GODDARD

NASA Technical Memorandum 87821

The Development of a Prototype Intelligent User Interface Subsystem for NASA's Scientific Database Systems

William J. Campbell, Larry H. Roelofs,
and Nicholas M. Short, Jr.

JUNE 1987

NASA

NASA Technical Memorandum 87821

The Development of a Prototype Intelligent User Interface Subsystem for NASA's Scientific Database Systems

William J. Campbell
Goddard Space Flight Center
Greenbelt, Maryland

Larry H. Roelofs
Computer Technology Associates
McLean, Virginia

Nicholas M. Short, Jr.
Science Application Research, Inc.
Lanham, Maryland

National Aeronautics
and Space Administration

Scientific and Technical
Information Office

1987

Table of Contents

Table of Acronyms

AI	Artificial Intelligence
ART	Analytical Reasoning Tool
CRUDDES	CRUstal Dynamics Database Expert System
DBMS	DataBase Management System
DIS	Data Information System
EOS	Earth Observation System
GSFC	Goddard Space Flight Center
KEE	Knowledge Engineering Environment
LISP	LISt Processing (a symbol manipulation computer language)
IDM	Intelligent Data Management
IDMC	Intelligent Data Management Controller
IUI	Intelligent User Interface
NSSDC	National Space Science Data Center
NLQP	Natural Language Query Processor
SAIS	Science and Applications Information System
SLR	Satellite Laser Ranging
VLBI	Very Long Baseline Interferometry

CHAPTER 1

INTRODUCTION

In the past decade, operations and research projects that support a major portion of NASA's overall mission have experienced a dramatic increase in the volume of generated data and resultant information that is unparalleled in the agency's history. The effect of such an increase is that most of the science and engineering disciplines are suffering from an information glut, which has occurred, not only because of the amount, but also because of the type of data being collected (generally spatial and most often continuous in nature such as images, maps, two and three dimensional drawings and figures).

This information glut is growing nonlinearly and is expected to continue to grow in this fashion for the foreseeable future. Consequently, it is becoming physically and intellectually impossible to identify, select, and access the most suitable information specifically applicable to the various engineering and research projects of interest. For example, in the earth sciences such vast amounts of data are now being collected and/or are available (e.g., satellite images) that it now exceeds the ability of the professionals in the field to process, manage, and study it . In addition, the number of professionals in the application disciplines is not expected to increase significantly enough to resolve this data problem in the foreseeable future. Thus, the dilemma arises that the amount and complexity of information has exceeded and will continue to exceed, using present information systems, the ability of the scientists and engineers to understand and take advantage of this information.

Based on the scope, expected growth and dominance of this problem, it is anticipated that the future ability of NASA to function and perform meaningful space and earth related research will be significantly affected by its inability to manage and use its collected information to derive knowledge. Consequently, it is envisioned that dramatically different approaches to data management will be needed if earth and space related operations and scientific investigations are ever to take full advantage of the information and data being collected and stored.

Considering the trends of present computer science technologies, it appears that an approach to data management that employes Artificial Intelligence (AI) coupled in a distributed environment with powerful super mainframe, and micro based computer systems offers a reasonable solution for resolving many present and future data management problems. It is expected that such a system will, when operational, be able to:

- Deal with the massive data ingest and update problem,
- Manage concurrently both spatial and meta data in a logical manner,
- Provide intelligent user access to database systems and services,

- Support large numbers of users interactively over a distributed environment.

Based on the above concept an approach for the development of such a system has been formulated and a demonstration prototype completed for an operational science research database. It is the purpose of this paper to present the results of the prototype demonstration effort.

CHAPTER 2

BACKGROUND

In October of 1984, the Office of Aeronautics and Space Technology provided funding to Data Management Systems Facility, under the National Space Science Data Center (NSSDC), to initiate a research effort to develop a new generation of data management technologies to support the needs of NASA's future operational, engineering, and scientific programs thorugh the beginning of the twenty-first century. The research effort is the Intelligent Data Management (IDM) project which has the following long term research goals:

- to develop very powerful database management systems using advanced technologies,
- to develop intelligent value-added services that will enable users to interact with the most complex database systems with minimal understanding of the systems architecture, stored data, or query language,
- to allow automatic data ingest and maintenance with minimal user guidance and interaction,
- to manage symbolic and spatial data in the same database systems.

The IDM project has been addressing those areas of data management that could potentially be improved by the applications of artificial intelligence technologies including:

- User Interfacing,
- System management and control,
- Spatial database management,
- Automatic data ingest and system maintenance,
- Dynamic database systems using AI design concepts.

A top level diagram of the IDM concept that includes each of the above areas, except dynamic database systems is shown in Figure 1.

Although each of the above topics are important to the overall goal of producing intelligent database services, it was recognized early in the project that some areas clearly are more difficult to develop than others and in fact the development of some areas is predicated on the previous development of others. For example, intelligent ingest and maintenance services must not only support the database, but also any intelligent user interface. This is true since to perform maintenance and updating properly on a database system with intelligent value added services, some or all of the services themselves must be updated in the process. The reason for this assertion is that there will exist in the knowledge base which controls the intelligent user interface, information about the objects in the database and their relationships to

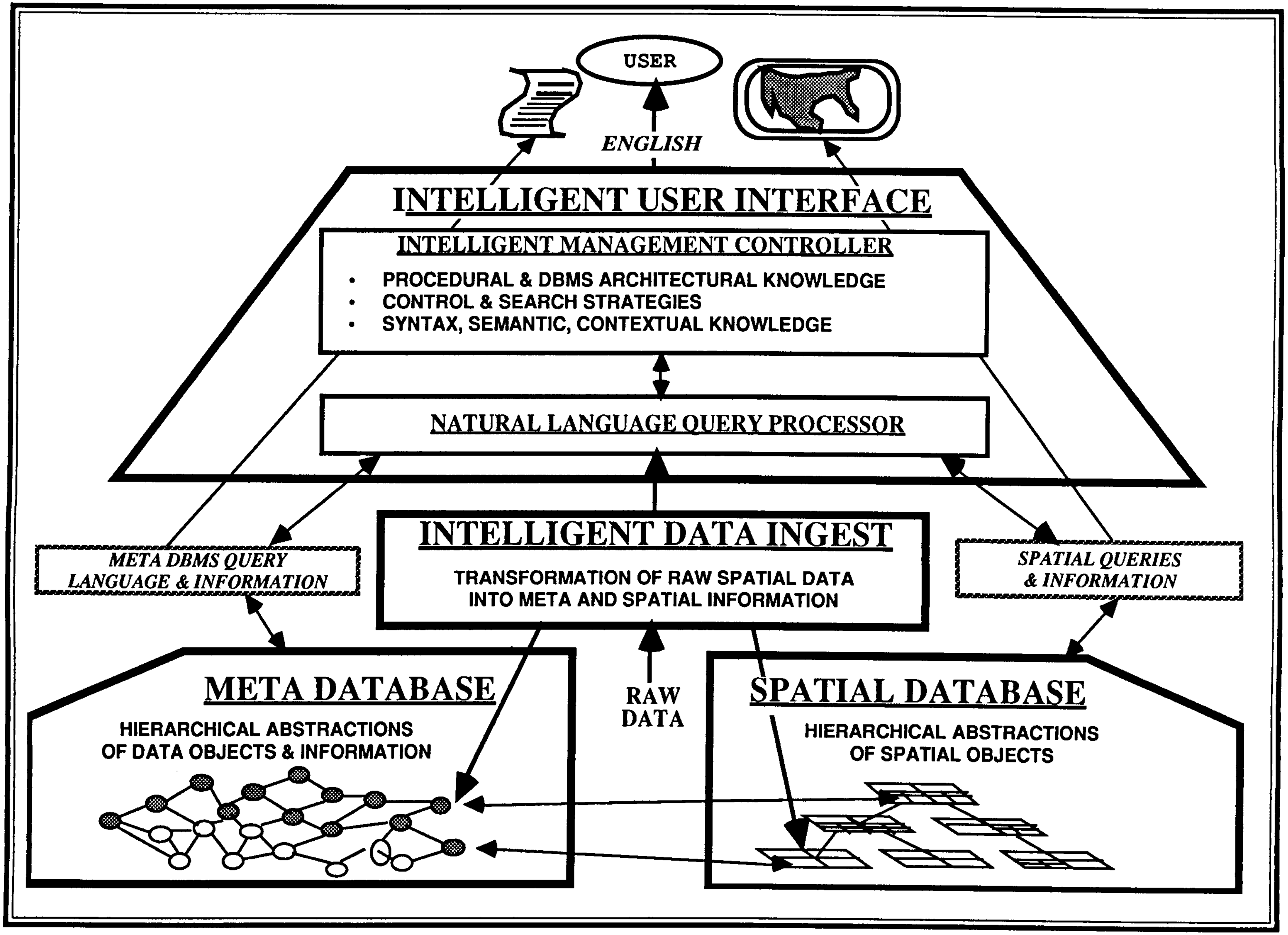
USER
ENGLISH
INTELLIGENT USER INTERFACE
INTELLIGENT MANAGEMENT CONTROLLER
• PROCEDURAL & DBMS ARCHITECTURAL KNOWLEDGE
• CONTROL & SEARCH STRATEGIES
• SYNTAX, SEMANTIC, CONTEXTUAL KNOWLEDGE
NATURAL LANGUAGE QUERY PROCESSOR
META DBMS QUERY LANGUAGE & INFORMATION
INTELLIGENT DATA INGEST
TRANSFORMATION OF RAW SPATIAL DATA INTO META AND SPATIAL INFORMATION
SPATIAL QUERIES & INFORMATION
META DATABASE
HIERARCHICAL ABSTRACTIONS OF DATA OBJECTS & INFORMATION
RAW DATA
SPATIAL DATABASE
HIERARCHICAL ABSTRACTIONS OF SPATIAL OBJECTS

Figure 1. Intelligent Data Management Concept

each other and to identified virtual objects (objects which do not exist in the database but which can be defined by the clustering and analysis of several existing database objects).

After careful consideration of the operation of database systems, it was determined that the development of an intelligent user interface offered the best chance for near term success while at the same time, providing technical direction for most of the other services in the overall concept.

This technical memorandum presents the results of a research and development effort related to user interfacing, called the Intelligent User Interface (IUI) task. The IUI task was initiated at the beginning of FY85 with the goal of developing a prototype intelligent value-added service that would allow users access to a given database with little understanding of the database architecture, data objects (data) or query language. The IUI development concepts are predicated on using powerful workstations, expert systems, and natural language processing technologies. It was envisioned that a database enhanced with a functioning IUI system would enable scientific and supporting (non-database) technical users to access and use the stored information with little or no understanding of the database's architecture, the actual data content or the system's query language.

It is believed that the development and implementation of such a prototype subsystem not only will demonstrate the concept of applying artificial intelligence to data management, but also will enable NASA to assess the long term applicability of AI technologies to its operational needs.

CHAPTER 3

TASK OVERVIEW AND TECHNICAL OBJECTIVES

The objectives of the initial IUI effort were to use existing NASA hardware and software systems to perform the functions of database management and to extend and enhance their performance by designing and developing intelligent value-added services that would give such systems the capabilities to "reason" and make decisions similar to a human expert with database experience in the knowledge domain of interest. The specific goals of the IUI Task for FY85 were as follows:

- The refinement of the IUI concept in the context of the IDM project goals,
- The review, selection, and acquisition of a low cost expert system development tool to support the development of a prototype IUI,
- The review, selection and acquisition of a functioning Natural Language Query Processing (NLQP) system to use in the prototyping of a IUI,
- The creation of an Intelligent Data Management Controller (IDMC) using expert system development tools for a selected operational database,
- The customization of the NLQP "THEMIS" (creation and installation of a dictionary of lexicons that is the same as the expert system) to support the interfacing of the IDMC to the selected operational database,
- The demonstration of a prototype IUI customized for the selected operational database.

During FY85, the IUI research effort successfully achieved all the above goals including the demonstration of a microcomputer based expert system that performed intelligent user interfacing to a NASA operational database. However, the IUI effort is still in the initial stages of concept formulation and system development. Therefore, it is expected that with the acquisition of advanced AI hardware and software tools (a LISP workstation with the expert system development tools), significant advances can be made in the development of a next generation of value-added services for NASA database operations.

CHAPTER 4

THE INTELLIGENT USER INTERFACE CONCEPT

The IUI, as envisioned, is a system that will "serve as an intermediary between a database and a user" who has little or no knowledge of the database's architecture, language, or content.[1]

Such an interface would allow the user to operate a database by locating, identifying and selecting specific information using the context of his particular knowledge domain by communicating in the medium that is most suitable to him (i.e. in the user's natural language, English). The advantages of such a concept are that:

- It can facilitate understanding by specifying the contents and meanings of a collection of data as well as the relation between objects within the data; [2]
- It will be able to support approximate reasoning to infer conclusions that are not explicitly stated by the user, such as an imprecise or fuzzy query which can be stated without mentioning file names or table joins;
- It can handle fuzzy concepts, expressed by using a fuzzy query;
- It will be able to handle information demands for which the database is used routinely, rapidly and efficiently with little understanding by the user;
- It will be able to communicate with the user in plain English text;
- It will serve as the semantic basis for understanding the user's data needs in conjunction with the natural language query system;
- It will provide a logical representation of the database architecture and information stored in the database to the casual user;
- It can be used to facilitate the identification and understanding of database information from the database **operational view** (the database design model which can be relational, network, etc.) to the user related views by using conceptual graphs that translate between the database's meaning and the user's meaning of data and sets of data [3];
- It will provide a physical and logical link between information contained in the meta-knowledge database and the spatial database (the actual data) in the overall IDM concept.

