HELP DESK SYSTEM WITH INTELLIGENT INTERFACE

BYEONG HO KANG and KENICHI YOSHIDA
Advanced Research Laboratory, Hatoyama, Saitama, Japan

HIROSHI MOTODA
Institute of Scientific and Industrial Research, Osaka University, Mihogaoka, Ibaraki, Japan

PAUL COMPTON
School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

Automated help desk systems should retrieve exactly the information required to assist a user as quickly and as easily as possible be it for a lay user who knows little about the domain or for an advanced user who requires more specialized information. Automated help desk systems should also be easily maintainable, as knowledge in domains where help is required often changes very rapidly, for example, help for computer users. The aim of this study was to develop a help desk information retrieval mechanism suitable for a wide range of users and to provide a way of easily maintaining the system. The prototype developed for use over the World Wide Web combines keyword search and case-based reasoning to provide both rapid focusing on a part of the help information and guided interaction when the user is unclear about appropriate keywords. Ease of maintenance is provided by using multiple classification ripple down rules (MCRDR) to maintain the domain knowledge in the system. Further issues that arise include the problem of inappropriate focusing by keywords and maintenance in a distributed environment.

In many areas, various forms of help desk service provide users with help. In conventional help desk services, groups of human experts who differ in their knowledge and expertise try to solve the customer's problems. Their roles are determined according to their problem-solving ability and the degree of difficulty posed by the problem. Thus, to provide help desk service of high quality, the availability of high-level experts is crucial. However, the number of such high-level experts is limited, and the demand for automated help desk systems is increasing.

An expert system approach is a feasible solution. In addition, worldwide computer networks such as the Internet are becoming the major communication medium. The rapid communication enabled by such computer networks has also increased the demand for efficient maintenance of the knowledge for the help desk services. The overall aim of this study is to develop better methods of maintaining knowledge bases for help desk systems while improving their usability.

Address correspondence to Professor Hiroshi Motoda, Institute of Scientific and Industrial Research, Osaka University, 8-1 Mihogaoka, Ibaraki 567, Japan 567. E-mail: motoda@sanken.osaka-u.ac.jp

For the discussion here, knowledge-based systems are roughly classified into two groups: rule based and case based. Although the rule representation is most popular in the expert system community, the case-based reasoning (CBR) approach has been frequently used to build help desk systems (Kriegsman & Barletta, 1993; Barletta, 1993*a;* Shimazu et al., 1994; Simoudis, 1992). Most of these systems, however, require a major effort to maintain the case base. We propose to use the multiple classification ripple down rules (MCRDR) method to reduce the case base maintenance cost and speed up the maintenance process. The MCRDR method is a case-based maintenance method with which the expert can develop and maintain the case base without the help of the knowledge engineers. In the MCRDR method, when the CBR system retrieves cases that are identified by the user as inappropriate, the system simply asks the expert to identify the important features that distinguish the incorrectly retrieved cases from the present case. The expert also adds the relevant information needed by the user to the new case, so it can be added to the case base for future retrieval. It is indexed using the distinguishing features identified by the experts and information from the previous incorrect retrieval of the other cases (Kang & Compton, 1994). This simple approach allows large systems to be easily built (Compton et al., 1993). The earlier simpler version of MCRDR, namely, ripple down rules (RDR), was used to maintain the pathology expert interpreting report system (PEIRS) in St. Vincent Hospital, Sydney. This system showed a very high level of performance and was developed and maintained by experts as part of their normal duties without any knowledge engineering support (Edwards et al., 1993; Compton et al., 1994; Kang et al., 1994; Preston et al., 1994).

In this study it is assumed that the help desk system is to be used by various users from experts to novices. It is also designed to be constructed and maintained through the World Wide Web (WWW) with remote users directly retrieving information. How to provide a suitable interface for the various types of the remote user, especially for the novice user, and how to maintain the consistency of the case base perhaps maintained by the multiple experts are important research issues in this study.

In some sense, a help desk system can be seen as an information retrieving system. The information is saved as cases. The major difference between the proposed method and the conventional information retrieving methods is that the proposed method focuses on using human expertise to develop and maintain the way in which a user interacts with the system to produce appropriate information retrieval.

RELATED WORK

A user's request to a help desk service can be classified into two types: information search and diagnosis of problems. One may simply seek new information. “What is WWW?” is a typical example of this type of information search. One may also seek a solution to a problem. “My printer does not work!” is a typical example of this type of diagnosis. The issues related to these requests are studied in the information retrieval area and the knowledge-based diagnosis area. In this section we briefly summarize these related issues together with knowledge maintenance issues and interface issues, which are also important when trying to develop a practical system.

Information Retrieving Studies

Many information retrieval studies focus on how to find the relevant information from a large text base. A simple approach is collecting related documents and providing a search engine with the collection. The major research issues in this area fall into (1) text representation, (2) user query representation, and (3) retrieving method.

Text representation is one of the classical issues in information retrieving studies. The simple approach extracts all words in the documents with exceptions such as pronouns and articles. Use of the statistical measurement of word appearance, i.e., the term frequency, is also used to supply additional information (Salton et al., 1994; Lewis, 1992).

User query representation is studied to accurately capture user requests. The simple approach is to capture the requests by keyword combinations. Natural language understanding and sophisticated interaction techniques (Callan & Croft, 1993) are also studied to provide a better interface.

The retrieval function (Salton & McGill, 1983) actually selects and ranks documents. The ranking method is particularly important when many documents are selected. Since simple boolean logic does not cover the ranking, various statistical methods such as k–nearest neighbors are used to provide the ranking for the selected documents.

These information-retrieving methods are useful in constructing a help desk system. This is particularly true if the task is to search for new information. Diagnosis-type help services can also be handled by providing the relevant documents. However, this type of system assumes that the user can specify the appropriate keywords to search for the related documents. If the user does not have the skills to provide proper keywords, the system may fail to retrieve the relevant documents or may find too many irrelevant documents. Since natural language understanding still has a performance problem, how to provide better information retrieval to the novice user who is lacking such skills remains a research issue.

Expert System Approach

Rule-Based Approach

Many expert systems have been developed for diagnostic problems (Boose, 1989). Rules are the most popular representation for the knowledge base in the expert system. The methods to obtain rules are classified into two categories: automated methods (machine learning) and manual methods (e.g., interviewing) (Boose, 1991). Regardless of which method is used, the rule-based approach constructs a knowledge base that interprets the problem and suggests solutions. Though rules in the knowledge base are a good source of help for the user, they are different from a set of documents that are used in an information retrieval system.

There are common criticisms about rule-based approaches (Barletta, 1993*b*). The first one is that it is hard to construct a knowledge base. The second one is that it is hard to maintain the knowledge base. Another criticism is that a rule-based system is brittle. The second criticism is particularly crucial for development of the help desk system, since it should be able to accommodate changing knowledge. Note that these criticisms are based on the classical rule-based approaches. There have been many attempts to solve these problems. The most common approach is based on the idea of a "knowledge level" (Newell, 1982) analysis of a situation in a software engineering type of approach to knowledge acquisition (Wielinga et al., 1992).

Case-Based Approach

Although many have worked on rule-based systems, a CBR approach is frequently used to build help desk systems. SMART (Acorn, 1992), CASCADE (Simoudis, 1992), and CARET (Shimazu, et al., 1994) are examples of help desk systems that use a CBR approach.

