ICIS 1987 Proceedings

International Conference on Information Systems (ICIS)

1987

A KNOWLEDGE BASED APPROACH FOR RESOURCE MANAGEMENT

Ajay S. Vinze
University of Arizona

Mari M. Heltne
University of Arizona

Minder Chen
University of Arizona

Jay F. Nunamaker
University of Arizona

Benn R. Konsynski
University of Arizona

Recommended Citation

Vinze, Ajay S.; Heltne, Mari M.; Chen, Minder; Nunamaker, Jay F.; and Konsynski, Benn R., "A KNOWLEDGE BASED APPROACH FOR RESOURCE MANAGEMENT" (1987). *ICIS 1987 Proceedings*. 42.
http://aisel.aisnet.org/icis1987/42

A KNOWLEDGE BASED APPROACH FOR RESOURCE MANAGEMENT

Ajay S. Vinze
Mari M. Heltne
Minder Chen
Jay F. Nunamaker, Jr.
Benn R. Konsynski*

Department of Management Information Systems
College of Business and Public Administration
University of Arizona
*Visiting Professor, Harvard Business School
for the academic year 1987-88.

ABSTRACT

This paper discusses the applicability of a knowledge based system to resource management in the context of information centers. The Information Center Expert (ICE) system has been developed in the MIS Department of the University of Arizona to support the consultation process of information center personnel. The system determines the (software) resource requirements of the end-users and makes appropriate recommendations. ICE further aids the management of the IC software resources by keeping track of user consultations and the recommendations made.

Issues of knowledge requirements, acquisition, representation and implementation of ICE are discussed. ICE is currently being tested at IBM/Endicott (New York), IBM/Tucson (Arizona) and the Center for the Management of Information at the University of Arizona. Preliminary feedback from users has confirmed the applicability of the knowledge based approach to resource management. The implications of this approach for future research are discussed.

1. RESOURCE MANAGEMENT

One indicator of the success of any organization operating in a competitive market is how effectively it uses its resources to support the production of goods and services (Porter 1985). The resource management process is "the set of all activities involved in the optimal allocation and administration of an organization's human, financial and physical resources, to fulfill the organization's mission and achieve its goals and objectives" (Bender 1983). The ability to identify resource needs and to acquire and allocate resources are basic requirements for effective management.

It is widely believed that the reuse of resources is a key to improving productivity and quality within organizations (Biggerstaff and Richter 1987). While this strategy holds great promise, it is one whose promise has been largely unfulfilled. To reuse resources, one first must be able to find them. Thus, an important aspect of the management of reusable resources is a method for identifying and accessing existing resources. It is a classification and retrieval problem (Prieto-Diaz and Freeman 1987).

Rowe, Mason and Dickel (1985) state that effective resource management requires tools for assessing resource requirements and decision aids for determining the best allocation. In the situation where the resources are reusable, then, we need tools that will help accomplish the process of classifying and retrieving existing organizational resources. There are two levels of reuse: (1) the reuse of data and knowledge, and (2) the reuse of specific objects or components. In both cases, the attributes of the reusable resource must be matched with the attributes of the new situation in which they are needed. One of the instances in which this matching process is done most often is the Information Center, where software resources are matched with end users and their needs.

1.1 Resource Management in the Information Center

There appears to be a consensus that, before long, end user computing will consume a majority of company computing resources. In places, this is already true. It is predicted that by 1990, end user computing will represent as much as 75% of the total computing capacity of the typical American corporation (Benjamin 1982). Consensus also suggests that the best general strategy for managing end user computing is to give end users adequate computing tools, establish standards, provide the necessary data resources, and encourage good computing practices

In many corporations, implementation of this strategy has been undertaken in an entity that was given the name Information Center (IC) (Røyksund 1987). Its mission is to "help users help themselves" by collecting and disseminating information about available computing resources (equipment, user developed systems, software packages, and data). Several types of service to end users are expected:

1. Consultation: Information Center personnel work with end users to help them analyze their problems and clarify their needs for computing resources.
2. Training: The IC functions as a center for learning about software and hardware products.
3. Technical Expertise: The IC provides technical assistance for the user in selecting hardware and software. Often there is an effort to establish policy for the standardization of these resources.

Whatever the demonstrated value and continuing need for the Information Center methodology for managing end user computing, there are hard questions to be answered if ICs are to continue to be successful. A recent study (Brancheau, Vogel and Wetherbe 1985) reported that end users expect to be even more dependent on the Information Center in the future than they now are, that they anticipate needing more support services and training, and that it will be more important than ever to remain "current" on new applications of technology. Thus, information centers are being subjected to increased user expectations, higher demand for integrated applications, and growing pressure to accomplish more with fewer resources.

Respondents to a survey reported in the 1986 *AMA Report on Information Centers* indicate that evaluations of software for end users are daily fare for 91.5 percent of their centers. Those ICs have put extensive effort into software evaluations because the consultants are so frequently asked for opinions by end users. "Requests for assistance in hardware and software selection come thick and fast, and require a matchmaking role between the end user's requirements and the capabilities of the technology" (Bohl 1986).

Information Systems people who still are dealing with backlogs of application development requests see a parallel overloading of demand on resources developing. Unless the organization is willing to repeat the bottleneck experienced by data processing departments, this time under the aegis of an Information Center, the IC must begin to offload some of its responsibilities for software evaluation, consulting and training. The task is to distribute the expertise provided by the Information Center throughout the organization, to leverage expertise about the technology by use of the technology itself (Røyksund 1987). The assumption is that at least most, if not all, staff members in an IC are more or less "expert" in the systems they support, but there are not enough experts to go around. So the expertise must be captured and applied in other ways. Harmon and King (1985) suggest that expert systems are particularly helpful in places where "a few key individuals are in short supply...[where] they spend a substantial amount of time helping others." Therefore, artificial intelligence is certain to find a place in information centers.

1.2 Information Center Expert (ICE) System

The Information Center Project at the University of Arizona Department of Management Information Systems, funded in part by the IBM Corporation, Endicott, New York, has resulted in the design and implementation of an expert system specifically for supporting the consultation activities of information center personnel. The system, known as ICE (Information Center Expert), is a rule-based knowledge system intended to be used in consultation with users who seek software and training resources (Heltne, et al. 1987).

The project undertaken was to build a system that could support the major activities of the IC consultants, including consultation, policy enforcement, tracking of end user needs, target marketing, and training of end users. ICE models the expertise of consultants at five information center locations: three at IBM/Endicott, one at IBM/Tucson, and one at the Center for Management of Information (CMI) in the College of Business and Public Administration at the University of Arizona. While the purpose of each of these ICs is "to provide you with tools and techniques that will allow you to retrieve, analyze, manipulate and present data more effectively..." (Wallace 1986), the centers differ in type of customers served, and therefore in the set of software resources supported. Clients of the information centers have ranged from application programmers with extensive skills in use of computers, to engineers and financial analysts who use computer packages as tools to do their jobs more efficiently and effectively, to students and staff who may never have used a computer before. To respond to such diverse users, ICE had to be designed with a flexible architecture that would allow the knowledge base to represent many unique sets of software tools without changes to the orginal rule base.

2. A KNOWLEDGE-BASED APPROACH TO RESOURCE MANAGEMENT IN AN INFORMATION CENTER

To develop a knowledge-based application, we first have to ask "What knowledge is required?" In Section 2.1, we therefore discuss what knowledge is required for resource management. In Section 2.2, the architecture of ICE is presented. Discussions of the knowledge system development and the design of the system components are found in Section 3.

2.1 Knowledge Requirements for Resource Management

The knowledge based approach, which captures the functionalities of resources, helps users identify what they really need. The approach is based on a classification and pattern matching process, as described by Clancey (1984). The knowledge required by an expert system for resource identification and assignment is as follows:

1. Knowledge of resources. To capture the knowledge of resources, a classification scheme is needed to identify the similarities and differences of a set of resources. The scheme should be based on salient features of the resources. Several schemes are possible, but the one chosen must be consistent, easy to use, accurate, and flexible. Knowledge of the software resources supported by the information center was captured and classified using a scheme based on Kelly's Repertory Grid, as described in Section 3.1.

2. Knowledge of the users of the resources. Users of corporate computing resources are at many different knowledge and skill levels. At one extreme, they know exactly the product or service they want. They already have the solution and only need help in securing it. At the other extreme, users have only abstract ideas about what they need. They know only the symptoms of their situation, not the solution, and need help determining how to fulfill their needs. In order for an expert system to make this determination, knowledge of users must be elicited that includes their background task environment, their skill level, and their preferences (Winograd and Flores 1986). ICE gathers this information during dialogue with each user and stores it in the User Profile and Problem Profile as described in Section 3.3.

