Help Desk System with Intelligent Interface.

Byeong Ho Kang : Visiting Research Scientist
e-mail : kang@harl.hitachi.co.jp

Kenichi Yoshida : Senior Research Scientist
e-mail : yoshida@harl.hitachi.co.jp
Advanced Research Laboratory
Hatoyama, Saitama, Japan
Zip Code : 350-03

Hiroshi Motoda : Professor
e-mail : motoda@sanken.osaka-u.ac.jp
The Institute of Scientific and Industrial Research
Osaka University
8-1 Mihogaoka, Ibaraki 567, Japan
Zip Code : 567

Paul Compton : Associate Professor
e-mail : compton@cse.unsw.edu.au
School of Computer Science and Engineering
University of New South Wales, Sydney, Australia
Zip Code : 2052

Abbreviated Title : HDS with Intelligent Interface

Author for correspondence:
Professor Hiroshi Motoda
e-mail : motoda@sanken.osaka-u.ac.jp
The Institute of Scientific and Industrial Research
Osaka University
8-1 Mihogaoka, Ibaraki 567, Japan
Zip Code : 567

Abstract

Automated help desk systems should retrieve exactly the information required to assist a user as quickly and as easily as possible either for a lay user who knows little about the domain or for an advanced user who requires more specialist 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 keyword and maintenance in a distributed environment.

1. Introduction

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 the problem difficulty. 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 the automated help desk systems is increasing.

An expert system approach is a feasible solution. In addition to this, world wide computer networks such as the Internet are becoming the major communication media. 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 system while improving their usability.

For the discussion here knowledge based systems are roughly classified into two groups, i.e., rule based and case bases. 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 and Barletta, 1993; Barletta, 1993a; 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 which are identified by the user as inappropriate, the systems simply asks the expert to identify the important features which 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 and Compton, 1994). This simple approach allows large systems to be easily built (Compton et. al., 1993). The earlier simpler version of MCRDR, 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; Perston 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.

2. Related Work

A user's request to a help desk service can be classified into two types: information search and diagnosis of his/her problems. One may simply seek for new information. "What is WWW?" is a typical example of this type of information search. One may also seek for a solution to his/her 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 we try to develop a practical system.

2.1. 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) the text representation, 2) the user query representation, and 3) the retrieving method.

The text representation is one of the classical issues in the 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).

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

The retrieval function (Salton and J. McGill, 1983) actually selects and ranks the 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 so 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 skill 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.

2.2. Expert System Approach

2.2.1. 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 which 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 which are used in an information retrieval system.

There are common criticisms about rule based approaches (Barletta, 1993b). 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 the help desk system development 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).

2.2.2. Case Based Approach

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

CBR builds expert systems using past cases to solve new problems (Sycara and 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 expert 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 and 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 the later use (Barletta, 1993b). While information retrieving studies concentrate on retrieval from large document data bases (Barletta, 1993b; Callan and Croft, 1993), CBR approaches try to represent the 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 can not solve a problem because it may find consistent, irrelevant or outdated cases.

2.3. Knowledge Maintenance by MCRDR

From a long experience of knowledge base 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 and Jansen, 1990; Compton et al., 1992). The basis of the MCRDR method is the maintenance and the retrieval of cases. It tries to use the historical way in which expert provides his expertise to justify system's judgments (Kang and Compton, 1992). The cases and associated justifications (rules) are added incrementally when a case is miss-classified in the retrieval process. This is similar to "failure-driven memory" which was introduced by Schank (Schank, 1982). Experts are very good at justifying their conclusions about a cases 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 and Shaw, 1990). The notion of on going 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 which 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., rule 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 rule 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 Case 5 and Case 7.

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 which 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 3.3 for details). This differentiation of cases ensures the knowledge in the MCRDR system to be always validated and verified.

2.4. 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 network, the WWW seems to be the best way of providing information to many users when they require it. We have therefore implemented a proto-type 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 which attempts to confirm everything, experienced users may get bored. On the other hands, 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.

3. 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 the MCRDR and keyword search mechanism is explained together with the current implementation of a help desk system for operating system support for personal computers.

3.1. 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 (plain line), the information retrieval engine (dashed line), or a combination of these two engines (bold line). The maintenance of the knowledge in the system is performed by 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 which is produced by the information retrieval engine when the combined method is used.

3.2. 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 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 may not want to follow this structure, particularly experienced users. 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 which are also stored in the MCRDR system. However, the corresponding rule conditions should be satisfied so that the case is appropriate for 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 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 which is shown in Figure 4 (b). The user now interacts with the MCRDR system using this tree (b). Note that the cases 3, 8 and 9 are selected by the information retrieval engine but this does not imply that the keywords which have been selected by user satisfy the conditions in the rules 3, 8 and 9. The keywords may be appropriate without relating to the rules. The MCRDR engine has to then check the rule conditions in the MCRDR knowledge base. The system first checks the conditions of the 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. Let's 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 short cut 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 7. The overall process is that the system regenerates rule tree (c) from (a) based on the cases selected by the information retrieval engine.

