Case-Based Reasoning : An Implemented Methodology

a

MCSE Design Project Report

by

Pradeep Raman
Auburn University

TABLE OF CONTENTS

ABSTRACT

Case Based Reasoning (CBR) is the process by which a computer system solves a given problem based on the knowledge gained from solving precedents in the past. It is a popular AI technique in the area of customer service or HelpDesks, whereby a system that uses CBR as its problem solving approach is utilized to solve most of the commonly occurring customer problems or questions. While the implementation techniques may vary, most CBR systems include the following five steps in some form or other: Representation where problem storage is handled, Retrieval where the closest-matching precedent is identified, Adaptation where a solution is generated from the retrieved problem, Validation where the accuracy of the solution is verified, and finally Update where the database is modified or updated with the information gained from this problem solving process.

This paper details the various implementation techniques for these five steps in the CBR process, while focusing on the methods used by one particular problem-solving application namely ***Smart USA.*** This system solves customer problems by filtering their problem descriptions through an "alias" table to handle abbreviations and non-key words and then matching the remaining keywords with those in the case database. It is an effective and user-friendly solution which successfully handles different descriptions of the same problem and allows for the case base to be built in free-formatted text. Major companies such as Compaq and NEC utilize a CBR based automated system to serve their customer service needs. These systems have played a significant role in reducing the overhead and the response times involved in the customer service departments of these organizations.

Chapter One: Introduction

1.1 Utilization of CBR :

Case Based Reasoning (CBR) is the method of solving problems based upon the knowledge gained from solving similar problems in the past. It is an Artificial Intelligence (AI) technique that could be very useful in any area where recurring problems or questions are the rule more often than the exception. As such, the customer service or HelpDesk department of an organization would be a good fit for this type of reasoning to be applied to the problem-solving process. Since most of the work would be repetitive in behavior i.e. when faced with a problem the problem solver would have to search for a precedent and then use the information available to present a possible solution to the customer, automation would be an ideal solution to these applications. If an organization was able to store all known problems/questions pertaining to their business in a computerized database and be able to search and retrieve from this database with reasonable success and ease, it would have successfully duplicated a customer service representative or HelpDesk personnel's primary tasks. Similarly CBR could be used as the problem solving approach in a variety of other applications since it closely resembles the thinking process of human beings and also improves or "learns" with each new case.

1.2 Different Types of CBR systems:

There are two types of CBR systems, namely those that perform semantic analysis of the given input and those that strictly perform syntactic analysis of the input. Obviously a CBR system that performs semantic analysis of the given input is much more impressive and thorough as it will be able to attain the “true meaning” of the description i.e. understand that the statement “Washington conducts talk with Beijing” really means that representatives from the United States and China conducted the talks and not the two cities themselves as that would be absurd. But this type of system, for all its advantages, is too difficult and expensive to develop into a practical problem-solving solution for organizations. On the other hand, systems that are restricted to performing syntactic analysis of the input descriptions, are practically of greater interest in applications where a reasonable rate of success is the ultimate goal. These type of systems would interpret the above example as a conversation between the two cities and therefore provide an invalid response or solution to the given input. But if the possibilities of encountering such ambiguity are small in the given domain or if most such relationships i.e. Washington = U.S. President, that exist within the domain, could be specified to the system, a CBR system performing syntactic analysis could be a powerful problem-solving tool to utilize. The idea is to develop a system in a short period of time that is easy to use and maintain, affordable and offers a reasonable rate of success in terms of problem solving in the given domain.

1.3 Goal of the Design Project :

This project report outlines one such system that uses CBR as its problem solving approach and focuses on primarily the syntactic analysis of input descriptions. The system, ***Smart USA,*** includes a simplistic attempt at a semantic analysis of the input along with a complete syntactic analysis of the same input. It is designed to be user-friendly i.e. easy to use both for the HelpDesk staff who will be responsible for creating and maintaining the case database as well as the end user or customer who will use the system to obtain a solution for their problem. All input is accepted in free text format i.e. the system does not restrict the amount or format of input received from the user in any manner, and the system is developed using relational database technology on popular PC and network operating platforms. The goal is to develop a system that is practical, inexpensive, and more sophisticated than a straightforward "spell-checker-type" syntactic analyzer, that would propose solutions for given problems with a reasonable rate of success. Furthermore, other AI learning techniques such as learning by parameter adjustment, are implemented to bolster the system's reasoning capabilities and performance on recurring cases.

Chapter Two: Literature Review and Project Description

2.1 What is Case-Based Reasoning ? :

Artificial Intelligence (AI), in its simplest form, is the study of teaching computers to solve problems in a manner similar to that of a rational human-being. Human reasoning can be viewed as being based on the collection of experiences and cases in a loosely organized dynamic memory [19]. More often than not, human beings solve problems by recollecting the knowledge gained from solving similar or related problems in the past in this dynamic memory. When faced with a new type of problem, we gain the knowledge required to solve the problem from other sources and then store this newly gained information in the dynamic memory to be utilized for future problem-solving endeavors. This method of solving problems based on the knowledge gained from solving precedents is formally known as Case-Based Reasoning (CBR) in the world of AI. It is the one AI technique, that when used in a comprehensive manner i.e. encompassing both the semantic as well as the syntactic meaning of cases, truly increases a computer's "experience" and moves it closer to human-like intelligence. CBR systems differ from the traditional systems in the sense that they focus on the transfer and reformation of knowledge about a particular domain and therefore are considered to be purely a problem-solving mechanism [13]. In theory, CBR can be considered as a five-step problem solving process [2] :

1. **Representation:** Representing case information i.e. building the knowledge base, and handling input of problems from the user.
2. **Retrieval:** Retrieving the closest-matching precedent to the current

problem from a database of cases.

3. **Adaptation:** Generating a solution using the information from the current problem description and the closest matching case.
4. **Validation:** Validating the solution through feedback from the user or the environment.
5. **Update:** Updating the case database, if appropriate, after analysis of validated response.

This chapter will detail the various possible implementations of these five steps while focusing on the implementation methods used in one particular system, namely ***Smart* USA**, which uses CBR as its problem solving approach. The goal of the system is to allow direct interaction with the end users and assist them in locating a solution for their problem. ***Smart* USA** focuses on the syntactic analysis of cases allowing decisions to be made on the closest-matching precedent which is derived from given problem descriptions.

