Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 1989 Proceedings

International Conference on Information Systems (ICIS)

1989

INTEGRATION OF INFORMATION SYSTEMS TECHNOLOGIES TO SUPPORT CONSULTATION IN AN INFORMATION CENTER

Sudha Ram
University of Arizona

Stephen Hayne
University of Arizona

David Carlson
University of Arizona

Jay F. Nunamaker Jr.
University of Arizona

INTEGRATION OF INFORMATION SYSTEMS TECHNOLOGIES TO SUPPORT CONSULTATION IN AN INFORMATION CENTER

Sudha Ram
Stephen Hayne
David Carlson
Jay F. Nunamaker, Jr.
The Department of Management Information Systems
College of Business and Public Administration
University of Arizona

ABSTRACT

This paper presents an approach for integrating different types of information systems technologies to support the functions of an Information Center (IC). A knowledge based system, Information Center Expert/Help Service (ICE/H), has been developed to provide support for the help services of an IC. A general process model to represent the consultation process in an IC is described. Based on this model, an architecture to support the consultation process has been developed. The architecture depicts the use of a knowledge management system, a data management system and a communication (E-mail) system to emulate the consultation process. The ICE/H system has been implemented using this architecture to support an IC with 5000 users.

1. INTRODUCTION

Information Centers (ICs) are organized to provide help to end users of computing resources. Typically ICs provide guidance in software and hardware selection and distribution. They also assist users in solving problems associated with the use of software and hardware. The rapid proliferation of personal computers and mainframe-based tools for end users has, however, created an unmanageable burden for many ICs. It has been demonstrated that providing knowledge based support for ICs can go a long way in easing this burden on IC consultants (Heltne et al. 1988). Past research has focused on providing knowledge based support for the software selection function of an IC (Vinze 1988). Our objective is to extend this research by considering the help services of an IC. Users approach the IC with specific problems related to the use of hardware and software; the IC consultants provide them with trouble shooting expertise. This is regarded as the most important function of an IC (Brancheau, Vogel and Wetherbe 1985). We have attempted to integrate different types of information systems technologies to support this function. Specifically, this research demonstrates the use of knowledge and database management integrated with communications technology to support the consultation process in an IC.

This paper describes Information Center Expert for Help (ICE/H), a knowledge based system that provides support for the help services of an IC. In section 2, we describe the environment of an IC and justify the need for ICE/H. In section 3, a process model for consultation is described. Based on this model, a generic architecture for a knowledge based system to support help services is developed in section 4. The components of ICE/H and the prototyping process used to develop the system are described in section 5. Issues related to validation of the system are addressed in section 6. Finally, section 7 presents conclusions and directions for continued research.

2. JUSTIFICATION FOR ICE/H

The Information Center Expert (ICE) project is an ongoing effort in the MIS department at the University of Arizona to provide knowledge-based support for ICs. Over the past two years, an Information Center Expert for Software Consultation (ICE/C) has been designed and prototyped (Heltne et al. 1988; Nunamaker et al. 1988). Our current research focuses on a general process model, architecture and implementation of a system to provide knowledge-based support for IC Help services (ICE/H).

ICE/H is unique among expert systems in several ways. First, it effectively demonstrates the integrated use of a knowledge management system, a database management system and an electronic mail facility to provide several advantages. The general architecture for consultation separates the static knowledge -- rules for guiding the problem diagnosis -- from the dynamic knowledge in the form of text for the solutions presented to the user. This approach maximizes the flexibility and maintainability of the system. Second, the system can be used simultaneously by a large number of users. Each user receives an answer suitable to his/her level of understanding. Third, a feedback mechanism in the architecture guarantees that the user receives an answer to his/her problem. The system provides a facility to forward users' comments to consultants via electronic mail whenever required. This feature illustrates the use of electronic mail to effectively capture and replicate the communication between a consultant and user. This capability also supports a semi-automatic evolution of the system in response to changes in user needs and the computing environment of the organization. More specifically, in many cases, it allows the system to be modified without requiring the services of a knowledge engineer. Fourth, user consultations are captured automatically by the system along with their comments. This provides the IC with data on the performance of ICE/H and information which can be used to plan future enhancements to the system. Finally, the system captures the expertise of multiple experts. Most other expert systems have been built using one or two experts (Buchanan et al. 1983). In addition, knowledge acquisition for ICE/H was achieved by making use of group support tools (Ram et al. 1989).

As stated earlier, the help service (help desk/technical support) provided by ICs is perceived to be the most important function of an IC. IC consultants are expected to assist end-users in solving problems associated with the use of hardware and software. Typically, IC consultants receive a high volume of calls every day, most of which deal with routine and repetitive problems. The need to be "on call" at all times prevents these consultants from effectively managing their time. Most ICs experience a high turnover rate because, as their expertise increases, the consultants are sought out and hired by functional departments in an organization. Thus the IC is frequently faced with the problem of hiring and training new personnel. In many cases, it is very difficult to attract senior experts to work in an IC.

A system such as ICE/H can enhance the effectiveness of an IC in several ways. Routine questions are addressed by the expert system with minimal participation required by the IC staff. Unlike a consultant, the system is available to the user at all times. End-users get immediate solutions to their problems rather than waiting for a consultant to return their call. End-users are furnished with a single source for dissemination of IC help (Rockart and Flannery 1983). In addition, consistent advice is supplied by the system. The problems caused by turnover in an IC are alleviated by the system. It essentially captures and preserves the expertise of senior level experts, and thus serves to train new consultants. Moreover, the system leaves the IC consultants more time to spend on non-routine tasks and to provide training and other services to end-users.