3. Knowledge of how to select among suitable or competing resources. Knowledge of the selection process is represented by decision rules and a selection algorithm which lists the preferable alternatives in linear order. The knowledge of the Resources Profile, the weighting of decision criteria, user requirements and preference must all be taken into consideration in the selection algorithm used to recommend resources to the user. These ideas are discussed in Section 3.3.

Resource identification and assignment can be thought of as a continuum, with supply and demand on the two ends and the matching process as the medium for determining how demand can be satisfied given the supply characteristics. The ability of IC consultants to perform this task by matching the skills and requirements of their clients to the resources of the IC is an instance of the resource identification and assignment problem. The next sections describe the application of expert system technology to this problem within the context of the corporate and academic information centers that participated in this project.

2.2 Information Center Expert System Architecture

The architecture of ICE, depicted in Figure 1, is built using ESE/VM, an expert system shell developed by IBM. In addition to the rule-based implementation, a database defining the software resources of the IC and a procedure for tool selection are also used.

There are five major components of the ICE architecture: (1) reasoning control, (2) intelligent dialogue, (3) selection algorithm, (4) maintenance tool for ICE, and (5) tracking facility. Purposes and implementation of these five components are discussed below.

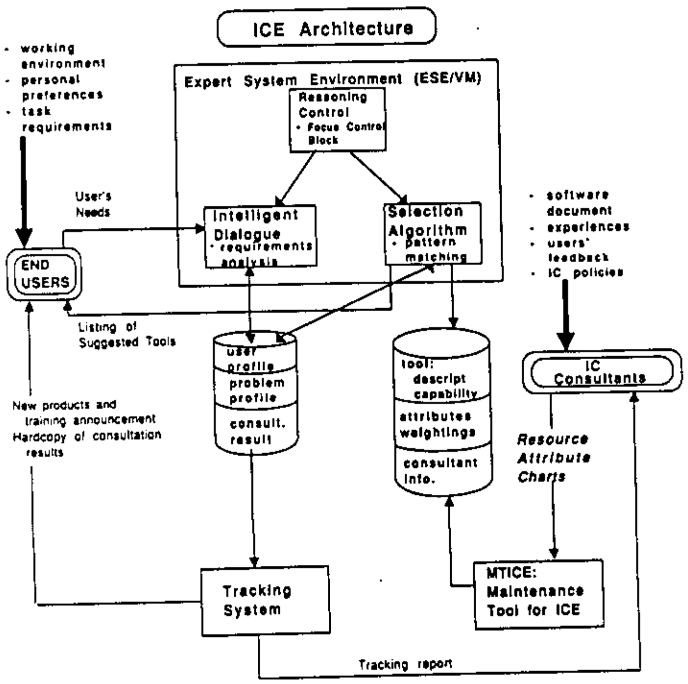


Figure 1. The Architecture of ICE

Reasoning Control: The reasoning control subsystem controls the process of user consultation. In a standard ICE consultation, the flow of control is as follows: collection of the user's background knowledge or profile, analysis of the user's current requirements, and initiation of the selection algorithm. The consultation concludes with software tools being recommended to the user.

Intelligent Dialogue: The inferencing mechanism uses rules to classify the user's tasks and requirements for software tools. The user's response to the system's queries determines the flow of the dialogue, i.e., ICE will ask relevant questions for collecting detailed functionalities of a high level requirement. Users' background knowledge and requirements for performing their tasks are acquired by this intelligent dialogue subsystem. Section 3.4 details the design and implementation of this subsystem.

Selection Algorithm: Once the user profile has been collected, the reasoning control subsystem passes control to the selection algorithm, which tries to match the user's needs and preference with the functionalities of tools. Tool recommendations are listed, ordered by their confidence levels in satisfying the user's needs. Users can check the tool descriptions and find the consultants responsible for the suggested tools. The algorithm of tool selection is described in Section 3.3.3.

Maintenance Tool for ICE (MTICE): MTICE is used to maintain information about tool resources. Consultants or IC managers use MTICE to add, update, or delete software tools supported by the IC. MTICE is described in Section 3.3.2.

Tracking Facility: A tracking system captures consultation results to support the following functions: print out the consultation result for a user, target marketing (e.g., use statistical consultation data to identify the target audiences for new products and training session announcements), and software purchasing and supporting decisions. The tracking facility is presented in Section 5.

3. BUILDING THE KNOWLEDGE SYSTEM

The task of building a knowledge system has been compared to tasks in mining (Hayes-Roth 1984):

Engineering Activities	*Knowledge Process Tasks*	*Engineering Products*
Mining	Knowledge Acquisition	Concepts & Rules
Molding	Knowledge System Design	Framework & Knowledge Representation
Assembling	Knowledge Programming	Knowledge Base & Inference Engine
Refining	Knowledge Refinement	Revised Concepts & Rules

3.1 Mining: Knowledge Acquisition

"Knowledge does not come off-the-shelf, prepackaged, ready for use" (Hayes-Roth 1984). The process of extracting knowledge, called knowledge acquisition, involves eliciting from experts or other sources the basic concepts of the problem domain, usually involving one or more of the following methods: interview, analogy, induction from example, observation or experimentation, prototyping, and reasoning from deep structure (Michalski, Carbonell and Mitchell 1983). A complete and correct description of the expert's knowledge must be integrated into an overall knowledge system architecture. This process has been highly labor-intensive, becoming a major difficulty for many expert system builders (Boose 1986b). New tools or aids are needed to assist in the knowledge acquisition process.

Knowledge for the Information Center Expert (ICE) was acquired by interviews, both informal and structured, by observation, and by example. Information about the consultation process, during which data is collected about the end user's background and current problem, was gathered through extensive interviewing of IC consultants, as well as observations of the process. This information was later represented as parameters and rules in the User Profile and Problem Profile of the knowledge base. Prototyping was used to iterate with the consultants to obtain feedback on the validity of the representation and dialogue.

In order to adequately elicit knowledge to build the Tool Profile, an extensive search was conducted of organizational documents and manuals, and structured interviews were held with consultants. Charts developed for this project, called Resource Attribute Charts, were used to structure the interviews to help the consultants classify and categorize the resources. To differentiate the software products in the knowledge base, all the software tools had to be compared. The Resource Attribute Charts are based on Kelly's Repertory Grid (Kelly 1955) in which knowledge is elicited about objects by asking the subject to compare the objects in groups of two or three. When groups of two are used, the subject is asked to name an attribute that distinguishes one from the others. When groups of three are used, the subject is asked to name an attribute that two of them have in common that distinguishes them from the third.

Our adaptation of Kelly's Grid, similar to Boose's ETS (1986a), allowed Information Center consultants to compare software within each of the software categories and define the attributes which distinguished them from one another. The grid methodology was especially appropriate because attributes that were common to all software products could be ignored.

The Resource Attribute Charts (RAC) elicited from the IC consultants the resource recommendations that were to be given by ICE. They also gathered vocabulary and identified attributes of the software resources and the relative weights of the attributes. RAC provided assistance in the knowledge engineering process by providing a structure for interviewing the expert, analyzing the information, and producing the resource profile for the knowledge base. Four steps are involved: initial knowledge elicitation, elicitation of attributes, attribute weighting, and attribute value assignment (Heltne 1987).

3.1.1 Initial knowledge elicitation

RAC first elicits from the expert conclusions, called elements, that should be determined by the Information Center Expert within the major categories of software previously agreed upon by the information center managers.

Major Software Categories
Data Management
Data Analysis
Graphics
Document Preparation
Project Management
Utilities
Programming
Integrated Packages

The expert is asked to list all the software resources that fall in a chosen category. Figure 2.1 shows the software that was listed for Data Analysis in one example. These software packages represent the possible solutions given by ICE in this category.

3.1.2 Elicitation of attributes

Next the expert compares successive groups of three packages and names an important attribute that distinguishes any two packages from the third (see Figure 2.2). By comparing three elements at a time, the consultant must at the same time think about both similarities and differences. This step is then performed iteratively until a list of classification attributes has been elicited for each category of software. Two final steps then establish priorities among the attributes and evaluate each of the elements (software packages) on the attributes.

3.1.3 Attribute weighting

The expert now must determine the relative importance of each attribute in selecting a recommended software package. For this purpose, Kelly's binary rating method has been extended to include scales, on which one end represents "crucial" attributes and the other "optional" attributes, with the following weights assigned: absolute or essential in importance (10, 9, 8), important but not absolutely essential (7, 6), moderate importance (5, 4), optional (3, 2, 1). Figure 2.3 shows the attribute weightings chosen by an IC consultant for Data Analysis tools, using the 1 to 10 scale and based on perceptions of the necessary attributes of software tools particular for a user.