CBR builds expert systems using past cases to solve new problems (Sycara & Ashley, 1991). It is based on the cognitive assumptions that real expertise comes from the experience of the expert and that episodic memory (Slade, 1991; Stottler et al., 1989) is an appropriate way to model the expertise. The approach of CBR is not to find appropriate rules in a knowledge base, but to find similar cases from the case base. CBR is appropriate when there is no formalized knowledge in the domain or where it is difficult for the experts to express their expertise in the rule format. In general, an expert is good at judging cases but not good at providing knowledge in the abstract (Manago & Kodratoff, 1987).

The functional similarity between CBR and information retrieval methods is that both methods carry out their task by retrieving the relevant cases or documents. Both methods maintain a set of cases/documents, and the new cases/documents are added into the database for later use (Barletta, 1993*b*). While information-retrieving studies concentrate on retrieval from large document databases (Barletta, 1993b; Callan & Croft, 1993), CBR approaches try to represent human problem-solving knowledge in the case representation.

Many CBR researchers claim that the knowledge acquisition bottleneck is solved by maintaining a case base, since the addition of new knowledge into the system can be performed by the simple addition of new cases. However, a CBR system needs a good case retrieval mechanism and a good case base maintenance method. If a CBR system lacks these methods, it cannot solve a problem because it may find inconsistent, irrelevant, or outdated cases.

Knowledge Maintenance by MCRDR

From long experience of knowledge-based maintenance (Compton et al., 1989), it was clear that experts never provide information on how they reach conclusions; rather, they justify that their conclusions are correct (Compton & Jansen, 1990; Compton et al., 1992). The basis of the MCRDR method is the maintenance and retrieval of cases. It tries to use the historic way in which the expert provides his expertise to justify the system's judgments (Kang & Compton, 1992). The cases and associated justifications (rules) are added incrementally when a case is misclassified in the retrieval process. This is similar to "failure-driven memory," which was introduced by Schank (1982). Experts are very good at justifying their conclusions about a case in terms of differences from other cases (Kolodner, 1985). When a case is incorrectly retrieved by an MCRDR system, the maintenance process requires the expert to identify how the case differs from the present case. This has a similar motivation to work on personal construct psychology, where identifying differences is a key strategy (Gaines & Shaw, 1990). The notion of ongoing refinement by differentiating cases seems a useful model of human episodic memory.

The MCRDR method also treats knowledge represented in the rule format as an important component. By using case differences, it can construct rules that summarize the essence of corresponding cases. When the MCRDR system tries to find an appropriate case from stored cases, the cases are retrieved using the information stored in the rule format. The case retrieval process of MCRDR can be seen as an inference process of a rule-based system.

The knowledge in MCRDR is stored in a tree structure (Figure 1). Each node of the tree is a rule with a corresponding case. The conditions in the rule are used to index the case. The MCRDR system evaluates all the rules in the first level of the tree (upper level rules, e.g., rules 2, 3 and 4 in Figure 1). After the user selects the rules whose conditions are satisfied by the current situation, the system evaluates the rules at the next level. The next-level rule is the refinement of the upper level rule. If the user specifies rules 2 and 4 as valid, the next rules are rules 5 and 6 (refinement of rule 2) and rules 7 and 9 (refinement of rule 4). This process stops when (1) there are no more refinement rules to be evaluated or (2) none of the refinement rules can be satisfied by the case in hand. It thus ends up with multiple paths, with each path representing a particular refinement sequence. Figure 1 shows the retrieval process for a particular case (test case a, d, c, f). In Figure 1 the conclusions of the inference are Cases 5 and 7.

Note that each case is represented by a set of keywords (i.e., a, d, c, etc., in Figure 1) and these keywords are used in the rule conditions. More precisely, the rule condition of MCRDR consists of a set of attribute and value pairs, e.g., "operating

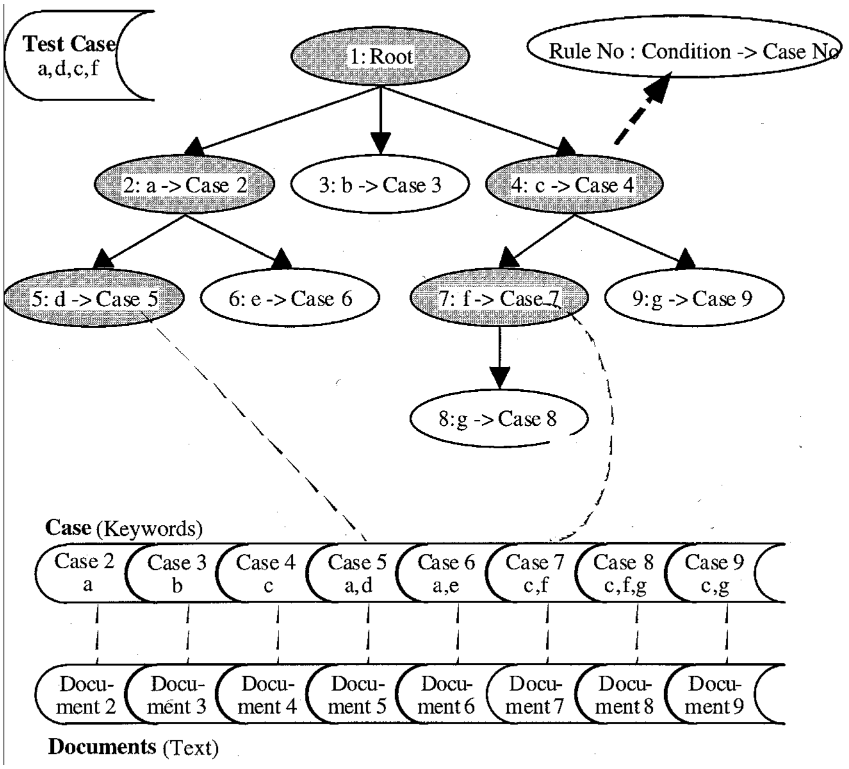


Figure 1. Knowledge structure for multiple classification ripple down rules. The rule tree in the system stores the information about how to retrieve cases in the case base. By using the rule tree, the system can find the appropriate cases from the case base by comparing the current case with conditions in the rule. To use the MCRDR method for the help desk system development, we also attach the documents to the case base. This document base is also shown in this figure. The shaded circles show the retrieving process for the test case (a, d, c, f).

system (attribute) is Linux (value).” Since both the attribute name, e.g., “operating system,” and the value, e.g., “Linux,” can be used as the keyword of the information retrieving engine (see section below, Combination of Keyword Search and MCRDR Inference), we can see that the rule condition and case of MCRDR are represented by a set of keywords. In this application, cases only hold essential information about each document and each document is attached to a case. Thus strictly speaking, a document and a case together can be understood as a case in other MCRDR/CBR systems. However, in the rest of this article a case may be understood, depending on the context, to include the attached document.

To add a new case into this tree structure, the system needs to identify the location and the condition of the rule. When the expert wants to add a new case into the MCRDR tree structure, there must be a case that is incorrectly retrieved by the system and not appropriate for the current problem. The system asks the expert to select conditions from the difference list between these two cases, i.e., new case and incorrectly retrieved case. Then the new case is stored as the refinement case with the rule whose condition part distinguishes these cases (see section below, Case Base Maintenance, for details). This differentiation of cases ensures the knowledge in the MCRDR system to be always validated and verified.

Variety of Users

The WWW has become the most popular Internet application today. While other applications such as E-mail and network news are still used to exchange information through computer networks, the WWW seems to be the best way of providing information to many users when they require it. We have therefore implemented a prototype of the help desk system using the WWW. By using the WWW, the user of the help desk system can retrieve information from a remote site.