3. CONSULTATION PROCESS MODEL

In order to develop a knowledge based system to support the help services of an IC, it is necessary to understand the consultation process. A typical consultation between an end-user and a consultant is described below and shown in Figure 1.

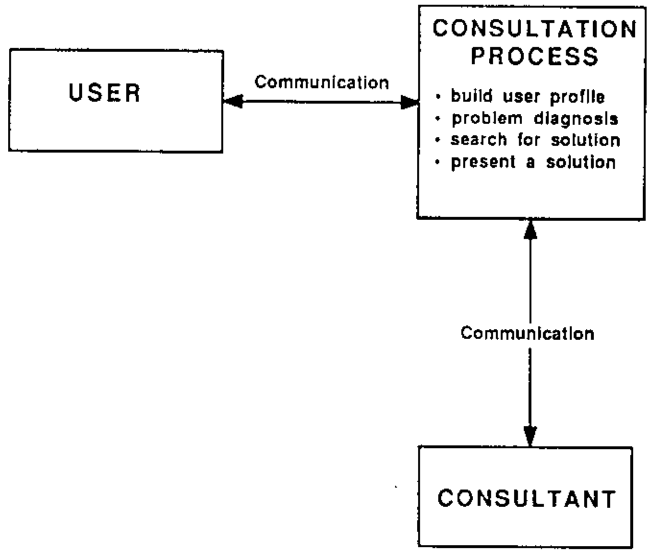


Figure 1. General Consultation Process

1. A user telephones the IC help number.

2. A general consultant answers and conducts a dialogue with the user in order to diagnose the problem and provide an answer.

3. During the course of the conversation, the consultant prompts the user with a few questions to discover:

   (a) the general problem area and
   (b) the user's level of expertise within that area.

4. If the problem is within the scope of the consultant's area of expertise, and is fairly simple, the consultant answers it right away.

5. For more complicated problems, the consultant may refer to software manuals to provide alternative solutions to the user. In some cases, the consultant may even discuss the alternative solutions with one or more other consultants to suggest the most suitable alternative(s) to the user.

6. If the problem is beyond the ability of the general consultant, the user is referred to a consultant who normally deals with problems in that area.

7. In some cases, no solution may be available for a user's problem. This may be because the user tried several alternatives and none of them worked. In such cases, the consultant learns by feedback from the user and attempts to formulate alternative solutions for the problem.

8. In all cases, the user is given the answer in terms that (s)he understands. For novice users, the consultation

process is more involved since the consultant asks more detailed questions to diagnose the problem. With expert users it is easier to identify the problem with fewer questions and a succinct answer usually suffices.

9. In all cases, the consultant may give the user an answer over the telephone, visit his office for a *hands-on* demonstration or notify the user through electronic mail when a solution to the problem has been found.

A knowledge based system that emulates the consultants and supports the process outlined above must address several aspects of the process. First, it must support different styles of dialogue with a user based on its ability to identify the expertise level of the user. Second, the user's profile must be stored in order to be retrieved for future consultations. Third, the heuristics used by a consultant for matching problems with solutions must be replicated in the system. Solutions which are readily available from the consultant's mind or from manuals need to be adequately represented. The ability to identify situations when a problem is beyond the scope of the system should be available. Lastly, the communication between the user and the consultant during different stages of the consultation process needs to be effectively duplicated. This communication should assist the system in learning when a solution is found to be ineffective or incorrect.

4. INTEGRATED ARCHITECTURE FOR CONSULTATION

A general architecture must define the system components needed to support the process model outlined above. This architecture utilizes a knowledge-based system to control the consultation between the system and the end-user. A database management system is integrated to improve the maintainability of the system and to record factual data. An electronic mail system is used to provide the communication links within the process model. Figure 2 shows the system consultation process model used to define this architecture. Section 5 describes a specific implementation of this architecture and discusses how a prototype development process was used to enhance this architecture in the course of our research.

Expert systems are generally composed of a knowledge base and an inference engine. The knowledge base may be further subdivided into the decision heuristics used by the expert (frequently rules) and the factual data. Knowledge provides the controlling and general information: typically complex and relatively small in volume. Data represents the manipulated and factual information: typically regular but voluminous (Wiederhold and Milton 1987). Knowledge relates to the general aspects of the data and, unlike data, should not vary rapidly over short periods of time. Knowledge bases excel at managing unstructured information. Databases, on the other hand, excel at the ability to insert, update, retrieve and delete the relatively dynamic data. Thus, this architecture uses a knowledge base to represent the static portion of knowledge (the rules used by experts to deal with situations within a domain) and a database to provide an efficient means of maintaining the dynamic portion of the knowledge (specific solutions to particular problems).

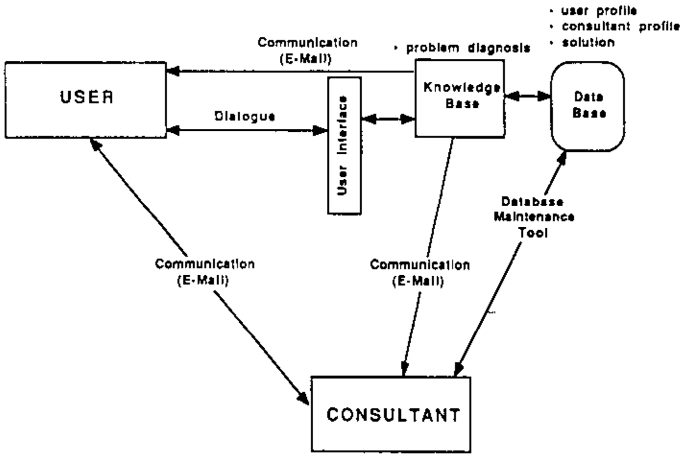


Figure 2. System Consultation Process

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/H is an example of an expert system using Inter-System Communication.