2.2 The five steps of the CBR process :

2.2.1 Representation: This is the initial and probably the most significant phase of CBR and it deals with the storage of information i.e. construction of the knowledge base. It is typically implemented through input gained from a domain expert. A case can be defined as a problem description coupled with the description of each possible solution for the problem [4]. Furthermore, a case, as stored in the computer, must include all the relevant domain knowledge such as category, application type, the context in which it exists, and its expiration date if any, required to identify this case as a precedent to the current problem. These kinds of domain knowledge are stored as attributes to the cases and primarily used in the process of filtering out the unqualified cases in an attempt to return the best case[15]. The method of storage is obviously crucial to the success of the search and retrieval phase and hence to the remaining steps of CBR.

2.2.1.1 Semantic Vs Syntactic Analysis : If semantic analysis were to be the approach of a CBR system, cases would have to be stored in some form of semantic networks or grammar trees[25]. It would have to include the context in which the language is used since the same words used in different contexts might not constitute a precedent to the current problem. This type of case analysis would require a significant effort on the part of the domain expert or knowledge base builder towards the construction of the semantic network . The system would also have to include rules for exceptions and the idea of what constitutes a "closest-match" might be difficult to develop and implement. The advantages of such a system are the fact that it will certainly be more comprehensive and human-like in its overall reasoning process than a CBR system based on syntactic analysis. Semantic analysis based systems can accommodate both episodic knowledge i.e. information about episodes or experiences, as well as conceptual knowledge i.e. definitions and rules. [16]

On the other hand, there is a little more flexibility in utilizing syntactic analysis as the approach for a CBR system, in terms of knowledge base construction. Such a system needs to handle storage of the entire problem description and then decide on indexing techniques and which pieces of information to index so that a quick and successful search is facilitated . The key will be to balance the need to index enough information to uniquely identify a case and the need to limit the amount of information to be indexed so that real-time search and retrieval is plausible [5]. CBR systems performing strictly syntactic analysis of cases can store case descriptions as free-text format, as database records, in a summarized format, or as some combination of the above three methods.

2.2.1.2 Smart USA's approach to Representation : ***Smart* USA** stores the complete problem description in a free-text format and, additionally, builds a "key-words-filled" field from the complete description which can then be used by the search algorithms to identify this problem. The database contains a record for each unique problem which in turn consists of a full description field (provided by the domain expert) and a brief description field (generated by the system). The key to a high rate of case retrieval success is the presence of "key-words" i.e. words which help uniquely identify the problem [9], in the indexed field and therefore ***Smart* USA** generates the brief description field by stripping out all "non-key" words such as "the", "when", "how", and, furthermore, abbreviating words wherever appropriate, in the given full description. The system "learns" the words to be stripped out as well as the words to be abbreviated from the knowledge base builder as cases are entered into the database. This information, gained as each case is entered into the database, is then stored in a table containing REAL_WORD ==> ALIAS_WORD relationships. Once all the cases have been entered into the knowledge base, the alias table will contain a list of all abbreviations and "non-key" words related to the domain. This allows the system to parse all user problems through the alias table and thus generate a "key-words-filled" brief description for each given problem description. As the number of entries in the alias table increase, the system's knowledge is increased and better brief descriptions are generated. After a relatively small number of case entries are stored in the knowledge base, the domain expert will have to edit or enhance the system generated brief description field only rarely and thus a significant amount of user friendliness is gained from the database builder's point of view. A sample problem is illustrated in Figure 2.1. A default brief description is generated by the system based on prior knowledge i.e. existing entries in the alias table, and if the database builder edits this brief description, the system attempts to learn new REAL_WORD ==> ALIAS_WORD relationships.

Problem Description: (Entered by domain expert)
The car does not start. When I turn the key, nothing happens. The car has been unused for the past three months.

Brief Description: (Generated by System)
car not start turn key nothing happens unused 3 mos

Relevant Alias Table Entries: (“” means Null character or omit)

The := “”	does := “”	when := “”
I := “”	has := “”	been := “”
for := “”	past := “”	three := 3
months := mos		

Figure 2.1: A sample problem entry in the case database of ***Smart*** **USA**

Smart **USA** learns the aliases and abbreviations from the database builder and then utilizes this table globally to convert all past and future problem descriptions into the appropriate brief descriptions. In this manner, data integrity is maintained and multiple abbreviations for the same word are avoided. The brief description generated in the example of Figure 2.1 contains a large number of the key words required to uniquely identify this problem for a successful search and retrieval. In this manner ***Smart*** **USA** mimics a human being’s analytical process as most problems are resolved by memory triggers on key words included in the problem’s description [7]. This makes the system’s CBR approach both efficient and quite comprehensive at the same time.

2.2.2 Retrieval: The representation of the system builds the foundation for a good CBR system but the end product is only as good as the quality of its search and retrieval mechanisms. These mechanisms in turn are highly dependent on the indexing scheme(s) used in the system. Many CBR systems use the technique of preindexing, whereby indices are fixed prior to case-retrieval time [10]. While this method is highly successful when the domain or purpose of the system is well-defined, it lacks the flexibility and ability to be used by multiple case-based or other systems [24]. The principal reason for its popularity is its simplicity. The other approach is the automatic indexing scheme, in which free-formatted text is automatically indexed. For example the Battle Planner system automatically induces the discrimination tree of indices, producing a better balanced tree that enables more efficient case retrieval [8]. The pros and cons of these types of indexing schemes is a topic for further research.

2.2.2.1 Characteristics of the Retrieval stage : The retrieval algorithm must not only search the indexed fields as developed in the representation phase but must further apply some domain expert provided rules to the case database in order to capture exceptions or common-sense type information that human beings implement almost unconsciously [6]. A successful case retrieval must not only retrieve the correct precedent if present in the database, but must also be able to admit its lack of knowledge in the event of the absence of a case precedent. A good AI system, given the choice, should prefer to admit its inability to provide a solution to the given problem over proposing an incorrect solution. The domain expert provided rules must therefore also include a qualification standard which specifies a match between a precedent and the current problem [17].

