3-7
rules (stored)	RG	IE	DISPLAY RULES function S8	3.2.1.1	3-6
rules (displayed)	IE	UOI	IE screen for rule entry S3	3.2.1.2	3-7
parameters (encoded)	IE	KBI	ENCODE GOALS function S9	3.2.1.4	3-11
parameters (decoded)	KBI	KB	DECODE GOALS function O2	3.2.2.3	3-14
guidelines (displayed*)	KB	UOI	DISPLAY GUIDELINES function O8	3.2.2.1	3-12
guidelines (stored)	KAS	KB	KB screen for data entry O4	3.2.2.2	3-13
user requests	UOI	KB	MONITOR function O3/O5	3.2.2.1	——

*guidelines are displayed as "frames" of up to 10 guidelines each

Figure 3-15 provides an overview of the HF-ROBOTEX operating modes, showing how they are interrelated. From the user's point of view, the system is basically divided into a Search Phase (via Insight 2+) on the left, and an Output Phase (via dBASE III) on the right. From the KE's point of view, the system provides an Update Phase which is similarly divided between Insight 2+ and dBASE III to update the IE and the KB, respectively. The dark arrows in Figure 3-15 indicate the critical path for a search through the system which is clearly destined to get the most frequent use. The light arrows indicate optional paths for the user to get supporting information when conducting a search, or for the KE to update the system with new data. The dotted arrows indicate optional paths for the KE to review data already existing in the system before entering new data.

The priority of the operating modes of Figure 3-15 have been ordered into two time-wise independent sets:

PRIORITY	INSIGHT 2+ modes		dBASE III	
A	UPDATE PHASE	INSERTION	UPDATE PHASE	STORAGE
B		RULE ENTRY		DATA ENTRY
C	SEARCH PHASE	SEARCH	OUTPUT PHASE	ACCESS
D		QUERY		GUIDELINE
E		EXPLANATION		CRITERIA

The criteria for this assignment of priority is based on two simple database management principles:

ONE-Updating a database must have a higher priority than searching the database, since all forms of "reading" a file must be locked out until "writing" to update the file has been completed. This ensures that the user is searching through the most up-to-date data, and that he does not attempt to further alter data that is in the process of being updated.

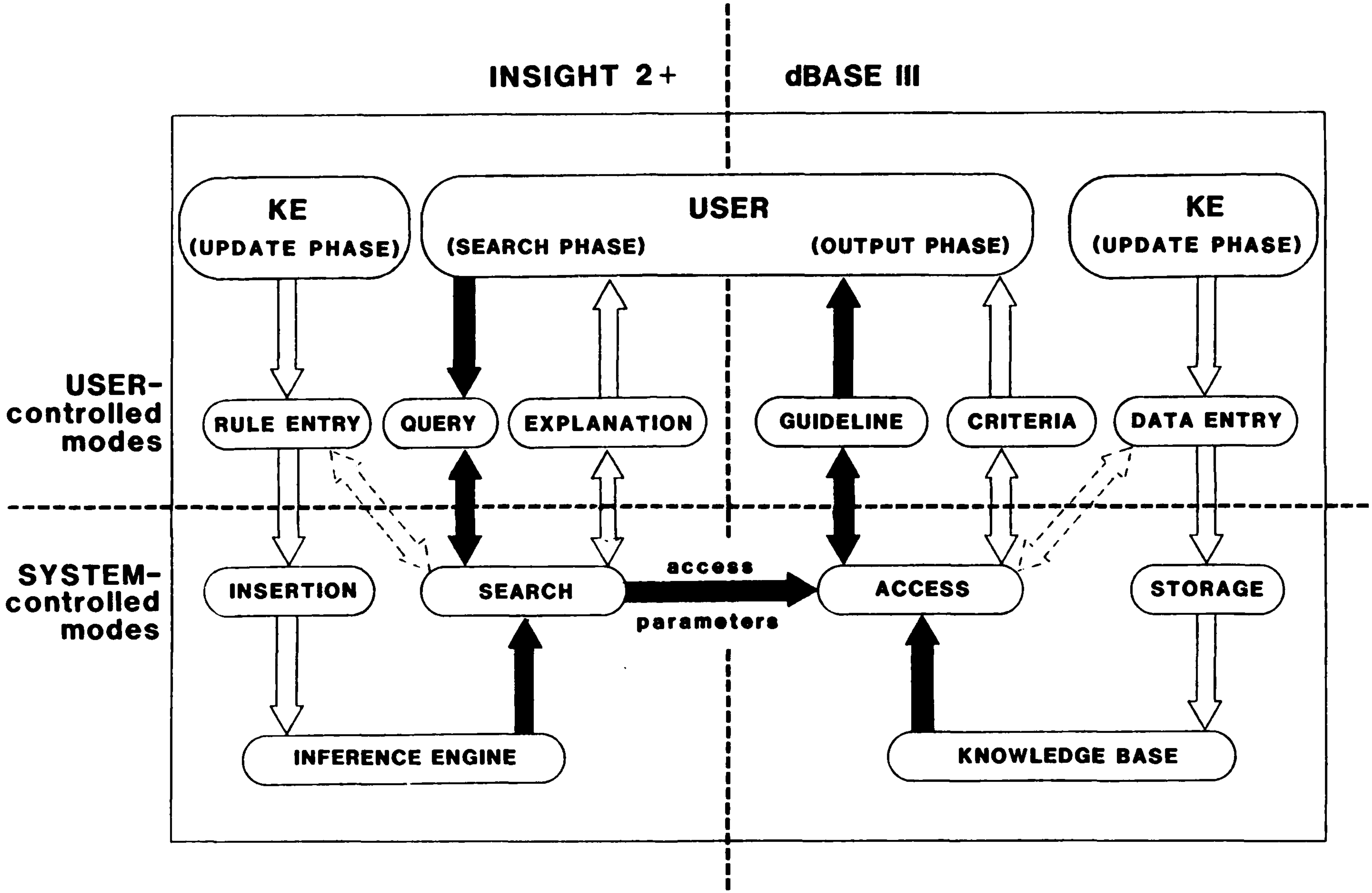


FIGURE 3-15. MODES OF OPERATION

TWO-Accessing a database must have a higher priority than entering new responses via the keyboard, since all forms of "input" requests must be locked out until "output" responses from the last request have been completed. This ensures that the user does not begin a new line of inquiry until he has considered the results of his current inquiry, and also that he properly "backs up" the system to a previous search node.

The scope and sequencing of the above operating modes will be discussed in conjunction with the functional flow diagrams in the next paragraph 3.4.1. The source and timing of program interrupts which pass control between the modes will be discussed in paragraph 3.4.2.

3.4.1 Functional Flow Diagrams

The functional flow of HF-ROBOTEX runs closely in parallel with the sequence of the original performance requirements (S1,..., S9) and (O1,..., O9) of paragraph 3.1. The functions associated with these requirements are summarized briefly at paragraph 3.1.1 and described in depth under paragraph 3.2; hence, they will not be discussed in great detail again in this paragraph.