3.3. 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 the user-selected conditions helps the expert to understand the user's problem, it may contain irrelevant conditions, or may be under specified. 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. So, 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 which 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 which 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 were correctly retrieved in a multiple fault situation. These are all cases which 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 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 which 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 which 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.

3.4. Current Status

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 on-going improvement of this operating system makes the stored information obsolete quite rapidly and thus makes it difficult to seek for appropriate information. This implies a need for better information retrieval facilities. The largest of the current implementations 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). 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 their contents are organized by the human experts, the structure of the new information is semi-automatically determined with the help of the MCRDR differentiation function.

4. 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.

4.1. 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 bi-directional 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 co-operation 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 which 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 he/she 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, it will be very interesting to observe that 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, 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 identify problems are common, or at least fairly unambiguous. Habits of experts such selecting poor conditions to differentiate cases, or over specialization, 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.

4.2. 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 Section 3, 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 key words that mach 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 which 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 was 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 (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 conditions candidates. We can use term frequency information (Salton et al. 1994) to extract typical words from the documents. This could then be used generate candidates for the 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.

4.3. 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 doesn't understand the current list of questions, he/she may again ask the help desk system what he/she doesn't 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 don't understand an expert's question, they ask him/her to clarify what is required.

As shown in Figure 8, we can set up a hyper text link inside the help desk WWW page. This mechanism makes the implementation of these idea simpler and it should be fairly easy to manage the recursive interaction with 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.

5. Conclusion

This study has shown that the MCRDR method (a kind of a case base 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 frame work is quite general, it could be applied to various kinds of help desk systems.

Reference

Acorn, T. L. (1992). SMART: Support Management Automated Reasoning Technology for Compaq Customer Service. In *IAAI-92*, (pp. 2-18).

Barletta, R. (1993a). Building a Case-Based Help Desk Application. *IEEE Expert* (December), 18-26.

Barletta, R. (1993b). 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 H. Motoda, R. Mizoguchi, J. Boose, and B. Gaines (Eds.), *Knowledge Acquisition for Knowledge-Based Systems* (pp. 25 - 62). Amsterdam: IOS Press.

Callan, J. P., and Croft, W. B. (1993). As Approach to Incorporating CBR Concepts in information retrieving systems. In *AAAI Spring Symposium*, (pp. 28-34). Stanford, California, USA: AAAI Press.

Compton, P., Edwards, G., Srinivasan, A., Malor, R., Preston, P., Kang, B., and Lazarus, L. (1992). Ripple Down Rules: Turning Knowledge Acquisition into Knowledge Maintenance. *Artificial Intelligence in Medicine*, 4, 47-59.

Compton, P., Horn, K., Quinlan, J. R., Lazarus, L., and Ho, K. (1989). Maintaining an Expert System. In J. R. Quinlan (Ed.), *Application of Expert Systems* (pp. 366-385). London: Addison Wesley.

Compton, P., and Jansen, R. (1990). A Philosophical Basis for Knowledge Acquisition. *Knowledge acquisition*, 2, 241-257.

Compton, P., Kang, B. H., Preston, P., and Mulholland, M. (1993). Knowledge Acquisition Without Analysis. In G. Boy and B. Gaines (Eds.), *Knowledge Acquisition for Knowledge Based Systems, Lectures Notes in AI (723)* (pp. 278-299). Berlin: Springer Verlag.

Compton, P., Preston, P., Kang, B. H., and Yip, T. (1994). Local Patching Produces Compact Knowledge Bases. In L. Steels, G. Schreiber, and W. V. d. Velde (Eds.), *A Future for Knowledge Acquisition* (pp. 104-117). Berlin, German: Springer-Verlag.

Edwards, G., Compton, P., Malor, R., Srinivasan, A., and Lazarus, L. (1993). PEIRS: a Pathologist Maintained Expert System for the Interpretation of Chemical Pathology Reports. *Pathology*, *25*, 27-34.

Gaines, B., and Shaw, M. (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 Compton, P. (1992). Knowledge Acquisition in Context : the Multiple Classification Problem. In *The 2nd Pacific Rim International Conference on Artificial Intelligence*, 2 (pp. 847-853). Seoul, Korea:

Kang, B. H., and Compton, P. (1994). A Maintenance Approach to Case Based Reasoning. In M. Keane, J.-P. Haton, and M. Manago (Eds.), *Advances in Case-Based Reasoning : EWCBR-94* (pp. 226-239). Paris, France: Springer.