3.1.4 Determining attribute values

The last step of RAC consists of completing a series of charts (Figure 2.4) to set the values of each attribute for each element. The elements are the software packages and values must be assigned to each attribute to indicate how much better one package is than the other with respect to that attribute. The following values are used: Absent to Poor (1, 2, 3), Acceptable but Below Average (4, 5), Average to Good (6, 7), Very Good to Excellent (8, 9, 10). These elements, attributes, weights, and attribute values are stored in four external files which together make up a Tool Profile that is searched during each consultation to match the most appropriate tool to the user's needs and skills. The Tool Profile is described in Section 3.3.1.

3.2 Molding: Knowledge System Design

Knowledge system design produces a framework or architecture for the knowledge system. Like architectural principles in housing construction, design principles suggest the broad outlines of a construction task without specifying the details (Hayes-Roth 1984). Based on these principles, an appropriate scheme is selected for representing the domain knowledge.

Two characteristics of knowledge important in expert systems development are context dependency and knowledge stability (Krcmar 1985). Context dependency describes how universal certain knowledge is, and stability is the change over time of the knowledge that is represented. These characteristics dictate two important design considerations for the Information Center Expert System: (1) maintainability and (2) transportability.

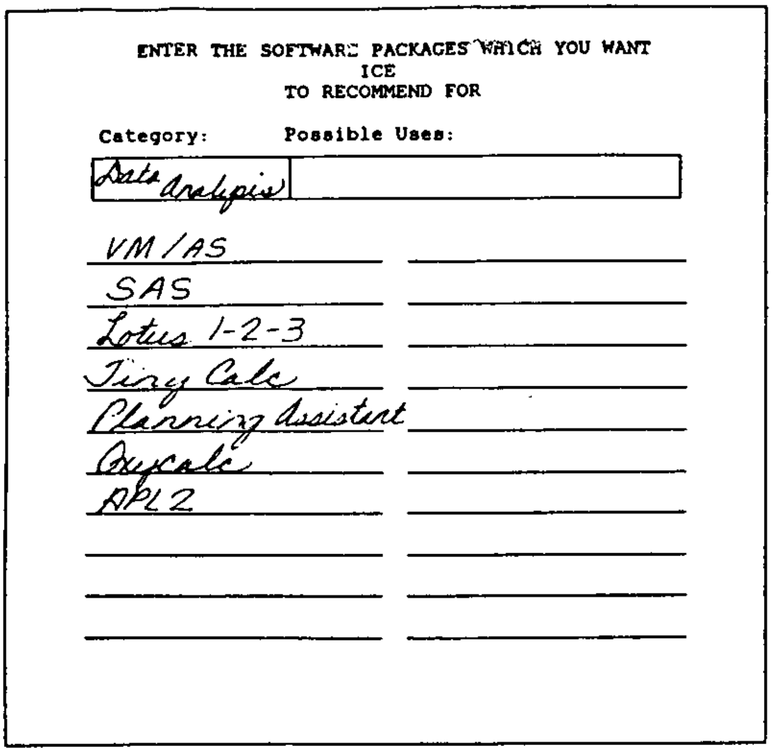
3.2.1 Maintainability

Maintainability is an extremely important issue because of the constant availability of new and different resources. Software tools are being introduced into the market at a very rapid rate; to stay competitive an IC must be able to continually adapt to this unstable and dynamic environment.

Most rule-based expert systems model problem solutions directly into the rules; that is, using IF/THEN statements, recommendations are "hard-coded" into the response portion of the rule. This method, however, is not appropriate in the dynamic environment of the IC. When new software tools are adopted for use, there must be an easy and efficient way to add them to the system without rewriting and recoding rules. By defining the tools in an external database, and using an external search algorithm to match those tools to the current problem definition, it is possible for the same set of selection rules to find the new software that may offer a better solution to the problem/tool match. A description of how these tools are maintained external to the knowledge base by the MTICE subsystem is presented in Section 3.3.2.

3.2.2 Transportability

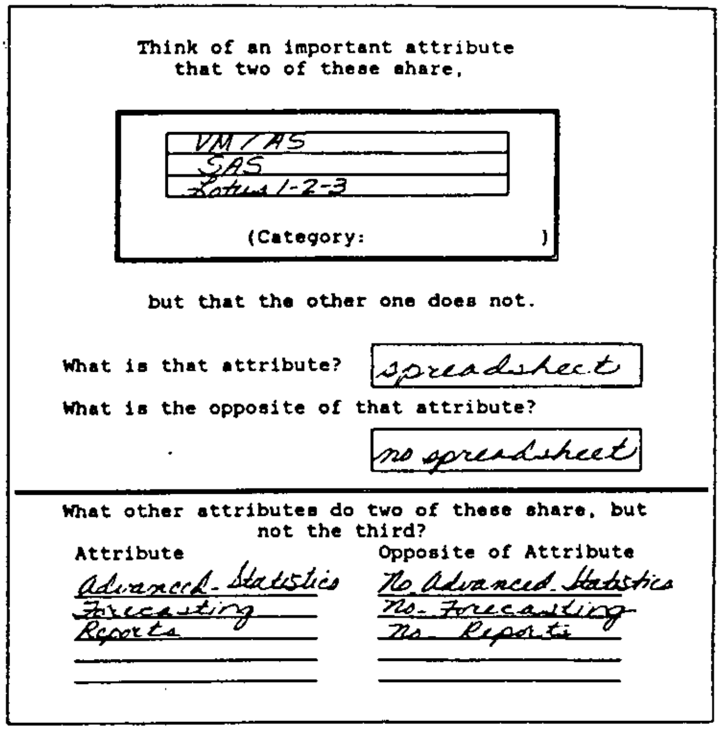
The second consideration, transportability, responds to the fact that knowledge is context dependent. This is an important issue because no two information centers are alike. In many organizations, more than one IC exists, each specializing to meet the requirements of a unique set of clientele. A consulting tool, therefore, must rest on a sufficiently flexible architecture to allow each Information



ENTER THE SOFTWARE PACKAGES WHICH YOU WANT
ICE
TO RECOMMEND FOR

Category:	Possible Uses:
Data Analysis	
VM/AS	
SAS	
Lotus 1-2-3	
Tiny Calc	
Planning Assistant	
Oxycalc	
APL2	

Figure 2.1. Elicitation of Elements



Think of an important attribute
that two of these share,

VM/AS
SAS
Lotus 1-2-3

(Category:)

but that the other one does not.

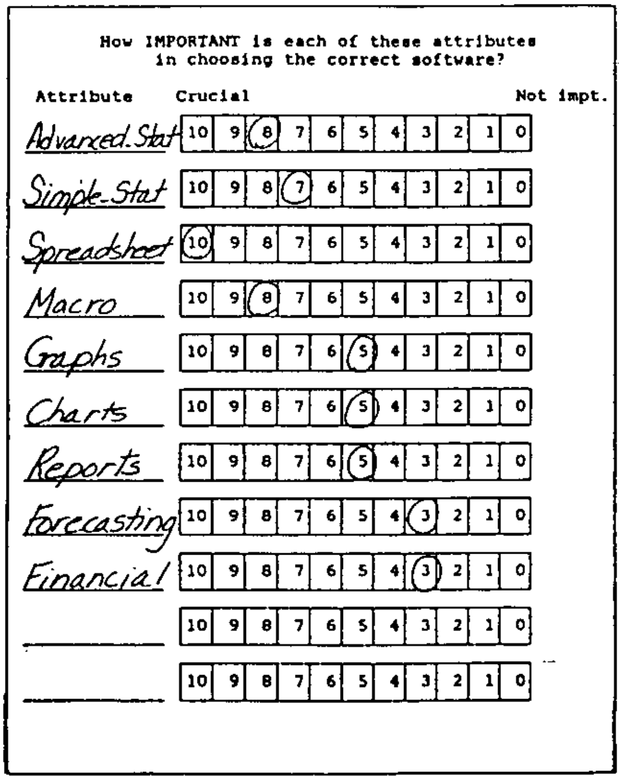
What is that attribute? spreadsheet

What is the opposite of that attribute? no spreadsheet

What other attributes do two of these share, but not the third?