Figure 3-16 (referred to hereafter as the "flowchart") is a general functional flow diagram showing the distribution and sequence of functions among the system modules for each of the above operating modes. Hence, this flowchart serves to correlate the following significant aspects of the HF-ROBOTEX system within a single drawing:

HF-ROBOTEX System Aspects	indicated on the diagram as:
System phase (SEARCH, OUTPUT, UPDATE)	large, circles with name in boldface type located in region of operation
System module (UII, UOI, IE, KB, etc.)	large, dotted-line boxes with module mnemonic in upper RH corner
Operating mode (QUERY, DATA ENTRY, etc.)	name in large type in upper LH corner of module boxes
Module functions (display, monitor, etc.)	small, solid-line boxes located within module boxes
Performance requirements (S1, O1, etc.)	requirement mnemonic on LH side of function boxes
Flow of execution control (user request,next rule,etc.)	small arrows drawn between function boxes

The following paragraphs trace the flow of execution control through the diagram within the Search Phase, Output Phase, and Update Phase, respectively.

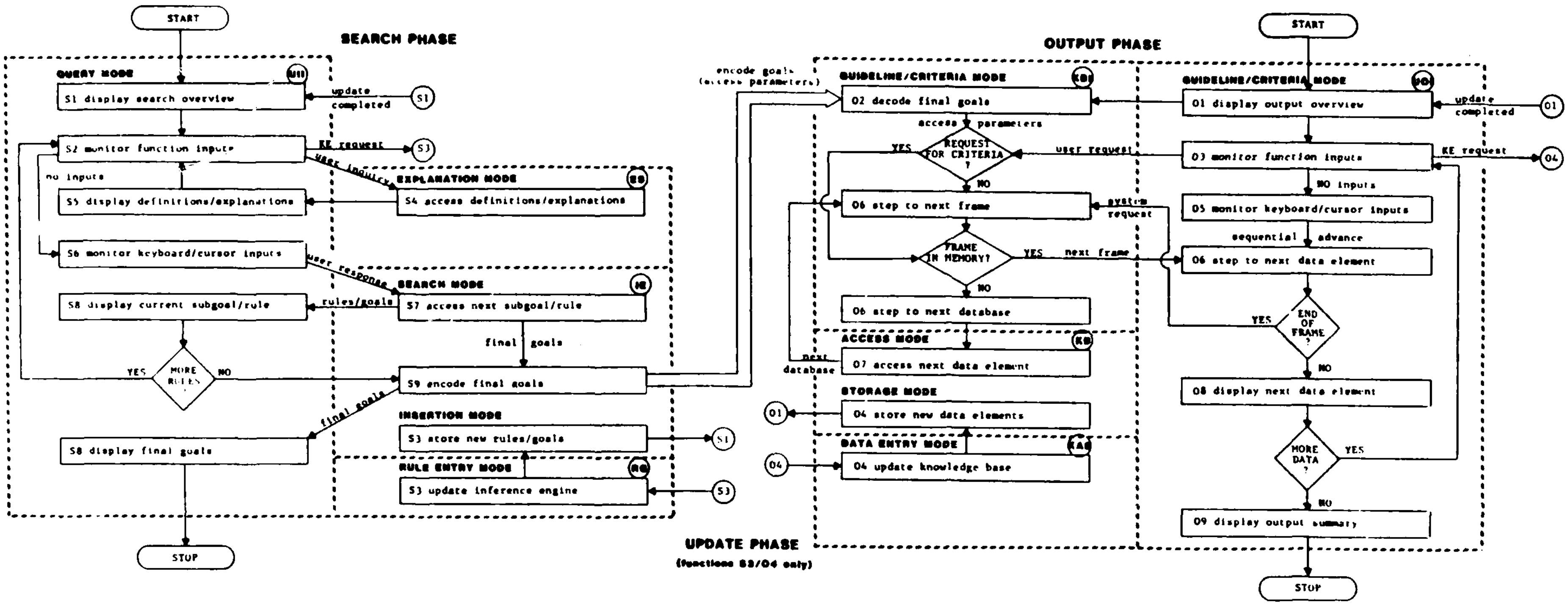


FIGURE 3-16. GENERAL FUNCTIONAL FLOW DIAGRAM

3.4.1.1 Search Phase. The flow of execution control begins with the user activating the UII at START (on the LH side of the flowchart). The UII begins by displaying the search overview (S1) which includes a main menu from which the user can select the next mode of operation. The system steps through the search overview under user control, allowing him to "escape" via the ESC key back to the main menu. At all times, the system must continuously monitor the function keys (S2) to allow the user to change system modes at his discretion. From the main menu, the user must use the function keys to select at least between conducting a search (via the QUERY mode) or updating the IE (via the RULE ENTRY mode).

Assuming the user wants the former option, then the UII responds by entering the QUERY mode and presenting the first rule. At any time from this point, the user may inquire (via the function keys) as to the definition of the current rule, an explanation of the search status, or a help message on a system feature. Upon such an inquiry, the UII responds by entering the EXPLANATION mode and transferring execution control to the explanation subsystem (ES) (on the RH side of the Search Phase in the flowchart). The ES responds, in turn, by accessing the appropriate definitions, explanations, or messages (S4) and sending them back to the UII for display (S5). Upon fielding a satisfactory explanation, the user "escapes" back to the current search via the ESC key, which triggers the UII to revert back to the QUERY mode both as to display screen and monitor function (S2). Thus, the entire EXPLANATION cycle is accomplished via functions (S2, S4, S5, S2) sequentially.

Assuming no inputs are made via the function keys (S2), the UII monitors the keyboard and/or cursor (S6), awaiting the user's response to the rule being displayed. Upon user request, the UII activates the IE in the SEARCH mode to enable it to access the next subgoal/rule (S7) that "matches" the user's response. Upon such a match, the IE sends the identified subgoal and/or the next rule along the path of the user's search (if any) back to the UII, whereupon execution control reverts back to the UII to display the next subgoal/rule (S8). If there are more rules along the search path, the UII returns back to monitor the function keys (S2) and start the QUERY cycle over again. Thus, the entire QUERY cycle is accomplished via functions (S2, S6, S7, S8, S2) sequentially.

This QUERY cycle is repeated for each rule for which the IE can find a "match" until all of the rules at all sublevels of system levels 1 and 2 are exhausted (see paragraph 3.2.1.4 for more description of the system levels). Finally, if there are no more rules along the user's search path, then the UII signals the IE to encode the last set of subgoals reached as the final goals (S9) for the search and sends them back to the UII for display (S8). Moreover, the IE must also transfer the encoded goals as access parameters to the Output Phase. Thus, the exit path for the QUERY mode is accomplished via functions (S8, S9, S8), whereupon the Output Phase is initiated. For more description of the Output Phase transfer mechanism and activation technique, see paragraph 3.2.1.4.

3.4.1.2 <u>Output Phase.</u> The flow of execution control continues with the user activating the UOI (on the RH side of the flowchart) via the main menu. As another activation technique, the IE might activate the KBI (on the LH side of the Output Phase) via the transfer of access parameters (see paragraph 3.2.1.4 for more description of both alternatives). In any event, the KBI must first decode the final goals (O2) received from the Search Phase into a format suitable for accessing the KB. The KBI performs this decoded (O2) entirely independent of the UOI which can, at the same time, be briefing the user on how to display his output frames.

The UOI begins by displaying the output overview (O1) which includes an output menu from which the user can select the next mode of operation. The system steps through the output overview under user control, allowing him to "escape" via the ESC key back to the output menu. At all times, the system must continuously monitor the function keys (O3) to allow the user to change modes at his discretion. From the output menu, the user must use the function keys to select at least between displaying the output (via the GUIDELINE mode) or updating the KB (via the DATA ENTRY mode).