Kang, B. H., Compton, P., and Preston, P. (1994). Multiple Classification Ripple Down Rules. In R. Mizoguchi, H. Motoda, J. Boose, B. Gaines, and P. Compton (Eds.), *Third Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop*, (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*, 1 (pp. 284-290). Los Angeles, USA: Morgan Kaufmann.

Kriegsman, M., and Barletta, R. (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 Kodratoff, Y. (1987). Noise and Knowledge Acquisition. In *The Tenth*

International Joint Conference on Artificial Intelligence, 1 (pp. 348-349). Milano, Italy: Morgan Kaufmann.

Newell, A. (1982). The Knowledge Level. *Artificial Intelligence*, *18*, 87 - 127.

Preston, P., Edwards, G., and Compton, P. (1994). A 1600 Rule Expert Systems without Knowledge Engineer. In *World Congress on Expert Systems*, (pp. 17.1-17.10). Lisbon, Portugal: Macmillan New Media License.

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

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

Schank, R. C. (1982). *Dynamic Memory : A Theory of Reminding and Learning in Computers and People*. Cambridge: Cambridge University Press.

Schatz, B. R., Johnson E. H., Cochrane P. A. (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, Maryland USA

Shimazu, H., Shibata, A., and Nihei, K. (1994). Case-Based Retrieval Interface Adapted to Customer-Initiated Dialogues in Help Desk Operations. In J. Mylopoulos and R. Reiter (Eds.), *The Twelfth National Conference on Artificial Intelligence*, 1 (pp. 513-518). Seattle, USA: 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., Henke, A. L., and King, J. A. (1989). Rapid Retrieval Algorithms for Case-Based Reasoning. In *The Eleventh International Joint Conference on Artificial Intelligence*, 1 (pp. 233-237). Detroit, Michigan USA: Morgan Kaufmann.

Sycara, K. P., and Ashley, K. D. (1991). *Case-Based Reasoning : Tutorial Note of the Twelfth International Joint Conference on Artificial Intelligence*. IJCAI-91.

Wielinga, B. J., Schreiber, A. T., and Breuker, J. A. (1992). KADS: a Modeling Approach to Knowledge Engineering. *Knowledge Acquisition*, *4*, 5-54.