To support such a design concept, two ideas need to be developed:

- First, the creation of multiple views of the database that are functionally appropriate from the perspective of a nonexpert database user and, at the same time, operationally necessary, from the database's perspective of supporting the physical storage and management of the data,

- Second, the creation of a means for moving from one type of view to another using logical inference statements (i.e., rules) called **conceptual graphs**, as proposed by Sowa [4].

A singularly important feature of such a concept is its role as a mediator between the user and the database. Such a system must deal with the duality of the views (or contexts) in order to maintain both the integrity of the database and the user's understanding of the needed information.

Using such a concept, the IUI deals with a user in his syntax and understanding of the objects in the database. Also, at the same time, the IUI understands the database's viewpoint, based on the physical and operational organization of the data and the relationships between objects within the data.

Based on our present understanding of database operations, as well as our reading of the supporting literature, there are at least three types of views required for a functionally complete IUI. These views are:

- **The Architectural view,**
- **Multiple application views,**
- **The Operational view.**

The first view, the architectural view, is the logical taxonomy of the objects contained in the database organized in a hierarchical schema. The purpose of this view is to organize the information contained in the database in a structure that is most commonly used by humans when dealing with large amounts of data. Such an approach will allow the search and identification of a desired data objects based on subject context, inherited object attributes, and search strategy heuristics.

The second view, the application view, is just a specific view point about the data objects in the database organized to support an application. What is unique about this view is that it will be constructed based on specific set of data object goals. These goals contain objects that are the result of the clustering or processing of real data objects into infered or virtual objects. In addition, this view will be supported by specific heuristics in an associated knowledge base that will enable the user to interact with the view as if he were the expert that constructed the view and is familiar with its operation.

The last view, database operational view, is the view that represents the actual database design (tables, key fields, join fields etc.,). This view is necessary because all other views must translate to it in order to get information from the database.

Based on the premise that the design of the IUI requires an awareness of all the above types of views, one can postulate a model that considers such views collectively linked by intelligent processes which translate between the views. Such a model is shown in Figure 2. and, as can be observed, the primary function of the IUI is to translate between the various database views and the user in a fashion that best supports the user's needs. The operational view must be included as part of the model even though this view, which is the actual DBMS, exists independently.

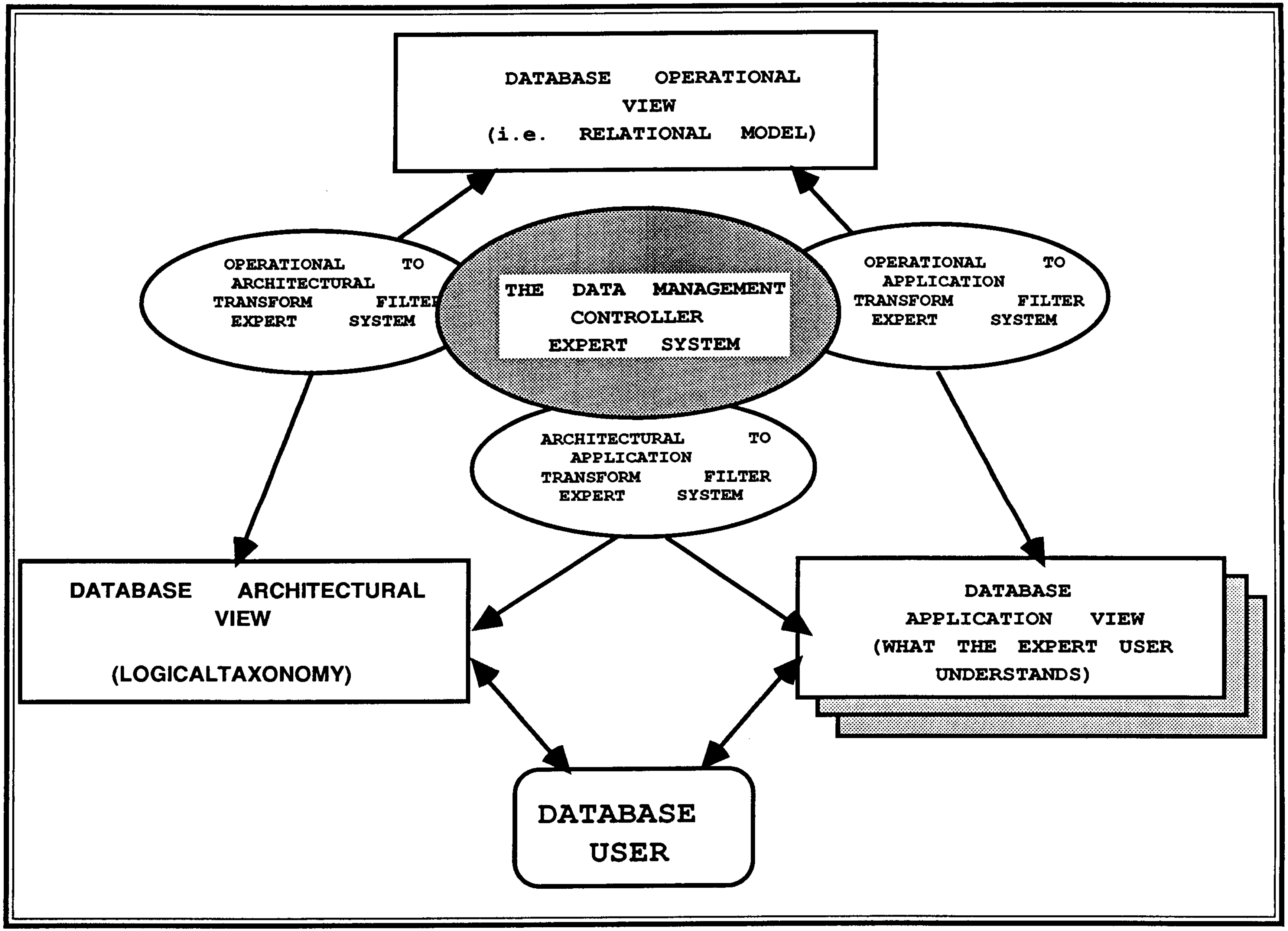
DATABASE OPERATIONAL VIEW (i.e. RELATIONAL MODEL)
OPERATIONAL TO ARCHITECTURAL TRANSFORM FILTER EXPERT SYSTEM
THE DATA MANAGEMENT CONTROLLER EXPERT SYSTEM
OPERATIONAL TO APPLICATION TRANSFORM FILTER EXPERT SYSTEM
ARCHITECTURAL TO APPLICATION TRANSFORM FILTER EXPERT SYSTEM
DATABASE ARCHITECTURAL VIEW
(LOGICALTAXONOMY)
DATABASE APPLICATION VIEW (WHAT THE EXPERT USER UNDERSTANDS)
DATABASE USER

Figure 2. IUI Concept Model

4.1 The Intelligent User Interface and Database Views

The multiple view concept upon which the IUI is based is an extension of an earlier concept presented in a popular text book by C. J.Date.[5] This concept involved the representation of database architectures using levels of abstraction. In Date's concept, the architecture is represented by three levels: internal, external, and conceptual. The internal level is concerned with the way data are stored and is the closest to physical storage; the external level is the view closest to the user's understanding of the data; and, the conceptual level is a "level of indirection" between the other two levels.

In the context of the IDM concept, the intelligent user interface serves as both conceptual and external level processes. Such an approach allows for powerful indirection by dealing with the user in his own syntax and logical processes while at the same time understanding the database's design, data content, and query language.

The IUI concept model shown in Figure 2. postulated two types of views to support the conceptual level: an architectural view and multiple user application views. However, because of the broad definitions given to the two types of views, part of the design formulation includes specifically limiting the view's scope in order to avoid the uncertainties and ambiguities that can occur when considering database concepts.

In the context of the IUI concept, a logical taxonomy of the database's organization, which includes the relationship of the database objects to the general knowledge of the expert database user, defines the architectural view. This view is organized to minimize processes that a user will apply when trying to locate a particular object or relationship between objects.

Although the architectural view is similar to some existing database design concepts, two differences exist. First, the objects and the language used to describe them in the architectural view are based on the user's language and understanding of the information of interest. Second, this view may contain objects which do not exist as real data objects in the actual database. These objects, known as **virtual objects**, result from the clustering of real data objects.

The second view, the domain-specific application view, includes all the knowledge, facts, and metaknowledge known to the expert user working in a particular technical area for which the database has been designed. The application view will include three different types of knowledge and facts:

- **Heuristic knowledge** related to obtaining particular information for specific applications;
- **Procedural knowledge** that is used by an expert user to obtain information from the database for a specific application;
- **Virtual objects** that are inferred objects resulting from the clustering of tuples based on application expertise.

Supplementary to the current application view, several user application views are expected to arise for different areas of database expertise. Because of this, formulation of an IUI design capable of easily representing and modelling an "expert" user's knowledge of the meaning of the collected data is necessary in the design phase. Such information will enable the expert system to include functional and semantic interrelationships among database entities, domain specific attributes, as well as content descriptions of objects.

The effect of the multiple view approach on the development of the IUI model is that when views are integrated with a database, two basic schemas of the database arise: a conceptual schema consisting of multiple logical views and the database architectural schema (i.e., relational model). [6] Both are necessary for a complete description of the database.

4.2 Conceptual Graphs

The use of the conceptual graph in the intelligent user interface process is presented in Figure 3. To better understand the application of the conceptual graph to the database problem the following example is presented.

In the Crustal Dynamics Project's Data Information System (DIS) [7], a database has been developed at the NSSDC to support the storage of collected and analyzed crustal motion data from sensing stations located at many sites over the earth's entire surface. The users of the Crustal Dynamics database fall into two classes: engineers and scientists. The latter group is primarily interested in the database for studying motion of the earth's crust and earth rotation. The area of science related to crustal motion is commonly called plate

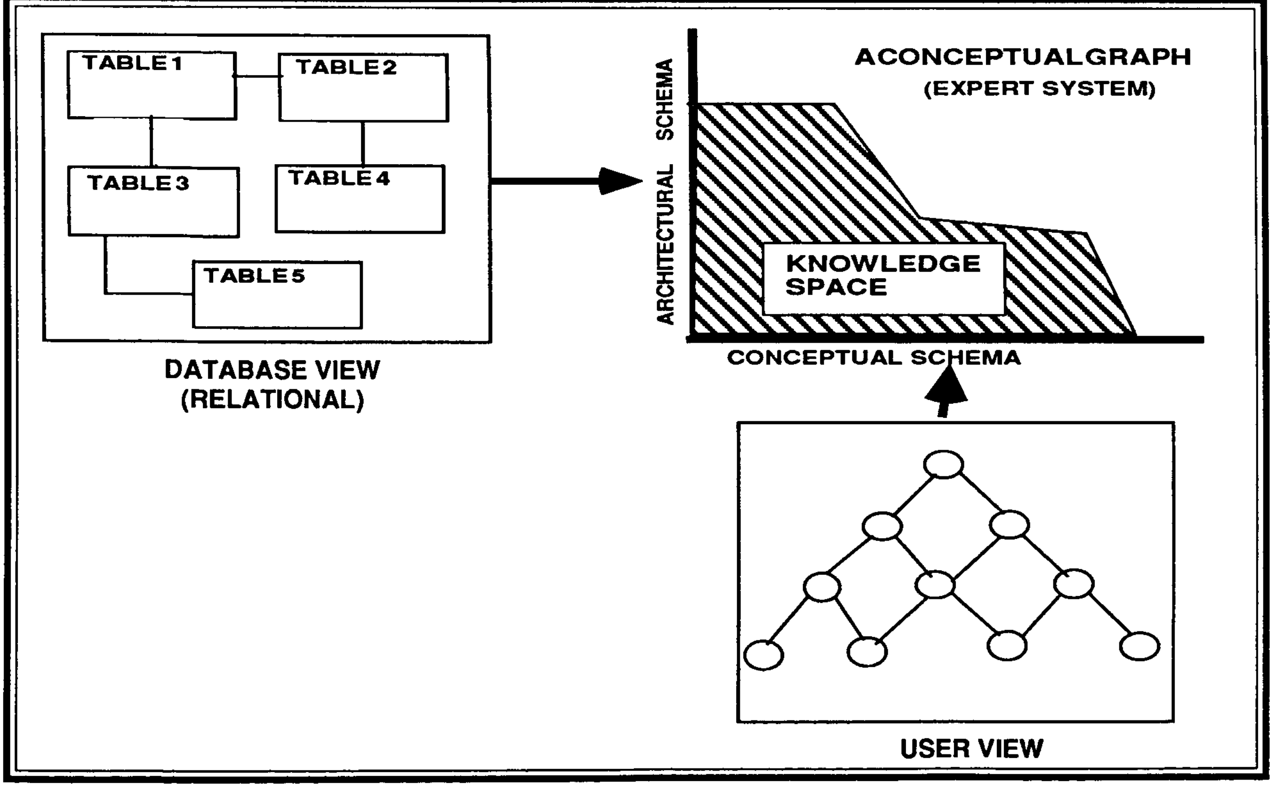

Figure 3. Representation of a Conceptual Graph

tectonics. To get the necessary information related to plate motion, an expert database user must query the database about a specific set of stations, calculate the chord between them for a specific period of time, and then determine the slope of the chord. However, the objects (the calculated slope of the chord between any two sites) are not stored explicitly in the database. Therefore, the only way to determine a specific plate's motion is to identify the stations that are located on the specific plate and measure the motion of those stations with respect to stations on another plate (i.e. baseline measurements). Thus, there exist data objects needed in the identification and selection of the appropriate database information that do not exist in the actual database. Furthermore, only the experienced database user can know how to calculate the chords between the stations and their slope.