In our architecture, the communication link between the knowledge base and the database is bi-directional. The database stores the factual data which is accessed by the knowledge base during inferencing. The expert system also updates these facts and inserts new entries into the database to store a profile of new users and to record a log of the session results. This log is then accessed by the knowledge base and used for eliciting feedback during future consultations.

The electronic mail communication links in the system consultation process model are unidirectional. Messages are generated by the knowledge base during a consultation with an end-user. The database supplies E-mail addresses for the sender and receiver of the message. These messages are inserted onto the communication network for delivery to either the end-user or to a consultant. No electronic mail messages are received by the knowledge base in our current design.

4.1 Knowledge Base

A classification scheme is used to organize the domain knowledge within the knowledge base and to coordinate the interaction of the knowledge and data bases. The problem diagnosis step of the consultation process model attempts to classify the user's question within this classification hierarchy. Figure 3 presents an excerpt of a classification tree for ICE/H. This classification tree satisfies the notion of a class hierarchy (lattice) and inheritance of properties along the class hierarchy. The class hierarchy captures the IS-A relationship between a class and its subclass (equivalently, a class and its superclass). Subclasses of a class not only inherit all properties defined for the class, but can have additional properties defined locally. The notion of property inheritance along the hierarchy facilitates top-down design of the knowledge base and database. For conceptual simplicity, "similar" objects (problem/solution pairs) are grouped together into a class. All objects belonging to the same class are called instances of that class. This class tree can also be viewed as a decision tree and, in order to reach a problem/solution leaf node, the tree must be traversed.

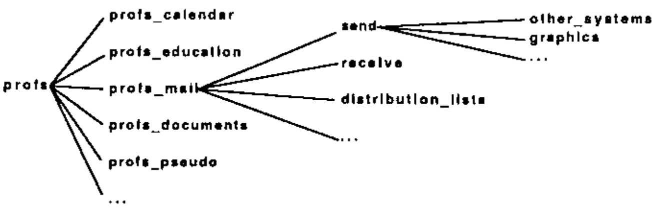


Figure 3. ICE/H Classification Tree (excerpt)

A classification tree for this architecture may be represented using any of the accepted knowledge representation models. An object-oriented programming language may be particularly suitable for implementing this representation. The final section of this paper elaborates on our plans for using object-oriented design in the future. Since most expert system development shells utilize rules, we use a rule representation in the following discussion. Once the knowledge has been acquired and a representation is defined, a method for traversing this knowledge must be chosen. Since our objective is to determine a solution to the classified problem, a key into the database of solutions must be generated. Backward chaining is the most appropriate inference mechanism for resolving this goal to generate the database key.

The problem classification key used to access the solution database is composed of three parts: a category, a subcategory, and a keyword string. The category and subcategory provide a definitive structure for the top two levels of the classification tree. The keyword string is accumulated as the remaining nodes of the class tree are traversed. Thus, from the example in Figure 3, we may have a category of "PROFS," a subcategory of "Mail," and a keyword string of "Notes/Send/Other Systems." When designing the dialogue for classifying the user's problem within the hierarchy, a serious attempt should be made to minimize the number of questions that must be answered by the user. Similar to the process model described in Section 3, the system should be designed to diagnose the user's general problem domain and then focus the consultation dialogue with more detailed questions. The current implementation of ICE/H (presented in the next section) contains four categories and an average of six subcategories within each category. Thus, two menu selections within the system dialogue can eliminate most problem classes, and further dialogue is conducted to construct a meaningful keyword string in a single subcategory. This separation of the category and subcategory from the keyword string facilitates the integration of the knowledge base with the database in the ICE/H architecture.