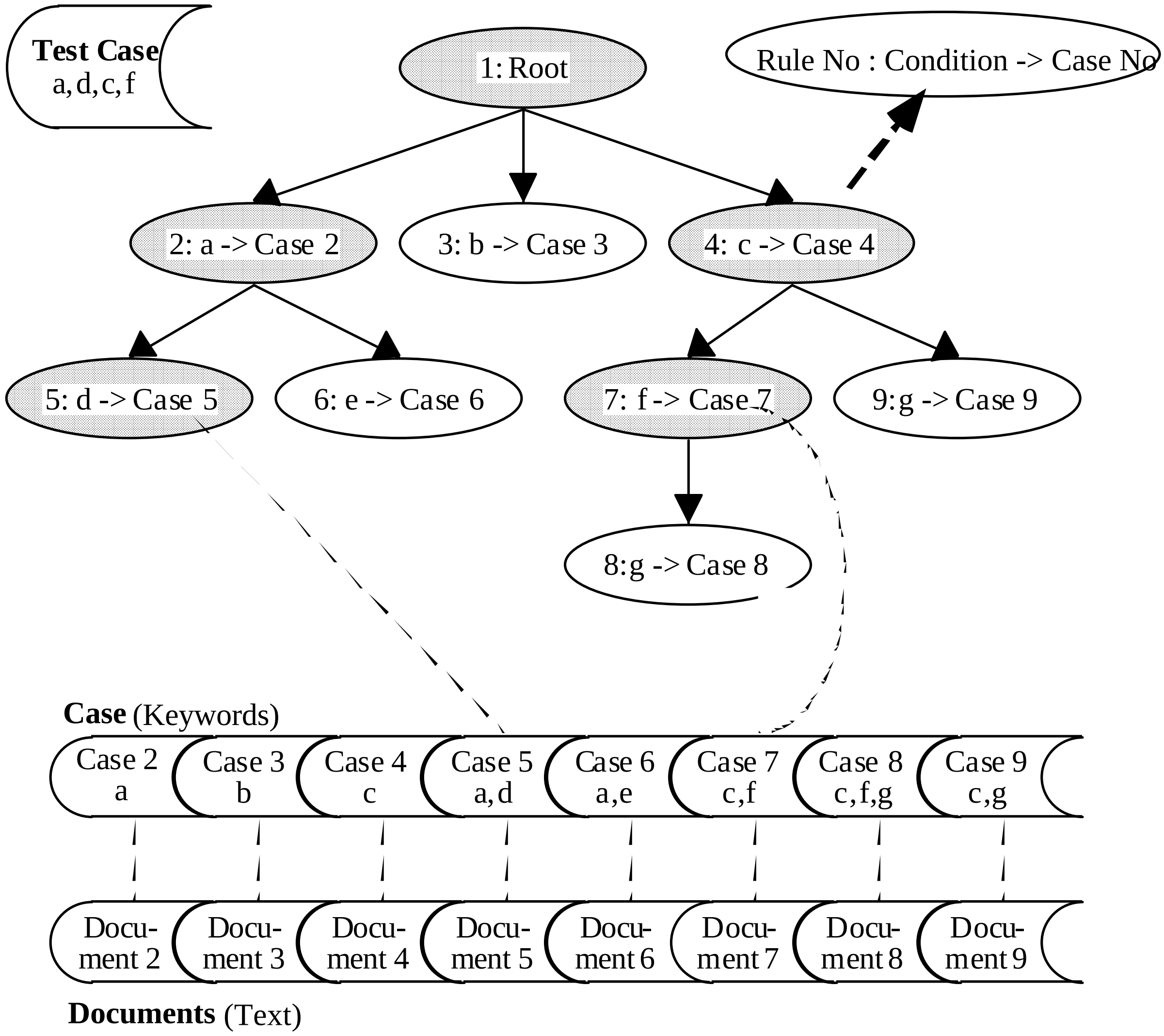

Figure 1 Knowledge Structure for Multiple Classification Ripple Down Rules.

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 store the documents as a part of the case base. This document base is also shown in this figure. The shaded circles shows the retrieving process for the test case (a, d, c, f).

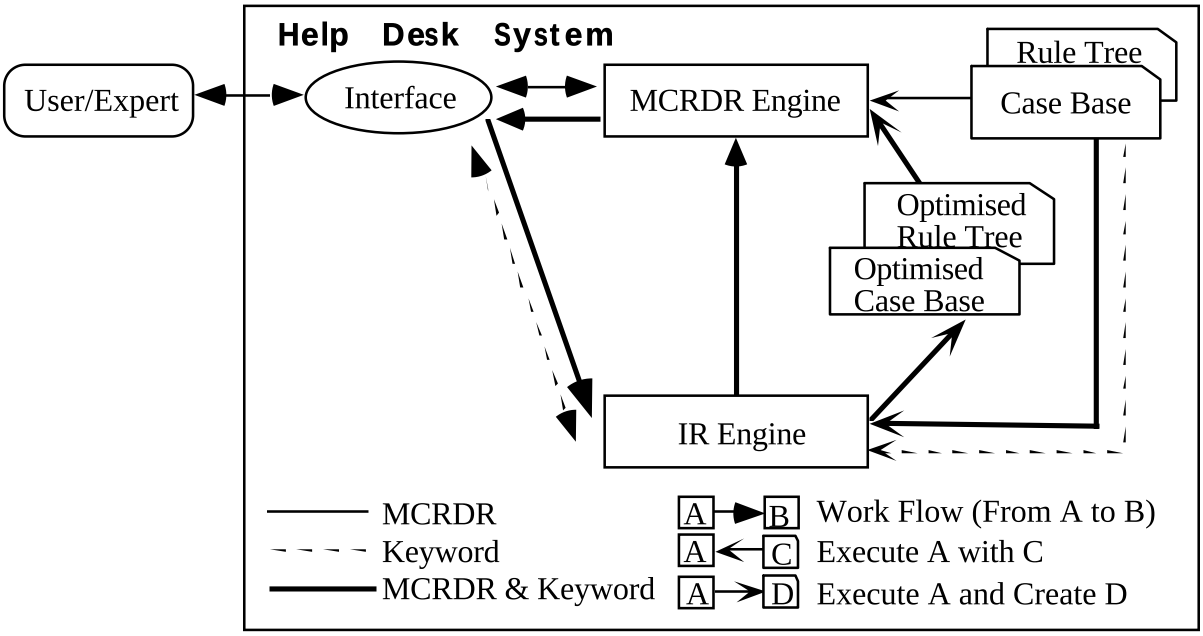

Figure 2. Overview of the Help Desk System.

A user can select one of the three retrieving methods, MCRDR method, keyword search method, 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.