Assuming the user wants the former option, then the UOI responds by entering the GUIDELINE mode and presenting the first frame of guidelines. At any time from this point on, the user may inquire (via the function keys) as to what supporting criteria there are for the current guideline (being displayed at the current cursor position). The UOI response to this will be discussed after first considering the typical functional flow through the system for reviewing guidelines.

Assuming no inputs are made via the function keys (S2), the UOI monitors the keyboard and/or cursor (S6), awaiting the user's response to the guideline being displayed. Upon user request, the UOI attempts to step to the next guideline (O6). However, in doing this, the UOI must determine whether the user has reached the end of the current frame. If not, the UOI merely displays the next guideline (O8), as usual, which may involve "scrolling" the earlier guidelines up the screen.

If it is the end of the frame, the UOI must obtain the next frame via the KBI before it can continue. Therefore, the UOI transfers control to the KBI with a request for next frame, whereupon the KBI attempts to step to the next frame (O6). If the next frame is located in memory (i.e., it lies within the last "overlay" brought into memory), then the "access parameter" for that frame is passed back to the UOI. This allows the UOI to step to the first guideline in the identified frame (O6) and continue on, as above, with display of that guideline (O8).

If the frame is not in memory, then the KBI must step to the next database (O6) and activate the knowledge base (KB) in the ACCESS mode (on the LH side of the Output Phase in the flowchart). The KB responds, in turn, by accessing the next database and transferring it to memory, whereupon the KBI can step to the first frame (O6) in that database. This cycle of "next" data segment access allows the UOI to continue on, as above, with display of the first guideline in the next frame (O8).

Thus, the entire GUIDELINE cycle is accomplished via functions (O3, O5, O6, O7, O6, O8, O3) sequentially. This cycle is repeated until each guideline scheduled for output in the Output Phase has been displayed; that is, until all access parameters received from the Search Phase (corresponding to frames "pending" display) have been exhausted. Moreoever, the user can accelerate through the scheduled output at any time by requesting "next frame" or "next database" via functions (O3, O6, O7, O6), as he wishes. If there are no more guidelines to display, then the UOI displays an output summary (O9) which capsulates the user's output summary descriptors. Thus, the exit path for the GUIDELINE mode is accomplished via function (O9), whereupon the Output Phase goes to STOP (at the lower RH side of the diagram).

As mentioned earlier, the user can at any time request the supporting criteria for the current guideline on display. The UOI responds to this request by entering the CRITERIA mode and activating the KBI to step to the next pertinent frame of criteria (O6) for subsequent display to the user (O8) by the UOI. Depending on program design, the KBI may have to step to the next CRITERIA database segment (O6) to enable this feature. Essentially, then, the CRITERIA cycle observes the exact same functions (O3, O5, O6, O7, O6, O7, O8, O3) in the same sequence as the GUIDELINE mode. As with he guidelines above, this cycle is repeated until all criteria pertaining to the current guideline have been displayed, whereupon the UOI reverts back to the GUIDELINE moc and display screen.

3.4.1.3 Update Phase. As mentioned above at paragraph 3.4.1.1, the flow of execution begins with the UII displaying the search overview (S1) including a search menu (see the LH side of the flowchart). The updating function is offered to the user as a first choice on the menu to promote timely and efficient data updates into the system. Therefore, because of its high priority, the Update Phase is performed offline independently, such that it preempts any user attempt to conduct a search or display output until all updating has been completed (see the priority of updating in paragraph 3.4).

The Search Phase starts by displaying the search menu, as described above for the QUERY mode. Assuming the user is a KE who wants to update the IE via the search menu (S1), then the UII responds by entering the RULE ENTRY mode and transferring execution control to the Rule Generator (RG) (on the lower LH central portion of the flowchart). Upon such activation, the RG interacts with the KE to update the IE (S3), as described at paragraph 3.2.1.2. The RG must, in turn, activate the IE in the INSERTION mode to enable it to store the new rules/goals (S3) that are subsequently submitted to it. Upon storage of all rules/goals, the IE reverts execution control back to the UII which resumes display of the main menu (S3). Thus, the entire UPDATE cycle for the IE is accomplished via functions (S1, S2, S3, S1) sequentially.

Updating in the Output Phase follows a functional pattern which is virtually a mirror image of the Search Phase process. Asssuming the user is a KE who wants to update the KB via the output menu (O1), then the UOI responds by entering the DATA ENTRY mode and transferring execution control to the data acquisition subsystem (KAS) (on the lower RH central portion of the flowchart). Upon such activation, the KAS interacts with the KE to update the KB (O4), as described at paragraph 3.2.2.2. The KAS must, in turn, activate the KB in the STORAGE mode to enable it to store the new data elements (O4) that are subsequently submitted to it. Upon storage of all new data, the KB reverts execution control back to the UOI which resumes display of the output menu (O4). Thus, the entire UPDATE cycle for the KB is accomplished via functions (O1, O3, O4, O1) sequentially.

3.4.2 Program Interrupt Control

This paragraph identifies all program interrupts that serve to effect execution control among the following classes:

> interrupts external to the system which permit extra system control (e.g., initial ACTIVATE signals, user mode changes, etc.)
>
> interrupts external to each module which permit inter-module control (e.g., next rule ready for display, next database ready for access, etc.)
>
> interrupts internal to each module which permit intra-module control (e.g., explanation completed, access parameters decoded, etc.)

Table 3-11 is a summary of all HF-ROBOTEX interrupts that fall into the above three classes. The table pulls together all pertinent information about each interrupt, including its associated system function (S1,..., S9) or (O1,..., O9), its input source, output destination, intended purpose, and the response expected from the interrupted module.

To permit convenient correlation with earlier drawings in this section, this table further associates the operating modes and priority levels (discussed in paragraph with respect to Figure 3-10) with each interrupt and shows which interrupts cause a change in modes. Moreover, each interrupt in this table corresponds to a specific "connecting" arrow in the general functional flowchart (Figure 3-16 of the preceding paragraph).

Likewise, every change of modes implied by the earlier description in this section is reflected by a separate interrupt in Table 3-11 dedicated to that purpose. For example, as can be seen in the table at a "KE request" to update the IE forces the system to change from QUERY mode (level D) to RULE ENTRY mode (level B). Furthermore, in response to the S2A interrupt, the receiving module RG is activated to begin "rule entry," thereby effecting transfer of execution control from the UII to another module, as well.