Attribute	Opposite of Attribute
Advanced Statistics	No Advanced Statistics
Forecasting	No Forecasting
Reports	No Reports

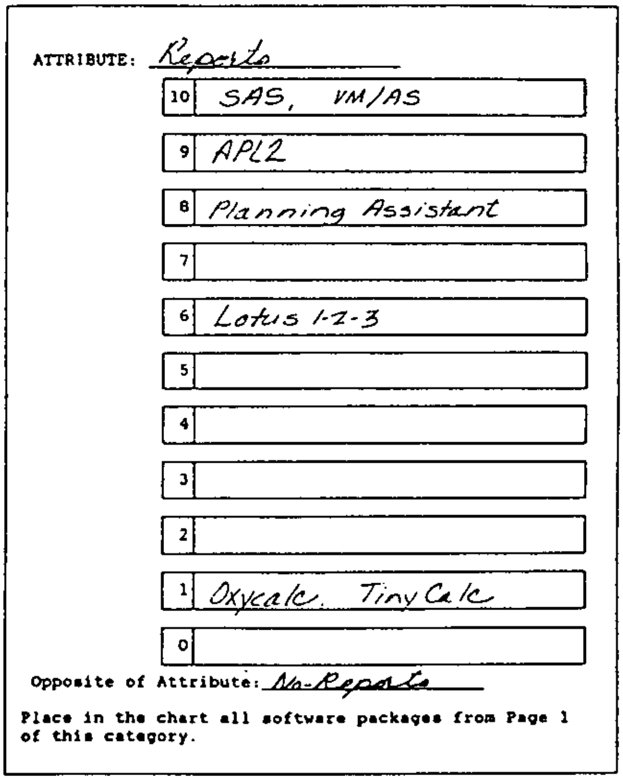
Figure 2.2. Elicitation of Attributes



How IMPORTANT is each of these attributes
in choosing the correct software?

Attribute	Crucial										Not impt.
Advanced Stat	10	9	[✓ 8]	7	6	5	4	3	2	1	0
Simple Stat	10	9	8	[✓ 7]	6	5	4	3	2	1	0
Spreadsheet	[✓ 10]	9	8	7	6	5	4	3	2	1	0
Macro	10	9	[✓ 8]	7	6	5	4	3	2	1	0
Graphs	10	9	8	7	6	[✓ 5]	4	3	2	1	0
Charts	10	9	8	7	6	[✓ 5]	4	3	2	1	0
Reports	10	9	8	7	6	[✓ 5]	4	3	2	1	0
Forecasting	10	9	8	7	6	5	4	[✓ 3]	2	1	0
Financial	10	9	8	7	6	5	4	[✓ 3]	2	1	0
	10	9	8	7	6	5	4	3	2	1	0
	10	9	8	7	6	5	4	3	2	1	0

Figure 2.3. Attribute Weighting



ATTRIBUTE: Reports

10	SAS, VM/AS
9	APL2
8	Planning Assistant
7	
6	Lotus 1-2-3
5	
4	
3	
2	
1	Oxycalc, TinyCalc
0	

Opposite of Attribute: No-Reports

Place in the chart all software packages from Page 1 of this category.

Figure 2.4. Attribute Values

Figure 2. Resource Attribute Charts

Center to individualize the system to meet site-specific needs.

The expert system must be adaptable enough to be implemented in different ICs with only minimal changes in the basic rule structures. The Maintenance Tool for ICE (MTICE) also makes transportability possible. Each IC can enter and define its own set of software tools and consultants.

3.3 Assembling: Knowledge Programming

Once the framework and knowledge representation have been selected, programming begins. Human know-how is transformed into a knowledge base that fuels the inference engine.

As mentioned previously, the ICE system is implemented on IBM 4381 in ESE/VM, an expert system development shell developed by IBM Corporation. ESE/VM is based on EMYCIN, and uses *rules* as the basis of its knowledge representation. ESE operates in the IBM mainframe environment. The shell provides users with convenient editors for representing the factual knowledge. The inference engine allows the use of both the backward and forward chaining inferencing techniques. ESE further provides exit and entrance points to the knowledge base so as to allow the access of external data/information/processes during the execution of a session.

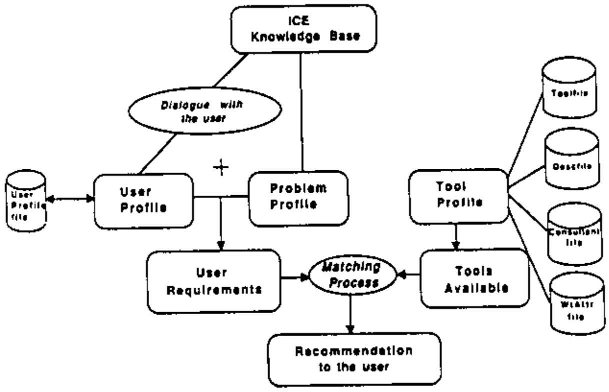


Figure 3. Conceptual Overview of ICE

Expert systems use several sources to populate their knowledge bases. Values for parameters are acquired from production rules, default values, interaction with users, or external storage, all appropriate under different circumstances. In the building of the Information Center Expert (ICE), each of these was used to some degree. The methods can be classified into internal and external methods.

The internal method of acquiring values for parameters in the ESE/VM environment are (1) the application of the rule base and (2) the use of default values. The external means of acquiring values for the knowledge base parameters of interest in this paper are (1) external storage, discussed in Section 3.3.1, and (2) interaction with users, the topic of Section 3.4.

3.3.1 External method: knowledge base/database issues

Expert systems, in very general terms, are composed of a knowledge base and an inference engine. The knowledge base is a collection of domain knowledge. Database is defined as "a collection of data representing facts. The amount of data is typically large, and these facts change over time" (Wiederhold 1984). The major difference between the knowledge base and the database approach is that a knowledge base contains information at a higher level of abstraction. Facts in a database are normally passive; they are either there or not there. A knowledge base, on the other hand, actively tries to fill in the missing information (Forsyth 1984).

Given the nature of a knowledge base and a database, we can say that the knowledge base tries to capture the expertise of the domain expert in the form of rules used by the expert to deal with certain situations. Knowledge relates to the general aspects of the data, and unlike data it should not change vary rapidly over time (Wiederhold 1984). The database, on the other hand, contains values for the parameters that are used to define the rules of a domain expert. Databases have, among other properties, the ability to efficiently insert, update, retrieve and delete data. Thus, a database could provide an efficient means of maintaining the values for the dynamic parameters of a knowledge base.

Zobaidie and Grimson (1987) describe a variety of ways in which an expert system might interact with a database system. In an **intelligent database**, the deductive component is embedded into the database management system. In an **enhanced expert system**, the inference engine of the expert system is provided with direct access to a generalized database. In **inter-system communication**, an expert system and a database management system co-exist with some form of communication between them. ICE is an example of an Enhanced Expert System. The ICE architecture divides the knowledge base into three primary groups: User Profile, Problem Profile, and Tool Profile.

1. **User Profile**: In the information center setting, users approach consultants with their own particular set of skills, computing environments, and biases. The user profile attempts to capture that knowledge.
2. **Problem Profile**: Each user approaches the information center with some perceived need which the resources in the information center will be able to satisfy.
3. **Tool Profile**: This is the information center's resource inventory. The tool profile further includes the ratings (weights) that the IC places on the various attributes used to define the resources (tools).

Two of the three groups, User Profile and Tool Profile, benefit from database concepts. Users consulting with the system have information about them stored in the User Profile database. The advantage is that it allows the user to make subsequent consultations with the system without having to re-enter the user profile information.

The Tool Profile of the software tools supported by the IC is maintained as a database; it consists of four files:

1. **Toolfile**: Contains the tools identification number and an array of its attribute ratings.
2. **Descfile** (Description file): Contains the tool identification number, tool name, tool description, and the employee number of the consultant who supports the tool.
3. **Consultant file**: Contains the consultant's name, employee number, and contact phone number.
4. **WtAttr** (Attribute Weighting file): Contains the name of each tool attribute, its definition, and its weighting.

These files are separate flat files in the current implementation of ICE, but the concepts used for access and maintenance are similar to those of a relational database, with each of the files representing one relationship.

One of the considerations in the design of ICE was that it had to be maintained in the dynamic environment of the information center, with a high turnover of resources. It is impractical for the maintainers of the system to continually update the rules every time a new tool is supported so that the knowledge base can reflect the current status of resources in the IC. Having the resource base separate from the rules is one method of dealing with this dynamic situation. Maintaining the resource base (the four files) then becomes an issue of database maintenance rather than knowledge base maintenance. The maintenance is carried out by MTICE.

3.3.2 Maintenance of the information center expert (MTICE)

The architecture of ICE is designed for relatively easy maintenance, because the stable knowledge is modeled internally in the rules of the knowledge base and the unstable, dynamic knowledge of the tool environment is maintained in external files that are simple to modify. The maintenance of these external files is controlled by a subsystem called MTICE -- Maintenance for ICE. This system is currently PC-based and allows for the creation of the four files necessary to describe the tool resources. These files are discussed in the Tool Profile in Section 3.3.1.