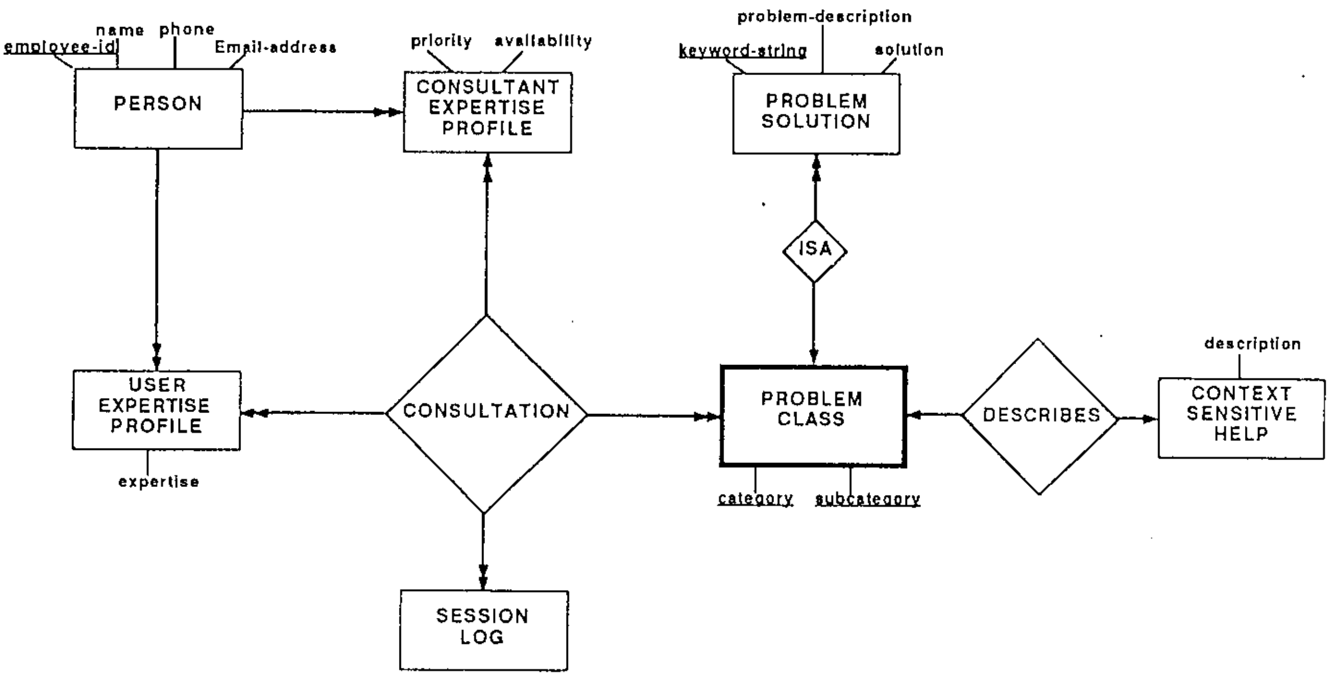
4.2 Data Base

The database component of the general architecture stores facts which are accessed by the knowledge base. Figure 4 displays an Entity-Relationship diagram showing the primary elements of the database design. The database relations support the consultation process in several respects. The Problem Class represents an abstract entity which relates many of the other entities in the design. The Problem Class in the database model has attributes in common with the top two levels of the classification structure in the knowledge base representation, thus providing a link for integrating the knowledge and database components of the architecture.

Each individual user has a profile which is divided into two parts: personal data and domain specific expertise at the subcategory level. An entity, Person, represents the personal data for each user of the system. The personal data, along with system-generated information about the problem diagnosis, is sent by the system to a consultant when a user is not satisfied with a solution. A similar personal data profile is stored for each consultant which provides an electronic mail ID for forwarding users' unresolved problems.

A Consultant Expertise Profile is defined which relates each consultant with one or more Problem Class(es) for which (s)he has expertise. Each consultant is also assigned a *priority* to select among those consultants proficient in a certain domain, and an *availability* to indicate whether that consultant is currently able to receive messages. Similarly, a User Expertise Profile is defined which stores the level of expertise for each user, relative to a given Problem Class.

The Problem/Solution entity contains specific instances of solution text for the classification key: category, subcategory, and keyword string. These solutions are retrieved during the consultation as the user's problem is diagnosed, and the text is presented to the user. The Context-



NOTE: Keys are underlined and only important attributes for each entity are shown

Figure 4. Database Model for the General Architecture

Sensitive Help entity supports descriptive information available on demand during a consultation session. By storing this help text in the database, rather than embedded in the knowledge base rules, IC consultants are able to customize term descriptions which clarify the dialogue questions.

The Session Log entity is used to capture several pieces of information about each user's consultation session and his/her satisfaction with the system. This information includes:

- Starting and ending date/time of each consultation (duration).
- Classification (database key) of the solution presented to the user.
- Outcome of the session (selected from: "None of the Above," no solution found, solution understood, solution not understood, *previous* solution not understood).
- User's remarks, if not satisfied with the solution.
- Employee ID of the consultant receiving the message from a dissatisfied user.
- User comments from the end-of-session questionnaire.

These data elements may be analyzed at a later date to ascertain the general user satisfaction with the system, to determine the problem areas which generate the most help queries, and to identify areas that are missing or poorly covered in the system. This capability to facilitate post-session analyses on the use of an expert system is a unique and valuable feature of this architecture.

A database maintenance tool, independent from the consultation knowledge base, must be provided for the Information Center staff to maintain the database contents.

4.3 Communication

The system consultation process model shown in Figure 2 indicates that four communication links are required to fulfill a user's request for help. A link required by all expert systems consists of a human-computer interface to conduct a dialogue with the user. However, when the expert system is implemented in a broad domain such as an IC, communication support should extend beyond the usual system dialogue. For IC help services, such support might include providing a copy of the expert solution to the end-user, forwarding a user's comments to a human consultant, and initiating communication directly between the end-user and the human expert whenever the user's problem is beyond the scope of the expert system. The architecture accommodates these three communication needs through the use of electronic mail in the integrated system.

Invariably there will be questions within a domain as large as the IC help desk for which the system has no acceptable answers. The question may be outside the scope of the system or the solution may not be specific enough to satisfy a particular user. Therefore, the expert system facilitates replies to unanswered questions and prompts the human experts to enhance existing solutions. The system prompts a dissatisfied user for comments describing his/her problem. These remarks, along with any problem classification accomplished by the system, are forwarded to an "appropriate" consultant. The problem classification is used to select a consultant from a database table which maintains a list of available consultants and their individual domains of expertise within the IC. This consultant receives an electronic mail message and replies to the user via electronic mail, telephone, or in person. The consultant then uses the database maintenance tool to enter an improved solution into the database or considers the unresolved problem for an extension of the knowledge base. For problems not within the domain of the system, knowledge engineers are required since support for automated extension of the knowledge base is not included in this architecture.