In the IUI concept, such missing information is accounted for so that, when information about a specific plate's motion is desired, the IUI knows how to calculate the motion based on the type of sensing system used (SLR or VLBI) [7]. However, to the casual database user, this intellectual transformation will be transparent. To the IUI developer, this transformation (also referred to as a **transform filter**) is in fact the conceptual graph which facilitates the understanding of the meaning of objects in the context of the user's needs and translates those objects into different objects with meaning to the database. In the formulation of the IUI, one of the primary functions of the expert system is the creation of such conceptual graphs.

Using the conceptual graph concept to support a multiview representation of the database, we believe that the IUI model allows for the creation of at least one logical translation path between a user's view of the data (domain specific) and the database management system's view. This path is, in fact, an expert system (see Figure 3).

4.3 The Application of Expert Systems to the IUI

The IUI design is predicated on the hypothesis that an expert system can be developed with the ability to reason about database information, both from the user's view and from the database's view. Such a design will allow the intelligent identification and selection of required information with only limited guidance by the user. Thus, the expert system will contain all the necessary facts and knowledge about the structures and contents of the various views of the database as well as the translation among the views. Such an expert system will actually emulate an expert database user within his limited expertise domain.

The decision to use an expert system as the operational environment for IUI software is based on the following considerations:

1. A database has a finite and rather limited knowledge space making the development of an expert system possible;

2. Certain important operations that are performed by an expert database user are based on **heuristics** (i.e., search

strategies used to find information based on past experience);

3. Although a database has a large number of potential data sets available, the actual number of sets is usually limited to a small number, compared to the large number of data objects that could potentially be extracted from the database;

4. Many of the operations performed by the expert user are procedural in nature;

5. Expert system development tools, especially the advanced ones that employ frame representation schemas (e.g., LISP with flavor extensions), allow not only the implementation of multiple database views but also the construction of a logical taxonomy of objects in the database and the subsequent reasoning about such views and objects in a very sophisticated manner.

In the history of expert systems, the most successful development efforts have involved goal-directed, backward chaining paradigms such as MYCIN [8] and PROSPECTOR [9]. In the case of the IUI, where domain-specific information is required, the same kinds of problem solving strategies, which were successfully used in the aforementioned examples, can also be applied to this development effort. Also, the ability to construct expert systems as conceptual graphs is based on the finite knowledge and syntax domains of a database.

The impact of the above assertions lies in the potential to develop intelligent processes for the IUI using expert system development tools for any prototyping effort.

CHAPTER 5

PROTOTYPE DEVELOPMENT AND IMPLEMENTATION

The approach used to develop a functioning IUI was based on a phased effort which included access to a functioning, operational database. Such a phased effort allowed the development of the IUI processes (database views and transform filters) to be done in a stepwise fashion so that performance, feasibility, and operability could be tested for each functional element. The development of the first prototype IUI had the following objectives and protocol.

5.1 Development Objectives

1. Formulate a design for a prototype Intelligent User Interface system that can be used to demonstrate the proof of the concept as well as support and guide the technical direction of the overall IDM project.

2. Design and implement an expert system development environment using an IBM PC AT computer with the capability of supporting all the necessary system hardware and application software needed for a prototype IUI.

3. Design, develop, and prototype an IUI for a selected operational database using an expert system development tool. The proposed IUI will consist of the architectural view, one application view, and supporting transform filters. The system will be technically sophisticated enough to interact with the selected database in a near real-time manner and allow the demonstration of the IUI concept.

5.2 Development Life-Cycle

The following is the planned protocol consisting of 12 tasks for the development of a prototype IUI system.

1. *Hardware & Software Selection* Select and acquire the proper development tools, both hardware and software, for the initial development effort.

2. *Database Identification & Selection* Select a functioning, operational database that uses a relational DBMS model for the prototype IUI.

3. *Database Evaluation & Characterization* `Study the selected database and develop a first order understanding of the database's architecture including the organization of tables, the information in the tables as well as the information objects that the database system manages.

4. *Evaluation of the Use of the Database by the Expert User* Interview the expert database users and determine the uses of the database that can be represented by domain-specific application views.

5. *IUI Prototype Design Formulation* Formulate a design for a prototype IUI

System based on the concepts presented in Chapter 4 of this Technical Memorandum and on the results of tasks 3 and 4.

6. *Formulate & Develop The Application View* Construct the selected application view of the database, assuming that there is more than one view. Based on the interviews with expert users, the selected view should include specific application goals, procedural knowledge, and identifiable virtual objects and be implemented using an expert system development tool.

7. *Formulate & DevelopThe Architectural View* Develop the architectural view of the database using the logical taxonomy schema formulated in task 3 of the protocol.

8. *Develop Transform FiltersTo Support TheTranslation Between The Application & Architectural Views &The Operational View* Develop a set of transform filters that will support the translation between the application view of the database and the operational (relational model)[10] view of the database.

9 *Interface Application & Architectural Views' Transform Filters ToThe NLQP* Develop a question/phrase construction process to interface between the NLQP and the application & architectural view transform filters.

10. *Customize NLQP To Support The Application & Architectural Views* Customize the natural language query processor so that it contains a dictionary of the lexicons used in the application view of the database.

11. *Develop Transform Filters To Support TheTranslation BetweenThe Application View & The Architectural View* Develop a set of transform filters that will support the translation between the architectural view and the application view of the database.

12. *IntegrateTransform Filters Into A Single Knowledge Base* Integrate the common elements of the three expert systems processes that perform the transformation filtering.

5.3 Development Task Effort

Based on the development protocol listed above, an operational database was selected and the first prototype intelligent user interface developed. The IUI System was named CRUDDES (CRUstal Dynamics Database Expert System) after the database selected. The following tasks, which follow the above protocol, describe how the prototype IUIS was developed and implemented.

TASK 1
HARDWARE AND SOFTWARE SELECTION

The selection of the hardware and software for the CRUDDES development effort was based on the availability of allocated funds for FY85. Because of these limitations, the hardware selected was an IBM PC AT, microcomputer system with a 20 megabyte hard disk drive, 512 Kbytes of RAM memory, two displays (mono screen and a color screen), and an external modem all utilizing the DOS operating system. A diagram of the system is presented in Figure 4.

Software selected for the project included the expert system development tool, M.1[10], developed by Teknowledge, the full screen editor, XYWRITE [11], from Xyquest Corporation, and the multiprocessing environment, DOUBLEDOS [12], by Softlogic. Specifically, M.1 was written using PROLOG 86 [13] (a popular IBM PC language that supports symbolic manipulation) and can support up to 200 facts and rules without segmentation (if the atoms are relatively small). Modelled somewhat after EMYCIN, the M.1 system performs only goal-directed backward chaining [14]. In fact, we found M.1 far superior to other IBM PC based development tools because of the way it handled the control structure, facts, variables, and meta-knowledge.

The procedure used to develop the expert system with the selected software depended on the installation of DOUBLEDOS. After DOUBLEDOS was booted, two separate operating system partition areas were created each with its own monitor, one with 256Kbytes of RAM and one with 200 Kbytes of RAM. In the first partition M.1 was installed and in the second XYWRITE. Both software packages have access to a common file area on the hard disk. This environment allows for the development of the expert system using the editor in the first segment and the concurrent testing of M.1 in the other segment. The development team found this configuration extremely valuable over normal single process DOS operations. This is because it allowed rapid develop-

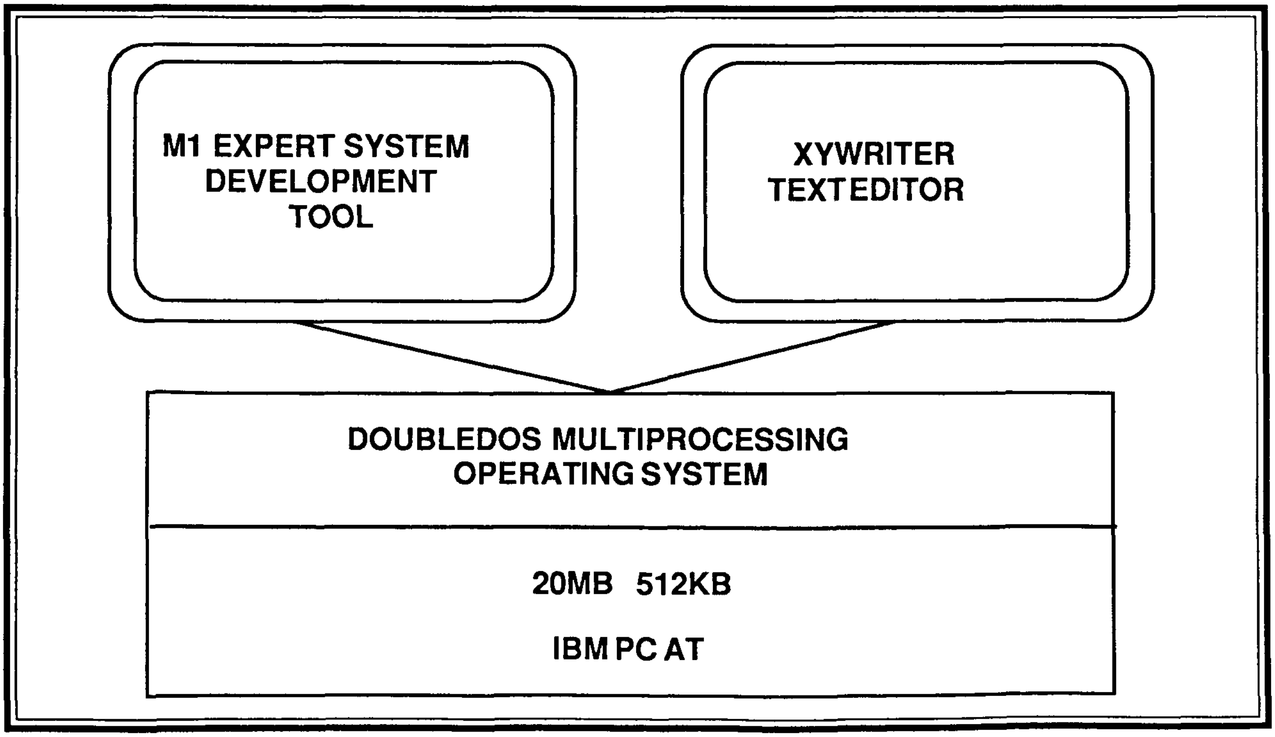

Figure 4. Development System Configuration

ment and testing of software without the constant changing from the editor to M.1 and back to the editor again. Furthermore, it was felt that the ability to develop software with an external editor allows for a great deal more flexibility in the development process thus creating a more robust expert system.

TASK 2
DATABASE IDENTIFICATION AND SELECTION

The database selected was one developed to support the Crustal Dynamics Project. Implemented in ORACLE [15], a relational DBMS[16] [17], and operating on a DEC VAX-11/780, the database consisted of 104 tables that supported the management of crustal movement , related engineering data collected by many sites over the earth's surface, and project management data. The sensors used to detect earth crustal movement are of two types: the first uses a satellite with a laser ranging device (SLR); the second uses a radio emitting stellar object and Very Long Baseline Interferometry (VLBI). Specifically, the VLBI method involves using two or more widely separated radio telescopes to observe radiation from extragalactic radio sources, typically quasars. In the Crustal Dynamics Project, SLR or VLBI is used to observe signals from various points on the surface of the earth, take a time lag measurement of their arrival with respect to a local clock at each site, and record the "time tagged" signals on tape in order to sense crustal movement.

A key feature in selecting the ORACLE DBMS over other systems was that an operational natural language query processor supported it. The need for the query processor is that the formulation of a desired database query in English is a much less complicated problem for an expert system than the translation from English to the query language SQL.[18] [19]

In some of our initial research, we determined that performing language parsing and translation in an expert system overly burdens the system (i.e., makes it too large and cumbersome) and reduces the ability of the IUIS to formulate the "best" query. In fact, we found such a task almost impossible to generalize without actually developing a natural language processor as part of the expert system task. Thus, the incorporation of parsing and translation was avoided in all parts of the development effort.

TASK 3
DATABASE EVALUATION AND CHARACTERIZATION

The evaluation and characterization of the Crustal Dynamics Project database was performed by the development of an Entity Relationship (E/R) [20] diagram (Figure 5. presents a simplistic example) which is commonly used in initially designing a relational database. In addition to creating the E/R diagram, it was necessary to develop a logical taxonomy of the objects in the database using an approach that is commonly found in a thesaurus. This logical taxonomy, along with conversations with the expert users, helped to identify virtual objects.