TABLE 3-11. SUMMARY OF PROGRAM INTERRUPTS

INTERRUPT FUNCTION/TITLE	INPUT SOURCE/MODE/LEVEL	OUTPUT DESTINATION/MODE/LEVEL	INTERRUPT PURPOSE	MODULE RESPONSE
S1 display search overview				
A ACTIVATE Search Phase	DOS — —	UII QUERY D	activate upon Insight 2+ load	display search menu
S2 monitor function inputs				
A KE request	UII QUERY D	RG RULE ENTRY B	KE request to update IE	activate RG
B user inquiry	UII QUERY D	ES EXPLANATION E	user request for explanation, definition, or help message	activate ES
S3 update inference engine				
A rule entry ready	RG RULE ENTRY B	IE INSERTION A	rule typed in for insertion into IE	activate IE to store
B update completed	IE INSERTION A	UII QUERY D	revert back to QUERY mode upon insertion of all rules entered by KE	display search menu
S4 access definitions/explanations				
A explanation ready	ES EXPLANATION E	UII EXPLANATION E	revert back to QUERY mode upon user review of retrieved explanation	display current rule
S5 display definitions/explanations				
A explanation completed	UII EXPLANATION E	UII QUERY D	requested explanation ready	display explanation
S6 monitor keyboard/cursor inputs				
A user response	UII QUERY D	IE SEARCH C	user response to last rule ready	activate IE to search
S7 access next subgoal/rule				
A next rule/goal	IE SEARCH C	UII QUERY D	next rule/goal from the last response ready	display next rule/goal
S8 display current subgoal/rule				
A final goals reached	UII QUERY D	IE SEARCH C	last set of subgoals are final goals	encode final goals
B EXIT from Search Phase	UII QUERY D	DOS — —	return to DOS to allow next load	load dBASE III
S9 encode final goals				
A final goals ready	IE SEARCH C	UII QUERY D	final goals encoded as access parameters	display final goals
B access parameters	IE SEARCH C	KBI ACCESS B	encoded goals ready for transfer	accept transferred goals
C ACTIVATE Output Phase	IE SEARCH C	UOI GUIDELINE D	activate upon completion of transfer	activate UOI
O1 display output overview				
A ACTIVATE Output Phase	UOI GUIDELINE D	KBI GUIDELINE D	activate upon initial dBASE III load	display output menu
O2 decode final goals				
A access parameters decoded	KBI ACCESS B	KBI GUIDELINE D	encode goals from Search Phase ready for decode into KB access parameters	decode final goals
O3 monitor function inputs				
A KE request	UOI GUIDELINE D	KAS DATA ENTRY B	KE request to update KB	activate KAS
B request for NEXT FRAME	UOI GUIDELINE* D	KBI GUIDELINE* D	user request to accelerate to next frame	step to next frame
C request for NEXT DATABASE	UOI GUIDELINE* D	KBI GUIDELINE* D	user request to accelerate to next database	step to next database
D request for CRITERIA	UOI GUIDELINE D	KBI CRITERIA E	user request to switch to CRITERIA mode to review criteria for current guideline	access criteria frame
E request for GUIDELINE**	UOI CRITERIA E	KBI GUIDELINE D	user request to revert back to GUIDELINE mode upon review of supporting criteria	display current guidline
O4 update knowledge base				
A data entry ready	KAS DATA ENTRY B	KB STORAGE A	data typed in for storage in KB	activate KB to store
B update completed	KB STORAGE A	UOI GUIDELINE D	revert back to GUIDELINE mode upon storage of all data entered by KE	
O5 monitor keyboard/cursor inputs				
A next data element ready	UOI GUIDELINE* D	UOI GUIDELINE* D	normal sequential advance by user	step to next element
O6 advance to next data element				
A request for NEXT FRAME	UOI GUIDELINE D	KBI GUIDELINE D	normal sequential advance by UOI	step to next frame
B request for NEXT DATABASE	KBI GUIDELINE* D	KB ACCESS* C	KBI request for next KB access	step to next database
C next frame ready	KBI GUIDELINE* D	UOI GUIDELINE* D	next frame from last request ready	step to first element
O7 access next data element				
A next database ready	KB ACCESS* C	UOI GUIDELINE* D	next database from last request ready	step to first frame
O8 display next data element				
A final data element reached	UOI GUIDELINE D	UOI GUIDELINE D	last guideline is final guideline	display output summary
O9 display output summary				
A EXIT from Output Phase	UOI GUIDELINE D	DOS — —	return to DOS as final exit from system	

*this interrupt pertains to prevailing output mode (GUIDELINE or CRITERIA)

**this includes initial request for first guideline out of output menu

Moreover, interrupt S2A provides an example of the HF-ROBOTEX priority interrupt scheme. Since the priority level of the RG in RULE ENTRY mode (level B) is higher than the UII in QUERY mode (level D), no further user inquiries will be honored from the function keys (e.g., via interrupt S2B) until the update has been completed by the KE (interrupt S3B) and the UII reverts back to the QUERY mode at priority level D.

The above examples represent the most significant impact that these program interrupts have on control design requirements. Namely, once the system has been initially activated, each interrupt must accomplish one of the following purposes:

ACTIVATE MODULES — activate another module for a specific function; or upon completion of that function, revert back to the calling module (typically, the UII or UOI);

and/or

CHANGE MODES — change operating modes to a higher priority level; or, upon completion of all higher-level functions, revert back to the original level (typically, level D for the QUERY mode or GUIDELINE mode);

and/or

USER REQUEST — honor any user request to shift to a higher level or to a different function on the same level; otherwise, inhibit any request from or to a lower level until all higher-level functions have been completed (typically, a user attempt to interrupt an IE access).

The above actions could take place either at user request, or at the end of the ordained sequence of functions along the system's functional flow (as shown in the earlier diagram). IN any event, the user can at any time on level D (i.e., while in QUERY mode or GIDELINE mode) exit from the current phase by hitting EXIT function key F7 which forces an immediate termination (for system EXIT procedures, see description of function F7 under paragraph 3.2.1.1 and 3.2.2.1).

3.4.3 Subprogram Reference Control

This paragraph describes the control logic involved in referencing each system module, as an outgrowth of the functional requirements discussed in paragraph 3.4.1 and timing constraints imposed by the program interrupt logic presented in paragraph 3.4.2. As a preface to this description, reference should be made to the earlier assignment of interrupt in Table 3-11. Hence, there is no need for further discussion here of priority assignments.

Reference should also be made to the cycle times imposed on the modules by the original performance requirements (S10, O10) delineated value-by-value in paragraph 3.1.1 and function-by-function in paragraph 3.2. However, it should be noted at the outset that, owing to the completely sequential flow of execution illustrated in Figure 3-16, there are no significant timing constraints that arise from the HF-ROBOTEX control logic suggested herein. Hence, discussion in this paragraph will focus on the few timing considerations that arise in transitioning between the Search Phase and the Output Phase.

3.4.3.1 Critical Path Flow Diagram. This paragraph presents a more detailed functional flow diagram to clarify the system control logic. The preceding description in paragraph 3.4.1 synopsized the functional flow of execution control and program data through the system. For emphasis and clarity here, this paragraph will focus on control logic governing the specific functions along the "critical path" through the system. This path was initially identified in Figure 3-15 and subsequently traced in Figure 3-16, function-by-function, as follows:

PHASE	MODE	MODULE	CRITICAL FUNCTIONS
SEARCH	QUERY	UII	S2 monitor function inputs
	QUERY	UII	S6 monitor keyboard/cursor inputs
	SEARCH	IE	S7 access next subgoal/rule
	QUERY	UII	S8 display current subgoal/rule
	SEARCH	IE	S9 encode final goals
OUTPUT	GUIDELINE	KBI	O2 decode final goals
	GUIDELINE	UOI	O3 monitor function inputs
	GUIDELINE	UOI	O5 monitor keyboard/cursor inputs
	ACCESS	KB	O6 advance to next data element
	GUIDELINE	UOI	O8 display next data element

Figure 3-17 shows the functional flow along the critical path through the system in greater detail. For example, after the initial function to "display search overview (S1)" in the upper LH corner of the Search Phase on Figure 3-17, there are four parallel cycles of "display" and "search" functions which follow in the center. These are actually constituent subfunctions of "display current subgoal/rule (S8)" for the UII and "access next subgoal/rule (S7)" for the IE, respectively. The interrupts out of the "display" functions are typically the user's keyboard response to the current rule being displayed (e.g., as a simple case, the user hits "RETURN" to send the statement at the current cursor position as a part of "search" command to the IE). The interrupts out of the "search" functions are typically the system's retrieval of the next rule based on the last user response (e.g., as a simple case, the IE returns to the UOI the next rule for which the proposition "matches" the last response).