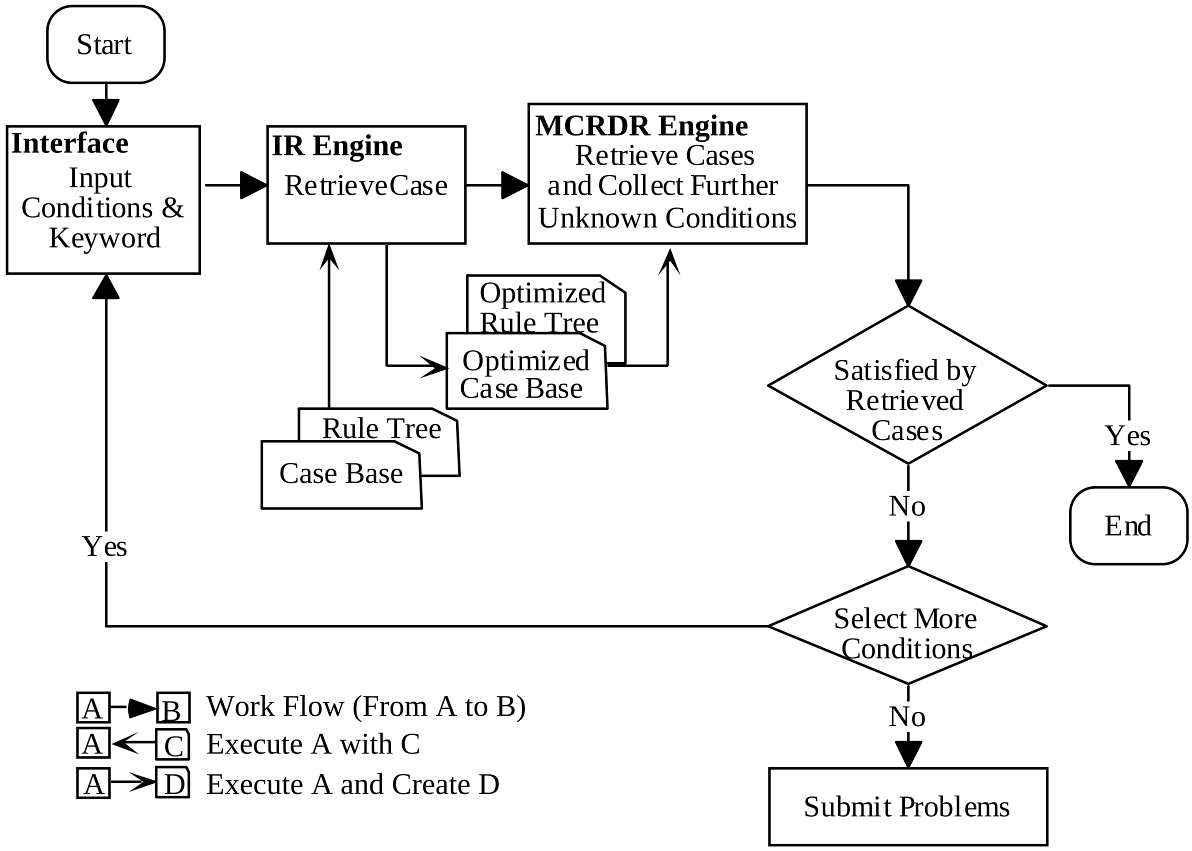

Figure 3 Integration of MCRDR and Keyword Search.

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.