If a user did not understand the solution which was presented, a more detailed solution for the next expertise level below his/her assigned level may be selected and presented. For example, if a user was originally classified as *intermediate* by the system, then solutions for a *novice* will be retrieved. This process of displaying solutions having successively less assumed expertise simulates the actual consultation dialogue between an end-user and a consultant. In addition, the ICE/H system tailors the dialogue questions to the user's level of expertise; novices may be asked additional questions to clarify the diagnosis. If the user does not understand the most detailed solution that is available, then (s)he is asked to enter a textual description of the unresolved problem. This problem description, along with the classification database key determined by the system, is forwarded to a consultant who replies with a solution to the user's problem and, if necessary, modifies the solution in the database.

The message built by the system should be inserted seamlessly onto the network as if the user had used the electronic mail system directly and actually sent the message. This message should contain the appropriate origin and destination address, header information, and message body text. The addresses of both sender and receiver are retrieved from the database of user and consultant profiles. By tagging the message with the return address, the consultant can correspond directly with the user through the electronic mail system and bypass the expert system after the initial consultation. A short subject header for the message, which depends on the outcome of the consultation, is inferred by the system. The electronic mail facility should support messages to remote nodes on the network. The final section of this paper discusses the implications of remote messages for a distributed implementation of ICE/H.

5. IMPLEMENTATION OF ICE/H

ICE/H is an integrated system implemented on an IBM 4381 mainframe. The system was built using three components. ESE/VM (IBM 1988a), an expert system development shell based on EMYCIN, uses rule-based inferencing as the basis for its knowledge representation. The shell provides developers with editors for maintaining the knowledge base and an interface to external routines and to SQL/DS (IBM 1988b), an IBM relational database management system. These two features allow electronic mail messages to be generated and sent by the system and provide convenient access to the facts in the database. The PROFS (IBM 1986) integrated office system was used to support electronic mail.

The domain knowledge is represented as a classification tree. ESE uses Focus Control Blocks (FCBs) to direct the flow of control for a given consultation. The concept of FCBs is similar to the "hypothesis" of NEOMYCIN (Clancey 1983). The FCBs in ESE/VM allow the tasks to be organized into a hierarchy which allows a convenient representation of the class tree in ICE/H. One of the properties of an FCB is that parameter/rules above a certain FCB are visible to the lower level FCBs, but the FCBs higher in the hierarchy cannot access the parameters/rules of the lower level FCB. This *inheritance* property is useful for reducing the number of parameters/rules that need to be resolved, yet still allow access to all applicable parameters/rules of the particular class.

The architecture of ICE/H, as implemented, is shown in Figure 5. Our objective was to incorporate features which would allow much of the system to be maintained by the domain experts without the support of a trained knowledge engineer. The static rules are contained in an ESE/VM knowledge base and the dynamic elements are stored in a SQL/DS database. In addition, a semi-automated feedback mechanism is incorporated which notifies the domain experts, via PROFS electronic mail, when an ICE/H user does not receive a satisfactory solution to his/her problem. This feedback mechanism is also activated when a user's question is beyond the scope of the ICE/H knowledge base.

5.1 ICE/H Consultation Process

The ICE/H system has been implemented to support the users as well as the consultants of an IC. The system serves to train new consultants in the IC. The consultation process was presented in Figure 2. A typical consultation with ICE/H follows.

ICE/H Architecture

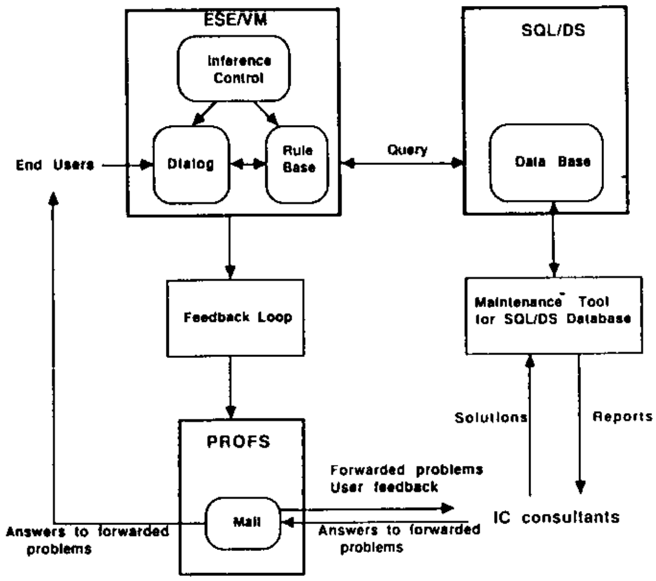


Figure 5. ICE/H Architecture

1. The user links to ICE/H and initiates a help session.

2. The user's profile is brought into the inferencing environment from SQL. If this is the first time that the user has accessed the system for this particular problem domain, his proficiency is gathered by the system and stored in the database.

3. Since the database contains all previous consultations, the user may view them and comment on their success or failure. These comments are logged and a message is forwarded to a consultant.

4. The system directs the dialogue with the user to fully diagnose and classify the problem.

5. A key to the solution database is generated and a query is made to the database. If a solution is not found, the dialogue with the user is captured along with the user's textual description of the problem and is forwarded to a consultant. Information on the domain of each consultant's expertise is maintained in the database. This is used to select the appropriate consultant when forwarding the message.

6. Solution(s) found are passed back to the inferencing environment for display to the user. If the user is not satisfied, (s)he may examine a solution appropriate for a lower level of expertise (the expanded text may address the dissatisfaction) or (s)he may send a message to a consultant.