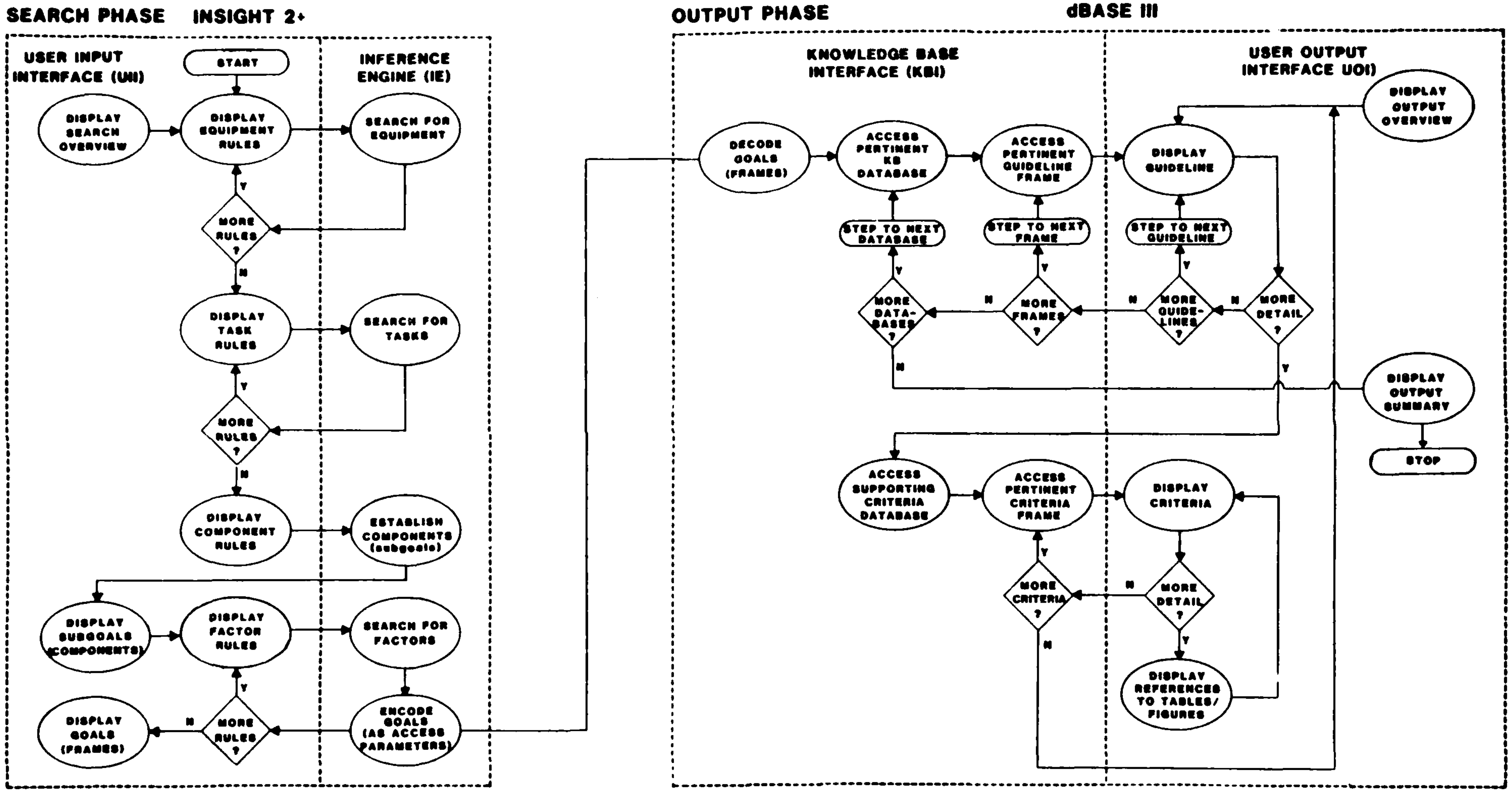


FIGURE 3-17. FUNCTIONAL FLOW DIAGRAM FOR CRITICAL PATH

Similarly, following the initial function to "decode final goals (O2)" in the upper LH corner of the Output Phase, there are two parallel cycles of "steps", "access", and "display" functions which follow in the center. These are actually constituent subfunctions of "step to next data element (O6)" for the KB, "access next data element (O7)" for the KBI, and "display next data element (O8)" for the UOI, respectively. The interrupts out of the "steps" functions are typically commands to locate data frames stored in memory (e.g., as a simple case, the KBI issues to the KB a SEEK command referencing a "memvar" storing the desired access parameters). The interrupts out of the "access" functions are typically output "pointers" to the desired frame (e.g., as a simple case, the KBI issues a RETURN back to the UOI upon loading the desired frame in a common display area). However, the specific control logic required for each for each situation is strictly a matter of program design choice.

Thus, Figure 3-17 serves not only to focus on the functional flow along the critical search/output path, but also, to break down the major functions involved into a coherent framework of subfunctions. As just illustrated, the interrupt mechanisms range from convential commands which transfer control to a system function for the duration of that command (e.g., via a SEEK command), all of the way to specific activation procedures which transfer control to a subprogram for the duration of several programmed functions (e.g., via an ACTIVATE command passing multiple data parameters).

Of particular note in this flowchart are the decision blocks (contained in the "diamond" symbols) which direct the flow from one subfunction to another, or from one mode to another. For example, the decision asking "more rules?" on the LH side dictates to what level in the rule hierarchy the UII and IE must move next (see Figure 3-8 for more detail). Similarly, the decisions asking "more (guidelines, frames, databases)?" on the RH side dictate to what segment in the data hierarchy the UOI and KBI must move next (see Figure 3-4 for more detail). Finally, the decisions asking "more detail?" on the RH side dictate to what major data level the UOI and KBI must move next (see Figure 3-5 for more detail). This illustrates still another mechanism for defining cyclic control logic on a modular and submodular basis.

3.4.3.2 Control Logic Timing Constraints. This paragraph describes the few timing constraints imposed by the program interrupt logic. The original performance requirements (S10, O10) which imposed MAX cycle times at paragraph 3.1.1, have already been addressed under paragraph 3.2 at each point where a function was indexed by an "S10" or "O10". Hence, the given timing constraints and their resolution in HF-ROBOTEX need not be discussed again here.

What remains as timing constraints for consideration here are the vital functions to ENCODE FINAL GOALS (S9) and to DECODE FINAL GOALS (O2) during the transition between Search Phase and Output Phase. These functions were time-constrained to 10 seconds each by requirements S10 and O10 respectively. However, both requirements also allocated an additional MAX cycle time to relate STOP/START functions (i.e., 10 seconds to DISPLAY FINAL GOALS (S8) and 5 seconds to DISPLAY OUTPUT OVERVIEW (O1)).