2.2.2.2 Implementation Issues : Case retrieval implementations can vary widely in CBR systems depending on the goals of the reasoning process [21]. In the case of semantic analysis, methods such as nearest-neighbor algorithms, decision trees, or connections associative memories will be best suited to traverse the semantic networks and grammar trees. Backtracking capabilities, early recognition of mismatches, handling different flavors of the same problem, and exception handling will be some of the concepts that must be taken into account in semantic analysis based applications. Syntactic analysis of cases usually involves a simpler retrieval mechanism such as the greedy algorithm. ***Smart* USA** improves the response time by indexing each word in the brief description field and using that to match cases to the current problem description. As stated earlier, the user's input of the current problem description is also filtered through the same alias table to come up with a temporary brief description field which is compared against all the indexed brief description fields in the database. A case is determined to be a match if any of the words in the brief description derived from the user's input match the indexed words in the database. The matched cases are ranked in decreasing order in terms of the number of words matched and then filtered through some domain expert provided rules that take into account factors such as category type, user's job description, and problem expiration date to further modify the rankings of the matched cases. The case that retains the highest ranking after this analysis is termed the "closest-matching precedent" to the current problem and the associated solutions are then provided to the user for validation. An example of this type of filtration through expert-provided rules as implemented in ***Smart*** **USA** is provided in Figure 2.2. This figure illustrates an example where three cases are found in the database containing some of the key words as entered in the end user's problem description. ***Smart*** **USA** now filters these three cases through the expert provided rules for ranking and then returns the highest ranked case as the closest-matching precedent. Similar rules may be applied for giving precedence to cases entered in the past six months, or cases pertaining to the end user's job, etc. So ***Smart*** **USA** would rank the accounting related cases higher when processing multiple cases that match the problem description given by an accountant. An important point to note is that this sort of filtration takes effect only when the difference in the number of words matched between two or more problems is marginal.

Problem Description: (Entered by End User from Dept. B)
Salesmen's commission report is generating an error.

Possible precedents retrieved from the case base: (by System)

#1. slsman commsn rpt #1234 error	**(Dept. A)**
#2. slsman rpt #9999 error	**(Dept. B)**
#3. slsman commsn rpt #5678 causing error	**(Dept. A)**

Relevant Expert-provided Rules:
a) Add two points to the ranking of the problem which pertains to the same division/department as the end user.

b) Add one point to the ranking for each key word matched

Case returned as closest-matching precedent is Case #2.
Although case #2 had one less key word matched to the user's problem description, the fact that it pertained to the end user's department caused the system to rank it higher than the others. Note that if another case had three more key words matched than case #2, it would have ranked higher even if it did not pertain to the end user's department. So the department match is of importance in deciding between cases with a difference of at most 2 in the number of key words matched.

Figure 2.2: Filtration of matched cases through expert rules in ***Smart*** **USA**

The retrieval mechanism used by ***Smart*** **USA** facilitates the grouping of multiple descriptions of the same problem thus proposing the same solution to all of them. Since a single word-to-word match in the brief description fields qualifies a case to move on to the next step i.e. filtering through the domain-relevant rules, the absence of some of the "key-words" in the user's description does not eliminate the possibility of a successful retrieval. In a typical customer support environment it is typical to receive differently worded versions of the same problem and therefore it becomes mandatory for an automated system to be equipped to handle these types of requests [20]. As long as all the users' problems filter through the domain rules in the same manner, ***Smart* USA** will recognize them as being the same problem simply if they all return the same record from the case database. The use of the alias table allows for the words such as "reports", "report", "rpt", and "rpts" to be processed as the same word in the search algorithm and thus increases the system's retrieval success rate without any additional burden or information to be carried. Figure 2.3 illustrates these concepts through an example. The example shows the structure of the case as stored in the knowledge base and a problem description as entered by the customer.

Problem Description: (Entered by domain expert)
The car does not start. When I turn the key, nothing happens. The car has been unused for the past three months.

Brief Description: (Generated by System and stored in case base)
car not start turn key nothing happens unused 3 mos

Current Problem Description: (Entered by customer)
I am having trouble with my car. It does not start at all and does not even make an attempt when I turn the ignition key. It has not been started in over 6 months.

Brief Description: (Generated by System and used in search for closest-matching precedent)
trouble car not start turn ignition key not started 6 mos

Figure 2.3: Details of Retrieval mechanism in ***Smart* USA**

***Smart* USA** begins the search by first converting the user's current problem description into a brief description using the same alias table as used to build the case database. Now each word in this brief description is used to search cases in the database. In the event the user's description generates more than one brief description (based on length), this process is repeated for each word in each brief description. A list of all case numbers that have at least one word-to-word match is generated, and then, used to filter through the domain-expert-provided rules and ranked to produce the closest matching precedent.

2.2.3 Adaptation: This phase of CBR deals with the proposal of a solution to the current problem. In most systems this phase is implemented in a simple manner in the sense that the solution of the retrieved "closest-matching precedent" is presented as the possible solution to the user's problem [3]. Nevertheless, in theory, significant intelligence can be added to the system at this stage of the process. For example, before proposing the solution, the computer may analyze factors such as the success rate of the solution in the past, alternative solutions, and compatibility with the current context and user. If any of these factors return a negative response, consideration must be given to override the retrieval phase and inform the user about the lack of a proper solution at this time. Such logic can be quite complex to implement and in many cases provide little gain.

2.2.3.1 Implementation : ***Smart*** **USA** takes a simplistic approach to adaptation by returning the highest ranking solution of the closest-matching precedent. It should be noted that each case may consist of more than one solution and the solution that has been the most successful historically is assigned the highest ranking and is therefore suggested first to the user. The user can, then, decide to accept the solution or check for the next solution. The system would suggest all documented solutions before informing the user that outside help is needed and the suggestions are made in decreasing order of ranking.

2.2.4 Validation : The proposed solution is validated through feedback from the user or the environment in this phase of CBR. This is part of the "learning" process of the system as consistently unsuccessful solutions must be weeded out from future analysis and good solutions must be rewarded by placing them at the top of the solution list.

2.2.4.1 Implementation Issues : Also, care must be taken to require as few keystrokes or mouse clicks as possible from the user in the form of validating responses in order to maintain the user friendliness of the system. As mentioned in the adaptation phase, ***Smart*** **USA** uses a ranking system for all solutions of each problem whereby the historically most successful solution for a problem is given the highest ranking. When a solution to the current problem is exercised, the user is asked to provide feedback by answering the following questions :

1. Is the assumed precedent truly related to the current problem?
2. If so does Solution A work? If not does Solution B work? . . .

The answers to these queries are passed on for processing in the Update phase of the CBR process.

2.2.5 Update : The final phase of CBR determines the performance of the system. After the discovery of an acceptable solution, the CBR system must determine whether or not this case is sufficiently different from others in the knowledge base to warrant its addition into its case database [22]. It receives the user's responses to the queries posed in the validation phase and makes any enhancements or corrections wherever appropriate so that the next occurrence of the same problem may be served by more accurate results. In addition, any new information is added to the case database and outdated information is weeded out. Learning by parameter adjustment, a machine learning technique whereby a weight is associated to each result depending on its quality or success [6], is thus implemented in this stage. For example, assume the described problem has three valid solutions. Solution A fails to solve this particular instance of the problem; Solution B succeeds in solving the problem; and Solution C has not been applied. In this situation, the weights associated with these solutions are manipulated in the following manner:

A constant value N is added to the weight of B and the same value N is subtracted from the weight of A. The weight of Solution C remains unchanged.