The result was the identification of three object classes in the architectural view of the database: experimental data, site data, and project management data. Because

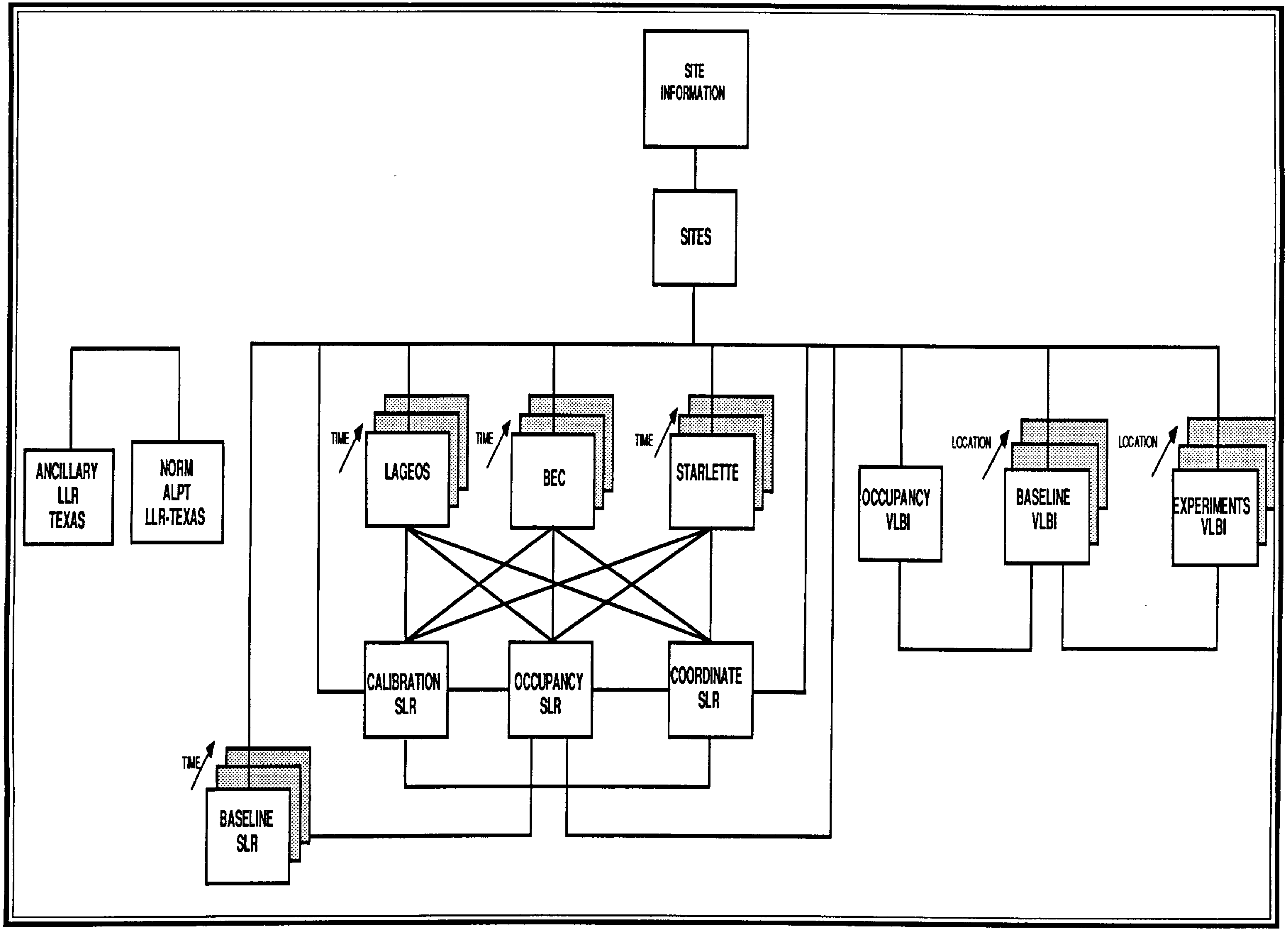

FIGURE 5. TOP LEVEL ENTITY RELATIONSHIP DIAGRAM

CRUDDES is a prototype, only the first two data classes, namely site data and experimental data, were dealt with in two separate steps for early validation in the development effort.

TASK 4
EVALUATION OF THE USE OF THE DATABASE BY EXPERT USERS

This portion of the development effort involved the application of knowledge engineering techniques to determine the utilization of database by the application database experts. Two scientists that support the Crustal Dynamics Project were identified and contacted in regard to collecting information related to the actual use of the project's database. The scientists were interviewed on several occasions and, based on the interviews, three domain specific application views were identified:

- Scientific applications,
- Engineering applications,
- Project management applications.

The information gained from this task determined the basis for the design of the prototype IUI formulated in task 5.

TASK 5
IUIS PROTOTYPE DESIGN FORMULATION

With the three application views identified from the expert users it was then possible to formulate a design for the initial prototype IUI, presented conceptually in Figure 6. Consisting of a single scientific application view, related to plate tectonics, and the architectural view, the design is founded on two sources of information: first, from the information gained during task 3 and, second, from the interviews with the expert users in task 4. Likewise, the basis for the design are the concepts discussed in Chapter 4.

Once the functional design was completed, a software design of the knowledge base evolved based on the the characteristics and limitations of the expert system development tool, M.1, and on the IBM PC AT computer system. A top level overview of the resultant knowledge base design is presented in Figure 7.

TASK 6
FORMULATE & DEVELOP THE APPLICATION VIEW

After finishing the overall prototype design, the development team conducted subsequent interviews with the expert users in order to specifically understand the database utilization and support the development of the application view. This resulted in a design concept that decomposed the scientific application view into two subviews: plate tectonics and earth rotation, each with specific sets of information extraction goals. For the subview plate tectonics, the information extraction goals were: whole plate motion, regional deformation, and plate stability; and for earth rotation: polar motion (precession) and rotational velocity. Each of the listed information extraction goals were further affected by the application of necessary modifiers such as year, site location, and measurement method. However, even when the **whenfound modifiers** (those procedures which set exception parameters like year,

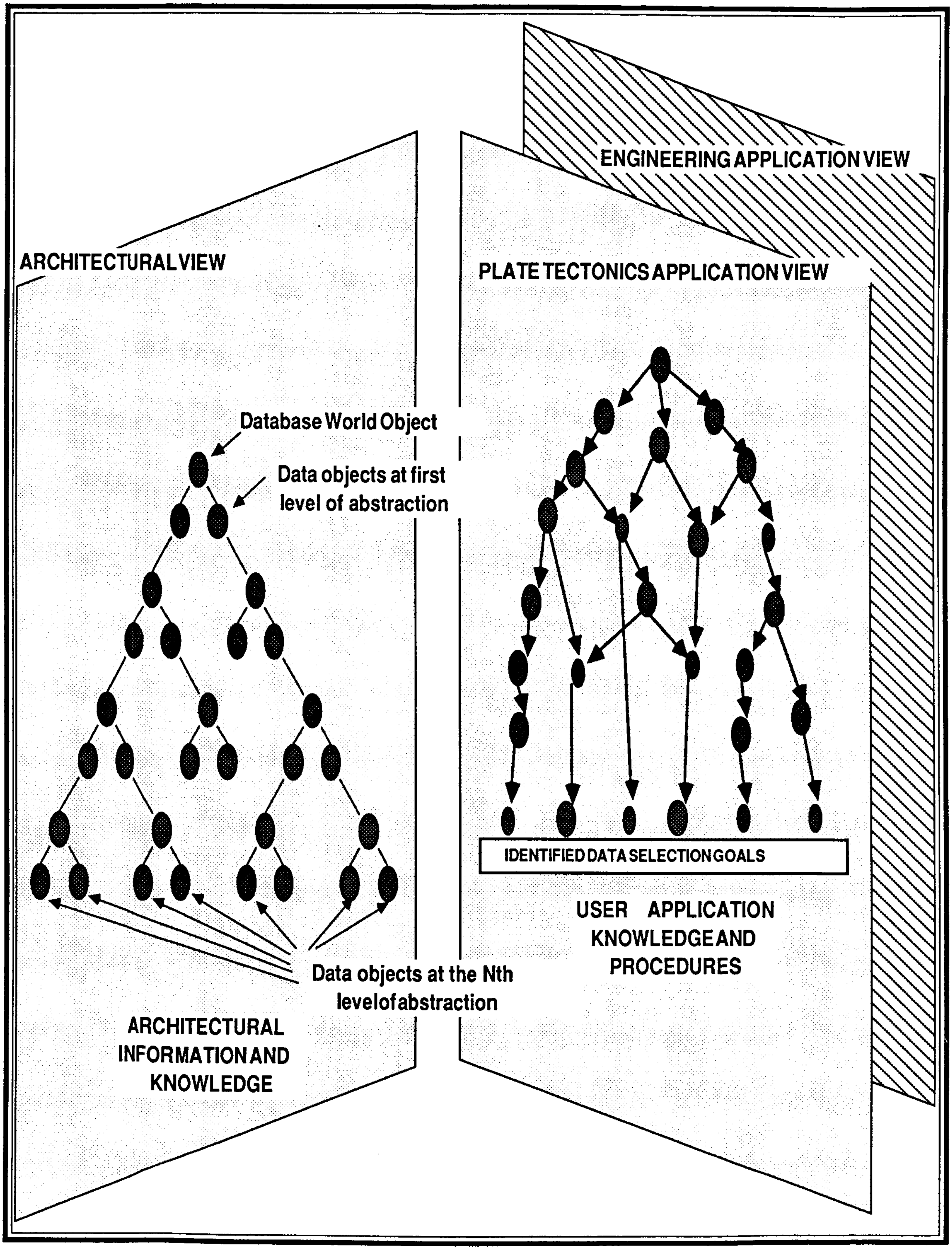
ENGINEERING APPLICATION VIEW
ARCHITECTURAL VIEW
PLATE TECTONICS APPLICATION VIEW
Database World Object
Data objects at first level of abstraction
IDENTIFIED DATA SELECTION GOALS
USER APPLICATION KNOWLEDGE AND PROCEDURES
Data objects at the Nth level of abstraction
ARCHITECTURAL INFORMATION AND KNOWLEDGE

Figure 6. IUI Prototype Conceptual Design

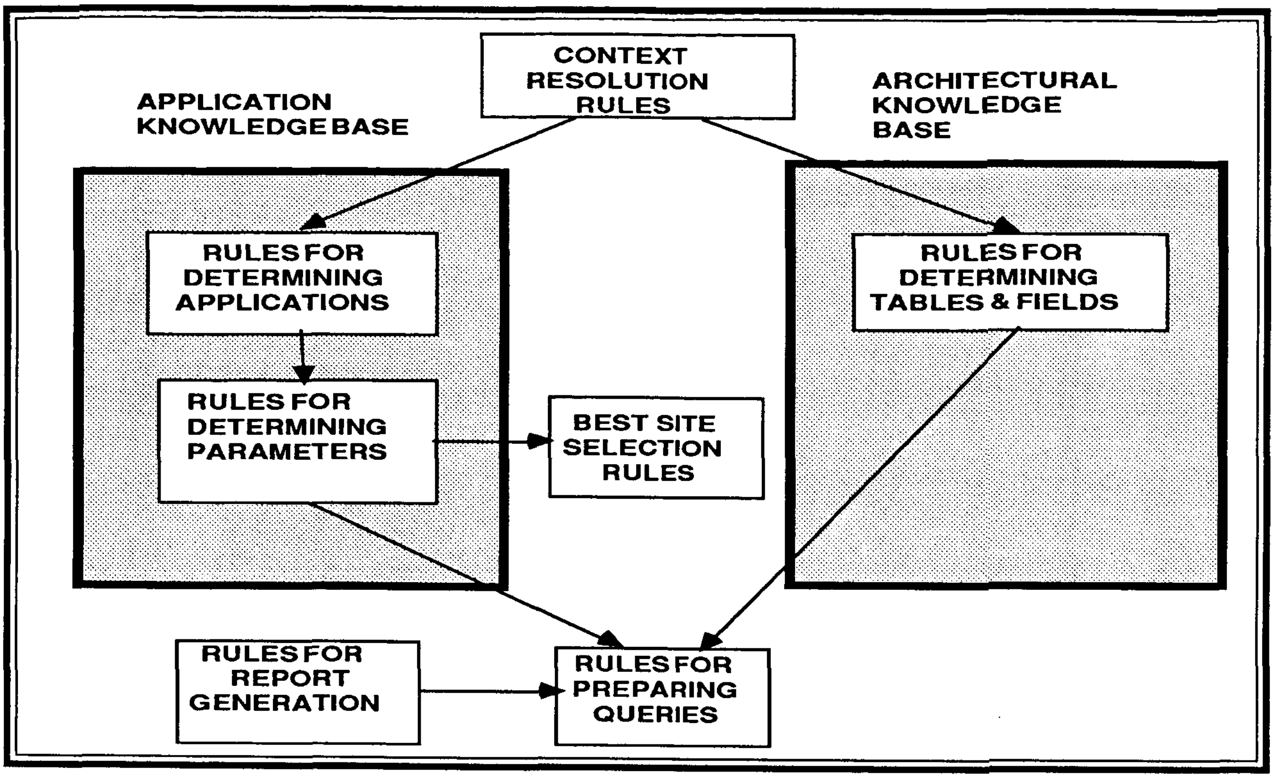
CONTEXT RESOLUTION RULES
APPLICATION KNOWLEDGE BASE
ARCHITECTURAL KNOWLEDGE BASE
RULES FOR DETERMINING APPLICATIONS
RULES FOR DETERMINING TABLES & FIELDS
RULES FOR DETERMINING PARAMETERS
BEST SITE SELECTION RULES
RULES FOR REPORT GENERATION
RULES FOR PREPARING QUERIES

Figure 7. Overview of Cruddes Knowledge Base Design

site location, etc.) are considered, the number of variables was fairly limited. Since the application view query usually included both procedural and heuristic knowledge, apparently, the development of a scientific application view was a fairly straightforward process using expert systems technologies.