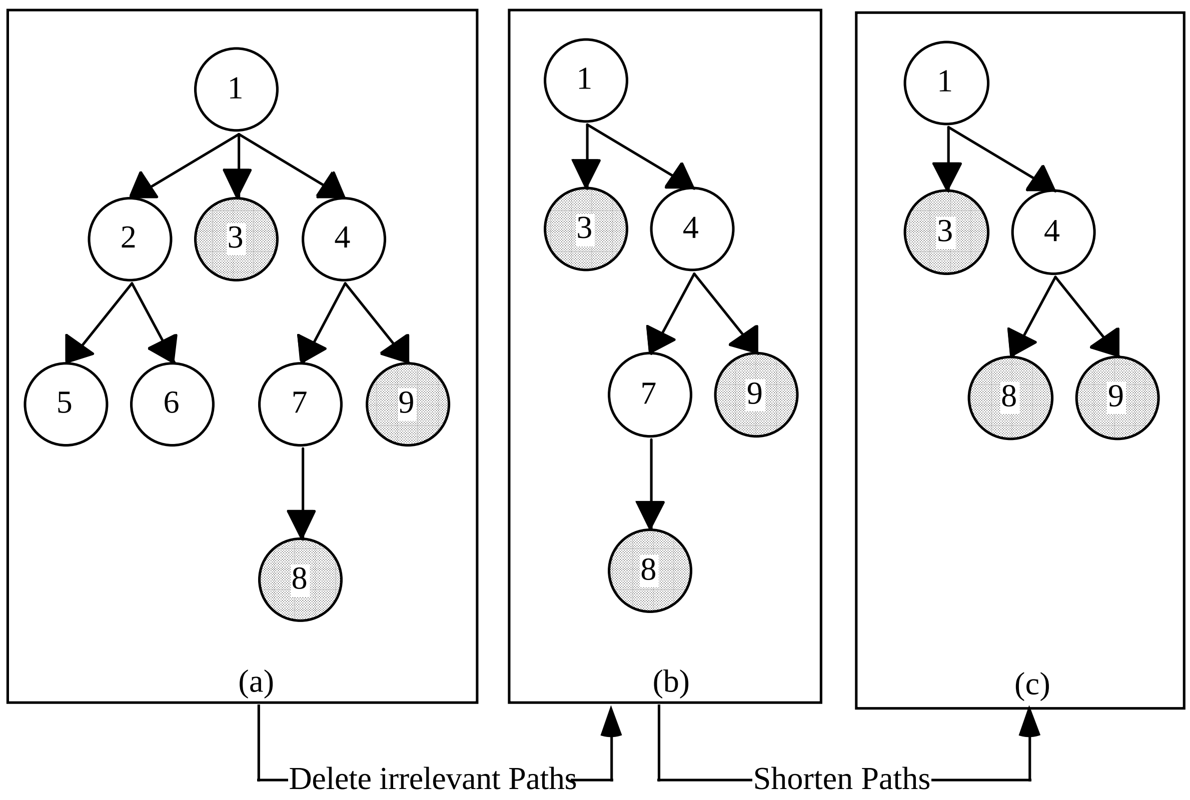

Figure 4 Optimizing the Rule Tree

The information retrieval engine selects the cases, and deletes their relevant 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.

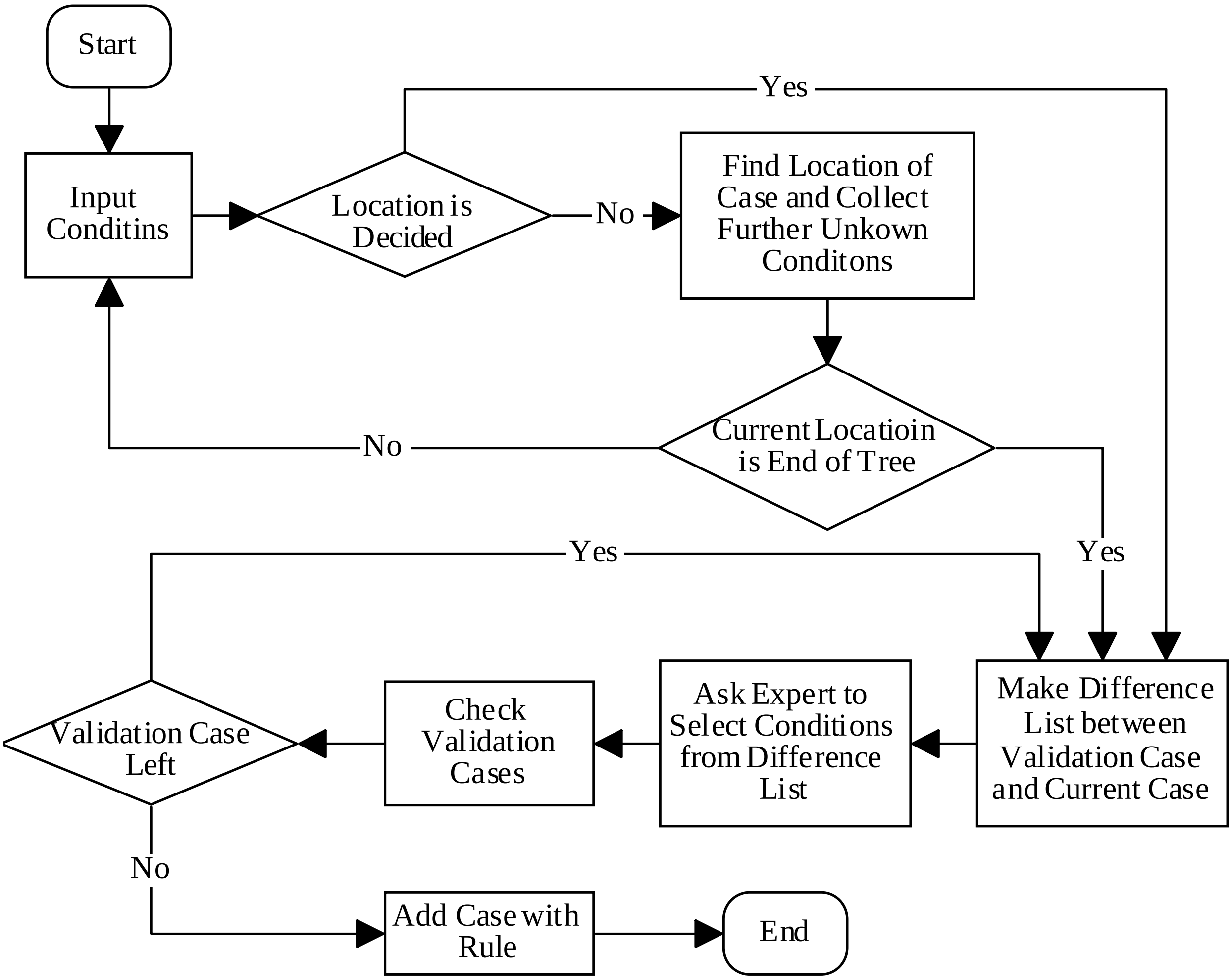

Figure 5 Knowledge Acquisition Process

The standard MCRDR knowledge acquisition method is used to maintain the knowledge.