The value of N must depend on the extent of the failure or success from the system's point of view. In this manner, the next time the same case is processed, the solution with the highest ranking i.e. the historically most successful solution is presented to the end user first. All solutions are provided with an initial weight or ranking (as determined by the domain-expert) when they are entered into the case base. Thereafter ***Smart*** **USA** maintains the success rate of all suggested solutions and by adjusting the associated weights, the system "learns" the better solutions after each processing iteration. In this manner, feedback is provided to the system maintenance group so that any changes to the database of cases may be made if necessary.

2.2.6 Summary of CBR steps : In practice, a CBR system may implement these five-steps in varying degrees of complexity and depth. While the methods used in each of these five-steps may differ, the presence of all five phases in a successful CBR system in some form is inevitable. The combination of these five steps builds a powerful and user-friendly analytical tool that is most suitable for environments where similar or related problems are faced on a repeated basis [14]. CBR systems, which base their intelligence and inferencing on known cases rather than rules, are considered to be more effective than rule-based or other types of expert systems. Researchers argue that a) the concrete examples presented by cases are easier for users to comprehend and apply than complex chains of reasoning generated by rule-based or model-based systems and b) record-like representation of cases in a CBR system allows for easy storage, entry, and update in relational databases [18]. The presence of validation and update steps based on feedback from the users increases the machine's "experience" and incorporates knowledge acquisition into the day-to-day use of a CBR system [11]. Figure 2.4, shown below, provides a step-by-step walk through of the entire reasoning process of ***Smart* USA.**

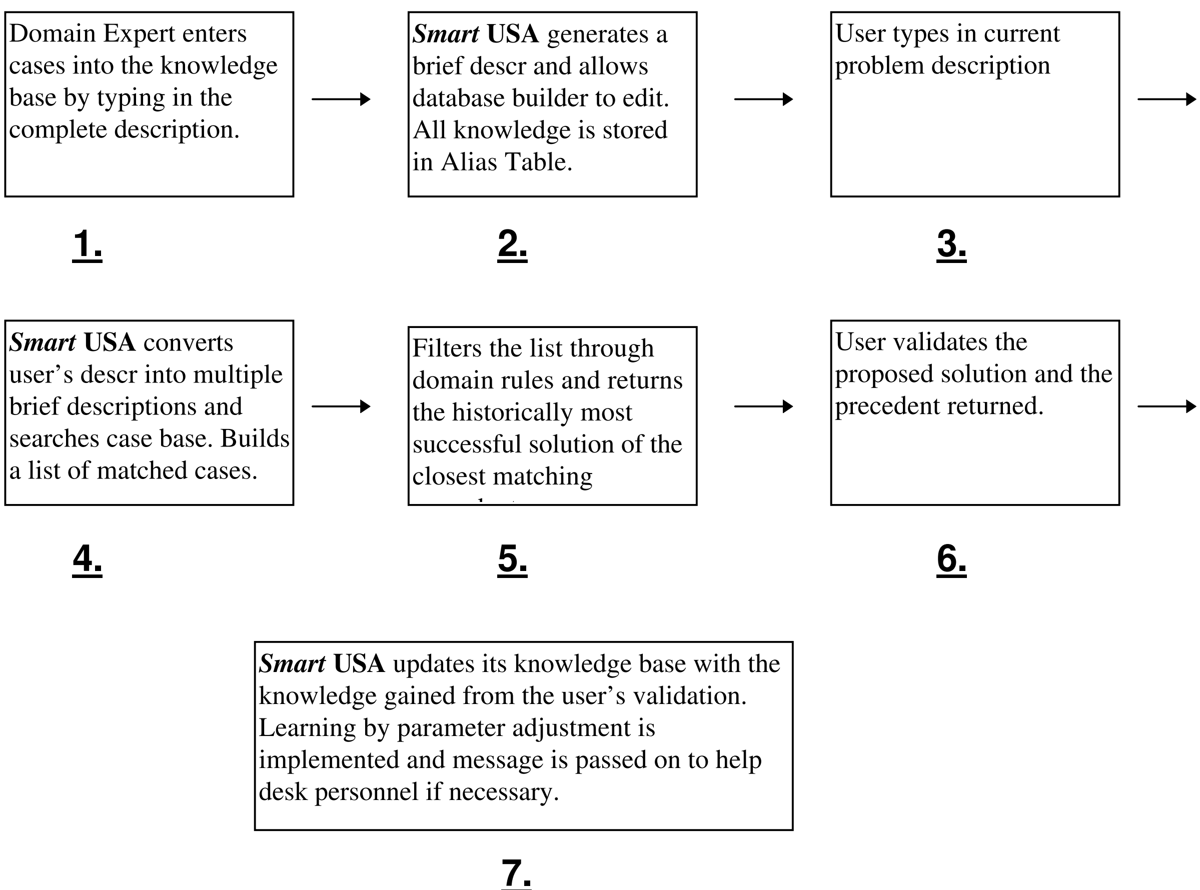

*Figure 2.4: Steps in the Analytical process of a CBR system- **Smart*** **USA**

2.3 Commercial CBR applications :

Although a wide range of potential applications have been explored, commercial CBR applications have focused mainly on using case retrieval in a decision support environment [20]. Customer service helpdesk application is a typical example of the commercial application of CBR [21]. Most customer service applications, such as ***Smart*** **USA**, tend to focus on retrieving the closest-matching precedent to the given problem without extensive adaptation logic being implemented. Since most help desk personnel report about 60% to 70% of their time being spent on solving repeat problems [21], there is a valid argument for the use of a CBR approach where data acquisition and distribution may be central to the daily operations.