MTICE addresses the two design issues of maintainability and transportability. The maintenance of the four files constituting the resource base has been approached by viewing the four files as relationships in a relational database for which the primary keys are the tool identification number and the consultant serial number. The relations have been normalized to the third normal form (Date 1986) to facilitate insertion, deletion and update. A consistency and completeness check is made each time any of the files are changed.

The transportability issue is addressed by allowing each IC the ability to maintain its own resource base. Maintenance of the resource base has two parts: first, maintenance of the actual set of tools supported by the IC, and second, considering the IC bias regarding those tools. The set of tools supported by the IC are maintained in the Toolfile, Descfile, and Consult files (see Section 3.3.1). The bias of the IC is built into the WtAttr file and is reflected in the form of attribute weighting.

MTICE also has a report and browse facility to help the IC consultants with record keeping of the software tools supported. A future modification currently in progress is to expand the browse facility into an IC resource "window shopping" facility.

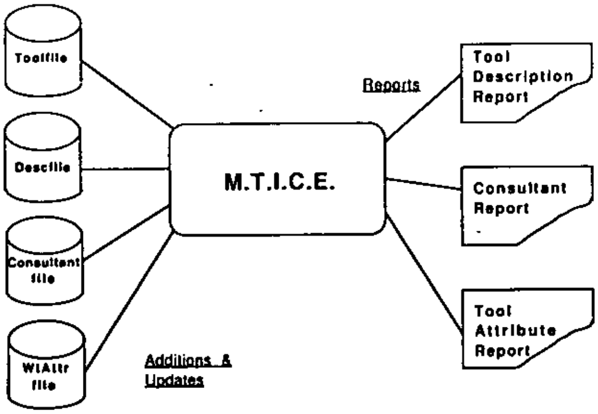


Figure 4. Maintenance of ICE (MTICE)

3.3.3 Selection algorithm

Once the user/problem profile has been collected through the intelligent dialogue subsystem, the ICE control subsystem calls the selection algorithm for a tool suggestion. The selection algorithm tries to match the user/problem profile with the tool profile and then make a suggestion of what tools are available for the user's task. This section provides details of the rationale behind the selection algorithm.

IC consultants define the functionalities of tools through a set of attributes. Each attribute has been assigned a weighting factor from 1 to 10 according to its importance in tool selection. Each tool's capabilities are evaluated by their relevant attributes in a scale from 0 to 10 (see Section 3.1). The attribute weightings and tools' capabilities are acquired by MTICE prior to the consultation. Users can only specify how important their needs are for certain functions. Under current implementation, users are given only a binary choice; they either need a functionality or they do not. To find tools that can cover all of the user's *must have* requirements, the capabilities must be greater than the value of *very capable.* The suggestions of which tools to use depend on the resulting comparison of the user's needs and tools' capabilities. Depending on how well we can cover a user's needs with the current tool repository, one of the following three situations will occur:

1. **First Choice:** As long as there are tools which are capable of covering all the user's must have needs, ICE will list up to nine such tools in the descending order of their coverage rates of the user's needs (also called confidence level).

2. **Second Choice:** When there is no tool qualified for the First Choice, ICE will list up to nine tools, by their confidence levels, that cover a portion of the user's needs. The confidence level in this situation has to be larger than some "cut off" point (called *low threshold*) set by the IC consultant.

3. **Last Resort:** When there is no tool in the First Choice and Second Choice categories, ICE will direct the user to appropriate consultants.

The selection algorithm takes the user's needs, weightings of attributes, and tool capabilities into consideration in making the selection. The following discussion shows how to calculate confidence level for tools in the First Choice situation:

For any attribute j,

```
IF user_need(j $\geq$ must_have  and
   tool(i,j) $\leq$ very_capable
THEN
   Rating_1(i) = 0

OTHERWISE
Rating_1(i) = $\sum_{j=1}^{n}$ user_need(j) x weight(j) x
                min(tool(i.j).user_need(j))
```

where:

Rating_1(i):	the rating of tool i for First Choice
user_need(j):	user's need for attribute j
weight(j):	weighting factor of attribute j
tool(i,j):	tool(i)'s capability in attribute j

The confidence level of the tool(i) in the First Choice situation is defined as the ratio of a tool's rating of an ideal tool:

Confidence_Level(i) = Rating_1(i)/Ideal_Rating

In the above formula, Ideal_rating is a tool which is capable of covering all the user's needs:

$$\text{Ideal\_Rating} = \sum_{j=1}^{n} \text{user\_need}(j) \times \text{weight}(j) \times \text{ideal\_tool}(j)$$

where:

Ideal_Rating:	The rating of an ideal tool which satisfies all the user's needs.
ideal_tool(j):	The capability of an ideal tool in attribute j, which is equal to the user_need(j).

3.4 Refining: User Interface/Dialogue Control

The fourth task of building a knowledge system was listed as Knowledge Refinement, resulting in revised concepts and rules. In the case of ICE, it also resulted in a refined and improved user interface.

Most expert systems never get beyond the research prototype stage (Waterman 1983). One reason for this is the lack of clear and concise interface with the user. Until recently, user interface was considered to be of secondary importance to the design of the knowledge base and inferencing mechanism. Berry and Broadbent (1987) explain this fact by pointing out that laboratory expert systems tend to be used by people who appreciate them and are tolerant of their idiosyncrasies. With the increasing acceptance of expert systems, the issues of user interface and dialogue control become more important.

The dialogue between ICE and users consists of three parts. As described in Section 2.2, the users are first queried about their skills and their work environments. The system queries the user with a series of standard questions which are asked of all users during their first interaction with the system. In instances where the user had previously consulted with the system, the user's profile is displayed. The user profile is built through static dialogue, i.e., the questions remain the same for all circumstances. Figure 5.2 presents a sample screen of the user profiling questions.

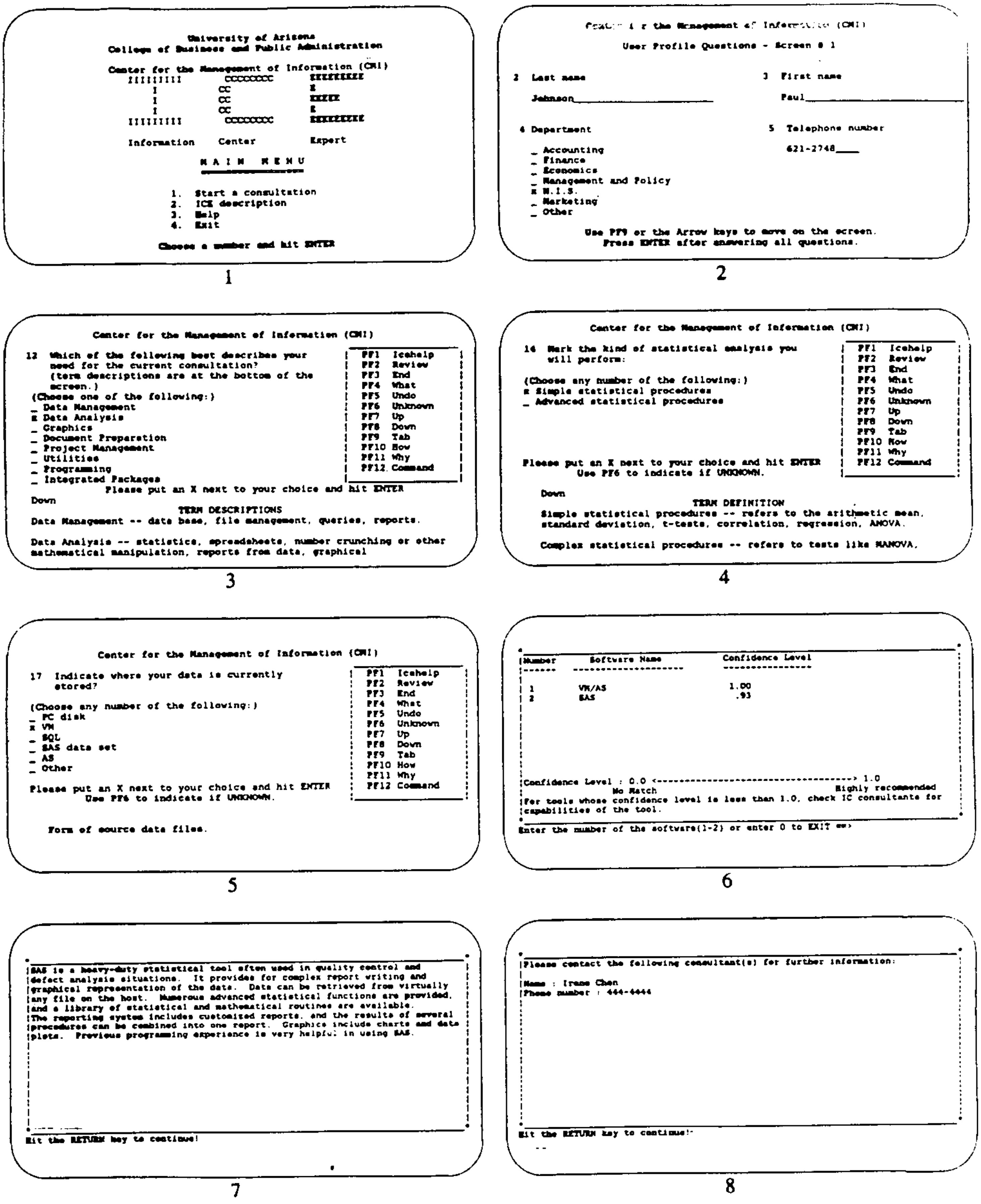
A second set of questions is used to determine the needs of the user. Once the general category of need is defined, further refinement is required for understanding the details of the user requirements. The general categories of user needs (Figure 5.3) were developed by extensive interviewing of IC consultants. The eight categories (see Section 3.1) cover all the software currently supported by the IC's in IBM/Endicott, IBM/Tucson, and CMI at the University of Arizona. Determining the user's need is accomplished through a backward chaining inferencing process. The querying process is strictly controlled to avoid both redundant and meaningless questions. Details about controlling the dialogue and the techniques used are discussed under query ordering (Section 3.4.1).