In view of this, a "workaround" option was recommended under paragraphs 3.2.1.1 and 3.2.2.1 in the event that either ENCODE or DECODE should exceed its 10-second MAX cycle time. The underlying idea was that the time-critical ENCODE/DECODE functions could "absorb" some of the additional STOP/START time allocated to the final UII function S8 and initial UOI function O1, respectively. Such a time overlap can be accomplished in a number of ways:

PARALLEL PROCESSING OPTION	Since the ENCODE/DECODE functions are performed by different modules (IE/KBI) than the STOP/START functions (UII/UOI), it may be more effective to activate the IE/KBI modules independent of, but in parallel with, the UII/UOI modules. This would permit the IE to ENCODE within up to 15 seconds and the KB to DECODE within up to 20 seconds.
TIME-SHARED PROCESSING OPTION	Since the STOP/START functions involve the user "paging" through a number of sequential display screens, it may be more effective to activate the IE/KBI modules upon each new dispay screen. This would permit the ENCODE/DECODE functions to "time-share" the initial 5- or 10-second display cycle, plus each additional 1-second display cycle allocated by S10/O10 thereafter.

SEGMENTED PROCESSING OPTION	Since the user may be "paging" through a number of screens for the STOP/START functions, as just mentioned, it may be advantageous to activate the ENCODE/DECODE functions sequentially after presenting each new display screen. This would permit the IE/KBI to effectively operate during the user portion of the display cycle, but at a lower level of priority to permit "next page"-type interrupts by the UII/UOI.
MULTIPLEXED PROCESSING OPTION	Finally, as a default to all of the alternative techniques above, the ENCODE/DECODE functions could be "multiplexed" across their successive IE/KB accesses made prior to ENCODE time and after DECODE time, such that the successive goals reached and access parameters requested would be processed as they arose in each access cycle. This would permit the IE/KBI to "absorb" some of the successive 3-second MAX cycle times allocated by S10/O10 to such accesses.

Beyond these ENCODE/DECODE considerations, there is but one other prospective candidate that may be time-constrained by requirement O10; namely, the 5-second MAX access time to access the entire CRITERIA database. A "workaround" option was recommended at paragraph 3.2.2.1 for UOI function 07, at paragraph 3.2.2.3 for the KBI, and at paragraph 3.2.2.4 for the KB. The underlying idea was that, during the time-critical ACCESS NEXT DATABASE function 07, the KB would regard the CRITERIA database as divided naturally into 7 segments, where each of the 7 segments represented a different Human Factor. To do this, the KBI would have to organize and maintain a simplified CRITERIA "output string" for the 7 segments, just as with the GUIDELINE output string described at paragraph 3.2.2.3. The net result would be a nominal CRITERIA segment size of 62K bytes which could be readily transferred within the 5-second time alloted (see Table 3-7 for supporting data).

3.4.4 Special Control Features

The HF-ROBOTEX system does not have any special control requirements that are outside of the normal operational functions already described above. Hence, this PDS paragraph is not applicable.

3.5 PROGRAMMING GUIDELINES

This section will describe the programming guidelines that should be observed by the system programmer when implementing the HF-ROBOTEX program modules. This section will further identify the programming language and supporting system recommended to implement the modules, including the mnemonic labeling conventions to be observed during system development.

The first of these considerations, programming guidelines for the programmer, is vastly simplified here by the fact that all such guidelines have already been integrated with the detailed description to which they pertain throughout the PDS. Hence, a detailed description is not needed here; rather, this secton will merely summarize the categories of guidelines already presented and indicate where they appear. Table 3-12 comprises such a programming guideline summary, providing cross-references to specific paragraphs of the PDS where each category of guideline can be found.

Likewise, the second of these considerations, programming language and supporting system, is also vastly simplified here by the fact that the most viable candidates for HF-ROBOTEX implementation, Insight 2+ and dBASE III, are self-contained systems with their own compilers, editors, utilities, etc. Hence, a detailed description is not needed here; rather, reference is made to the reference manual for Insight 2+ and dBASE III cited earlier as applicable documents under Section 2.

It should be noted that Insight 2+ has its own PASCAL compiler called DBPAS, which allows the programmer considerable flexibility in programming, for example, an independent ENCODE routine at the end of the Search Phase. It should also be noted that the cited dBASE III reference manual has a number of "canned" subroutines in its appendices which may be useful in program development and/or in generating overview/summary display screens.

TABLE 3-12 HF-ROBOTEX PROGRAMMING GUIDELINES (cross-references)

MODULE/ PARAGRAPH	PERTINENT FUNCTION	PROGRAMMING GUIDELINES (INTEGRATED WITH DESCRIPTION)
UII	S1 display search overview	HF display considerations
3.2.1.1	S2 monitor function inputs	HF keyboard considerations
	RESTART (F2)/EXIT (F7)	user abort safeguards
	S6 monitor keyboard inputs	S10 timing constraints
	S8 display inference rules	S10 timing constraints
	S9 encode final goals	S10 timing constraints
IE	S7 search component goals	ENCODE algorithms
3.2.1.4	S9 encode final goals	parameter transfer mechanisms
UOI	O1 display output overview	O10 timing constraints
3.2.2.1	O2 decode final goals	O10 timing constraints
		parameter transfer mechanisms
	O3 monitor function inputs	HF keyboard considerations
	RESTART (F2)/EXIT (F7)	user abort safeguards
	BACKUP (F1)/NEXT DB (F3)	CRITERIA DB segments
		memory overlays
	O5 monitor keyboard inputs	user response configuration
	O7 access knowledge base	CRITERIA DB segments
	O8 display guidelines/criteria	O10 timing constraints
KBI	O2 decode final goals	dBASE III considerations
3.2.2.3		dBASE III acivate mechanism
		DECODE algorithms
	O7 access knowledge base	DB integrity safeguards
		O10 timing constraints
KB	knowledge base description	CRITERIA segments
	O7 store retrieve data records	DB integrity safeguards
	O7 search guideline frames	O10 timing constraints
		memory overlays
3.3	IE/KB storage allocation	memory overlays
3.3.2	KB estimates	CRITERIA segments
3.4	Program functional flow	update/access priority
3.4.3.2	Control logic timing constraints	S10 ENCODE constraints
		O10 DECODE constraints

Finally, after preliminary implementation tradeoffs are analyzed, it may prove more performance-effective to implement the HF-ROBOTEX modules by programming them rather than relying on off-the-shelf components like Insight 2+. If such is the case, then the following unique mnemomic prefixes should be affixed to any external subprogram titles (followed by unique name) and any internal statement labels (followed by a unique number):

SEARCH MODULE PHASE	LABEL PREFIX	OUTPUT MODULE PHASE	LABEL PREFIX
UII	II	UOI	OI
RG	RG	KAS	KS
ES	ES	KBI	KI
IE	IE	KB	KB

4.0 QUALITY ASSURANCE (QA)

This section defines review procedures for verifying , as an assurance of quality, that the program design conforms to the requirements set forth in this PDS. Such procedures (referred to hereafter as "QA testing") are actually tests run on the finished product, or parts of the product, after the programmer has designed, implemented, debugged, and tested the program to his satisfaction. It should be established at the outset that a separate quality assurance (QA) manager, independent of the program team, should perform all QA testing at all levels to help ensure strict compliance of the final product. Also, as much as possible, QA testing should be performed on a module-by-module basis, as each module emerges from the development cycle to help ensure the full scope of each module's functionality prior to system-level integration.