During this task effort, it became apparent that more consideration for the development of the overall design of the expert system processes would be required. Initially, it was thought that the application view could be developed separately from the architectural view. However, it became evident that any single view lacked the minimum necessary content required to assist the nonexpert database user. The reason for this conclusion was that nonexpert users understood little about the information in the database. Consequently, they might think there was, or should be, more information extraction goals than had been provided. In other words, the IUIS has to accommodate for the "I don't know what to select" response.

This additional complication resulted in the development of a design, called the user **context resolution** process, which attempts to resolve what view best serves the user's needs. The resolution of the context could only be assured by concurrently providing an application view and an architectural view. A model of the proposed final design concept is presented in Figure 8.

The final step in this task involved determining all of the possible combinations of unique data extraction goals (the database objects with their wherefore clauses) required in getting the desired information from the database for the scientific view.

After all the information extraction goals for the scientific domain (application view) had been identified and characterized and a strategy for resolving view conflict resolution determined, it was then possible to begin development of the expert system. The goal of the first expert system was to support the automated reasoning required in the determination of a user's desired knowledge from his interaction with the system. This is done in the context of the various information extraction goals available in the scientific application view or, where applicable, in the architectural view.

This first expert system dealt with the knowledge area involving plate tectonics. The system's inference process, based exclusively on backward chaining, dealt with the various alternative selections available in the query formulation process by using generalized rule forms with variables. The expert system, in a planned structured selection process, identified the various components that would make up a question and instantiated them into the variables where rules were used to construct the desired English question. The resultant expert system enabled a user scientist, with little experience with, or knowledge about the database, to easily get desired scientific information.

To elucidate for the reader a better understanding of how the system supported its database interface function with a non-expert user, a sample consultation is preented in Appendix A. In addition, Appendix B.presents a listing of the first segment of the knowledge base that involves context resolution and the selection of data objects related to plate (tectonics) motion.

TASK 7
FORMULATE & DEVELOP THE ARCHITECTURAL VIEW

The development of the architectural view of the database is based on the premise that it is necessary to provide to the user a generalized view of the database which is a logical organization of the objects that exist in the database and their interelationships. With the knowledge gained from this view, one could identify information that is not available through the other views (application and operational) or an extension of an already available application formed query.

As discussed in task 6, one of the primary reasons for the architectural view is to support the "I don't know what I want" response expected from a new database user. The approach taken in designing the CRUDDES architectural view was predicated on the concept of object relationships that will be required in developing the next generation IUI system using a frame based expert system. The decision to mimic a frame type organization came from an initial development effort that organized the database into a tree structure.

The impact of selecting the frame over the tree approach was twofold. First, a frame provided the most complete relationship organization possible including virtual objects, and second, it makes the view reasonably easy to port to a new expert system development tool being acquired as part of the FY86 project effort.

The development of the rules that created the architectural view was segregated into

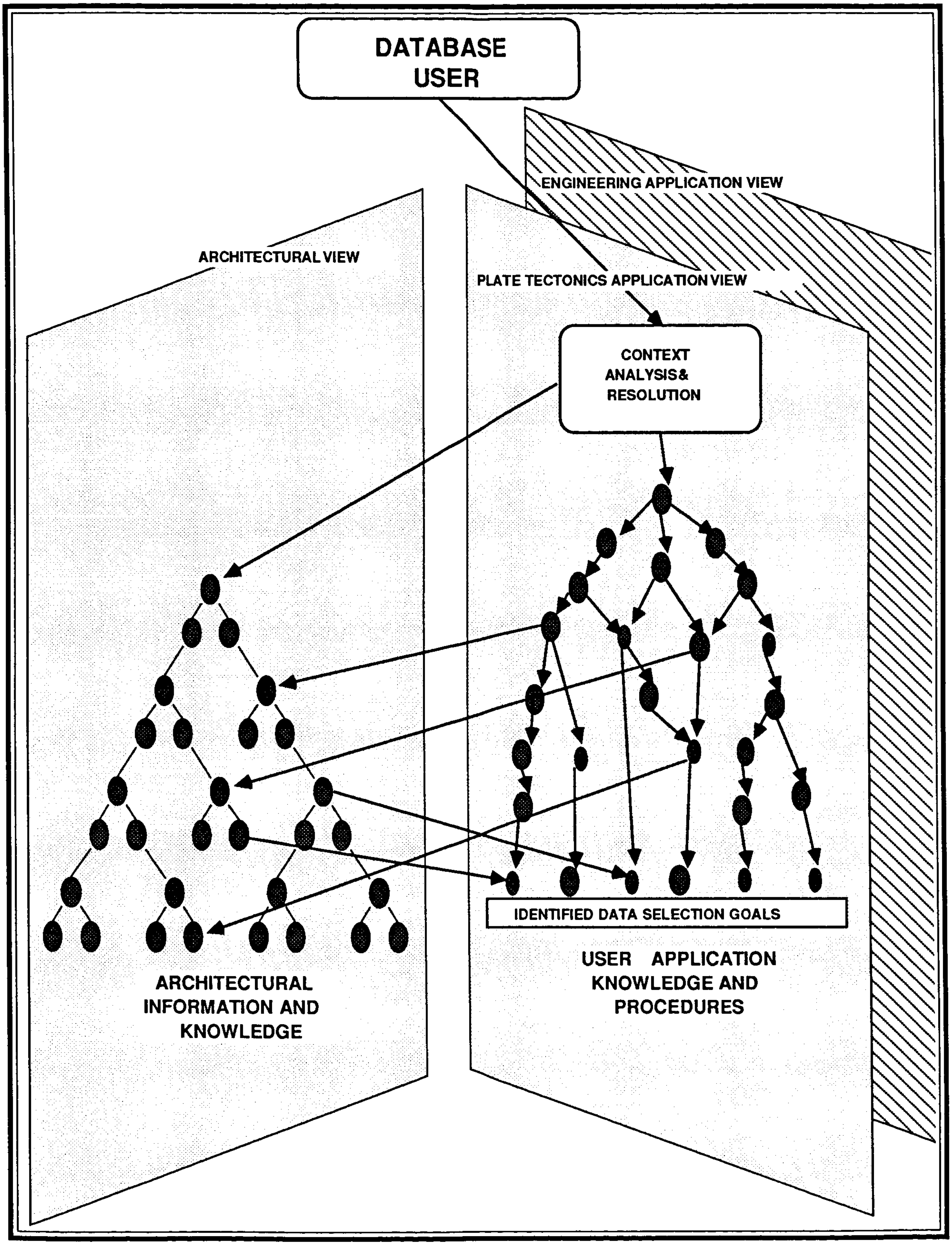
DATABASE USER
ENGINEERING APPLICATION VIEW
ARCHITECTURAL VIEW
PLATE TECTONICS APPLICATION VIEW
CONTEXT ANALYSIS& RESOLUTION
IDENTIFIED DATA SELECTION GOALS
ARCHITECTURAL INFORMATION AND KNOWLEDGE
USER APPLICATION KNOWLEDGE AND PROCEDURES

Figure 8. Finalized IUIS Prototype Conceptual Design

two stages. Taking about six weeks to complete, the first stage involved the creation of two alternative view structures and an assessment of their performance to provide the necessary information to the user about the objects in the database. The first architectural view structure was based on a functional representation of the actual database design and the second, a structure using the object relationship concept. The results of an evaluation of the two design alternatives indicated that the first was incapable of supporting the needs of the CRUDDES effort because it did not include all the necessary objects. The second would be impossible to implement with M.1 because the software did not include the LISP flavors feature that enables object/subobject inheritance to be defined.

The organization for the architectural view is show in Figure 9. As can be observed, the highest level object is the database world in which every other object is a component. The next level of decomposition in the architectural view is site information and experiment information. At this level, objects are clustered by their relationship to the ordering. The beauty of this type of view structure is that when inheritance is considered, objects will be able to receive new attributes by the very fact that features have been added to objects at lower levels. Thus, inheritance allows a system to respond as if it were learning automatically.

Because of the limited capabilities of M.1, no possibility existed to create a true frame-based structure for the architectural view of the Crustal Dynamics Project Database. However, it was possible using M.1 to organize the objects within the view in a manner similar to what would have been possible with frames. The exception, however, is that no inheritance exists. This organization was carried out and demonstrated to a limited extent in about three weeks.

TASK 8

DEVELOP TRANSFORM FILTERS TO SUPPORT THE TRANSLATION BETWEEN THE APPLICATION & ARCHITECTURAL VIEWS & THE OPERATIONAL VIEW

The development of expert system based transform filters functioning as the interface between the architectural view and the operational view did not require a great deal of development because only a portion of the architectural view was implemented in the first phase of the CRUDDES development effort.

The approach for the development of transform filters between the architectural view and the operational view were very similar to those developed from the application view to the operational view. For both cases, once objects and cluster object relationship were determined, it was only a matter of creating the necessary rule base to provide a functioning transform filter capability.

In this task, identified objects were translated from the application and architectural view to objects that exist in the operational view and then translated into a syntax that could be understood by the database (SQL).

The most dificult problem in the prototype development effort was the formulation of a SQL query, once the English question

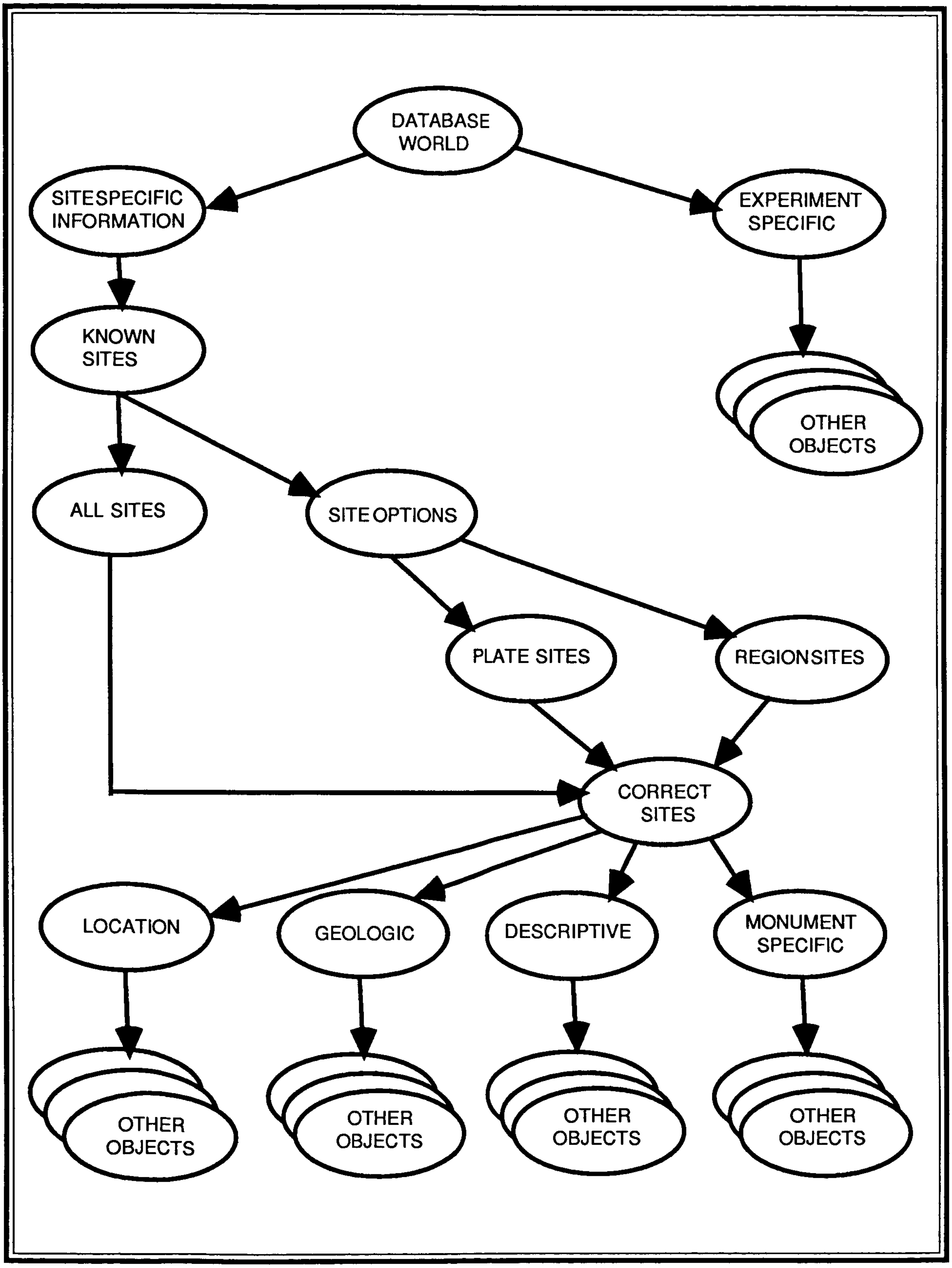

Figure 9. Architectural View

phrase had been constructed. The reason for the difficulty is twofold. First, it was necessary to translate the object information identified in the application and archiectural views to objects that existed in the operational view. Second, once an object translation had occurred that resolved all operational view ambiguity, the query then had to be translated into the database's query language (SQL).