7. The user exits the system, either satisfied with the solution, or awaiting the response of the consultant.

8. When a consultant receives a message describing an unsolved problem or an unsatisfactory solution, (s)he replies to the user (via electronic mail, phone or in person) and then updates the solution in the database by using the maintenance tool. This ensures that a current and understandable solution is found whenever the same problem is encountered again. It also ensures that a user with an unresolved problem gets through to a consultant. The user and consultant may continue their dialogue through the electronic mail system.

The ICE/H prototype is being tested at an IC Help Desk which supports a base of 5000 end-users, receives approximately 3000 calls per month, and is staffed by twelve consultants.

5.2 Prototype Development Process

A prototyping process (Boehm, Gray and Seewaldt 1984; Lantz 1984) was used to establish the detailed requirements for the ICE/H system and to validate and improve the knowledge base rules and database solutions. In addition to the typical benefits derived from developing system prototypes (user interface design, clarification of original design requirements, etc.), we were able to define several of the architectural features described in the previous section.

A group decision support (GDSS) environment was used to assist in the process of system development. Initially, the GDSS tools helped in determining the scope of the system and in eliciting the process model used for consultation. This process model was then emulated in the system using stepwise refinement. Three prototype systems were developed prior to the final implementation. Table 1 traces the evolution of three prototypes during the development of ICE/H. The first column of the table represents the main features of the consultation process such as problem diagnosis approach, the domain covered by each prototype, etc. The first prototype was developed primarily to illustrate the architecture of the system and did not support all features of the consultation process model. During the development, existing features of the system were modified to incorporate the process model or new features were added to completely replicate the consultation process model. Most of the enhancements made to the system were essentially to incorporate different types of feedback from the users. These features were aimed at building a semi-automatic learning feature in ICE/H. The final prototype incorporated all the various types of feedback mechanisms using electronic mail for communication.

Table 1. Evolution of ICE/H

Feature	ProtoType ONE	ProtoType TWO	ProtoType THREE
Domain Categories	• PROFS	• Other domains included such as File Transfers, Application System, Printing, etc.	• Same as TWO - some domains reclassified
Dialog for Problem Diagnosis	• Dialog tailored to type of user: Novice, Intermediate, Expert	• Same as ONE	• Reduced to two levels of expertise: Novice, Expert
User Profile	• Overall Measure of user expertise from proficiency in use of Hardware/Software	• User expertise determined according to domain category	• Users allowed to move down level of expertise if solution not understood • User expertise determined according to domain category, and subcategory
Consultant Profile	• Only one consultant designated to receive ALL user feedback	• Different consultants for each problem area receiving feedback	• More sophisticated algorithm to choose consultant (based on priority and availability)
Feedback Mechanism	• Feedback mailed to consultant only if solution NOT found	• Users given the option of mailing solutions to themselves • User feedback sent to consultants if solution NOT understood • Feedback solicited on most recent previous consultation	• Seamless integration with PROFS electronic mail facility • Feedback solicited on ALL previous consultations
Miscell-aneous		• Multiple consultations during the same session • Context-sensitive help built into knowledge base	• Complete record of consultations captured • Context-sensitive help enhanced by incorporating help text into database • Feedback on overall system solicited

Since ICE/H was designed not only for end-users, but also for IC consultants, feedback from both sets of users was required during the prototyping process. The GDSS tools were particularly useful in resolving conflicts among the various users. For instance, one major issue causing conflict was the number of levels of expertise to be addressed by the system. This issue was resolved amicably using the Issue consolidation and Voting tools (Applegate, Konsynski and Nunamaker 1986). A permanent record of feedback was easily obtained using these tools, which further assisted development during the feedback process.

5.3 Knowledge Acquisition

Most expert systems reflect the knowledge of one or, at most, a few experts. Even when several experts are used, one of these is usually the primary contributor who resolves any conflicts which arise during knowledge acquisition (Gaines and Boose 1988; Prerau 1985). However, this approach is not suitable for developing a system such as ICE/H. The breadth of knowledge required by IC consultants presents a difficult problem for designing and maintaining a knowledge-based support tool.

When an end-user queries the IC Help Desk, (s)he expects this single contact point to lead to a solution to the problem. No single human expert possesses all of the knowledge required to answer the full range of end-user problems. In practice, an individual with broad, but shallow, knowledge would answer the routine questions and would forward the more specialized questions to an expert consultant. A knowledge-based system for an IC Help Desk, therefore, must represent knowledge from multiple experts.

Our current implementation of ICE/H utilized approximately 12 experts who were all located at one site within the organization. However, it is possible that the experts for the system could be scattered nation-wide. Acquiring knowledge from multiple experts involves the issue of interpreting alternative points of view and varying methods for problem solving. Chorafas (1987) recommends using group discussions as one way to approach this issue. Since knowledge acquisition consumes approximately 50 percent of the total development time, efforts were made to facilitate this process. Accordingly, we used several GDSS tools to conduct the knowledge acquisition. The details of this process are summarized in another paper (Ram et al. 1989).

By using these tools, experts were able to work simultaneously, adding to each others comments while considering others' points of view. The major contribution of using the GDSS tools was the reduction in total time taken for the entire process and the ease of conflict resolution.

6. VALIDATION

Expert systems are evaluated primarily to test for program accuracy, completeness and utility (Gaschnig et al. 1983; O'Leary 1988). Evaluations by domain experts help to determine the accuracy of the embedded knowledge and the accuracy of any advice or conclusions that the system provides. These experts can also determine the boundaries of system knowledge. Evaluations by users help to determine the utility of the system: whether it produces useful results, the extent of its capabilities, its ease of interaction, the intelligibility and credibility of its results, and its speed, efficiency and reliability.