2.3.1 Implemented Systems : CBR systems based on semantic analysis are few in number in the commercial arena. The existing ones are limited in the domain and the number of cases they can handle efficiently, due to the obvious physical limitations in the number of concepts and episodes they can store. TOLTEC, a case-based planner performing semantic analysis, was developed in 1993 to solve complex manufacturing problems by storing old plans in a dynamic memory organization [23]. Its domain is limited to the different manufacturing processes and the system uses a complex indexing scheme and retrieval algorithm to model its reasoning after the way human experts reason. Caper is another such planning system utilizing CBR as its plan generating approach [12]. Its domain is limited to car assembly and transportation logistics. It has the capability of breaking a given goal into subplans and retrieving multiple cases to achieve these subgoals and then merging them into a composite plan that solves the global problem. The number of CBR systems based on syntactic processing available commercially is much larger as this type of analysis is more suitable to adaptation in many different domains. The indexing scheme and search and retrieval mechanism will work irrespective of the context or “true meaning” of the cases. There are a few companies that market CBR based application development tools that allow IS professionals to develop a system that utilizes, at least partially, a CBR approach to problem solving. The indexing schemes are flexible in these types of systems, i.e. the developer chooses the features to be indexed, but the retrieval and adaptation capabilities are minimal at best. It provides default database schemas for case representation, graphical editing of cases, importing from and exporting to relational databases, and uses a pre-defined problem-solving flow for decision support. This is typical for most development tools available in the market today [21]. Bendata Management Systems’ Helpdesk Expert Automation Tool (HEAT) is one such system that uses a combination of CBR and simplistic browse and lookup techniques in a PC LAN environment to store cases and propose solutions. HEAT provides a graphical user-interface thus allowing help desk personnel to quickly browse through related problems and propose solutions. It is not a robust CBR system as there is not much of a search and retrieval mechanism besides categorizing cases. The system does not process user's problem descriptions for matches but simply allows for a friendly interface for help desk personnel to retrieve the precedent.

Compaq Computer's SMART system is an integrated call-tracking and problem resolution system which digs deeper into the CBR approach for its problem solving methodology [1]. It is utilized by the company's customer service department. It generates promising solutions to the users' problem by using the closest-matching precedent approach. SMART, which runs on a 386-based PC under the Microsoft Windows environment, has produced a significant increase, from 50% to 87%, in the percentage of user problems resolved on the first call [1].

Chapter Three: Specific Examples

3.1 Alias Table Construction :

When Smart USA is used for the first time i.e. before the first case has been entered into the case database, the alias table is empty. Entries are added to it as illustrated by Figure 3.1.

Problem Description: (Entered by the database builder)
Report BSR600 is giving a security violation error message when run from the menu.

Brief Description: (Generated by the system)
Report BSR600 is giving a security violation error message w

Modified Brief Description: (After being edited by the database builder)
rpt BSR600 security violation err msg

Entries in the Alias Table: (After the above problem is saved)

Report ::= rpt	is ::= “”	giving ::= “”
a ::= “”	error ::= err	message ::= msg
when ::= “”	run ::= “”	from ::= “”
the ::= “”	menu ::= err	

*Figure 3.1: Alias table and case base creation in **Smart USA***

Note that, in Figure 3.1, since the brief description field is of fixed length (60 characters), the initial description generated by the system simply pulls down the first sixty characters. Thereafter, as the case base builder deletes and abbreviates words, all the words in the problem description are processed and appropriate entries added to the alias table. Note also in Figure 3.1, that the words “BSR600”, “security”, and “violation” do not create entries in the alias table since they remain unmodified by the case base builder.

3.2 Search and Retrieval Example :

Once the problems have been entered into the case base, ***Smart USA*** is ready for use by the end users in search of answers to their problems. The user is provided with a data entry screen to describe their problem in free-formatted text. This full description is then parsed through the alias table and a new key-words-filled description is obtained in the following manner: Using the alias table of Figure 3.1, all occurrences of “is”, “a”, “giving”, “when”, “run”, “from”, “the”, and “menu” would be eliminated and all occurrences of “report”, “error”, and “message” would be abbreviated by the parsing process which creates the key-words-filled description. Note that the words “BSR600” and “security” and “violation” would appear unchanged in the key-words-filled description since there are no entries for them in the alias table. ***Smart USA*** now searches for the occurrence of each word in this key-words-filled description in the case database against the Brief Description field (Figure 3.1). Two sample user descriptions and the corresponding key-words-filled descriptions as created by the system are shown in Figure 3.2. For the sake of convenience, they both describe the problem documented in the case base as shown in Figure 3.1.

User #1's Description (in free-format text):
Hi! there. I am getting a big box with an error message inside saying that I have violated the security of the system. This is happening when I run the rpt BSR600.

Key-words filled description (created by Smart USA):
err msg security violated system happening rpt BSR600

User #2's Description (in free-format text):
error message about sec violation of the system when running rpt BSR600.

Key-words filled description (created by Smart USA):
err msg sec violation system rpt BSR600

Figure 3.2: Two sample user problem descriptions and corresponding key-words filled descriptions created by Smart USA .

3.2.1 Building List of Possible precedents : In Figure 3.2, the first user's lengthy and conversational description of the problem is stripped and only the key words are retained i.e. either words that had abbreviations in the alias table or did not have an entry in the alias table at all. The assumption is made in this example that the case base is comprehensive enough that entries for words such as "I", "am", and "Hi!" are available in the alias table with a null equivalence. Thus as more entries are made into the alias table, this parsing process results in a better key-words-filled description. Now each word in the key-words-filled description is searched for in the brief descriptions stored in the case database and it is obvious that the descriptions obtained in Figure 3.2 will both return the problem described in Figure 3.1 as the closest-matching precedent. Note that although "violated" in User #1's description and "sec" (an abbreviation for security) in User #2's description are not found in the problem's brief description (Figure 3.1), the system does not discard the case from being a precedent. It is important to note that not all words found in the case's brief description have to appear in the user's description, rather all cases containing word matches are compiled into a list and passed on to the next phase of processing.

<u>3.2.2 Extracting the Closest-matching precedent from list :</u> Once a list of all possible precedents has been compiled, certain expert-provided rules are applied to alter the ranking based on the number of words matched. These rules generally affect the outcome only when there are two or more cases with a small difference in the number of words matched. In that event, the case that satisfies the expert-provided rules would emerge on top. These rules take into consideration factors such as the user's department/division, the age of the cases as entered into the database, any specific references to the user's plant or location, etc. They are predetermined by the domain expert and hard coded into the system. For example, assume that the same error message is encountered in a Billing application and in a Order-Entry application. Naturally, these would generate two independent cases in the knowledge base, since their solutions, dates of occurrence, expiration date, and other such properties may differ. Now, if an end-user, working in the Billing department, requests a solution for this error message, the system will return both the cases stored in the knowledge

base that match this problem as the possible precedents, namely, the cases entered for the Billing and Order-Entry applications respectively. The expert provided rules would require the system to increase the weight associated with the case entered for the Billing application by a fixed number since the end user works in the Billing department. If the number of words matched were the same or the difference was less than the fixed number, for these two cases, the Billing problem would be returned as the closest-matching precedent since it would now rank higher after the expert-provided rules have been applied. The rules themselves as well as the fixed number involved are provided by the domain expert. Note once again, that this would only happen if the differences in the number of word matches were less than the fixed number. So if the fixed number were two and a completely unrelated Billing problem contained one of the words present in the user's description while another Sales Service problem in the case base contained seven word matches against the user's description, then despite the user being a Billing person, the system would return the Sales Service problem as the closest-matching precedent. This is because an increase in the Billing problem's ranking by two gives it a ranking of three whereas the Sales Service problem has a ranking of seven thus making it a better precedent. In the example of Figure 2.2, the chosen precedent is case #2 i.e. the case related to Department B, in the list of possible precedents.