The first translation process is one of the most important components of the IUI concept because this is where the expert database user is needed in order to translate from what the user comprehends and needs to what the database understands. This assertion is based on the hypothesis that there exists in the application and architectural views many objects that do not exist singularly in the operational view. These consist of the clustering or grouping of several objects that exist in the operational view. It is this expertise along with the procedure knowledge that is required to form the proper query that signifies the expert database user. In the prototyping of CRUDDES, such knowledge was identified and captured in the expert system.

TASK 9

INTERFACE APPLICATION & AND ARCHITECTURAL VIEWS TRANSFORM FILTERS TO THE NLQP

The translation, from English to SQL, was a more difficult problem to resolve. It became evident early in the development effort that an approach which supported the translation from English to SQL would be an extremely complex and difficult task involving the development of parsers that could convert from English to SQL. The reason this approach was undertaken was that the natural language query processor had not yet been customized to interface with the Crustal Dynamics database due to the lack of necessary computer resources. Therefore, an alternative approach had to be taken to allow the parsing of an English query into SQL. The result of the initial development effort was a version of CRUDDES that performed inadequately. Although it worked well for the limited information goals in the application view that had been included in the knowledge base, it lacked the robustness for dealing with the broader range of English to SQL parsing problems that would have to be resolved when the application and the architectural views were extended and other application views added.

One important result of this effort was the appreciation for the robustness of the English language in being able to formulate, in a very compact way, an expression that communicated the desired information. It would appear that, when dealing with or reasoning about an object (or information about the object) that exist in a domain-specific space, it is best to formulate the questions in the context of that domain using the syntax applicable to it. The result of this conclusion for the IUI design is that, it is best to stay in an object's conceptual domain until the entire query expression has been formed and only then translate the question to a database query using its language. It is easy to see how a simple English question quickly becomes very long and complex when converted to a SQL expression.

Because of the complexity of developing an effective English to SQL conversion process, it was decided to terminate that part of the task before it was completed. This decision was made for two reasons; first, because the project's primary focus is on the development of intelligent processes using expert systems and not in developing natural language processors; second, because the project had the foresight to acquire a commercially available natural language query processor that was able to translate easily from English to SQL.

Given that CRUDDES would not have to contend with the database's language, whatsoever, the creation of the interface between the NLQP and the transform filters became a fairly straight forward process. What was required was a means of storing, in a temporary file, the English question formed by CRUDDES so that it could be passed to the NLQP via a communications process. Since M.1 supported such a capability, this process was coded into CRUDDES, tested, and demonstrated.

TASK 10
CUSTOMIZATION OF THE NLQP FOR THE APPLICATION & ARCHITECTURAL VIEWS

The customization of the natural language query processor, THEMIS, was a very simple and straight forward process because the system is a functioning commercial product that has been built to support rather large and complex relational database systems. In addition, since CRUDDES is a prototype system that deals with only a portion of the selected database, the number of English queries it is capable of generating is rather small. Consequently the dictionary of lexicons that had to be added was also rather small (due to the limited number of information goals coded into the knowledge base). However, because the IUIS had to resolve ambiguity in the user to system communication, special attention was paid to the formulation of English queries on the CRUDDES side and their response on the database side.

If the CRUDDES system were expanded, the ability of the NLQP to handle imprecise queries would become a more important issue. Consequently, how the system will deal with imprecision is unclear. However, in the long term, it appears that any IUI system must be able to understand the user based on more than syntax. Presently, creating systems that understand queries at the contextual and semantic level of abstraction is quite difficult because of the limitation of M.1. However, with the use of third generation expert system development tools (e.g. ART, KEE) it will be possible to capture database context and semantics in the expert system knowledge base.

5.4 The Completion of CRUDDES

The last two tasks in the protocol are presently under development. Currently, the intention for CRUDDES is to serve as both a learning tool and a demonstration system.

After the current effort's completion, further development is not planned for the following reasons:

1. The CRUDDES effort demonstrated that expert systems can be used to develop

intelligent user interfaces to database systems,

2. More can be learned about developing an IUIS by moving as quickly as possible to a more robust expert system development environment.

Given the above, we believe that we now have the technical basis to begin the design and development of a next generation IUI system that should be able to function very effectively in many NASA operational database environments where research scientists need better and more responsive data management support.

CHAPTER 6
RESEARCH RESULTS AND FUTURE DIRECTION

6.1 Research Results

Based on the research to date, the following important results have been determined:

1. It is best to represent a database as a collection of logical views;
2. At a minimum the collection of views must include an architectural view and at least one application view;
3. The architectural view provides a logical taxonomy of the database;
4. The application views are where real database user expertise is captured because this is the view dominated by procedures and heuristics;
5. The context and semantics of a database can be captured and represented in an expert system;
6. The expert system must be linked to a natural language query processor to support the translation of the English formed question about the data into a query in the database's language;
7. For both development and explanation purposes, the expert system should reason in the domain of the user, using English;
8. The most efficient method for translating between views is by creating conceptual graphs (or transform filters) to allow translation amongst the various views;
9. Virtual objects, created via the clustering of real database objects, can only be represented in the knowledge base of the expert system.

6.2 Future Directions

The near term IDM effort plans are to extend the capabilities of the intelligent data management controller using an advanced expert system development tool like ART[22], or KEE[23] and to begin the development of some of the other IDM subsystems that were presented in a summary form in Chapter 2. Based on our present understanding of the capabilities of the more advanced development tools we plan to begin development of a second prototype IUI that supports, to some degree, the following capabilities:

- Structured object relationships with inheritance using a frame based schema,
- Concurrent data-driven and goal-driven data object identification and query formulation,
- Concurrent multiview representation of the database,
- Learning by example using the frame based schema to add new objects and virtual objects,

- Multiple application views for different domain-specific information gathering tasks.

As in the original design the NLQP, THEMIS, will be interfaced to the IUIS. However, in this case, the communications will be supported by a broad band local area network using the communication protocol TCP/IP[24].

In addition to the IUI technical effort, the project will also be formulating and assessing alternative designs for an advanced dynamic DBMS architecture that can create "on the fly" search structures and strategies to support a near real-time dynamic database system.

Over the next 3 to 5 years, it is the goal of the IDM effort to design, develop and implement a completely functioning intelligent data management system that will support:

- A sophisticated system interface that will be able to interact with and understand a user's data needs based on the users English and graphical representations of information entirely in a application context.
- An intelligent automatic data ingest and maintenance system that will allow spatial and non-spatial data to be automatically input into the various databases that the system will interface;
- A spatial database system that will be able to manage and reason about spatial information to support domain specific applications;
- A dynamic database system that will be able to perform complex data management operations by using pattern matching and "on the fly" network and tree structure construction to perform optimal object identification and selection.

Concurrent with the above technical development effort, the project will select a suitable candidate test bed project, such as Science and Applications Information System (SAIS) or Earth Observation System (EOS) to use for test and evaluation. Once selection has been made, a test bed IDMS will be customized based on the data maagement needs and technical direction of the selected project.

REFERENCES

1] Shi-Kuo Chang, Jyh-Sheng Ke, "Translation of Fuzzy Queries for Relational Database Systems", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-1, No. 3, July 1979.

[2] ibid.

[3] Campbell, W. J., Roelofs, L. H., "Artificial Intelligence Applications Concepts for the Remote Sensing and Earth Science Community",
Proceedings of the IX Pecora Conference, IEEE Publication Catalog No. 84CH2079-2, October 2 - 4 1984, PP 232.

[4] Sowa, J. F. "Conceptual Graphs For A Data Base Interface," *IBM J. Res Develop.,* July 1976.

[5] Date, C. J. "An Introduction to Database Systems", Third Edition, Addison-Wesley Publishing Co., Reading, MA, 1981, pp. 13 - 17.

[6] op. cit. Campbell, W. J., Roelofs, L. H., "Artificial Intelligence Applications Concepts for the Remote Sensing and Earth Science Community"

[7] Noll, C. E., "The Development of Selected Database Applications for the Crustal Dynamics Data Information System", NASA Technical Memorandum 83886, December 16, 1981.

[8] Buchanan, B. G., Shortliffe, E. H., "Rule-Based Expert Systems, The MYCIN Experiments of the Stanford Heuristic PROGRAMMING Project," Addison-Wesley Publishing Co., Reading, MA, 1984.

[9] Duda, R. O., Gaschnig, J. G., Hart, P. E., "Model design in the PROSPECTOR consultant system for mineral exploration. In. D. Machine, ed., Expert system in the micro-electronic age. Edinburgh; EDINBURGH University Press, pp.153-167, 1979

[10] D'Ambrosio, B., "Building Expert Systems with M1,"BYTE Magazine, June 1985.

[11] XyWrite Reference Guide," XYQUEST Corp, BEDFORD MA, 1985.

[12] SoftLogic Solutions, Inc. DoubleDos Reference Manual, Manchester, N. H. 1984.

[13] "Prolog - 86 Technical Summary," Solution Systems Inc. Norwell MA, 1984.

[14] op. cit., Shortliffe, E. H., Buchannan, B. G., "Rule-Based Expert Systems, The MYCIN Experiments of the Stanford Heuristic PROGRAMMING Project," pp. 302-313.

[15] "SQL/UFI Reference Manual, Version 4.0," ORACLE Corp., June 1984, D002-0684-840521.

[16] Codd, E. F., A Relational Model of Data for Large Shared Data Banks," *Commun. Ass. Comput.* Mach., Vol. 13, June 1970.

[17] op. cit. Date, C. J. "An Introduction to Database Systems", pp. 73 - 81.

[18] Denny, G. H. "An Introduction to SQL: A structured query language,", IBM Tech. Rep. RA93(28099), May 1977.

[19] Chamberlin, D. D., Boyce, R. F., SEQUEL: A structured English query language," *ACM-SIGMOD Workshop on Data Description, Access, and Control,* May 1974.

[20] Chen, P. P. S., "The entity-relationship model: Toward a unified view of data," *ACM Trans. Database Syst.,* Vol. 1, pp. 9-36, 1976.

[21] Davis, D. B., English: The newest computer language," High Technology, Vol 4, No.2, Feb. 1984.
pp. 59 - 65.

[22] Clayton, B. D., "ART Programming Primer," Inference Corp., Los Angles, CA, 1985.

[23]Kinnucan, P., "Software Tools Speed Expert System Development," High Technology, Vol 5, No. 3, March 1985, pp. 22 - 27.

[24] Tannenbaum, A. S., "Computer Networks," Prentice-Hall Inc. Englewood Cliffs, NJ, 1981, pp. 371 - 377.

APPENDIX A

APPENDIX A: SAMPLE DIALOG FOR APPLICATON CASE OF WHOLEPLATE MOTION

ENGLISH QUERY CONSULTATION

```
                          WELCOME to the

          CRUSTAL DYNAMICS DATABASE EXPERT SYSTEM (CruDDES)

                      Developed as part of the

              Intelligent Data Management Project

 This system  provides assistance for supporting access to the crustal
dynamics database at GSFC using the ORACLE database management system.
If you are certain of your scientific application and we support it,
CruDDES will direct you quickly to the information needed to generate a
query.  Otherwise, CruDDES will help you determine what information you
require to prepare a query and then assist you in generating a query.

NOTE--if you do not understand any question throughout the session,
       type "uncertain".
NOTE--if you are uncertain of how to respond please type "options".
       This phase of the consultation is application oriented.
Do you want:
     pt) plate tectonics info. or
     er) earth rotation  info. or
type "uncertain".
>> pt
 Do you want information related to:
    r) regional deformation
    p) plate stability
    w) whole plate motion
>> w
    We support two applications areas related to whole plate motion.
We can supply you with information either for the motions between
two regions on the respective plates or for the motions between entire
plates relative to each other.  The latter choice will supply information
needed to implement a least square method ,using all baselines, for the
rates of change between the entire plates.

Do you want:
   p)  plate motions between regions
   e)  entire motion between plates

>> e
 Choose two plates from this list:
  p)  pacific         n) nasca      na) north american c)  carribean**
  sa) south american  e) eurasian   a)  australian     af) african**
```

```
**no data from these plates, yet.

What is the letter of the reference or "fixed" plate?
>> p
 What is the letter of the "moving" plate?
>> na
 ->enter in the first year for your baseline measuring time
 (78 to 83).
>> 78
 -> enter in the final year for the time range (78 to 83)
>> 83

  Do you want data acquired through the satellite laser ranging (slr)
method or through the very long baseline interferometry (vlbi) method?

Type slr or vlbi.
>> slr
                        CRUDDES THEMIS SUB-EXPERT SYSTEM

  You have now entered a generic subroutine which determines how you want
your query presented.  The following English question has been formulated from
this consultation and will be presented to the database management system via
the natural language query interface:

Get all slr baselines between the pacific plate and the north american plate
during 1978 and 1983.

Please type "n" for the next page!
>> n

NOTE: remember to type "log off" if you logged printer earlier!!

Are you satisfied with the results?
>> yes

Thank you for your CRUD-es
M.1> log off
```