6.1 Knowledge Accuracy

Both static and dynamic evaluations were performed with the experts:

- **Static evaluation:** Each consultant was given a copy of all solutions existing in the database along with a description of the path followed to reach the solution. By reviewing this information, the consultant was able to identify and correct inconsistent, or incomplete, solutions within his/her area of expertise.
- **Dynamic evaluation:** Each consultant was required to travel down every single path of the decision tree. This allowed him to compare the system's line of reasoning with his own. Each consultant was also asked to use a number of "real" cases to test the system. Suggestions were solicited to improve the knowledge base and the decision tree. These suggestions were gathered during the feedback sessions conducted in conjunction with the knowledge acquisition. A *gripe* facility was made available to the consultants. This facility allowed the comments from the consultants to be forwarded directly to the system designers via E-mail.

6.2 Problem Domain Completeness

Even though the scope of each problem category had been specified in the first knowledge acquisition session, it was necessary to ensure that the problems handled by ICE/H did indeed cover all problems within each category. Testing for this type of completeness of the problem domain is difficult. One method is to build a flow chart displaying the sequence of each condition and the consequence of each pair of rules. This method was followed to test the completeness of the decision tree in ICE/H.

6.3 System Utility

We are currently in the process of validating the system more thoroughly. At the end of every consultation, users are required to fill out an electronic questionnaire. This information is included in the session log. Preliminary analysis of log records has shown that users are satisfied with the automated consultation process. It also appears that users receive solutions to their problems in less time by using ICE/H. This may be due to the fact that there is no "telephone tag." Informal feedback from consultants indicates that they are less interrupt driven; they prefer electronic mail to the telephone.

7. DISCUSSION AND FUTURE RESEARCH

In this paper, we have justified the need for a knowledge based system to support the help services of an Information Center. We have discussed a general process model and generic architecture for such a system and described a prototype that implements this model. This research has made the following contributions. We have demonstrated the use of an integrated environment to support one of the most important functions of an IC: help services. The system has been designed to provide maximum flexibility and to support the changing needs of end-users and the IC. The development of the system followed the standard prototyping approach; however, the process was facilitated by using a group decision support environment.

Unlike most knowledge based systems, ICE/H captures the knowledge from more than ten experts. The static portion of this knowledge has been separated from the dynamic portion by integrating the knowledge base with a database system. This allows for easy modification of the system without the services of a trained knowledge engineer. Unlike the earlier ICE/C system, a relational database management system (DBMS) supports the representation of the dynamic knowledge. This enables ICE/H to be used simultaneously by multiple users. Issues of concurrency and recovery are managed by the DBMS. Use of a DBMS also allows context sensitive help to be provided to the user whenever required. This help can be tailored to the user's level of expertise. The DBMS allows logging of all user consultations, providing statistics and descriptive feedback on the usage of the system.

Most important of all, the system has effectively captured the communication process between the consultant and user. As discussed in Section 4, this communication using electronic mail is the key to the system. ICE/H also uses the electronic mail facility to incorporate a semi-automatic *learning* feature in the system. This is done by means of the immediate feedback to IC consultants regarding solutions. Although the system does not have the capability to automatically update solutions, it triggers the consultants to do so.

While this research has addressed several issues, the following topics need to be addressed in the future.

7.1 Distributed ICE/H

Frequently, organizations are forced to operate multiple IC facilities each with its own set of consultants to provide help service. These facilities are required in order to support different offices, branches or regions. This support can now be provided by means of a system such as ICE/H implemented in a distributed environment. The system could be divided into a number of subsystems each operating at one or more sites connected together by means of a communication network.

Issues that need to be addressed in such a design include techniques to integrate independent knowledge bases. While the issue of integrating independent databases has been examined in great detail in the past (Ceri and Pelagatti 1984), integration of knowledge bases is currently being studied (Carlson and Ram 1989). By developing a distributed ICE/H system, users from any site in an organization could be supported with little or no incremental cost. The system would automatically refer to the appropriate knowledge base after conducting a preliminary consultation with the user. The individual knowledge base would then be used in the manner described in Section 4 to solve the user's problem. This would ensure location transparency to the end-user. Further, by integrating individual knowledge based systems, we can achieve synergy in solving a problem that no single system can solve. Typically, smaller systems can be built faster and more easily and integrated to solve problems that are out of the scope of any one system. Typically this "teamwork" is also followed by human IC consultants when they are faced with a complicated problem. In many cases, one expert is able to understand and solve a problem partially, and often refers the user to one or more other consultants for further clarification.

The distributed approach can also be used to integrate the knowledge base system for software and hardware recommendations with ICE/H.

7.2 Brittleness of ICE/H

Most of today's knowledge based systems cannot determine if the problems they are trying to solve are outside the scope of their expertise. This is referred to as the *brittleness* problem (Lenat 1987). In addition, systems expect inputs in a tightly defined environment and changes in syntax can cause problems. Currently, the brittleness problem is addressed to a very limited extent. Most of the burden of determining whether the problem is within the scope of the system is on the end-user. The user is provided with context sensitive help to assist in deciding if the problem can be addressed by the system. If the user decides it is not, feedback is obtained and forwarded to an appropriate consultant.

Human IC consultants can often solve problems using common sense or analogy. An important extension to the current ICE/H system would involve capturing "analogous knowledge" for a subset of non-routine problems. By building "analogical reasoning," the problem solving scope of the system could be extended, thereby addressing the brittleness issue.