3.2.3 Presenting solutions based on precedent : Once the precedent has been identified, all its possible solutions, as stored in the case database, are gathered along with their associated rankings. The solutions are proposed to the end user in decreasing order of ranking until a positive response is received from the user indicating that the problem has been solved. If one of the solutions elicits this positive response, then the technique of learning by parameter adjustment is applied to modify the rankings of all proposed solutions. The rankings of the solutions which were proposed to the end user but failed to elicit a positive response are decreased by a fixed number and the solution that succeeded in doing so is given an increase in ranking. The solutions that were not proposed to the end user before the receipt of the positive response remain unchanged. In this manner ***Smart USA*** maintains the list of solutions for each problem ranked by their historical success rates. The user's feedback triggers the update phase of the CBR process and is the measure of a particular solution's ranking. The initial rankings for each solution as well as the weights (or fixed numbers) involved are provided by domain expert at the time the case database is built. In the event that none of the solutions elicit a positive response from the user, the rankings remain unchanged and the user is asked to await human intervention. The customer service staff who comes into the picture at this point is armed with the knowledge about all proposed solutions and therefore has a head start on the problem-solving process. Figure 3.3 illustrates the process of proposing multiple solutions, and updating the respective rankings after receiving feedback from the user. All solutions are assigned an initial ranking which is bolded and enclosed in parentheses in the solution list of Figure 3.3.

Problem Description: (Entered by End User)

I am getting a general protection fault error in KERNEL.EXE when I was in the middle of a MS-Word session. Running MS-Windows 3.1 on HPVectra 486 PC.

Closest-matching precedent retrieved : (by System)

Any type of general protection fault error message in MS-Windows 3.1 operating system.

Solution list : (in decreasing order by rank, as proposed to user)

*a)***(20)** A quick solution is to shut down any other applications that are currently running and try again. If you are unable to do this or if this does not work then try the other solutions.

*b)***(15)** Go to the 386-Enhanced part of the Control Panel and change your virtual memory settings to use a Temporary swap file with the maximum recommended size. Then exit MS-Windows and reenter and try again.

*c)***(10)** Exit MS-Windows. Check the available base memory by typing MEM at the DOS prompt. Try to increase this number by moving more programs into the Upper Memory area. You can optimize it by running the Memmaker program. Type MEMMAKER to do this and select the "optimizing memory for running under Windows" option. REM out any TSR's if they are not needed. Then reboot PC and try again.

*d)***(5)** In the event of constantly reoccurring memory problems, it is best to increase the amount of memory on your machine. For running new software, it is advisable to have a minimum of 8M and preferrably 12M of memory in your PC.
NOTE: Make sure the memory sims are compatible with your hardware. Call the MIS dept at 912-966-4286 if you need help.

User's Responses to proposed solutions & the new rankings:

a) DID NOT WORK **(18)**
b) DID NOT WORK **(13)**
c) WORKED **(12)**
d) NOT PROPOSED TO USER **(5)**

Figure 3.3: Proposal of multiple solutions and Rankings Updates in ***Smart*** **USA**

The amount of decrease and increase in the rankings of unsuccessful and successful solutions respectively is provided by the domain expert. Figures 3.4 and 3.5 show the results from two sample runs of ***Smart* USA.**

Problem Description: (Entered by user in Cust. Service Dept.)
I sent a report to the printer and it is not printing. The last time I printed was on January 12, 1995. Not getting any error messages either.

Brief Description: (Generated by System and used to search)
rpt prtr not prtg prtd jan 12, 1995 err msg

List of possible precedents: (Retrieved by System)

1. *Full Descr:* january report of BSR601 generating error due to change in year. Message is : Invalid Date value
Brief Descr: jan rpt BSR601 err chg msg invalid date
Department: Operations   Expiration Date: None
words matched: 4

2. *Full Descr:* sent output to printer and it is not printing
Brief Descr: output prtr not prtg
Department: Sales & Customer Service   Expiration Date: None
words matched: 3

3. *Full Descr:* Network printer not accessible. Errors appearing for all clients attempting to print. Cannot PING either.
Brief Descr: network prtr not err clients prt cant ping
Department: Communications   Expiration Date: None
words matched: 3

List of precedents after applying expert rules:

1. *Brief Descr:* output prtr not prtg
Ranking: 8 (3 + 5 for dept match + 0 for furthest expiration date)
2. *Brief Descr:* jan rpt BSR601 err chg msg invalid date
Ranking: 4 (4 + 0 for dept match + 0 for furthest expiration date)
3. *Brief Descr:* network prtr not err clients prt cant ping
Ranking: 3 (3 + 0 for dept match + 0 for furthest expiration date)

Case returned as closest-matching precedent & its solution list:
Case Descr: sent output to printer and it is not printing
Solution List with Rankings in parentheses:

1. (20) Reset printer and try again. If other people are getting output then check your program and display output to the screen. If it displays on the screen OK, check to ensure that the cables are plugged in correctly.
2. (10) If on the HP3000 network, type <STARTSPOOL dev#> where dev# is the device number of the printer. Then check for output. If on a PC network and you are running MS-Windows, go to Print Manager in Control Panel and restart printing if the printer is in stalled status. Check for any output again.
3. (1) Call MIS Technical Services at 912-966-4288.

User's Feedback and Updated Rankings for the Solutions:

1. DID NOT WORK (18)
2. WORKED (12)
3. NOT PROPOSED (1)

Figure 3.4 : A sample run of Smart USA

Problem Description: (Entered by user in Operations/Data Entry Dept.)
I am getting an out of balance error in the EDI customer shipments on my daily sales report.

Brief Description: (Generated by System and used to search)
bal err edi cust shipments dly sls rpt

List of possible precedents: (Retrieved by System)

1. *Full Descr:* how to process a new edi customer shipment type
Brief Descr: new edi cust shipment type
Department: Sales Service Expiration Date: 12/31/99
words matched: 3

2. *Full Descr:* Out of balance error in monthly Sales statistics
Brief Descr: bal err mo sls statistics
Department: Operations Expiration Date: None
words matched: 3

List of precedents after applying expert rules:

1. *Brief Descr:* bal err mo sls statistics
Ranking: 5 (3 + 0 for dept match + 2 for furthest expiration date)
2. *Brief Descr:* new edi cust shipment type
Ranking: 3 (3 + 0 for dept match + 0 for furthest expiration date)

Case returned as closest-matching precedent & its solution list:
Case Descr: Out of balance error in monthly Sales statistics
Solution List with Rankings in parentheses:

1. (20) Run the correction job on the Sales History system and then rerun the monthly Sales statistics report.
2. (1) Call Programmer/Analyst responsible and hold off all other monthly processing until a solution is given.