WWW accessibility enables novice users to directly access the remote system without the help of retrieval experts, i.e., customer support persons. Most conventional help desk systems are operated by these retrieval experts, who know how to extract information from the help desk system. A help desk system with WWW access should have a good interface not only for the expert user but also for the novice user, who knows little about the retrieval mechanism.

In the development of a conventional help desk system, the user's behavior model is analyzed and the interface is designed based on the analysis. However, it is very hard to design an interface for a novice user who tries to use the system without any help from operators. Designing such an interface is not an easy task, and the performance of the system is highly influenced by the design. If we design an interface that attempts to confirm everything, experienced users may get bored. On the other hand, novice users may have difficulties in using a system if they have to specify detailed information by themselves.

Providing a good service for novice users is also difficult for an MCRDR system. Previous studies of MCRDR have been applied to domains where the system can get information from experts or automated measuring systems rather than directly from users. Although knowledge maintenance is an important issue for our study, the interface issue is also of critical importance.

HELP DESK SYSTEM FOR UNIX

In this study, we use a CBR approach with MCRDR to maintain the index structure for case base retrieval. We also combine a keyword search mechanism with the MCRDR system to realize an interface suited to different levels of expertise. In this section the combination of MCRDR and the keyword search mechanism is explained, together with the current implementation of a help desk system for operating system support for personal computers.

System Overview

Figure 2 shows the overview of the proposed help desk system. The arrows represent the different methods of information retrieval with this system. The user can use the MCRDR engine (thin line), the information retrieval engine (dashed line), or a combination of these two engines (heavy line). The maintenance of the knowledge in the system is performed by the MCRDR engine. The interface enables the interaction between the system and its user through the WWW.

The current implementation of the information retrieval engine uses a simple keyword search technique. The help desk system also maintains a keyword file for each case (document). The information retrieval engine tries to find cases whose keyword file includes the specified keywords. Keywords can be combined with disjunction and conjunction operators. The retrieved documents are provided to the interface directly when only the keyword search has been used. If the combined method is used, the information retrieval engine selects a subset of the original case base, which includes the specified keywords and constructs the optimized case base (see next section for details).

The function of the MCRDR engine is as described previously. However, it uses the optimized case base that is produced by the information retrieval engine when the combined method is used.

Combination of Keyword Search and MCRDR Inference

The MCRDR engine has two problems as an information retrieval engine. The first one is the number of conditions that are to be reviewed by the user. The second

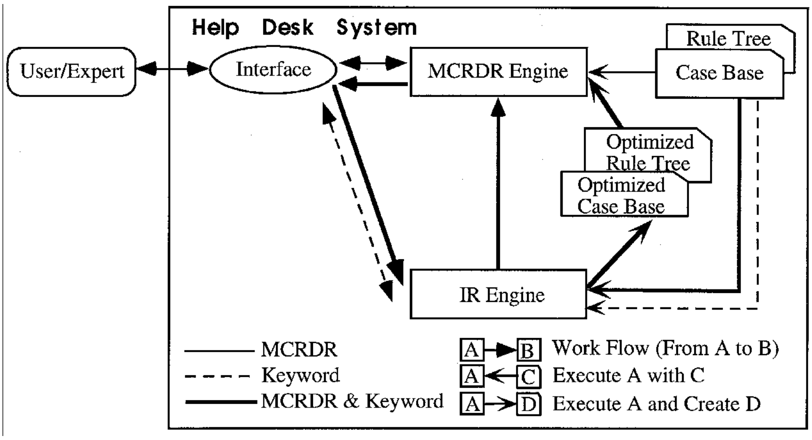


Figure 2. Overview of the help desk system. A user can select one of the three retrieving methods: MCRDR, keyword search, or a combination of both methods. If the combined method is selected, the system produces an optimized case base and rule tree from the original case base and rule tree using the information retrieval engine. The optimized case base and rule tree are then processed by the MCRDR engine.

one is the number of interactions between the user and the system. Even if the MCRDR tree structure (see Figure 1) is well organized by the human expert, the user, particularly experienced users, may not want to follow this structure. Thus the experienced user may only use the information retrieval method. However, the information retrieval method may not be able to distinguish the appropriate document from documents that are very similar. A certain level of semantics is required to solve this problem.

The proposed system is based on the idea of overcoming this problem by taking advantage of the benefits of both the MCRDR method and the information retrieval method. The system first retrieves documents by the information retrieval engine. At this stage, only syntactic information is used, i.e., keywords. Then the MCRDR engine selects the appropriate cases from the retrieved cases, i.e., the optimized case base in Figure 2, using the knowledge (semantics) acquired from the human experts (Figure 3).

When the information retrieval engine selects cases by the given keywords, it does not care about rules that are also stored in the MCRDR system. However, the corresponding rule conditions should be satisfied so that the case is appropriate for

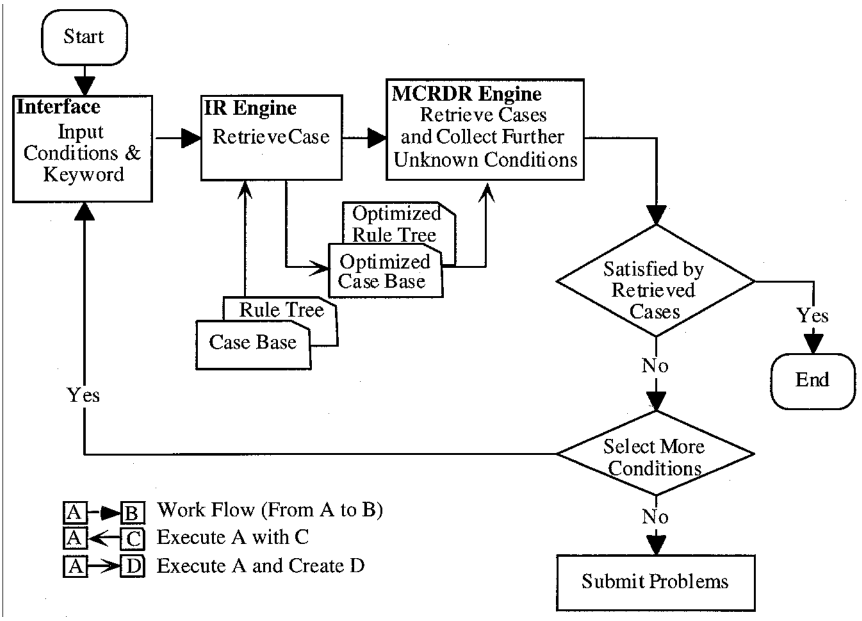


Figure 3. Integration of MCRDR and keyword search. Both the keywords for the information retrieval engine and the conditions for the MCRDR engine can be specified in any order. If a new keyword is specified, the optimized rule tree and case base will be regenerated. If a new condition is specified, MCRDR inference will proceed. A user can specify a keyword by filling the empty box area on the WWW screen. He/she can also specify a condition by clicking on terms shown on the same screen.

the user's request. Suppose the original MCRDR rule tree is shown in Figure 4*a*. The user then specifies keywords, by which the cases corresponding to the shaded rules are selected. If we believe that the user has selected appropriate keywords, then we do not need to check the conditions of rule 2 in the MCRDR inference process. This can dramatically reduce the number of conditions the user is asked. We assume here that the main problems to be addressed are the size of the search space and keywords that are too general. We do not address here the problem of incorrect or missing keywords, which would focus the search on another part of the case base.