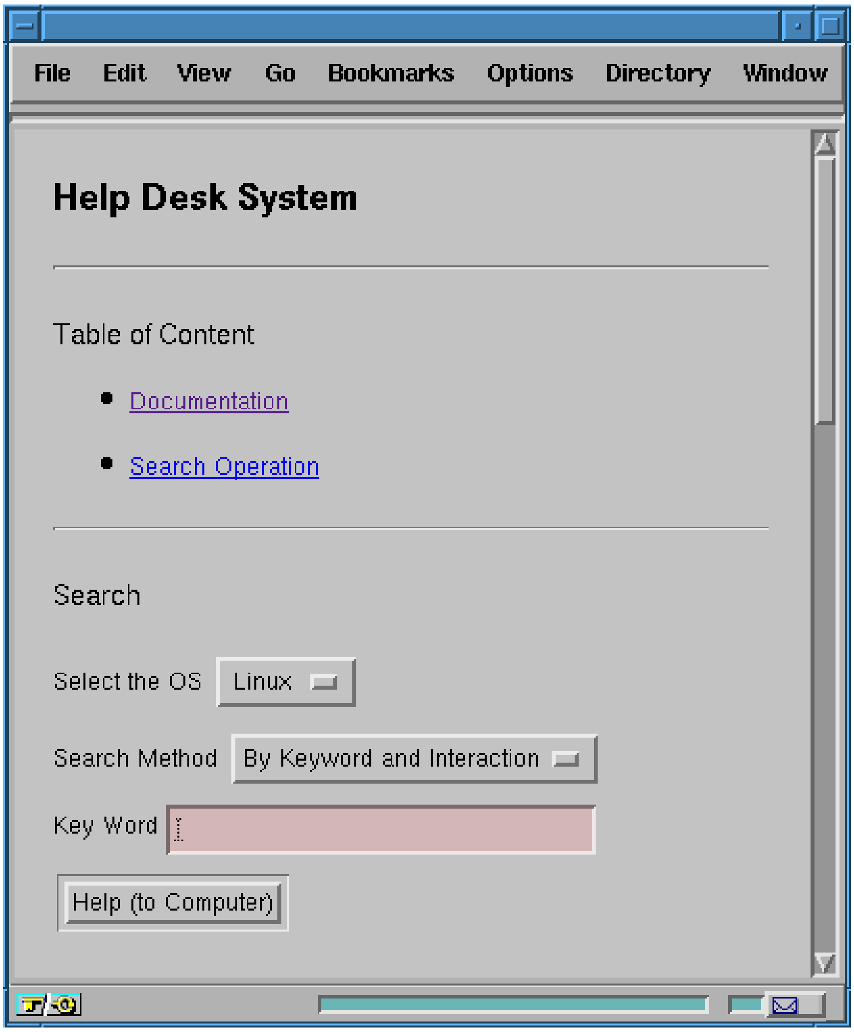

Figure 6

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.