User's Feedback and Updated Rankings for the Solutions:
INCORRECT CASE RETURNED AS PRECEDENT .
Therefore the solution rankings remain the same.
Message is displayed to user to call the MIS Helpdesk at 912-966-4286.

Figure 3.5 : A sample run of Smart USA

Chapter Four: Conclusion

4.1 Implementation Environment of Smart USA :

***Smart* USA**, is developed using the PC database product Microsoft FoxPro for Windows and runs in a client-server environment under the MS-Windows and Novell Netware operating environments. Its primary benefit would be to eliminate or drastically reduce the number of calls placed to the help desk personnel of an organization. Union Camp Corporation, where this system was deployed, is providing a self-assisting tool to the end users in an attempt to funnel help desk resources into other much needed areas. The system has been used by the help desk personnel for some time now and has amply demonstrated its capability for increasing productivity. Problems that used to be resolved in tens of minutes prior to the system's use are now being resolved in a matter of seconds by presenting it to the CBR engine. The company expects a 60% to 70% decrease in the number of calls from end users once deployment is complete due to the fact that most calls are similar in nature (see Appendix A). A survey of the Help Desk and IS personnel reveals the significance and effectiveness of ***Smart* USA**'s capability to handle multiple interpretations of the same problem and the "word-aliasing" concept (also see Appendix A). While certainly not comprehensive, in the sense that not all queries may be resolved successfully by the system, ***Smart* USA** is certainly a practical, user-friendly, and intelligent solution to handling large volumes of cases, many of which may be repetitive. The technique of "word-aliasing" serves a dual purpose in the sense that it allows for the system to process two different user descriptions in the same manner and also recognizes the precedent of a given problem as all non-key words are stripped out prior to the processing phase.

4.2 Smart USA Vs Other CBR based systems :

Smart **USA** compares favorably with the other CBR-based customer support systems in use today. The significant advantage of using ***Smart*** **USA** over the other systems is its simplified method of handling multiple descriptions of the same problem. While most customer-support systems have the ability to process two different descriptions of the same problem and return the same case as the closest-matching precedent, the techniques used require a significant amount of work and time on the part of the case base builder and maintenance group. NEC Corporation's customer support system, SQUAD, [21] contains a pre-determined list of attributes, a list of possible values for each attribute, a thesaurus of keywords to be the value of each attribute and a hierarchy formation that defines relationships among keywords and attributes. This hierarchy is then mapped into a relational database which is used to store the cases. This strategy, while certainly effective, is a lot more cumbersome to develop and maintain compared to ***Smart*** **USA**'s strategy of maintaining a dynamic table of aliases and abbreviation for words used in each case. There is also a lot more overhead and less flexibility in terms of handling free-formatted text in SQUAD. Compaq's SMART [1] is used by the company's support staff, who interpret the end user's problem and enter the problem into the system in a familiar manner. There is also not any automated validation or update phase as the staff members sort through the "similar cases" returned by SMART and use them to resolve the problem. ***Smart* USA** on the other hand allows the end user to directly input the problem in free-formatted text and furthermore attempts to resolve the problem by presenting a single precedent case which, if accepted as such by the user, is then followed by the different solutions available for that case in the appropriate order. Bendata's HEAT is less sophisticated than that, as it simply allows the helpdesk staff to search for a precedent using graphical browsing tools and categorized filing methods.

4.3 The future of Case-Based Reasoning :

More sophisticated approaches to case representation and indexing are in the near future for CBR system development methods and tools. They will also allow the creation of problem-solving applications that utilize a combination of cases, rules, and models in their dynamic memory [26]. As companies begin to think and operate on a global basis, it will become increasingly crucial for all information to be available to personnel throughout the organization and for personnel to find more efficient approaches to problem solving. CBR systems facilitate the smooth transfer of knowledge throughout the company and ensure that solutions are not regenerated from scratch and proven reasoning methods not recreated. Research in information filtering and retrieval [5] will lead to automatic generation of indexing and case representation methods from unstructured text. Case-based reasoning when combined with information retrieval will replace the reference manuals and diagnostic episodes or experiences available to a customer service representative today and ably serve as an automated helpdesk to a global organization [21].

References

1. Acorn, T., and Walden, S., “SMART: Support Management Automated Reasoning Technology for Compaq Customer Service,” *Proceedings of the Fourth Innovative Applications of Artificial Intelligence Conference*, AAAI Press, 1992.

2. Allen, B.P., “Case-Based Reasoning: Business Applications,” *Communications of the ACM*, Vol. 37, No. 3, March 1994, pp. 40-42.

3. Alterman, R. “An adaptive planner,” *Proceedings of the AAAI*, 1986, pp. 65-69.

4. Barletta, R., “Case-based reasoning and information retrieval: Opportunities for technology sharing,” *IEEE Expert*, Vol. 8, No. 6, December 1993, pp. 2-4.

5. Belkin, N.J., and Croft, W.B., “Information filtering and information retrieval: Two sides of the same coin,” *Communications of the ACM*, Vol. 35, No. 12, December 1992.

6. Carbonell, J.G., “Learning by analogy: Formulating and generalizing plans from past experience,” *Machine Learning, An Artificial Intelligence Approach*, R.S. Michalski, J.G.Carbonell, and T.M. Mitchell, Eds. Palo Alto, California, Tioga, 1983.

7. Freeman, M., “HSTDEK: Developing a Methodology for Construction of Large-Scale, Multi-use Knowledge Bases,” *NASA Conference Publication 2492*, NASA/Marshall Space Flight Center, 1987.

8. Goodman, M., “CBR in Battle Planning,” *Proceedings of the DARPA Case -Based Reasoning Workshop*, Morgan Kaufmann, San Mateo, California, 1989, pp. 264-269.

9. Hammond, C., “Case-Based Planning: An Integrated Theory of Planning, Learning, and Memory,” *Ph.D. Thesis*, Yale University, 1986.

10. Hammond, K.J., “Case-Based Planning: A Framework for Planning from Experience,” *Cognitive Science,* Vol. 14, 1990, pp. 384-443.