By ignoring the unnecessary cases, we can obtain the subset of the MCRDR rule tree that is shown in Figure 4*b*. The user now interacts with the MCRDR system using this tree (Figure 4*b*). Note that cases 3, 8, and 9 are selected by the information retrieval engine, but this does not imply that the keywords that have been selected by the user satisfy the conditions in rules 3, 8, and 9. In other words, there is no guarantee that all the conditions of the rules in the path up to the point are satisfied. The keywords may be appropriate if we ignore the rules. The MCRDR engine then has to check the rule conditions in the MCRDR knowledge base. The system first checks the conditions of rules 3 and 4. Although rule 4 was not selected by the keywords, its conditions have to be satisfied to reach rules 8 and 9, so that in a large knowledge base, such a rule may usefully be evaluated to exclude multiple rules below (i.e., if the user judges the condition of rule 4 as invalid, cases 8 and 9 can be

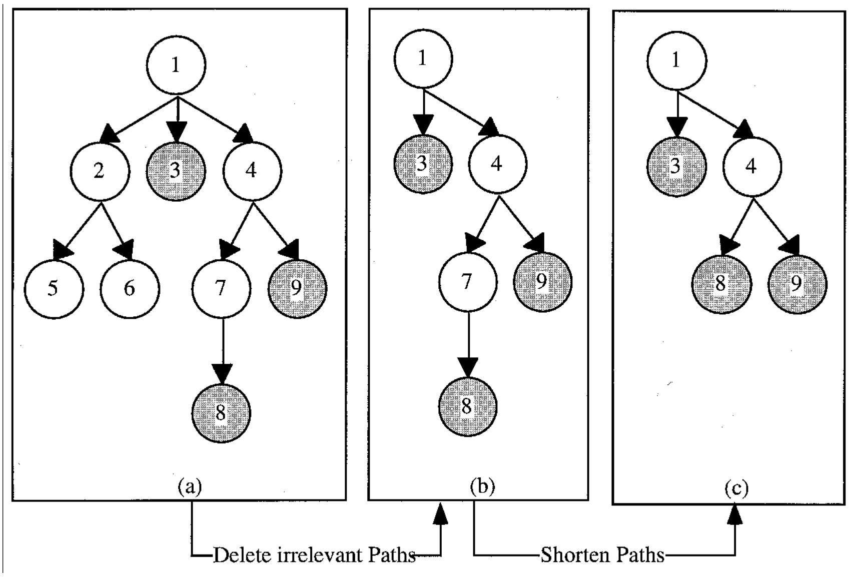


Figure 4. Optimizing the rule tree. The information retrieval engine selects the cases and deletes the irrelevant paths from the rule tree. This decreases unnecessary interaction between the system and its user. In this figure, each circle corresponds to one rule. Since each rule has a corresponding case, this tree can be seen as a case tree.

pruned). Suppose that the user selects the condition of rule 4 as valid. In a conventional MCRDR system, the system then asks about the conditions of rules 7 and 9. If rule 7 is satisfied, the system then has to check rule 8, since it was satisfied by the keyword. A shortcut therefore is to not ask about the conditions in rule 7, as knowing this information does not have the potential to reduce the search. Since the system has to ensure about either 7 alone or both 8 and 7, it is simpler to just ask about 8. This discussion applies when there are more than one node between rules 4 and 8. The overall process is that the system regenerates the rule tree in Figure 4*c* from that in Figure 4*a* based on the cases selected by the information retrieval engine.

Note that the proposed method is used through the WWW system and that the WWW server (i.e., our help desk system) cannot hold the information of previous client access (i.e., information about the user problem). To use the information specified by the user in the later processing, the WWW server sends the new questions together with the previously specified information, and the client sends back the answers together with the previous information. For example, if the user first specifies “print” as a keyword, the server generates a new WWW page with new questions. “Print” is also added in this WWW page as the already specified information. When the user specifies new information, the client sends back this new information together with the already specified information. This mechanism slightly slows down the system performance by reconstructing the optimized tree each time. However, this gives the user freedom to add new keywords during the MCRDR inference. If the user thinks that the current keywords are insufficient, he/she can add new keywords, retaining the already specified MCRDR rule conditions.

Case Base Maintenance

When the user is not satisfied with the retrieved cases, the submission module of the MCRDR engine can be invoked, so that the knowledge base can be improved. In this situation, the information entered by the user is stored as a new unsolved case. When an expert retrieves one of the unsolved cases, he/she is required to add the solution to the problem and check the conditions that were selected by the user.

Though the list of user-selected conditions helps the expert to understand the user’s problem, it may contain irrelevant conditions, or it may be underspecified. In such a case, the expert must correct the conditions. The list of conditions specified by the user is also influenced by the retrieval method. If the user uses the MCRDR method alone, the user has in fact already selected all the conditions in the rule paths to the retrieved cases. Thus when the user submits the case, the selected conditions (all the conditions in the path) are submitted. In this situation, the conditions provided by the user tend to contain rich information. However, if the user chooses the combined method, only some conditions are specified, with others implicit in the optimized case base. Since the system only includes the conditions that are actually specified by the user, the expert is normally required to supply additional information.

The MCRDR engine also supports this checking process by the experts. When the expert tries to add a new case to the knowledge base, the system tests the case against the current knowledge in the system. This is the same process as the MCRDR inference process. The expert can select the conditions from the provided list until it reaches the end of the tree, or he/she may decide to add new conditions at some branch of the tree.

After the location of the rule in the tree structure is fixed, the system checks the validation cases to maintain the consistency of the knowledge (Figure 5). Cornerstone cases, of which validation cases are a subset, are those cases that have required a rule to be added to the system because the system has not provided a satisfactory solution for them. The cornerstone cases are stored with rules that were added to deal with them and also with any other rules that gave a correct solution for part of the problem where there were multiple faults. Validation cases are the subset of the cornerstone cases that may be incorrectly retrieved by the new rule. The new rule has to be made specific enough to exclude such cases. The validation cases for a rule are its parent, its siblings, and the children of its siblings, unless any of these

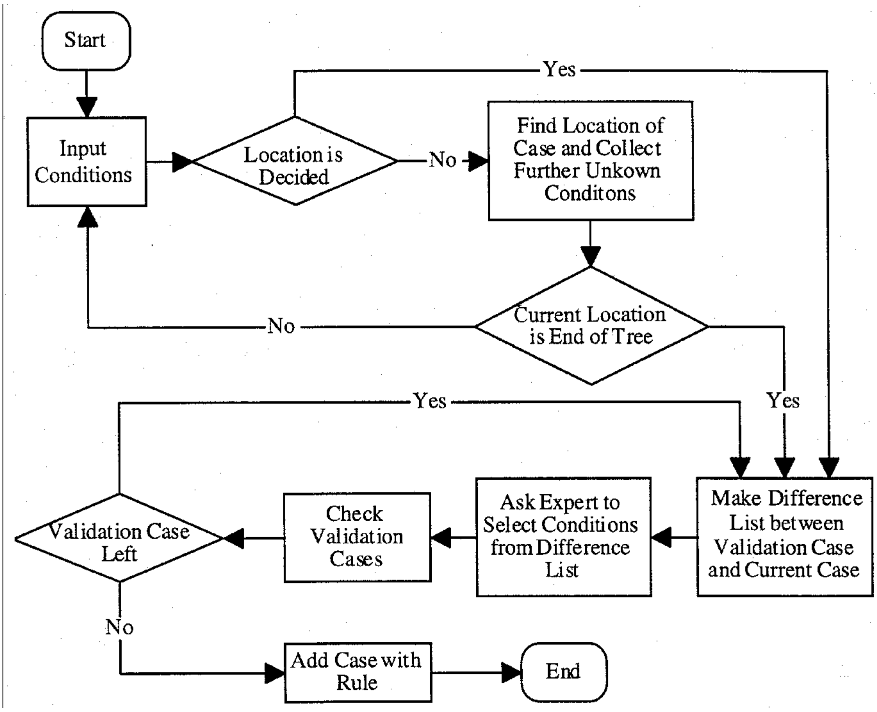

Start
Input Conditions
Location is Decided
Yes
No
Find Location of Case and Collect Further Unkown Conditons
Current Location is End of Tree
No
Yes
Yes
Make Difference List between Validation Case and Current Case
Ask Expert to Select Conditions from Difference List
Check Validation Cases
Validation Case Left
No
Add Case with Rule
End