7.3 Object-Oriented Design for ICE/H

While it was possible to build the decision tree using the ESE expert system shell, a truly object-oriented expert system shell would allow the Problem Class (see Figure 4) to be mirrored directly in the class hierarchy. These domains could be treated as discrete objects, referencing other objects only as needed. To add a new domain into the classification tree, one would merely instantiate an existing class. This abstraction mechanism would help the knowledge engineer capture more generic knowledge with higher reusability.

Further, it is necessary to decouple the control logic (order of execution of profile retrieval, message generation, etc.) from the problem diagnosis. Declarative and procedural knowledge, as well as the structure of a target domain, could be easily organized around objects. This would facilitate the modularity, modifiability, and maintainability of the knowledge based system.

8. ACKNOWLEDGEMENTS

This research was supported in part by grant WG-488116 from IBM, Gaithersburg, Maryland.

9. REFERENCES

Applegate, L. M.; Konsynski, B. R.; and Nunamaker, J. F. "A Group Decision Support System for Idea Generation and Issue Analysis in Organizational Planning." In *Proceedings of the Conference on Computer-Supported Cooperative Work*, 1986, pp. 16-34.

Boehm, B. W.; Gray, T.; and Seewaldt, T. "Prototyping versus Specifying: A Multi-project Experiment." *IEEE Transactions on Software Engineering*, December 1984, pp. 473-484.

Brancheau, J. C.; Vogel, D.; and Wetherbe, J. C. "An Investigation of the Information Center from the User's Perspective." *Data Base*, Fall 1985, pp. 4-17.

Buchanan, B.; Barstow, D.; Bechtel, R.; Bennett, J.; Clancey, W.; Kulikowski, C.; Mitchell, T.; and Waterman, D. "Constructing an Expert System." In *Building Experts Systems*, Reading, Massachusetts: Addison-Wesley, 1983, pp. 127-167.

Carlson D., and Ram, S. "An Object-oriented Design for Distributed Knowledge Based Systems." In *Proceedings of the 22nd Hawaii International Conference on System Sciences*, Kona, Hawaii, January 1989, pp. 55-64.

Ceri, S., and Pelagatti, G. *Distributed Database: Principles and Systems*. New York, New York: McGraw-Hill, 1984.

Chorafas, D. N. *Applying Expert Systems in Business*. New York, New York: McGraw-Hill, 1987.

Clancey, W. J. "Heuristic Classification." In *Knowledge-based Problem Solving*, Englewood Cliffs, New Jersey: Prentice-Hall, 1983, pp. 1-67.

Gaines, B., and Boose, J. *Knowledge Acquisition for Knowledge-Based Systems*, Volume 1. San Diego, California: Academic Press, 1988.

Gaschnig, J.; Klahr, P.; Pople, H.; Shortliffe, E.; and Terry, A. "Evaluation of Expert Systems: Issues and Case Studies." In *Building Experts Systems*, Reading, Massachusetts: Addison-Wesley, 1983, pp. 241-280.

Heltne, M.; Vinze, A. S.; Konsynski, B. R.; and Nunamaker, J. F. "ICE: Information Center Expert, A Consultation System for Resource Allocation." *Database*, Summer 1988, pp. 1-16.

IBM. *Expert System Development/Consultation Environment/VM*. Reference Manual, Program Number 5798-RWQ, 1988a.

IBM. *Professional Office System*. Reference Manual, Program Number 5848-439, 1986.

IBM. *SQL/DS Database Management System*. Reference Manual, Program Number 5998-RXQ, 1988b.

Lantz, K. E. *The Prototyping Approach*. New York, New York: Prentice Hall, 1984.

Lenat, D. "Overcoming the Brittleness Bottleneck." In *Proceedings of the Third IEEE Conference on Artificial Intelligence Applications*, February 1987, pp. 64-72.

Nunamaker, J. F.; Konsynski, B. R.; Chen, M.; Vinze, A.; Chen, Y. I. L.; and Heltne, M. "Knowledge-based Systems Support for Information Centers." *Journal of Management Information Systems*, Volume 5, Number 1, Summer 1988, pp. 6-24.

O'Leary, D. "Methods of Validating Expert Systems." *Interfaces*, Volume 18, Number 6, November-December 1988, pp. 72-79.

Prerau, D. S. "Selection of an Appropriate Domain for an Expert System." *AI Magazine*, Volume 6, Summer 1985, pp. 26-30.

Ram, S.; Nunamaker, J.; Liou, I.; Carlson, D.; and Hayne, S. "Using Group Decision Support Systems for Knowledge Acquisition: An Information Center Application." In *Proceedings of the Ninth International Conference on Decision Support Systems*, June 1989, pp. 87-102.

Rockart J. F., and Flannery, L. S. "The Management of End-User Computing." *Communications of the ACM*, Volume 26, Number 10, October 1983, pp. 776-84.

Vinze, A. *Knowledge Based Support for Software Selection in Information Centers: Design Criteria, Development Issues, and Empirical Evaluation*. Ph.D. thesis, Department of MIS, University of Arizona, Tucson, 1988.

Wiederhold, G., and Milton, J. *A Precis of Research on Knowledge-Based Management Systems*. Working Paper, Department of Computer Science, Stanford University, June 1987.

Zobaidie, A., and Grimson, J. B. "Expert Systems and Database Systems: How Can They Serve Each Other?" *Expert Systems*, Volume 4, Number 1, February 1987, pp. 241-280.