11. Hennessy, D., and Hinkle, D., “Applying Case-Based Reasoning to Autoclave Loading,” *IEEE Expert*, October 1992.

12. Kettler, B.P., Hendler, J.A., Andersen, W.A., and Evett, M.P., “Massively Parallel Support for Case-Based Planning,” *IEEE Expert,* Vol. 9, No.1, 1994, pp. 8-14.

13. Kitano, H., Shibata, A., Shimazu, H., Kajihara, J., and Sato, A., "Case-Method: A Methodology for Building Large-Scale Case-Based Systems," *Proceedings of the Tenth AAAI Conference*, July 12-16 1992, AAAI Press, San Jose, California, pp. 303-308

14. Kitano, H., Shibata, A., Shimazu, H., Kajihara, J., and Sato, A., "Building large-scale and corporate-wide case-based systems: Integration of organizational and machine executable algorithms," *Proceedings of the Tenth AAAI Conference*, AAAI, San Jose, California, 1992.

15. Kolodner, J.L., "Judging Which is the Best Case for a Case-Based Reasoner," *Proceedings of the Case-Based Reasoning Workshop*, Morgan Kaufmann, San Mateo, California, 1989, pp. 77-81.

16. Redmond, M., "Distributed Cases for Case-Based Reasoning: Facilitating Uses of Multiple Cases," *Proceedings of the Eighth National Conference on Artificial Intelligence*, MIT Press, Cambridge, Massachusetts, 1990, pp. 304-309.

17. Riesbeck, C., and Schank, R., "Inside Case-Based Reasoning," Lawrence Erlbaum Associates, 1989.

18. Riesbeck, C.K., "An interface for case-based knowledge acquisition," *Proceedings of the DARPA Workshop on Case-Based Reasoning*, DARPA, Washington, D.C., 1988.

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

20. Shimazu, H., Kitano, H., and Shibata, A., "Retrieving Cases from Relational Database: Another Stride Towards Corporate-Wide Case Based Systems," *Proceedings of the IJCAI-93*, 1993.

21. Simoudis, E., "Using case-based retrieval for customer technical support," *IEEE Expert*, Vol. 7, No. 5, October 1992.

22. Stanfill, C., and Waltz, D., "Toward Memory-Based Reasoning," *Communications of the ACM*, Vol. 29, No. 12, December 1986, pp. 1213-1228.

23. Tsatsoulis, C., and Kashyap, R.L., "Case-Based Reasoning and Learning in Manufacturing with the TOLTEC Planner," *IEEE Expert*, Vol. 23, No. 4, August 1993, pp. 1010-1025.

24. Waltz, D., “Is Indexing Used for Retrieval?”, *Proceedings of the DARPA Case-Based Reasoning Workshop*, Morgan Kaufmann, San Mateo, California, 1989, pp. 41-45.

25. Wilkins, D.R., “Domain-independent planning: Representation and plan generation,” *Artificial Intelligence*, Vol. 22, 1984, pp. 269-301.

26. Zito-Wolf, R., and Alterman, R., “A framework and an analysis of current proposals for the case-based organization and representation of procedural knowledge,” *Proceedings of the Eleventh AAAI Conference*, AAAI Press, Washington, D.C., 1993.

<u>Appendix A</u>

The author distributed a questionnaire among the users of ***Smart* USA**, i.e. both IS staff as well as end user community, and analyzed the results to come up with the following conclusions:

- There are currently about 500 (five hundred) cases in ***Smart* USA**'s case base and more are being added every day.
- Workload for the database builders has drastically decreased since, the system has learnt all the common abbreviations and aliases and therefore generates an accurate "key-word-filled" brief description field automatically. This conclusion was reached based on the responses to the penultimate query in the questionnaire (see attached sample).
- The users have listed the quick response time, the fact that the solution tends to remain in their memory longer since they feel like they solved the problem themselves, the invaluable service provided to people with average or less than average memory, and the freedom to type in any of the key words for a problem in order to retrieve it from the database as some of the benefits of using the ***Smart* USA** system.
- Although the author intended for a descriptive input to be provided by the user in order to maximize matches, there seems to be a trend in typing in only the words that must identify the problem. This works well as there is

less typing for the users and less processing and thus greater speed for ***Smart*** **USA**

- The overall rating was quite high, as database builders felt like the learning facilities for the system were effective and easy to use. The same database builders also gave the system a high rating in the level of “intelligence” it displayed both in the generation of the key-words-filled brief description as well as in the retrieval mechanism. The help desk staff, who are the most frequent users of the system, have projected atleast a 60% to 70% reduction in calls once all the cases are entered into the database and the system is implemented at more end user locations.

Appendix A

Smart USA User's Questionnaire

Name :

Position/Title:

How long have you used the ***Smart* USA** system?

How many problems have you entered into the knowledge base (approx.)?

1-10 10-50 50-100 100-150 150-200

How many times have you used the system to solve problems?

1-10 10-50 50-100 100-150 150-200

How would you rate the system's case retrieval capability?

Poor Satisfactory Excellent

1 2 3 4 5 6 7 8 9 10

What is the sytem's rate of failure i.e. % of times it either returned a completely unrelated problem or returned nothing when a similar or related problem was present in the database?

1-10% 10-25% 25-50% 50-75% 75-100%

How does ***Smart* USA** benefit you in performing your job directly?

How would you rate the ease of use and overall user friendliness of the system?

Poor Satisfactory Excellent

1 2 3 4 5 6 7 8 9 10

How often do you change the system generated brief description when you enter a problem into the case database? Select a % from the options below:

1-10% 10-25% 25-50% 50-75% 75-100%

How would you rate the overall "intelligence" of ***Smart* USA**?

Poor Satisfactory Excellent

1 2 3 4 5 6 7 8 9 10

Appendix B - Sample Entries in the alias table

Note: A blank alias word represents the null equivalence of the corresponding real word

Real Word	Alias Word
the	
a	
and	
it	
is	
by	
of	
on	
this	
that	
when	
where	
why	
how	
problem	prob
solution	soln
report	rpt
reports	rpt
printer	prtr
in	
if	
very	
to	
too	
much	
but	
(	

Appendix B - Sample Entries in the alias table

Note: A blank alias word represents the null equivalence of the corresponding real word

Real Word	Alias Word
must	
maximum	max
number	nbr
programs	pgm
be	
change	chg
from	
status.	status
are	
for	
each	
as	
sales	sls
salesman	slsman
customer	cust
does	
plant	plt
operation	op
dir	
center	ctr
print	prt
system	sys
package	pkg
multiwall	mw
table	tbl
tables	tbl
month	mo
production	prod