Figure 5. Knowledge acquisition process. The standard MCRDR knowledge acquisition method is used to maintain the knowledge.

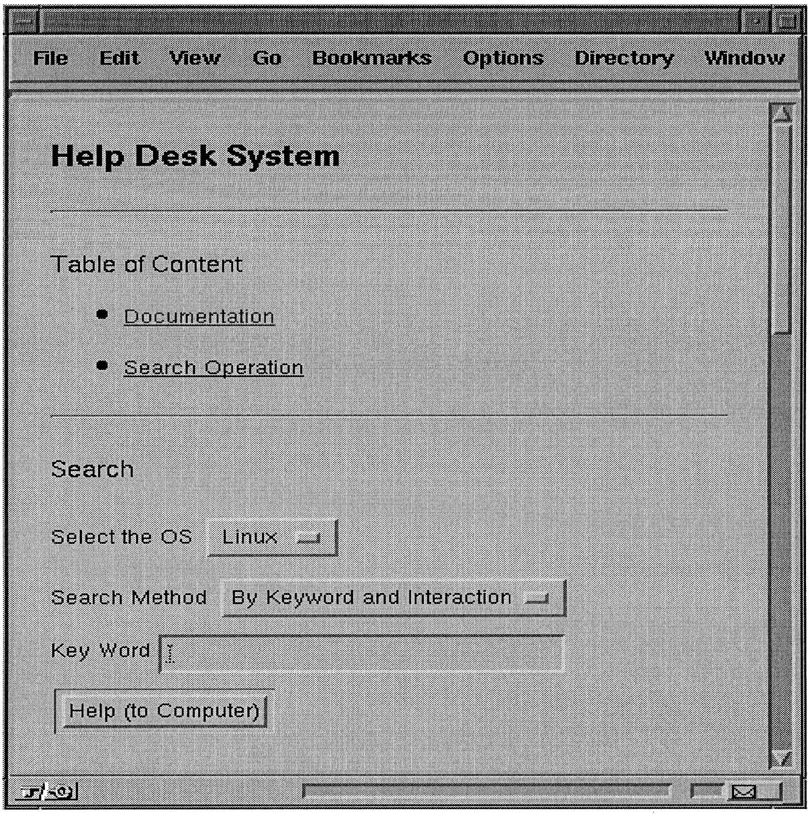
were correctly retrieved in a multiple-fault situation. These are all cases that, if processed by the system, would reach the new rule, but should not satisfy it and be retrieved along with the new case that has been added. Of course, this does not exclude them from being later retrieved together in a multiple-fault situation. For example, if the system adds a new rule below rule 8 in Figure 1, the only validation case is rule 8. However, if a rule is added below rule 4, i.e., a sibling to rules 7 and 9, then cases 4, 7, 8, and 9 are validation cases.

If a validation case satisfies the condition of the new rule, the system displays the differences between the validation case and new case. The experts are required to select additional conditions for the new rule from these differences, so that the validation case will not be retrieved. This is repeated until no validation case satisfies the new rule conditions. In this process the number of interactions can be a problem when the system finds many validation cases that satisfy a new rule's conditions. However, a previous evaluation study of MCRDR shows that this is a relatively minor issue and even a simulated expert can eliminate all the validation cases in a few interactions (Kang et al., 1994). In unpublished studies a suitable rule is even more rapidly arrived at by human experts. The expert's task in this is simply to identify those features of the case that distinguish it from other cases and is prompted in this by the presentation of these other cases. It seems that in this environment the expert rapidly identifies the few key features necessary to differentiate the case from related cases. The maintenance environment then provides for further refinement as required with later rules.

Current Status and Example

We have implemented a prototype help desk system using the WWW. The first page for the system is shown in Figure 6. The current knowledge base has information extracted from SGML documents on Linux, a UNIX clone for personal computers. Although the SGML documents are a rich source of valuable information, their richness prevents novice users from making the best use of the stored information. In spite of the contribution of many expert users, one still finds many frequently asked questions on network news and elsewhere. The rapid ongoing improvement of this operating system makes the stored information obsolete quite rapidly and thus makes it difficult to seek appropriate information. This implies a need for better information retrieval facilities. The current implementation accessing SGML Linux documents has about 2000 rules/cases. The basic knowledge structure follows the structure of the SGML help documents provided with Linux.

As described in the previous sections, the user can search for information by answering questions, guided by the system. The sequence of questions is based on the MCRDR rule tree and reflects the original document structure. The user can reduce the interaction with the system by supplying additional keywords (Figure 7). After the system has acquired enough information from the users about their requests, it comes back with the related portion of the SGML document (Figure 8).



Help Desk System

Table of Content

- Documentation
- Search Operation

Search

Select the OS: Linux

Search Method: By Keyword and Interaction

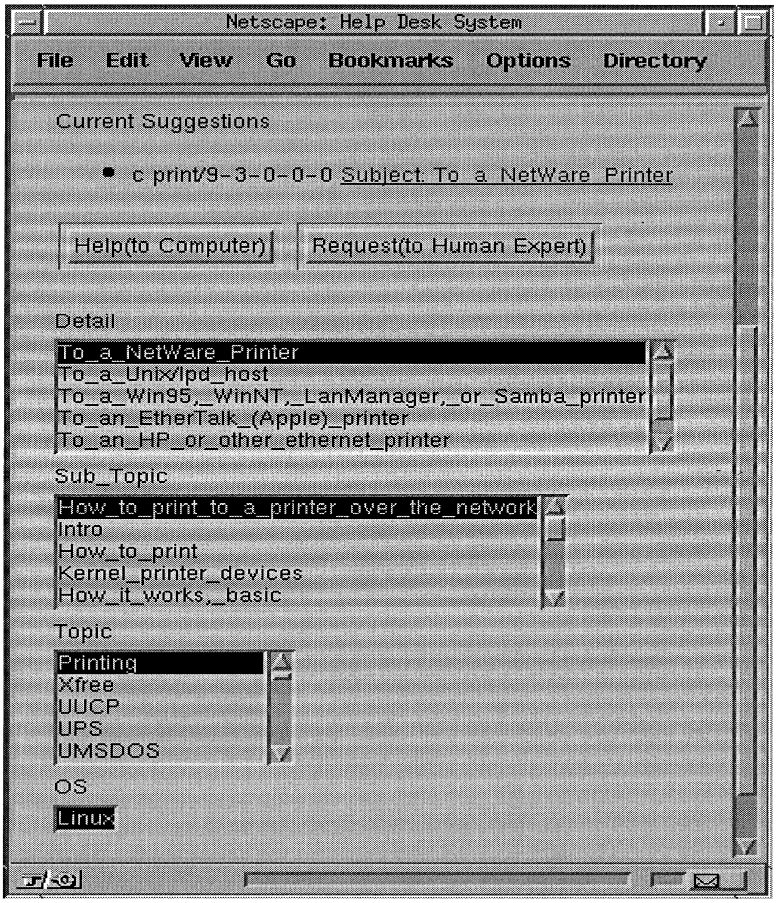
Key Word

Help (to Computer)

Figure 6. First page of the help desk system. Although the current system only has information on the Linux operating system, the user should specify the name of the operating system he uses as the first input for the system. This is for future extensions. If the user selects keyword search or the combination of MCRDR and keyword search, he could also specify a keyword with the name of the operating system.