A third set of questions, referred to as the "common set," follow the need determination. These questions are not *need specific*, but must be asked for almost every consultation.

Dialogue control refers primarily to the way in which the system and the user interact. Two important aspects are Query Ordering and Screen Layout. It should be mentioned here that the first prototype of ICE did not use the techniques described. The result, as can be expected, was user dissatisfaction with the interface and the dialogue.

3.4.1 Query ordering

Two techniques of ESE are used to control the ICE questioning. The first is Focus Control Blocks (FCBs), in which each FCB represents a subtask of the ICE application. The concept of FCBs is similar to the "hypothesis" of NEOMYCIN (Clancey 1983). FCBs in ESE/VM allows for the organization of ICE



```
                  University of Arizona
        College of Business and Public Administration

        Center for the Management of Information (CMI)
           IIIIIIIII       CCCCCCCC       EEEEEEEEE
               I          CC             E
               I          CC             EEEEE
               I          CC             E
           IIIIIIIII       CCCCCCCC       EEEEEEEEE

           Information    Center         Expert

                       M A I N   M E N U

                  1.  Start a consultation
                  2.  ICE description
                  3.  Help
                  4.  Exit

               Choose a number and hit ENTER
```

1

```
        Center for the Management of Information (CMI)

             User Profile Questions - Screen # 1

2  Last name                          3  First name

   Johnson____________________           Paul______________________

 4 Department                          5  Telephone number

   _ Accounting                           621-2748____
   _ Finance
   _ Economics
   _ Management and Policy
   x M.I.S.
   _ Marketing
   _ Other

          Use PF9 or the Arrow keys to move on the screen.
            Press ENTER after answering all questions.
```

2

```
          Center for the Management of Information (CMI)

12  Which of the following best describes your        |  PF1   Icehelp  |
    need for the current consultation?                |  PF2   Review   |
    (term descriptions are at the bottom of the       |  PF3   End      |
    screen.)                                          |  PF4   What     |
 (Choose one of the following:)                       |  PF5   Undo     |
 _ Data Management                                    |  PF6   Unknown  |
 x Data Analysis                                      |  PF7   Up       |
 _ Graphics                                           |  PF8   Down     |
 _ Document Preparation                               |  PF9   Tab      |
 _ Project Management                                 |  PF10  How      |
 _ Utilities                                          |  PF11  Why      |
 _ Programming                                        |  PF12  Command  |
 _ Integrated Packages                                |_________________|
            Please put an X next to your choice and hit ENTER
 Down
                       TERM DESCRIPTIONS
 Data Management -- data base, file management, queries, reports.

 Data Analysis -- statistics, spreadsheets, number crunching or other
 mathematical manipulation, reports from data, graphical
```

3

```
          Center for the Management of Information (CMI)

14  Mark the kind of statistical analysis you         |  PF1   Icehelp  |
    will perform:                                     |  PF2   Review   |
                                                      |  PF3   End      |
(Choose any number of the following:)                 |  PF4   What     |
x Simple statistical procedures                       |  PF5   Undo     |
_ Advanced statistical procedures                     |  PF6   Unknown  |
                                                      |  PF7   Up       |
                                                      |  PF8   Down     |
                                                      |  PF9   Tab      |
                                                      |  PF10  How      |
                                                      |  PF11  Why      |
Please put an X next to your choice and hit ENTER     |  PF12  Command  |
        Use PF6 to indicate if UNKNOWN.               |_________________|

   Down
                          TERM DEFINITION
   Simple statistical procedures -- refers to the arithmetic mean,
   standard deviation, t-tests, correlation, regression, ANOVA.

   Complex statistical procedures -- refers to tests like MANOVA.
```

4

```
          Center for the Management of Information (CMI)

17  Indicate where your data is currently             |  PF1   Icehelp  |
    stored?                                           |  PF2   Review   |
                                                      |  PF3   End      |
(Choose any number of the following:)                 |  PF4   What     |
_ PC disk                                             |  PF5   Undo     |
x VM                                                  |  PF6   Unknown  |
_ SQL                                                 |  PF7   Up       |
_ SAS data set                                        |  PF8   Down     |
_ AS                                                  |  PF9   Tab      |
_ Other                                               |  PF10  How      |
                                                      |  PF11  Why      |
Please put an X next to your choice and hit ENTER     |  PF12  Command  |
        Use PF6 to indicate if UNKNOWN.               |_________________|


   Form of source data files.
```

5

```
|Number      Software Name             Confidence Level
|------    --------------------       ----------------
|
| 1         VM/AS                       1.00
| 2         SAS                          .93
|
|
|
|
|Confidence Level : 0.0 <-------------------------------------> 1.0
|                No Match                                  Highly recommended
|For tools whose confidence level is less than 1.0, check IC consultants for
|capabilities of the tool.
Enter the number of the software(1-2) or enter 0 to EXIT ==>
```

6

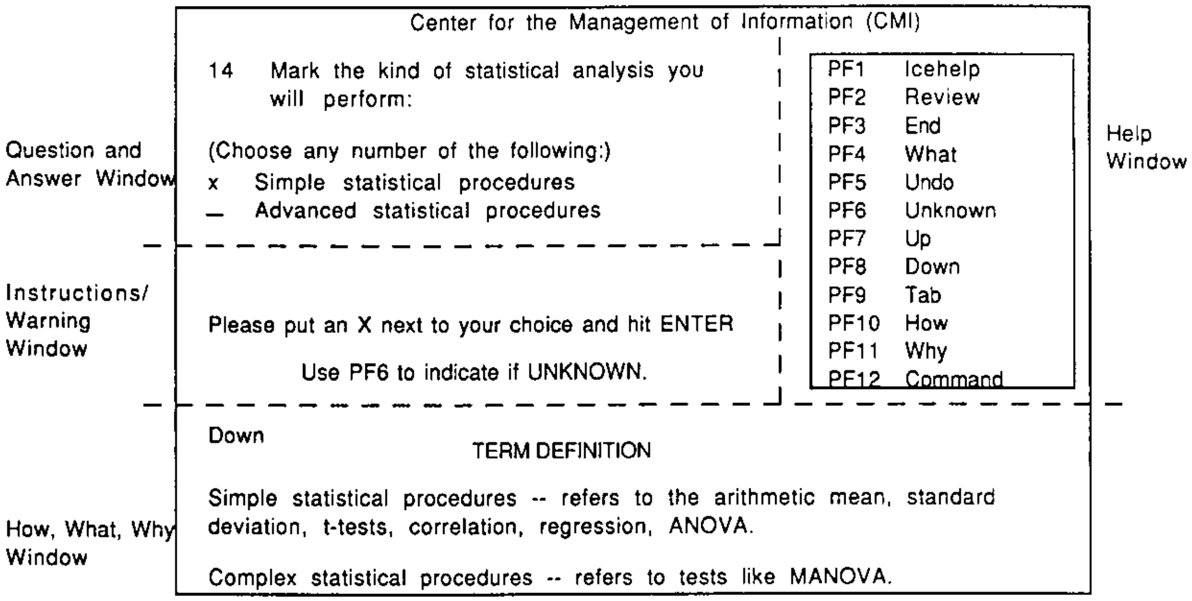
```
|SAS is a heavy-duty statistical tool often used in quality control and
|defect analysis situations.  It provides for complex report writing and
|graphical representation of the data.  Data can be retrieved from virtually
|any file on the host.  Numerous advanced statistical functions are provided,
|and a library of statistical and mathematical routines are available.
|The reporting system includes customized reports, and the results of several
|procedures can be combined into one report.  Graphics include charts and data
|plots.  Previous programming experience is very helpful in using SAS.
|
|
Hit the RETURN key to continue!
```

7

```
|Please contact the following consultant(s) for further information:
|
|Name : Irene Chen
|Phone number : 444-4444
|
|
Hit the RETURN key to continue!
   --
```

8

Figure 5. Sample Screens of Dialogue



Center for the Management of Information (CMI)

14 Mark the kind of statistical analysis you will perform:

(Choose any number of the following:)
x Simple statistical procedures
_ Advanced statistical procedures

Please put an X next to your choice and hit ENTER

Use PF6 to indicate if UNKNOWN.