SQL QUERY CONSULTATION

```
M.1> load[crudds:sgl].
M.1> go.
                           WELCOME to the

          CRUSTAL DYNAMICS DATABASE EXPERT SYSTEM (CruDDES)

                      Developed as part of the

              Intelligent Data Management Project

 This system  provides assistance for supporting access to the crustal
dynamics database at GSFC using the ORACLE database management system.
If you are certain of your scientific application and we support it,
CruDDES will direct you quickly to the information needed to generate a
query.  Otherwise, CruDDES will help you determine what information you
require to prepare a query and then assist you in generating a query.

NOTE--if you do not understand any question throughout the session,
      type "uncertain".
NOTE--if you are uncertain of how to respond please type "options".
      This phase of the consultation is application oriented.
Do you want:
     pt) plate tectonics info. or
     er) earth rotation  info. or
type "uncertain".
>> pt.
Do you want information related to:
    r) regional deformation
    p) plate stability
    w) whole plate motion
>> w.
   We support two applications areas related to whole plate motion.
We can supply you with information either for the motions between
two regions on the respective plates or for the motions between entire
plates relative to each other.  The latter choice will supply information
needed to implement a least square method ,using all baselines, for the
rates of change between the entire plates.

Do you want:
   p)  plate motions between regions
   e)  entire motion between plates

>> e.
Choose two plates from this list:
  p)  pacific         n) nasca      na) north american c)  carribean**
  sa) south american  e) eurasian   a)  australian     af) african**

 **no data from these plates, yet.
```

```
What is the letter of the reference or "fixed" plate?
>> p.
What is the letter of the "moving" plate?
>> na.

  Do you want data acquired through the satellite laser ranging (slr)
method or through the very long baseline interferometry (vlbi) method?

Type slr or vlbi.
>> slr.
    timelegvals entered at the end of the knowledge base.
->enter in the first year for your baseline measuring time
 (78 to 83).
>> 78.
-> enter in the final year for the time range (78 to 83)
>> 83.

QUERY INFORMATION:

--You have two choices in accessing your desired information:
  either you can use a batch or canned query, if it exists, on the
  ORACLE system by typing "staboth5.ufi" or you can type the query by
  using the below information to generate 8 queries.

There are several alternatives for obtaining the proper information.
The recommended sources in order of preference are:
   a)  the canned query generation information before the
          parameter list
   b)  the printed version of the required query
   c)  the multiple screen presentation of required query information.

Which do you want?

>> b.

******NOTE: TURN ON YOUR PRINTER******

Please type "n" for the next page!
>> n.

COLUMN F_STATION HEADING "FROM" JUSTIFY CENTER FORMAT 99999
COLUMN S_STATION HEADING "TO" JUSTIFY CENTER FORMAT 99999
COLUMN CUR_NAME FORMAT A15 TRUNC
COLUMN PLATE FORMAT A15 TRUNC
SELECT F.STATION,F.CURR_NAME,F.PLATE,S.STATION,S.CUR_NAME,S.PLATE,
       BASELINE,CHORD,GEODESIC
FROM BASELINE78_SLRGSFC,SITES F,SITES S
WHERE           F_STATION = F.STATION
            AND S_STATION = S.STATION
            AND F.PLATE = pacific
            AND S.PLATE = north american
            AND F_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE83_SLRGSFC)
```

```
           AND S_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE83_SLRGSFC)
SELECT F.STATION,F.CURR_NAME,F.PLATE,S.STATION,S.CUR_NAME,S.PLATE,
       BASELINE,CHORD,GEODESIC
FROM BASELINE83_SLRGSFC,SITES F,SITES S
WHERE          F_STATION = F.STATION
           AND S_STATION = S.STATION
           AND F.PLATE = pacific
           AND S.PLATE = north american
           AND F_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE78_SLRGSFC)
           AND S_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE78_SLRGSFC)
SELECT F.STATION,F.CURR_NAME,F.PLATE,S.STATION,S.CUR_NAME,S.PLATE,
       BASELINE,CHORD,GEODESIC
FROM BASELINE78_SLRGSFC,SITES F,SITES S
WHERE          F_STATION-9000 = F.STATION
           AND S_STATION-9000 = S.STATION
           AND F.PLATE = pacific
           AND S.PLATE = north american
           AND F_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE83_SLRGSFC)
           AND S_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE83_SLRGSFC)
SELECT F.STATION,F.CURR_NAME,F.PLATE,S.STATION,S.CUR_NAME,S.PLATE,
       BASELINE,CHORD,GEODESIC
FROM BASELINE83_SLRGSFC,SITES F,SITES S
WHERE          F_STATION-9000 = F.STATION
           AND S_STATION-9000 = S.STATION
           AND F.PLATE = pacific
           AND S.PLATE = north american
           AND F_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE78_SLRGSFC)
           AND S_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE78_SLRGSFC)
SELECT F.STATION,F.CURR_NAME,F.PLATE,S.STATION,S.CUR_NAME,S.PLATE,
       BASELINE,CHORD,GEODESIC
FROM BASELINE78_SLRGSFC,SITES F,SITES S
WHERE          F_STATION-9000 = F.STATION
           AND S_STATION      = S.STATION
           AND F.PLATE = pacific
           AND S.PLATE = north american
           AND F_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE83_SLRGSFC)
           AND S_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE83_SLRGSFC)
SELECT F.STATION,F.CURR_NAME,F.PLATE,S.STATION,S.CUR_NAME,S.PLATE,
       BASELINE,CHORD,GEODESIC
FROM BASELINE83_SLRGSFC,SITES F,SITES S
WHERE          F_STATION-9000 = F.STATION
           AND S_STATION      = S.STATION
           AND F.PLATE = pacific
           AND S.PLATE = north american
           AND F_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE78_SLRGSFC)
           AND S_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE78_SLRGSFC)
SELECT F.STATION,F.CURR_NAME,F.PLATE,S.STATION,S.CUR_NAME,S.PLATE,
       BASELINE,CHORD,GEODESIC
FROM BASELINE78_SLRGSFC,SITES F,SITES S
WHERE          F_STATION      = F.STATION
           AND S_STATION-9000 = S.STATION
           AND F.PLATE = pacific
           AND S.PLATE = north american
           AND F_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE83_SLRGSFC)
           AND S_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE83_SLRGSFC)
SELECT F.STATION,F.CURR_NAME,F.PLATE,S.STATION,S.CUR_NAME,S.PLATE,
       BASELINE,CHORD,GEODESIC
FROM BASELINE78_SLRGSFC,SITES F,SITES S
WHERE          F_STATION      = F.STATION
           AND S_STATION-9000 = S.STATION
           AND F.PLATE = pacific
           AND S.PLATE = north american
           AND F_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE83_SLRGSFC)
           AND S_STATION IN
                 (SELECT S_STATION
                   FROM BASELINE83_SLRGSFC)
```