Figure 9 compares the number of items the user should confirm with each method, i.e., Keywords, MCRDR, and the combination of both methods. If the user selects keyword search and specifies "print" as the search key, the system retrieves 74 documents as the result. In other words, the user has to check 74 items to reach an appropriate document. Although the user can proceed with the search by specifying the conjunction or disjunction of keywords, selecting good keywords becomes an important user role. If the user only uses the MCRDR method, the system first displays the section name of the Linux SGML document. The succeeding items displayed on the screen are the name of subsections. In this case, the user has to check 65 items to reach the same document.

The user can reduce the items to be checked with the combined method. If the user specifies "print" as the keyword and uses the combined method, the first page


Netscape: Help Desk System
File Edit View Go Bookmarks Options Directory
Help(to Computer)
Request(to Human Expert)

Current Suggestions

● c print/9-3-0-0-0 Subject: To a NetWare Printer

Detail

To_a_NetWare_Printer
To_a_Unix/lpd_host
To_a_Win95,_WinNT,_LanManager,_or_Samba_printer
To_an_EtherTalk_(Apple)_printer
To_an_HP_or_other_ethernet_printer

Sub_Topic

How_to_print_to_a_printer_over_the_network
Intro
How_to_print
Kernel_printer_devices
How_it_works,_basic

Topic

Printing
Xfree
UUCP
UPS
UMSDOS

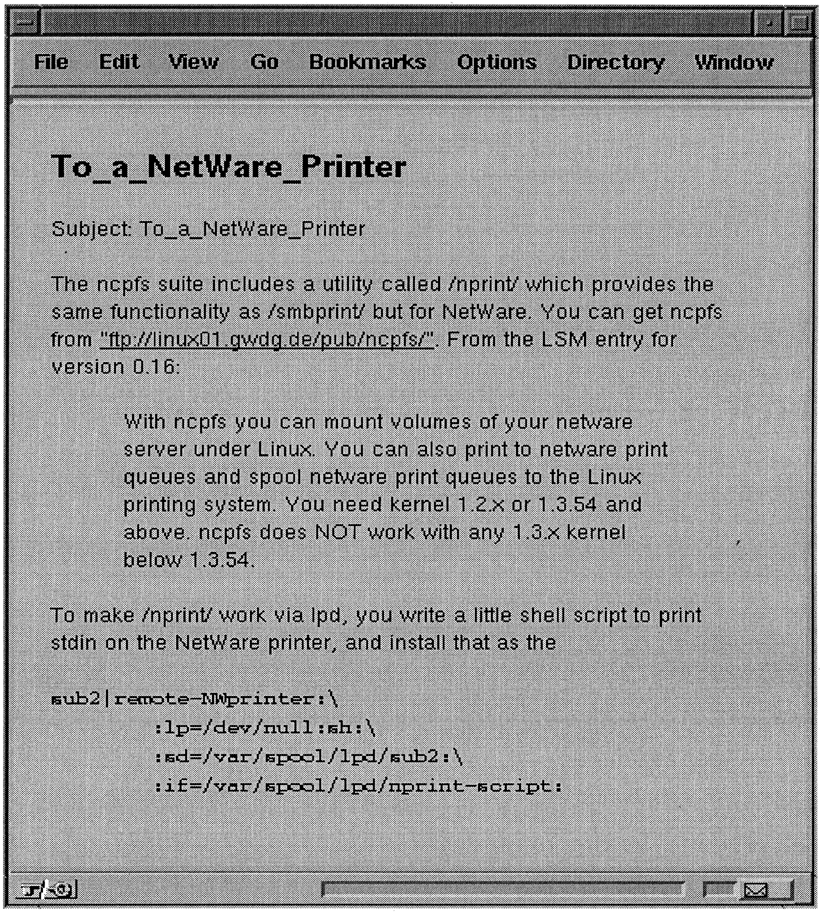
OS

Linux

Figure 7. After the user specifies the keywords (for keyword search) and conditions (for MCRDR inference), the system shows the selected keywords/conditions with the name of the documents that may fit the user's request.

has nine items. Although the original Linux SGML document contains 45 sections, the combined method can eliminate unnecessary sections from the list. Note that not only is the total number of items the smallest with the combined method, but the number of new items on the screen is also smallest. Since finding an appropriate item from many items is not an easy task for the user, we believe this is another advantage of the proposed method.

The knowledge maintenance function for this system is also implemented as a WWW page by which an expert for this operating system can construct new rules. Unlike the current SGML documents, where contents are organized by human experts, the structure of the new information is semi-automatically determined with the help of the MCRDR differentiation function.



To_a_NetWare_Printer

Subject: To_a_NetWare_Printer

The ncpfs suite includes a utility called /nprint/ which provides the same functionality as /smbprint/ but for NetWare. You can get ncpfs from "ftp://linux01.gwdg.de/pub/ncpfs/". From the LSM entry for version 0.16:

> With ncpfs you can mount volumes of your netware server under Linux. You can also print to netware print queues and spool netware print queues to the Linux printing system. You need kernel 1.2.x or 1.3.54 and above. ncpfs does NOT work with any 1.3.x kernel below 1.3.54.

To make /nprint/ work via lpd, you write a little shell script to print stdin on the NetWare printer, and install that as the

```
sub2|remote-NWprinter:\
        :lp=/dev/null:sh:\
        :sd=/var/spool/lpd/sub2:\
        :if=/var/spool/lpd/nprint-script:
```

Figure 8. Documents selected by the user. Note that the result is just a standard WWW document, and we can also use hypertext links, as commonly used in WWW pages. For example, ftp://linux01.gwgd.de/pub/ncpfs/ in the figure is a link to another Internet resource.

DISCUSSION

The research here is not complete, and there are a number of issues where further research is required. The key research issue is an evaluation of the advantages of this approach. This will be a major task in this sort of interactive domain. Other research issues are outlined below.