Down

TERM DEFINITION

Simple statistical procedures -- refers to the arithmetic mean, standard deviation, t-tests, correlation, regression, ANOVA.

Complex statistical procedures -- refers to tests like MANOVA.

Figure 6. Screen Layout

subtasks into a hierarchy. One of the properties of an FCB is that parameters/rules above a certain FCB are visible to the lower level FCBs, but the FCBs higher up in the hierarchy cannot access the parameters/rules of the lower level FCBs (Hirsch et al. 1986). Therefore, in the ICE context, the user requirement is determined early in the questioning, and the consultation is directed to the relevant lower level FCB. This approach reduces the number of parameters/rules that need to be resolved, and therefore causes more meaningful questions to be asked. The use of FCBs greatly enhances the ability to understand the dialogue.

The second method of controlling query ordering is to divide the parameters of the knowledge base into two groups (Vinze 1987): the Dialogue Control Parameters (DCPs) and the Attribute Setting Parameters (ASPs). This division helps control the dialogue by utilizing two types of rules: Inference Rules and Monitor Rules. DCPs are parameters that the knowledge engineer specifies as the questions for which the domain expert wants a response. They can be in the form of multiple choice, boolean, string, or numeric.

The backward chaining inference engine is implemented only on the DCPs. Each FCB has associated with it its own group of DCPs, and ICE tries to determine the entire group when a particular FCB is initiated. Associated with each of the DCP options is a monitor rule. Monitor rules are those for which the action part of the rule is executed if the premise of the rule becomes true. The inference engine ignores these rules during their processing (IBM 1986). The monitor rules related to DCPs are of the form "if *condition* then don't consider *certain* DCPs." This type of rule further helps reduce the number of rules in an FCB that the inference engine needs to consider. It was earlier pointed out that the use of FCBs helps the inference engine by requiring it to look only at a subset of all the rules in the rule base. ASPs use monitor rules to set values of particular attributes in both the user profile and the problem profile. The ASPs do not have any say in the dialogue control process. These parameters acquire value through the use of monitor rules of the form "if *condition* then ASP1 = 1.0." It is the values of the ASPs that are used in the tool selection algorithm discussed in Section 3.3.3.

3.4.2 Screen layout

The screen format plays an important role in user acceptance of a system. Several guidelines exist for constructing an effective screen layout for interactive systems. In ICE, the authors use the guidelines provided by Cole, Lansdale and Christie (1985), which specify four key aspects for screen design: (1) content of display, (2) format, (3) coding, and (4) use of color.

Each screen in ICE (with the exception of the user profiling and the final recommendation screens) is divided into four windows (see Figure 6). The first window consists of the question and answer, the second is the PF key (predefined function key) or the help window, the third is the instruction/warning window, and at the bottom of the screen is the "How, What and Why" window. The standardized format of screens helps users become more familiar with the system.

4. VALIDATION OF ICE

The ICE system is developed to meet the design criteria of transportability and maintainability. As previously discussed, the system is not meant to replace the existing consultation process in the IC, but rather serves as a technological support. In its current implementation, ICE deals with only the software recommendation aspect of the consultation process. The current consulting methodology for software recommendation is shown in Figure 7

Figure 7. Current Consultation Process

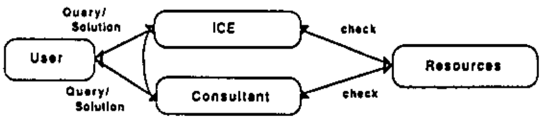


Figure 8. Change in the Consulting Process

With the introduction of the ICE system, the consultation process is altered by the addition of another possible channel to match users' needs with the organization's software resources. The changes caused by the introduction of the ICE system are shown in Figure 8.

It is assumed that the ICE system is introduced in the organization to reduce the IC consultant's workload without decreasing the utility of the IC to the end user community it supports. Given this assumption, ICE should be tested both for the validity of its recommendations and the comparative merits of each of the channels of consultation that may be used to map the user's needs with the resources of the organization.

Confirmation of the validity of the ICE model was sought by utilizing the "blind" validation procedure based on the work of the mathematician A. M. Turing (1950). The procedure consisted of presenting scenarios (in case form) to consultants. The cases reflected possible consultation sessions that the consultants are likely to encounter. The cases were also solved using the ICE system. The two sets of solutions -- those of consultants and of ICE -- were presented to experts for judging the appropriateness of the solution to the case. Results of the validation study are currently being analyzed.

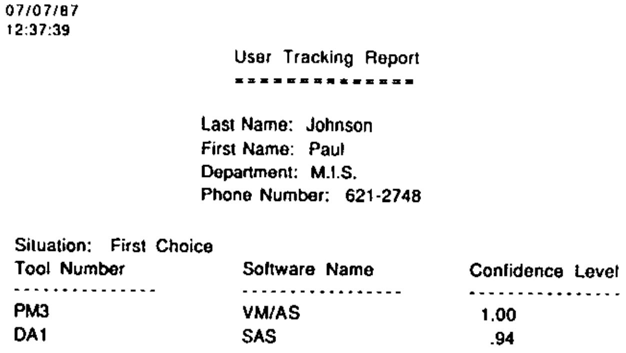
An experiment to test the comparative merits of the different consultation channels for obtaining appropriate software tools is also being conducted. The approach uses a hypothetical construct called "Consultation Effectiveness," which includes a measure of the "user satisfaction" with the process, as well as measures of time and cost of conducting a consultation session. The traditional mode of consultation will also be compared with the changes caused in the consulting environment by the ICE system implementation. Results of the experiment will be available in a few months.

5. TRACKING

The ICE system aids the organization with collecting data on users of the information center, and on the ability of the IC to meet the computing requirements of the users. A tracking subsystem captures the basic attributes of the user of the system as well as the recommendations made by ICE to the user.

This information makes possible a profile of end users in the organization. Most successful ICs have been able to identify key users who develop systems that provide large company payoffs. They have concentrated on helping those users choose application approaches and have provided necessary training to them. ICE can provide information that would help the ICs to differentiate their user population, providing some services to all and specialized services to certain targeted populations.

The information center's ability to manage its software resources is enhanced by the tracking report. The system keeps track of the situation under which a recommendation is made to the user: (1) all critical requirements are met, (2) majority but not all needs of the user are met, (3) none of the available software meets the requirements (see Section 3.3.3).



```
07/07/87
12:37:39
                            User Tracking Report
                            ====================

                         Last Name:  Johnson
                         First Name:  Paul
                         Department:  M.I.S.
                         Phone Number:  621-2748

Situation:  First Choice
Tool Number              Software Name             Confidence Level
--------------           ----------------          ---------------
PM3                      VM/AS                      1.00
DA1                      SAS                         .94
```

Figure 9. Sample Tracking Report

The fact that IC records of the situation exist allows the IC to evaluate its software inventory and make effective updates to it. The tracking report can be checked by IC consultants and managers on a periodic basis to determine if the software tools they support meet the needs of the end-user population. If a large number of consultations conclude with situation 1 (the tool recommended met all the critical needs of the user), then the IC is supporting appropriate software tools for its user population. If, however, a large number of consultations with ICE end with situation 3 (no software can be recommended), then the IC management needs to be concerned and must re-evaluate the software resource inventory. Frequency of consultations ending with situation 2 should send warning signals to the IC management indicating that critical user requirements are not being met, and temporary solutions are being used to meet the immediate needs of the users.