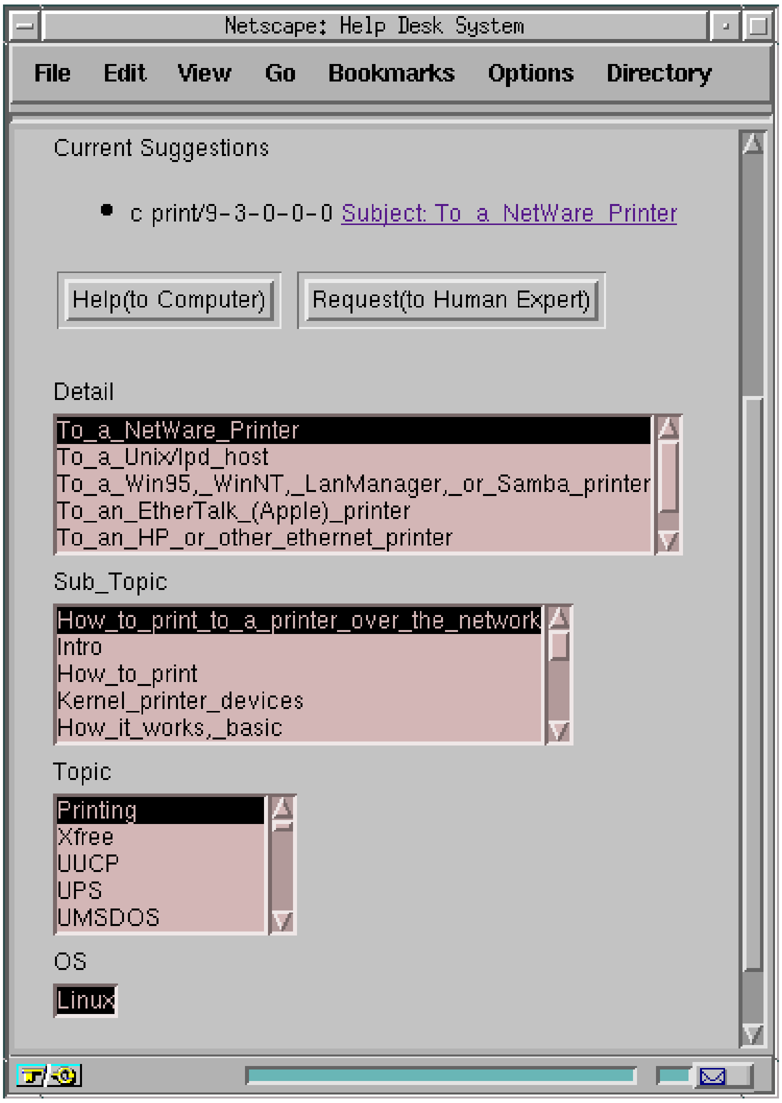

Figure 7

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 which may fit the user's request.

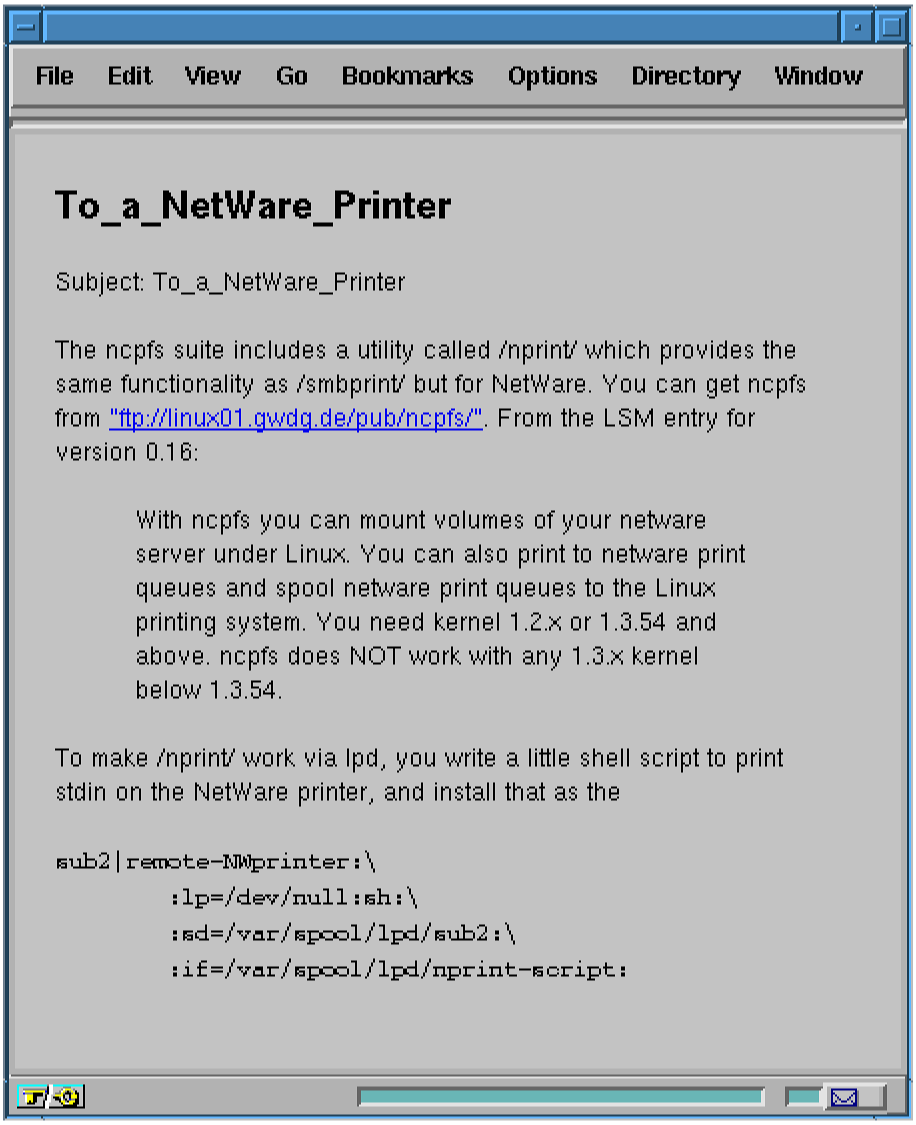

Figure 8

The selected documents are shown in this figure. Note that the result is just a standard WWW document, and we can also use hyper text links as commonly used in WWW pages. For example, "ftp://linux01.gwgd.de/pub/ncpfs/" in the figure is a link to another Internet resources.