Knowledge Maintenance in Distributed Environment

We use the WWW mechanism to let a remote user get information from the system. However, WWW is not a simple one-way information service, but provides

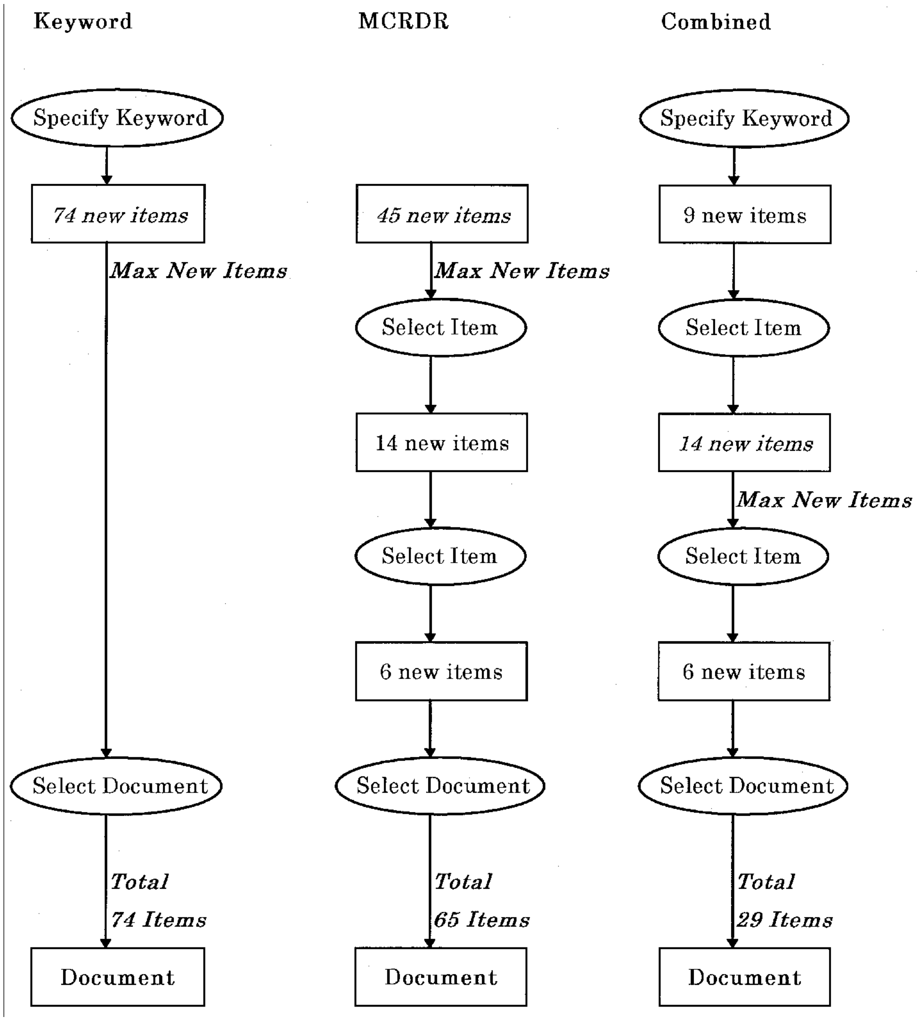


Figure 9. Comparison of number of items the user should confirm for each method. The user can reduce the interaction with the combined method. Not only is the total number of items the smallest with combined method, but the number of new items on the screen is also smallest with the combined method.

bidirectional communication. This feature enables the remote maintenance of the help desk system.

The WWW provides us only with the communication service. Maintenance of the contents (i.e., knowledge in the system) in a distributed environment is another issue. The cooperation of multiple experts to maintain a single knowledge base seems to raise difficult problems, particularly where remote experts are involved over the WWW.

In this study, we use MCRDR to maintain the consistency of the knowledge base, although the system may be maintained by multiple experts. When an MCRDR system is used by a single expert, the system supports him/her by checking the difference between cases. By requiring the expert to select features that differentiate the cases, the maintenance process cannot corrupt the previous performance of the knowledge base. We believe that the same mechanism works even if we have multiple experts. The singular advantage of MCRDR is that each expert has only to focus on the difference between two cases, i.e., a new case that he/she wants to add and one of the old cases that satisfies the rule's conditions but should not be retrieved. In this sense, the expert does not have to consider how the existing knowledge base was built and what other terminologies were used. We have not evaluated this, and it remains a significant research issue. However, if the experts do not have to consider the actual structure of the knowledge base, but simply have to differentiate cases based on existing terminology or new terms they add, it will be very interesting to observe whether there are any issues at all in having multiple experts. There will, of course, be domains where this is a major issue, but in this domain the documents to be retrieved are already specified, and the issue is simply to specify sufficient conditions to differentiate their retrieval. It also seems that the terms used to identify problems are common, or at least fairly unambiguous. Habits of experts such as selecting poor conditions to differentiate cases, or overspecialization, or minimal specialization to exclude validation cases, will simply show up as requirements to add or refine more rules than might otherwise be the case. The use of an MCRDR approach avoids the issue that arises in conventional knowledge engineering of how experts go about their problem solving. CBR, in general, attempts to avoid this problem, but MCRDR provides the further advantage of built-in maintenance and incremental development via the simple task of asking the expert to identify features that differentiate cases.

Combination of MCRDR and Information Retrieval Techniques

We have added a keyword search mechanism to the MCRDR system to give the help desk system a better interface. From a converse viewpoint, we have added an MCRDR inference mechanism to an information retrieval system. The central issue for an information retrieval system is the effective use of the various statistical indices to find an appropriate document from a large collection of documents. In our study, we support this syntactical analysis of documents by the knowledge stored in the case base. As explained in the section above, Help Desk System for UNIX, our current implementation uses a simple information retrieval mechanism, i.e., a simple keyword search mechanism. However, the performance of this simple mechanism is enhanced by using the MCRDR engine to further refine the search. Other, more sophisticated information retrieval mechanisms could also be used, with MCRDR again providing the final focusing and dealing with the user's inability to sufficiently specify keywords.

The problem in our approach is that we assume that a lack of knowledge will result in keywords that match the desired document but are too few in number or too general, resulting in the retrieval of too many documents. However, the problem arises that the user may also use keywords that result in an optimized case base that does not include the relevant document. We have not addressed this issue in the current implementation. The simplest solution is, of course, to revert to a pure MCRDR interaction if a relevant document is not found. However, since some information has already been obtained from interaction with the user in terms of rule conditions, these could be used to select a second optimized case base. Schatz et al. (1996) propose a method to extract co-occurring words from documents as alternative retrieval keywords. This could also be used to form a further optimized case base. There are a number of variants on such strategies, which will be investigated.

Another interesting use of information retrieval techniques is to provide advice as to rule condition candidates. We can use term frequency information (Salton et al., 1994) to extract typical words from the documents. This could then be used to generate candidates for MCRDR's rule conditions to suggest to the expert. This may also be used to order the interaction with the user in working through rule conditions for the optimized case base.

The aim of the combined use of MCRDR and keyword or related information retrieval techniques is to facilitate the user's interaction with the system. However, it has a second key advantage. Information retrieval techniques depend on the algorithm used, and the normal solution to an imperfect algorithm is to find another algorithm. Here the inclusion of MCRDR allows the ready addition of expert knowledge to refine the information retrieval engine performance without replacing it. The expert gradually fixes the inadequacies of the retrieval system, by the addition of knowledge. This also obviously allows local specialization. However, there is a third likely advantage in a combination system: that keywords and various information retrieval measures of their utility could support the expert in making useful rules.

Other Issues

What would happen if the level of the user's knowledge was too low to understand the questions being asked? The simple solution is to keep an extended explanation related to each rule condition for the novice user. This would be similar to bubble help in Macintosh. Another and more attractive approach would be to use the current system recursively. This could handle information more dynamically. If the user does not understand the current list of questions, he/she may again ask the help desk system what he/she does not understand. For example, if the user is asked to specify the window management system in his/her computer, the system assumes that the user knows terms related to the window management system. However, the user may not know the term (e.g., twm) that is presented, and may again ask the help desk system what this is. This recursive approach seems to parallel human behavior. When people do not understand an expert's question, they ask him/her to clarify what is required.