The underlying concept behind the QA procedures outlined herein is to test through isolation, as much as possible, the functions of each module on a modular basis, even when integrated into the full system. The idea behind this is that, if the modules still check out individually even when integrated, then they are more likely to perform satisfactorily throughout the subsequent system-level tests of all the modules in the chain. This same philosophy should be carried down to the submodule level, again, as much and as soon as possible, particularly along the system's critical path (e.g., for development and preliminary QA testing of the critical ENCODE/DECODE functions prior to full IE and KBI testing at the module level). This will help to ensure that at least the functions along the predominantly-used critical path are in order long before full system testing has begun. Thus, the level and scope of QA testing is commensurate with the modular development of each submodule prior to integration into the parent module, and, in turn, each module prior to integration into the overall system.

The underlying concept behind this modular QA approach is to exhaustively test each function as soon as possible, particularly along the critical path, in an effort to reduce the need for more costly, exhaustive testing at the system level later. That is, by testing a given function exhaustively earlier in the development cycle, the chances of encountering "combinatorial" and "cumulative" errors between and among multiple interactive modules (which are much more difficult to detect, diagnose, trace, and resolve) is drastically reduced. Thus, in light of the exhaustive testing of isolated functions at the submodule level, there is less need for more costly and time-consuming testing of the same functions at the module level; similarly, in light of exhaustive testing of internal functions at the module level, there is less need for even more costly and time-consuming testing of modular functions at the system level.

The net effect of the underlying QA concepts taken together is that, upon final system integration (when the modules are finally working as a single, coherent Expert System), the most difficult system-level QA testing can be confined to exhaustive testing of far fewer system-level functions. Such QA testing would primarily address "interactive" functions that involve cooperation among two or more modules (e.g., UOI access of the "next database" from the KB via parameters from the KBI), and "parallel" functions that involve time synchronization of independent modules operating in parallel toward a mutual "deadline" (e.g., UOI displaying the output overview while the KBI decodes access parameters prior to the UOI's initial request to access the first guideline).

4.1 SUBMODULE-LEVEL QA TESTING

QA testing at the submodule level is in one regard the most difficult because, invariably, a given submodule does not have inputs and/or outputs that can be conveniently generated, modulated, or even recognized by the QA manager. That is, a typical output function such as DISPLAY CURRENT SUBGOAL/RULE (S8) conveniently shows its outputs but, without some other external control to manipulate what rule becomes "current", does not provide a view of its inputs. Similarly, a typical input function such as MONITOR FUNCTION INPUTS (O3) conveniently shows its inputs but, without some other external control to monitor what each function key "stimulates" as a succeeding function, does not provide a view of its outputs. Moreover, a typical processing function such as ACCESS KNOWLEDGE BASE (O7), without some other external controls, obviously does not reveal its inputs or its outputs. Hence, for submodule QA testing, the QA manager must often devise special input and/or output controls to manipulate the input to a given submodule and/or to monitor its resulting output.

Once such external controls are in place, the QA manager must devise a test scheme that varies each input through the range of its expected values (e.g., entering a series of guidelines of 1 to 200 characters in length), and thereafter, to the point just beyond its "legal" range (e.g., a guideline with no characters and another with 201 characters). Such testing is considered "exhaustive" because it "pushes" the given submodule right to, and just beyond, its limits. Once this level of testing is accomplished, the same type of test does not have to be applied at the module level.

As an example of testing at this level, the IE module is designed to ENCODE FINAL GOALS (S9) as described in detail at paragraph 3.2.1.4. The ENCODE submodule within the IE would be tested in part by "exercising" the ENCODE function across the entire scope of the Logic Table for Component Subgoals (Table 3-4) recommended also at paragraph 3.2.1.4. Such a QA test would vary the input across the X-axis (EQUIPMENT categories). Each matrix slot in Table 3-4 contains 2-3 specific COMPONENT categories which correspond to the given X/Y input "combinations". The QA manager would have to verify that each successive X/Y inputs produced their expected COMPONENT outputs exactly as shown in the Table.

Once such testing is successfully completed, the QA manager could "sign off" on the ENCODE function, and move on to module-level testing.

4.2 MODULE-LEVEL QA TESTING

QA testing at the module level has now become much less encumbered by virtue of exhaustive testing at the submodule level. The QA manager should now have a fully coherent module to test, and, with strategic "input" and "output" functions already tested, he no longer has to devise special I/O controls to "stimulate" and "monitor" the tests.

Nevertheless, the QA manager must still devise a test scheme that, once again, varies each input to the module through the range of its expected values (e.g., entering a series of guidelines that reference from 1 to 20 criteria in a CRITERIA DB comprising 20 elements), and again, thereafter, to the point just beyond its legal range (e.g., a guideline referencing no criteria and another referencing 21 criteria). Such testing is considered exhaustive, again, because it "pushes" the given module right to, and just beyond, its limits. Once this level of testing is accomplished, the same type of test does not have to be applied at the system level.

As an example of testing at this level, the IE module is designed to SEARCH EQUIPMENT/TASK RULES (S7) as described in detail at paragraph 3.2.1.4. The IE module could be tested in part by "exercising" its SEARCH function across the entire set of EQUIPMENT and/or TASK rules by simpy responding with "don't know" or "don't care" responses at each sublevel of rules. This should force the IE to return, as a series of "next rules" to the user, every rule that has been entered at the next lower sublevel (which the QA manager has complete control over).

This testing is vastly simplified by the fact that, if the submodule testing has been correctly sequenced, then the KAS can be used to enter "dummy" rules into the IE (via function S3) and the UOI can be used to display the "dummy" test results (via function S8). Once such testing is successfully completed, the QA manager can "sign off" on the IE module, and move on to the system-level testing.

4.3 SYSTEM-LEVEL QA TESTING

QA testing at the system level has now become dramatically less encumbered by virtue of exhaustive testing at, first, the submodule level and, secondly, the module level. The QA manager should now have a fully coherent Expert System to test, and, with all modular I/O functions exhaustively tested, he no longer has to devise a test scheme that varies each possible input across its range of expected values. In fact, the QA manager can now simply apply the earlier-devised input tests to the integrated system and monitor its response acrosss the entire network of modules. Obviously, the results should remain the same since these were, by definition, not "interactive" or "parallel" functions in the first place. And, if they are not the same, then the "culprit" module causing the error is readily identified.

As an example of testing at this level, the IE module is designed to warn the user during the above SEARCH function (07) that his output is becoming excessive (i.e., it is exceeding 20 frames) so that he can selectively narrow down his query. The system could be tested in its entirety in part by "exercising" the IE up to and past the point of this warning. Once again, the QA manager could "stimulate" this test by "don't know" responses, but, this time, the test would span all three system levels and, more importantly, the IE would be operating on the final rule structure accessing a facsimile of the actual KB. This should force the IE to return a warning each time the "expanding" query stimulated a set of goals equivalent to more than 20 frames of output.

This test procedure should continue expanding the scope of the query until the search fails entirely at the IE's "cutoff" limit of 40 frames. At this "disastrous" extreme of system performance, the QA could examine the IE's recovery procedures, as well. Once such testing is successfully completed, the QA manager can "sign off" on the entire HF-ROBOTEX system and submit it to the intended user.