The tracking program provides two services to the users of ICE. First, the user is provided with a hard copy of the recommendations made by the system. The user may use the suggestions made by the system or, alternatively, may acquire a second opinion from the IC consultant. Second, since the program keeps track of the users and recommendations made, the user can be kept up-to-date on the software used by forwarding any notices of updates concerning the software recommended.

6. FUTURE RESEARCH ISSUES

Since the initial implementation of ICE, several research possibilities have been identified. Much can be done to enhance its capabilities and its architecture can be applied to other resource management issues.

6.1 Maintenance and Flexibility of the Knowledge Base

Future efforts will (1) increase the ease with which ICE is maintained as new and different software tools are added to the Information Center and (2) enhance its versatility in offering different categories of software depending upon the nature of the end users' demands (Heltne et al. 1987). MTICE should be rewritten into a rule-based "intelligent MTICE" that assigns values to tool attributes based on dialogue with the IC consultant. The dialogue would allow for comparison of the new tool with software previously defined and adjustments to their ratings as well as assignment of values to the attributes of the new tools. This would involve automating the RAC (Resource Attribute Charts) methodology, the outputs of which would then feed into the MTICE program.

6.2 Training

ICE, as it exists today, supports only the *consulting* function of IC personnel. It is currently limited to two kinds of resource recommendations: (1) software tools and/or (2) information about consultants who have expertise in their problem areas. We foresee a much more varied and sophisticated role for such an expert system. Just as the IC staff must provide multiple kinds of services, other dimensions can be added to ICE to increase its support of the IC environment. In addition to recommending a set of tools, ICE could maintain a schedule of classes or workshops for those tools and offer online information or enrollment to the user. A fully developed system will surely include, as part of the interface with users, a demonstration of capabilities of the tools recommended, or even online training for selected software.

6.3 ICE as a Decision Support Tool for Software Purchasing

ICE can be used by the IC consultant to help make software purchasing decisions. When new tools or tool descriptions arrive for evaluation in the Information Center, the consultant can enter a consultation, fitting the attributes of a user to those of the new tool. If software already in the system meets those needs, it is possible that both tools do not need to be supported, and a comparative evaluation can be made. Consultants also should study stored consultations for which no software tool could be recommended to solve a user's problem. If the need can be verified as one that must be met, the consultant can search for the necessary tools to add to the the Information Center. Research documenting this use needs to be undertaken.

6.4 Reusable Code

If the knowledge based approach is used in the management of reusable code, the problems of finding, understanding, modifying, and composing components must be addressed (Biggerstaff and Richter 1987). The knowledge based system must capture not only the syntax of each module, but also the semantics and the coupling between modules. The ICE architecture might be applied to a heterogeneous library of codes and high level requirements of a system.

An executive information officer from a major computer company who saw the ICE system has proposed adapting the architecture to aid in the process of preparing proposals for customer orders. The immediate applications of the knowledge based approach for improving resource identification and selection are enormous. The generality and adequacy of the knowledge representation and structure, and the proposed knowledge acquisition process, must be examined in domains beyond the IC. We have demonstrated the viability of a knowledge based system as a mechanism for matching user demand with resource supply. The specific domain of the users and resources was the Information Center, which deals with end user computing resources. It is probable that the architecture of the Information Center Expert system will facilitate the management of resources in other problem environments as well.

7. ACKNOWLEDGEMENTS

The authors would like to thank IBM Corporation for their support of this project. We especially acknowledge the excellent contributions of Fred Wallace and Al Smith, IBM/Tucson; Yiu Leung, IBM/Endicott; and Kendall Cliff and Irene Chen CMI/University of Arizona.

REFERENCES

Bender, P. S. *Resource Management: An Alternative View of the Management Process.* John Wiley and Sons, New York, 1983.

Benjamin, R. I. "Information Technology in the 1990's: A Long Range Planning Scenario." *MIS Quarterly*, Vol. 6, No. 2, June 1982, pp. 11-31.

Berry, D. C., and Broadbent, D. E. "Expert Systems and the Man-Machine Interface. Part Two: The User Interface." *Expert Systems*, Vol. 4, No. 1, February 1987.

Biggerstaff, T., and Richter, C. "Reusability Framework, Assessment, and Directions." *IEEE Software*, March 1987, pp. 41-49.

Bohl, D. L. (Ed.) *The AMA Report on Information Centers.* AMA Membership Publications Division, American Management Association, New York, 1986.

Boose, J. H. "ETS: A System for the Transfer of Human Expertise." In J. S. Kowalik (ed.), *Knowledge Based Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ, 1986a.

Boose, J. H. *Expertise Transfer for Expert System Design.* Elsevier Science Publishers, New York, 1986b.

Brancheau, J. C.; Vogel, D. R.; and Wetherbe, J. C. "An Investigation of the Information Center from the User's Perspective." *Data Base*, Vol. 17, No. 1, Fall 1985, pp. 4-17.

Clancey, W. J. "The Advantages of Abstract Control Knowledge in Expert System Design." *Proceedings AAAI-1983*, 1983, pp. 74-78.

Clancey, W. J. "Classification Problem Solving." *Proceedings of the National Conference on Artificial Intelligence*, Austin, 1984.

Cole, I.; Lansdale, M.; and Christie, B. "Dialogue Design Guidelines." In B. Christie (ed.), *Human Factors of Information Technology in the Office*, John Wiley and Sons, New York, 1985.

Date, C. J. *An Introduction to Database Systems*. Volume 1, Addison-Wesley, Reading, MA, 1986.

Forsyth, R. "The Architecture of Expert Systems." In. R. Forsyth (ed.), *Expert Systems Principles and Case Studies*, Chapman and Hall, New York, 1984.

Harmon, P., and King, D. *Expert Systems: Artificial Intelligence in Business*. Wiley, New York, 1985.

Hayes-Roth, F. "The Knowledge-Based Expert System: A Tutorial." *IEEE Computer*, September 1984, pp. 11-28.

Heltne, M. M. "Resource Attribute Charts: Structured Knowledge Acquisition for Resource Identification." CMI Working Paper Series, MIS Department, University of Arizona, Tucson, 1987.

Heltne, M. M.; Vinze, A. S.; Konsynski, B.; and Nunamaker, J. F., Jr. "A Consultation System for Information Center Resource Allocation." In Elias M. Awad (ed.), *Proceedings of the 1987 SIGBDP-SIGCPR Conference*, 1987.

Hirsch, P.; Katke, W.; Meier, M.; Snyder, S.; and Stillman, R. "Interfaces for Knowledge-Base Builders' Control Knowledge and Application-Specific Procedures." *IBM Journal of Research and Development*, Vol. 30, No. 1, January 1986.

IBM. *Expert System Development Environment/VM, Reference Manual*. Program Number 5798-RWQ, 1986.

Kelly, G. A. *Psychology of Personal Constructs*. Norton, New York, 1955.

Krcmar, H. A. O. "Enterprise-Wide Information Management: Expert Systems for Information Management." IBM Los Angeles Scientific Center, Report Number G320-2767, July, 1985.

Michalski, R. S.; Carbonell, J. G.; and Mitchell, T. M. (Eds.). *Machine Learning*. Tioga, 1983.

Porter, M. E. *Competitive Advantage*. The Free Press, New York, 1985.

Prieto-Diaz, R., and Freeman P. "Classifying Software for Reusability." *IEEE Software*, January 1987, pp. 6-16.

Rowe, A. J.; Mason, R. O.; and Dickel, K. E. *Strategic Management and Business Policy*, Addison-Wesley, Reading, MA, 1985.

Royksund, C. "Critical Success Factors in Information Centers." CMI Working Paper Series, MIS Department, University of Arizona, Tucson, 1987.

Turing, A. M. "Computing Machinery and Intelligence." *Mind*, October 1950; reprinted in *Creative Computing*, Vol. 6, No. 1, January 1980, pp. 44-53.

Vinze, A. S. "A Dialogue Control Technique for Rule Based Systems." CMI Working Paper Series, MIS Department, University of Arizona, Tucson, 1987.

Wallace, F. Information Center, IBM Corporation, Tucson, AZ. Interview with the authors, June, 1986.

Waterman, D. A. *A Guide to Expert Systems*, Addison-Wesley, Reading, MA, 1983.

Wiederhold, G. "Knowledge and Database Management." *IEEE Computer*, January 1984, pp. 63-73.

Winograd, T., and Flores, F. *Understanding Computers and Cognition*. Ablex Publishing Co., Norwood, NJ, 1986.

Zobaidie, A. A., and Grimson, J. B. "Expert Systems and Database Systems: How Can They Serve Each Other?" *Expert Systems*, Vol. 4, No. 1, February 1987.