As shown in Figure 8, we can set up a hypertext link inside the help desk WWW page. This mechanism makes the implementation of these ideas simpler, and it should be fairly easy to manage the recursive interaction with the user. In contrast, when one uses a pure keyword search to find out how to find documents, one can very rapidly become lost. This extension remains to be implemented.

CONCLUSION

This study has shown that the MCRDR method (a kind of a case-based reasoning system) can provide an effective framework to develop a help desk system. The contents of the case base can be easily maintained by a human expert with the help of MCRDR functions, as MCRDR can keep track of the problem-solving contexts in the past and store them in the knowledge base. The combined use of keyword search and MCRDR seems likely to provide an excellent interface for a range of users for the help desk system by reducing unnecessary interaction between the system and the user. Since the framework is quite general, it could be applied to various kinds of help desk systems.

REFERENCES

Acorn, T. L. 1992. SMART: Support management automated reasoning technology for Compaq customer service. In *IAAI-92,* pp. 2–18.

Barletta, R. 1993*a.* Building a case-based help desk application. *IEEE Expert* (December):18–26.

Barletta, R. 1993*b.* Case-based reasoning and information retrieval: Opportunities for technology sharing. *IEEE Expert* (December):2–8.

Boose, J. H. 1989. A survey of knowledge acquisition techniques and tools. *Knowledge Acquisition* 1:3–37.

Boose, J. H. 1991. Knowledge acquisition tools, methods, and mediating representations. In *Knowledge acquisition for knowledge-based systems,* eds. H. Motoda, R. Mizoguchi, J. Boose, and B. Gaines, pp. 25–62. Amsterdam: IOS Press.

Callan, J. P., and W. B. Croft. 1993. An approach to incorporating CBR concepts in information retrieving systems. In *AAAI Spring Symposium,* pp. 28–34. Stanford, Calif.: AAAI Press.

Compton, P., G. Edwards, A. Srinivasan, R. Malor, P. Preston, B. Kang, and L. Lazarus. 1992. Ripple down rules: Turning knowledge acquisition into knowledge maintenance. *Artificial Intelligence in Medicine* 4:47–59.

Compton, P., K. Horn, J. R. Quinlan, L. Lazarus, and K. Ho. 1989. Maintaining an expert system. In *Application of expert systems,* ed. J. R. Quinlan, pp. 366–385. London: Addison Wesley.

Compton, P., and R. Jansen. 1990. A philosophical basis for knowledge acquisition. *Knowledge Acquisition* 2:241–257.

Compton, P., B. H. Kang, P. Preston, and M. Mulholland. 1993. Knowledge acquisition without analysis. In *Knowledge acquisition for knowledge based systems: Lecture notes in AI,* vol. 723, eds. G. Boy and B. Gaines, pp. 278–299. Berlin: Springer-Verlag.

Compton, P., P. Preston, B. H. Kang, and T. Yip. 1994. Local patching produces compact knowledge bases. In *A future for knowledge acquisition,* eds. L. Steels, G. Schreiber, and W. V. D. Velde, pp. 104–117. Berlin: Springer-Verlag.

Edwards, G., P. Compton, R. Malor, A. Srinivasan, and L. Lazarus. 1993. PEIRS: A pathologist maintained expert system for the interpretation of chemical pathology reports. *Pathology* 25:27–34.

Gaines, B., and M. Shaw. 1990. Cognitive and logical foundations of knowledge acquisition. In *The 5th Knowledge Acquisition for Knowledge Based Systems Workshop,* pp. 9.1–9.25. Banff, Alberta, Canada: SRDG Publications, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada.

Kang, B. H., and P. Compton. 1992. Knowledge acquisition in context: The multiple classification problem. In *The 2nd Pacific Rim International Conference on Artificial Intelligence,* vol. 2, pp. 847–853.

Kang, B. H., and P. Compton. 1994. A maintenance approach to case based reasoning. In *Advances in case-based reasoning: EWCBR-94,* eds. M. Keane, J.-P. Haton, and M. Manago, pp. 226–239. Paris, France: Springer.

Kang, B. H., P. Compton, and P. Preston. 1994. Multiple classification ripple down rules. In *Third Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop,* eds. R. Mizoguchi, H. Motoda, J. Boose, B. Gaines, and P. Compton, pp. 197–212. Hatoyama, Japan: Japanese Society for Artificial Intelligence.

Kolodner, J. L. 1985. A process model of case-based reasoning in problem solving. In *The Ninth International Joint Conference on Artificial Intelligence,* vol. 1, pp. 284–290. Los Angeles, Calif.: Morgan Kaufmann.

Kriegsman, M., and R. Barletta. 1993. Building a case-based help desk application. *IEEE Expert* (December): 18–26.

Lewis, D. D. 1992. An evaluation of phrasal and clustered representations on text categorization task. In *SIGIR'92,* pp. 37–44.

Manago, M. V., and Y. Kodratoff. 1987. Noise and knowledge acquisition. In *The Tenth International Joint Conference on Artificial Intelligence,* vol. 1, pp. 348–349. Milano, Italy: Morgan Kaufmann.

Newell, A. 1982. The knowledge level. *Artificial Intelligence* 18:87–127.

Preston, P., G. Edwards, and P. Compton. 1994. A 1600 rule expert system without knowledge engineer. In *World congress on expert systems,* pp. 17.1–17.10. Lisbon, Portugal: Macmillan New Media License.

Salton, G., J. Allan, C. Buckley, and A. Singhal. 1994. Automatic analysis, theme generation, and summarization of machine-readable texts. *Science* 264:1421–1426.

Salton, G., and J. M. McGill. 1983. *Introduction to modern information retrieval,* pp. 52–99. New York: McGraw-Hill.

Schank, R. C. 1982. *Dynamic memory: A theory of reminding and learning in computers and people.* Cambridge: Cambridge University Press.

Schatz, B. R., E. H. Johnson, and P. A. Cochrane. 1996. Interactive term suggestion for users of digital libraries: Using subject Thesaurus and co-occurrence lists for information retrieval. In *ACM International Conference on Digital Libraries,* pp. 126–133. Bethesda, Md.: ACM.

Shimazu, H., A. Shibata, and K. Nihei. 1994. Case-based retrieval interface adapted to customer-initiated dialogues in help desk operations. In *The Twelfth National Conference on Artificial Intelligence,* eds. J. Mylopoulos and R. Reiter, vol. 1, pp. 513–518. Seattle, Wash.: AAAI Press.

Simoudis, E. 1992. Using case-based retrieval for customer technical support. *IEEE Expert* (October): 7–12.

Slade, S. 1991. Case-based reasoning: A research paradigm. *AI Magazine* 12:42–55.

Stottler, R. H., A. L. Henke, and J. A. King. 1989. Rapid retrieval algorithms for case-based reasoning. In *The Eleventh International Joint Conference on Artificial Intelligence,* vol. 1, pp. 233–237. San Mateo, Calif.: Morgan Kaufmann.

Sycara, K. P., and K. D. Ashley. 1991. Case-based reasoning: Tutorial note of the Twelfth International Joint Conference on Artificial Intelligence. In *IJCAI-91.*

Wielinga, B. J., A. T. Schreiber, and J. A. Breuker. 1992. KADS: A modeling approach to knowledge engineering. *Knowledge Acquisition* 4:5–54.