APPENDIX B

APPENDIX B: CRUDDES LISTING OF SEGMENT 1

(NOTE: this represents 19 out of 200 rules)

```
/*
                              <m.1> crudds.sgl

                         CRUSTAL DYNAMICS DATABASE
                              EXPERT SYSTEM

                             FIRST PROTOTYPE

                              JANUARY, 1986

                             Segment #1 OF 8

     This segment primarily deals wtih USER RESOLUTION AND PLATE TECTONICS
     (NOT including Whole Plate Motion between specific regions).
*/

initialdata = ['consultation over seg 1'].

/*   "Initialdata" is what the system is searching for. What this means is that
a session  will be over when the system has found the value of the initialdata
to be true. In this case it is "consultation over seg 1".
*/

/*============================= INITIALIZATION ==============================*/

/* The following allows for easy modification of dynamic parameters in the
system (i.e. file names that change) because the entire system has been divided
into eight (8) segments.

NOTE: Since this is a generic module, any change to this requires a similar
change in all other segments.
*/

segment-number = 1.

nocache(file-number-N).

whenfound('display shown') = do(set segment-number = 1).

noautomaticquestion(objects-N).


file-name-1 = 'V3CRUD.SG1'.
```

```
file-name-2 = 'V3CRUD.SG2'.

file-name-3 = 'V3CRUD.SG3'.

file-name-4 = 'V3CRUD.SG4'.

file-name-5 = 'V3CRUD.SG5'.

file-name-6 = 'V3CRUD.SUB'.

file-name-7 = 'V3CRUD.SQL'.

cache-name-1 = 'initdis.che'.

cache-name-2 = 'arvw.thm'.

/**************************INITIAL DISPLAY MODULE***************************/
/*   This area displays the Crustal Dynamics header for this system.
*/
nocache(initial-display-N).

rule-a1:  if   cache-name-1 = NAME and
 *             do(loadcache NAME) and
               initial-display-1= DESCRIP and
               initial-display-2= DESCRIP2 and
               initial-display-3= DESCRIP3 and
               display(DESCRIP) and
               display(DESCRIP2) and
               display(DESCRIP3)
          then 'display shown'.

/*
    This rule loads the external file initdis.che into cache. But due to
    the fact that the three descriptive sections in the file have been
    "nocached", the information will be displayed without expending cache.
*/

/*************************USER RESOLUTION MODULE*********************************/

rule-a2:  if 'display shown' and
               not(resolution = uncertain)
          then applications-view.

explanation(rule-a2) =
[nl,'The database has two basic types of scientific information',nl,
'which can be accessed by the user, which is either Plate techtonics',nl,
'or earth rotation information.  If the user is unsure of what he wants',nl,
'then an "uncertain response" will invoke the presentation of an',nl,
'architectural view of the database which can be used to identify',nl,
'and determine what information might be of interest',nl,nl].


rule-a3:  if 'display shown' and
             resolution = uncertain
          then  architecture-view.

explanation(rule-a3) =
[nl,'This rule is used to control the selection of a view of the',nl,nl,

'database that represents a logical taxonomy of architectural design',nl,nl,
'which can be used to understand, identify and select database infomrmation'].

/*=================== QUESTIONS: USER RESOLUTION ============================*/

question(resolution) =
['Do you want:',nl,
'     pt) plate tectonics info. or',nl,
'     er) earth rotation  info. or',nl,
'type "uncertain".'].


legalvals(resolution) = [pt, er, uncertain].

/*****************************APPLICATIONS-VIEW********************************/

/*====================== RULES: DETERMINE APPLICATION ======================*/

rule-a4:  if applications-view and
                resolution = er and
                file-name-3 = NAME and
                do(load NAME) and
                'consultation over seg 3'
          then 'consultation over seg 1'.

rule-a5:  if applications-view and
                resolution = pt
          then 'tectonics option'.

/*    !!!!!COMMENT!!!!!!:
     this rule is used to select the plate tectonics application option
*/

/*===================== RULES: REGIONAL DEFORMATION =======================*/

rule-a6:  if 'tectonics option' and
                select_plates_option = r
          then 'regional deformation'.
```

```
/*--------------- THIS SECTION IS NOT COMPLETED YET -------------------------*/

/*   !!!!!COMMENT!!!!!!:
     NOTE:  a search kb subroutine is coded to keep facts from cluttering
     up the cache or main memory.  It must search through
     best-site and best-station file.  subroutine name = CRUDDS.SUB.
*/

rule-a7:  if 'regional deformation' and
               select-reference-plate = PLATE and
               select-reference-region-from-PLATE = REGION
          then chosen-region-reference-plate = REGION.

rule-a8:  if chosen-region-reference-plate = REGION and
               file-name-6 = NAME and
               do(load NAME) and
               do(set subroutine_iteration_flag = 1) and
               'subroutine done'
          then 'sub called from reg def'.

rule-a9:  if 'sub called from reg def' and
               chosen-reference-station = STATION and
               display(['this is a test to determine the correct',
               ' station number--station = ',STATION,nl])
          then objects-1 = sorry_not_completed_yet.

rule-a10: if 'regional deformation'
          then fields-1 = bye_bye.

/*======================== RULES: PLATE STABILITY ========================*/

rule-a11: if 'tectonics option' and
               select_plates_option = p
          then 'plate stabilization'.

rule-a12: if 'plate stabilization' and
               select-reference-plate = REF
          then select-moving-plate = REF.

rule-a13: if 'plate stabilization' and
             select-moving-plate = REF and
             low-time = LOW and
             high-time = HI and
             exper-type = TYPE
          then ref-plates-known.

/*======================== RULES:INTERPLATE MOTION ========================*/

rule-a14: if 'tectonics option' and
```

```
               select_plates_option = w
          then 'whole plate motion'.

explanation(rule-al4) =
[nl,'Within the plate techtonics application area there are three general',nl,
'areas of information: the motion of whole Earth plates relative to one',nl,
'another (whole plate motion), the deformation of a single Earth plate',nl,
'(regional deformation), and the vertical motion of a plate as it floates',nl,
'on the mantle (plate stability).',nl,nl].

rule-al5: if 'whole plate motion' and
             select_plates_motion_type = e
          then 'entire plate motion'.

rule-al6: if 'whole plate motion' and
               select_plates_motion_type = p and
               file-name-2 = NAME and
               do(load NAME) and
               'consultation over seg 2'
          then 'consultation over seg 1'.

/*==================== RULES: SELECT PLATES PARMS  ===========================*/

rule-al7:    if 'entire plate motion' and
               select-reference-plate = REF and
               select-moving-plate = MOVING and
               low-time = LOW and
               high-time = HI and
               exper-type = TYPE
            then both-plates-known.

/*============================================================================*/
/*========================== RULES:  POLAR MOTION ===========================*/

/*
                               THIS SECTION IS IN SEGMENT THREE
                                  (CRUDDS.SG3)
*/
/*==================== RULES: FIND objects AND FIELDS ========================*/

rule-al8:   if ref-plates-known or
               both-plates-known
            then objects-1 = 'parms determined'.

rule-al9:   if ref-plates-known or
               both-plates-known
            then fields-1 = 'parms determined'.
```

```
/*================== QUESTIONS:  PLATE TECTONICS ========================*/

question(select_plates_option) =
'Do you want information related to:
    r) regional deformation
    p) plate stability
    w) whole plate motion'.

legalvals(select_plates_option) = [r, p, w, unknown].

question(select_plates_motion_type) = [
'   We support two applications areas related to whole plate motion.',nl,
'We can supply you with information either for the motions between',nl,
'two regions on the respective plates or for the motions between entire',nl,
'plates relative to each other.  The latter choice will supply information',nl,
'needed to implement a least square method ,using all baselines, for the',nl,
'rates of change between the entire plates.',nl,nl,
'Do you want:',nl,
'    p)  plate motions between regions',nl,
'    e)  entire motion between plates',nl].

legalvals(select_plates_motion_type) = [p, e, uncertain].

question(select-reference-plate) =
['Choose two plates from this list:',nl,
'  p)  pacific          n) nasca      na) north american c)  carribean**',nl,
'  sa) south american   e) eurasian   a)  australian     af) african**',nl,nl,

'**no data from these plates, yet.',nl,nl,

'What is the letter of the reference or "fixed" plate?'].

legalvals(select-reference-plate) = [p,sa,n,na,e,a,c,af].

question(select-moving-plate) =
'What is the letter of the "moving" plate? '.

legalvals(select-moving-plate) = [p,sa,n,na,e,a].

question(low-time) = '->enter in the first year for your baseline measuring time
 (78 to 83).'.

legalvals(low-time) = integer(78,83).

question(high-time) = '-> enter in the final year for the time range (78 to
83)'.

legalvals(high-time) = integer(78,83).
```

```
/*================ FACTS: PLATES =========================================*/

/*    !!!!!COMMENT!!!!!!:
      These parameters are used in the query format for whole plate
      motion and plate stability (specifically in the plate fields
*/

noautomaticquestion(plate(P)).

plate(p) = 'pacific'.

plate(sa) = 'south american'.

plate(n) = 'nasca'.

plate(na) = 'north american'.

plate(e) = 'eurasian'.

plate(a) = 'australian'.

plate(c) = 'carribean'.

plate(af) = 'african'.

/*================ QUESTIONS:  DETERMINE REGION ==============================*/
/*    !!!!!COMMENT!!!!!!:
      NOTE:  These regions were taken from table = sites in field = region.
*/

question(select-OBJECT-region-from-na) =
['Which region do you want from the north american ',OBJECT,' plate:',nl,
 '       ca) california',nl,
 '       al) alaska',nl,
 '       na) north america',nl,
 '       pa) pacific',nl,
 '       at) atlantic',nl].

question(select-OBJECT-region-from-sa) =
['Which region do you want from the south american ',OBJECT,' plate:',nl,
 '       sa) south america',nl,
 '       pa) pacific',nl,
 '       at) atlantic',nl].

question(select-OBJECT-region-from-p) =
['Which region do you want from the pacific ',OBJECT,' plate:',nl,
 '       pa) pacific',nl,
 '       ca) california',nl,
 '       na) north america',nl].
```

```
/* NOTE: nasca plate has one region*/

select-OBJECT-region-from-n = pa.

question(select-OBJECT-region-from-e) =
['Which region do you want from the eurasian ',OBJECT,' plate:',nl,
 '     eu) europe',nl,
 '     pa) pacific',nl,
 '     me) mediterranean',nl,
 '     as) asia',nl].

question(select-OBJECT-region-from-a) □
['Which region do you want from the australian-indian ',OBJECT,' plate:',nl,
 '     a)  australia',nl,
 '     as) asia',nl].

legalvals(select-OBJECT-region-from-na) = [ca, al, na, pa, at].

legalvals(select-OBJECT-region-from-sa) = [sa, pa, at].

legalvals(select-OBJECT-region-from-p) = [pa, ca, na].

legalvals(select-OBJECT-region-from-e) = [eu, pa, me, as].

legalvals(select-OBJECT-region-from-a) = [a, as].

/*================= WHENFOUND CLAUSES:  APPLICATIONS VIEW ====================*/

whenfound(exper-type = slr) = do(set process-location = gsfc).

whenfound(exper-type = slr) □ do(set proc-loc-needed □ no).

/*   !!!!!COMMENT!!!!!!:
     The next meta proposition allows for experiment analysis location
     to be defaulted to gsfc when user doesn't care about process location.
*/

whenfound(station_type_needed = no) = do(set station_type = fixed).

/******************************************************************************/

/*====================== QUESTIONS: PARAMETERS ===============================*/

question(satellite) =
'From which satellite do you want the data (lageos,bec, or starlette)?'.

legalvals(satellite) = [lageos,bec,starlette,unknown].
```

```
question(exper-type) =
[nl,'  Do you want data acquired through the satellite laser ranging (slr)',nl,
 'method or through the very long baseline interferometry (vlbi) method?',nl,nl,
 'Type slr or vlbi.'].

legalvals(exper-type) = [slr, vlbi, uncertain].

question(time) =
'From which year do you want the data (77-81)?'.

timelegvals: legalvals(time) = integer(76, 85).

question(station_type_needed) =
'Does it matter if you choose fixed or mobile stations? (yes or no)'.

legalvals(station_type_needed) = [yes, no].

question(station_type) =
'Then, which do you want (fixed or mobile)?'.

legalvals(station_type) = [fixed, mobile].

/*============== WHENFOUND CLAUSES:  RANGE PARAMETERS ON TIME ===============*/

/*    !!!!!COMMENT!!!!!!:
      This sections handles the variability in the time ranges for the satell-
      ites catalogues. This is a generic module used in all the segments.
*/

whenfound(satellite = lageos) = do(add
                                  timelegvals: legalvals(time)=integer(76,85)).

whenfound(satellite = bec) =  do(add
                                   timeleglvals: legalvals(time)=integer(76,81)).

whenfound(satellite = starlette) = do(add
                                   timeleglvals: legalvals(time)=integer(76,81)).

whenfound(exper-type = slr) □ do(add
                                timelegvals: legalvals(time) = integer(78,83)).

/*======================= MISCELLANEOS ====================================*/

noautomaticquestion(display_flag).

/*    !!!!!COMMENT!!!!!!:
      These nocache meta-facts allow the system to switch between segments
      several times.  This must be present in every segment.
*/
```

```
nocache('consultation over seg 1').

nocache('consultation over seg 2').

nocache('consultation over seg 3').

nocache('consultation over seg 4').

nocache('consultation over seg 5').

/*================== RULES:  CONSULTATION OVER SEG 1 ========================*/

rule-a20:  if segment-number = N and
              objects-N is known and
              fields-N is known and
              file-name-7 = NAME and
              do(load NAME) and
              'subroutine done'
            then 'consultation over seg 1'.

explanation(rule-a20) =
[nl,'You dummy. You ought to know better than to ask why.',nl,nl].

/*================= QUESTIONS:  DISPLAY QUERY INFO ==========================*/

question(next_page) = [nl,'Please type "n" for the next page!'].

nocache(next_page).

/*******************SEGMENT SWITCHING CONTROL MODULE*************************/

/*    !!!!!COMMENT!!!!!!:
     NOTE:  architecture-view is a logical break rather than a physical one as
     in 'consultation over seg 2'.

     NOTE:  the segment switches to a specific segment for a logical reason.

*/

rule-a21: if architecture-view and
               display([nl,'wait a second while I load in some more',
                    ' knowledge...',nl]) and
               do(set applications-view = unknown) and
               file-name-4 = NAME and
               do(load NAME) and
               'consultation over seg 4'
          then 'consultation over seg 1'.
```

```
rule-a22: if objects-1 is sought and
                fields-1 is sought and
                display([nl,'wait a second while I load in some more',
                     ' knowledge...',nl]) and
                do(set applications-view = unknown) and
                file-name-4 = NAME and
                do(load NAME) and
                'consultation over seg 4'
          then 'consultation over seg 1'.

/***************************************************************************/

question(user_satisfied) = [nl,
'NOTE: remember to type "log off" if you logged printer earlier!!',nl,nl,
'Are you satisfied with the results?'].

legalvals(user_satisfied) = [yes, no].

/*======================================================================
```

The following are some of the M.1 functions used in this system:

nocache(EXPRESSION) :

Values of nocache expressions will not be stored permanently in the cache. When such expressions are evaluated, the results are temporarily cached and the proposition that depends on the results is tested. Then, before proceeding, M.1 removes the facts from the cache. If the value of a nocache expression is needed later in a consultation or session, M.1 seeks it again.

whenfound(EXPRESSION) □ LIST or
whenfound(EXPRESSION □ VALUE) = LIST :

A whenfound knowledge base entry allows you to modify the M.1 inference process by designating what M.1 should do whenever a value for the EXPRESSION is found.

noautomaticquestion(EXPRESSION) :

This prevents an automatic question from being generated for the expression even if there are no other knowledge base entries for the expression. If the expression is a variable, then it turns off the automatic question facility for all expressions.

explanation(LABEL) = TEXT :

Explanation displays the string TEXT in response to "why" queries regarding a knowledge base entry whose label matches LABEL. The text must be either an atom or a list whose elements are atoms and formatting commands("nl" and "tab"). This meta-fact enables knowledge engineers to substitute customized explanations or to offer acceptable explanations about a path of reasoning that contains proprietary information. LABEL can be any expression and may contain variables.

Ordinarily, rules, presuppositions, whenfounds, and initialdata are the entries used with the "explanation" meta-fact.

question(EXPRESSION) = TEXT :
The given text is used to ask the end-user for the value of the expression. The text must be either a string or else a list of strings, variables, and special formatting commands. The answer provided will be checked against the "legalvals" for the expression, if any have been provided by the engineer.

legalvals(EXPRESSION) = integer :
M.1 will only allow the end-user to answer with integers for the value of this expression.

legalvals(EXPRESSION) = integer(LOW, HIGH) :
An integer between LOW and HIGH inclusive is a suitable response as a value for this expression.

legalvals(EXPRESSION) = LIST :
The elements of the list are acceptable values for the expression. M.1 will attempt "auto completion" (meaning that the answer must be one of a set of fixed responses and only sufficient letters to unambiguously distinguish the response need be typed or M.1 will repeat the question) on responses to identify an element from this list.

legalvals(EXPRESSION) = number :
Any number is an acceptable value for EXPRESSION.

legalvals(EXPRESSION) = number(LOW, HIGH) :
The real or integer numbers given as responses must lie within the range LOW to HIGH, inclusive.

legalvals(EXPRESSION) = real :
Any real (floating point) number is a suitable response. A real number must include a decimal point with at least one digit on either side, unless an exponent is used. Either fixed or floating point notation is acceptable.

legalvals(EXPRESSION) = real(LOW, HIGH) :
The real number given as a response must lie within the range of real numbers, LOW to HIGH, inclusive.

*

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.
NASA TM-87821

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
The Development of a Prototype Intelligent User Interface Subsystem for NASA's Scientific Database Systems

5. Report Date
June 1987

6. Performing Organization Code
634

7. Author(s)
William J. Campbell, Larry H. Roelofs, and Nicholas M. Short, Jr.

8. Performing Organization Report No.
87B0266

10. Work Unit No.

9. Performing Organization Name and Address
Goddard Space Flight Center
Greenbelt, Maryland 20771

11. Contract or Grant No.

13. Type of Report and Period Covered
Technical Memorandum

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D. C. 20546-0001

14. Sponsoring Agency Code

15. Supplementary Notes

William J. Campbell: Goddard Space Flight Center, Greenbelt, Maryland.
Larry H. Roelofs: Computer Technology Associates, McLean, Virginia.
Nicholas M. Short, Jr.: Science Application Research, Inc., Lanham, Maryland.

16. Abstract

The National Space Science Data Center (NSSDC) has initiated an intelligent Data Management (IDM) research effort which has as one of its components the development of an Intelligent User Interface(IUI). The intent of the IUI effort is to develop a friendly and intelligent user interface service that is based on expert systems and natural language processing technologies. The purpose of such a service is to support the large number of potential scientific and engineering users that presently have need of space and land related research and technical data but have little or no experience in query languages or understanding of the information content or architecture of the databases of interest. This technical memorandum presents prototype Intelligent User Interface Subsystem (IUIS) using the Crustal Dynamics Project Database as a test bed for the implementation of CRUDDES (Crustal Dynamics Expert System). Presently, the knowledge base has more than two hundred rules and represents a single application view and the architectural view. Operational performance using CRUDDES has allowed non-database users to obtain useful information from the database that previously required an expert database user or the database designer.

17. Key Words (Suggested by Author(s))

Intelligent Data Management, Conceptual Views, Expert Systems, Natural Language User Interface, Artificial Intelligence, LISP, Symbolic Processing. Metadata

18. Distribution Statement
Unclassified - Unlimited

Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	~~18~~	~~A02~~

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, Virginia 22161-2171

NASA-Langley, 1987