5.0 PROGRAM DEVELOPMENT NOTES

This PDS was developed around Insight 2+ and dBASE III, using the best information available at the time of writing. As with many new software products on the market, the Insight 2+ Expert System continues to evolve as a commercial product of greater appeal to a wider audience. Many of the configurations and constraints written into this PDS were based on the features projected by the company developing the Insight Expert System, Level Five Research. However, the newest version, just released as Insight 2+, differs to some degree from what was announced by the company earlier.

Hence, there are some variances between Insight 2+ and this PDS in the type and format of functions dedicated to each operating modes and in the system's operating limits, as reflected by the letter regarding Insight 2+ configuration in Figure 5-1. Some of these variances have already been incorporated in this PDS (such as an increase from 400 rules MAX searched at a nominal rate of 200 rules per second, up to 2000 rules MAX searched at a rate of 400 rules per second). Other variances, such as the type and format of certain functions in each operating mode, could not be incorporated prior to publishing this PDS. However, as reflected by the letter in Figure 5-1, any particular functional variance can be reconciled by working directly with the manufacturer, Level Five Research, should the need arise.

PERSON-SYSTEM INTEGRATION

Human Factors - Systems Analysis

PSI

2401 HUNTINGTON AVENUE ALEXANDRIA, VIRGINIA 22303-1531

(703) 960-5555

April 3, 1986

Mr. Cornelius Willis
Level Five Research
503 - 5th Avenue/Suite 201
Indialantic, FL 32903

Dear Mr. Willis:

We are excited about your new upgrade product, Insight II Plus, which was released this week. Late last year, we decided to employ Insight II Plus in the design of a prototype expert system for applying human factors to robotics design. Our decision was based on crucial improvements over Insight II, represented to us by your lead designer Mr. Henry Seiler; and the anticipation that it would be released in January.

We fully understand the evolving nature of your new product and welcome the advantages that arise from each improvement; however, many of these improvements have affected our design dramatically. These variances generally center around the type and format of functions dedicated to each operating mode, and the system's operating limits such as the number of rules and levels allowed, number of goals to each level, number of nominal rules fired per second, number of parameters that can be passed externally, etc. While some of these improvements can be readily accommodated in our system design specification, others are not so easily changed to the configuration specified.

As you and I discussed today, it may become necessary to redesign some features of Insight II Plus should our customer desire to maintain the specified configuration (e.g., the type and format of certain functions (F1 - F7) dedicated to selected operating modes). Should this need arise, we will, of course, employ you as a consultant at a reasonable fee to help reprogram the Insight II Plus subroutines that will implement the desired functional variations from Insight II.

Thank you for your continuing cooperation with us. We wish you success with the new release.

Sincerely,

Jan E. Rhoads
Senior Programmer/Analyst

FIGURE 5-1. LETTER REGARDING INSIGHT 2+ CONFIGURATION

APPENDIX A

Glossary

NOMENCLATURE

ABBREVIATIONS & ACRONYMS

AI	Artificial Intelligences
AVG	Average Value
CTL	Control Key on Keyboard
DB	Database
DBMS	Database Management System
DBPAS	Extended PASCAL Compiler for Insight 2+
DOS	Disk Operating System
EOF	End-of-File indication
ES	Explanation Subsystem
ESC	Escape Key on Keyboard
F1-F7	Function Keys F1 through F7 on Keyboard
flowchart	General Functional Flow Diagram (Figure 3-16)
HF	Human Factors
IE	Inference Engine Module or Concept
IE screen	Screen Format for IE Rules (Figure 3-7)
I/O	Input/Output
K	Thousand Units (as in KB=1000 bytes)
KAS	Knowledge Acquisition Subsystem
KB	Knowledge Base Module or Concept
KB screen	Screen Format for KB Data (Figure 3-13)
KBI	Knowledge Base Interface Module
KE	Knowledge Engineer
LH	Left-Hand Side
M	Million Units (as in MB=mega bytes)
MAX	Maximum Value
memvars	Memory Variables in dBASE III
O1-O10	Performance Requirement (1,...,10) for Output Phase
O-A-V	Object-Attribute-Value Triplets
PCO	Suffix for PASCAL-compiled Program
PRL	Suffix Insight 2+ data file (uncompiled)
PDS	Program Design Specification
PRG	Suffix for dBASE III command file
PSI	Person-System Integration
QA	Quality Assurance
RD	Robotics Design
RETURN	Carriage Return on Keyboard
RG	Rule Generator
RH	Right-Hand Side
ROBOTEX	Robot Expert System (also HF-ROBOTEX)
S1-S10	Performance Requirement (1,...,10) for Search Phase
UII	User Input Interface Module
UOI	User Output Interface Module

DISTRIBUTION

Facility	Copies
Department of the Navy DTNSRDC Attn: Robert G. Stilwell Code 296 Naval Ship Research and Development Center Bethesda, MD 20084	1
Naval Sea Systems Command Attn: LCDR Bart Everett Office of Robotics and Autonomous Systems SEA-90G Washington, D.C. 20362-5101	1
Naval Sea Systems Command Attn: SEA-55W11 (Robert Bost) Habitability Branch Ship Arrangements Division Washington, D.C. 20362-5101	1
Office of Robotics and Autonomous Systems Attn: SEA-90G (Bill Butler) Naval Sea Systems Command Washington, D.C. 20362-5101	1
Naval Air Systems Command Department of the Navy Attn: AIR-351 (CDR David A. DuVal) Washington, D.C. 20361-5101	1
Naval Data Automation Command Attn: CAPT Ken VanLue Code 40 Washington Navy Yard Washington, D.C. 20374-1662	1
Naval Explosive Ordnance Disposal Technology Center Attn: John R. Butler Mechanical Engineering Indian Head, MD 20604	1

Facility	Copies
David Taylor Naval Ship R&D Center Attn: Harvey F. Knowles, Director Robotics Research and Development Bethesda, MD 20084-5000	1
Naval Post Graduate School Attn: Dr. David Smith Code 69HX Halligan Hall Room 207A Mechanical Engineering Dept. Monterey, CA 93940	1
Automation Technology Branch Attn: Walter W. Hankins III Nancy Orlando-Silwa Kathy Abbott NASA-Langley Research Center Hampton, VA 23665	 1 1 1
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Martin Marietta Aerospace Attn: Roger T. Schappell P.O. Box 179 Denver,CO 80201	1
Automation Technology Corporation Attn: Eugene B. Silverman 5457 Twin Knolls Rd. Columbia, MD 21045	1
American Robot Corporation Attn: Dr. Romesh Wadhwani, President 121 Industry Drive Pittsburgh, PA 15275	1

DISTRIBUTION

Facility	Copies
Litton-Ingalls Shipbuilding Attn: John M. Sizemore P.O. Box 149 Pascagoula, MS 39567	1
Odetics Inc. Attn: Tom Bartholet 1515 S. Manchester Ave. Anaheim, CA 92802	1
University of Cincinnati Attn: Ernest L. Hall, Ph.D. Director, Cincinnati Center for Robotics Research Department of Mechanical and Industrial Engineering 659 Baldwin Hall Mall Location #72 Cincinnati, OH 45221	1
Internal Distribution:	
E35	1
E231	1
E232	1
R402 (S. Hogge)	20

NOTES

END
DTIC
8-86