# A Self-improving Helpdesk Service System Using Case-Based Reasoning Techniques

Pradeep Raman, Kai H. Chang. W. Homer, Carlisle, and James H. Cross
Department of Computer Science and Engineering
Auburn University
Auburn, AL 36849-5347
Pradeep.Raman@BLOCKBUSTER.sprint.com, {kchang, carlisle, cross}@eng.auburn.edu

## ABSTRACT

Case-Based Reasoning (CBR) is the process of solving a given problem based on the knowledge gained from solving precedents. It is an effective technique in the area of customer services or helpdesks. That is, a CBR system is used to solve most of the commonly occurring customer problems. While the implementation techniques may vary, most CBR systems include the following five steps: case representation and storage, precedent matching and retrieval, adaptation of the retrieved solution, validation of the solution, and finally, casebase update to include the information gained from the new problem. This paper details the various implementation techniques for these five steps, while focusing on a particular helpdesk system, namely SmartUSA, developed for the Union Camp Corporation. This system solves a customer's problem by filtering the problem description through an alias table to generate a brief description and then matching the brief description with the cases in the database. It has proved to be an effective and user-friendly system that has successfully handled different descriptions of the same problem and allowed for the casebase to be built in free-format (plain) text. This system has significantly reduced the workload and the response time in the customer services department of the Union Camp Corporation.

**KEYWORDS**: Case-based reasoning, machine learning, expert system, artificial intelligence, helpdesk service automation.

## 1. INTRODUCTION

Artificial Intelligence (AI), in its simplest form, is the study of utilizing computers to solve problems in a manner similar to that of a rational human being. Most of human reasoning is based on the collection of experience and cases in a loosely organized dynamic memory [20]. That is, human beings solve problems by recollecting the knowledge gained from solving similar or related problems in the past. 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) [18]. When used in a comprehensive manner, i.e. encompassing both the semantic as well as the syntactic meaning of cases, CBR 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 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 problem descriptions from the user.
2. *Retrieval:* Retrieving the closest-matching precedent to the current problem from a database of cases.
3. *Adaptation*: Selecting a solution based on 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, after analyzing the validation responses.

This paper describes the implementation of these five steps while focusing on a particular customer support system, namely **SmartUSA** [16] of the Union Camp Corporation. The goal of the system is to provide the end users a direct access to SmartUSA and assist in locating a solution for their problems.

## 2. CBR DEVELOPMENT AND APPLICATIONS

Although a wide range of potential applications have been explored, commercial CBR applications have focused mainly on decision support environments [21]. Problem resolution in customer service helpdesks has also been reported [22]. Most customer service applications, such as SmartUSA, tend to focus on retrieving the closest matching precedent to the given problem without extensive adaptation and validation capability. Since most helpdesk personnel report about 60% to 70% of their time being spent on solving repeated problems [22], there is a valid argument for the use of a CBR system in the daily operations.

There are two basic CBR approaches, semantic analysis and syntactic analysis. CBR systems based on semantic analysis are few in number in the commercial arena. These systems are limited in their domains and the number of cases they can handle efficiently. TOLTEC, a semantic analysis based planner was developed in 1993 to solve complex manufacturing problems by storing old plans in a dynamic memory [24]. Its domain is limited to the manufacturing processes and the system uses a complex indexing scheme and retrieval algorithm. Caper is another planning system utilizing CBR as

its plan generation approach [12]. Its domain is limited to car assembly and transportation logistics. It has the capability of breaking a goal into subgoals and retrieving multiple cases to achieve these subgoals and then merge them into a composite plan that solves the global problem.

The number of commercially available CBR systems based on syntactic analysis is much larger as this type of analysis is more suitable for adaptation to different domains. The indexing scheme and search and retrieval mechanism function independently from the meaning of the cases. The indexing schemes are flexible in this type of systems, i.e. the developer chooses the features to be indexed. Bendata Management Systems' Helpdesk Expert Automation Tool (HEAT) is one such application that uses a combination of CBR, simplistic browse, and lookup techniques in a PC LAN environment to store cases and propose solutions. HEAT provides a graphical user interface thus allows helpdesk personnel to quickly browse through related problems and solutions. The system does not process user's problem descriptions for matches but simply allows for a friendly interface for helpdesk personnel to retrieve the precedents.

Compaq Computer's SMART system is an integrated call tracking and problem resolution system [1]. It is used by Compaq's support staff, who interpret the end user's problems and enter the descriptions into the system in a predetermined manner. There is no automated validation or update phases as the staff members sort through the similar cases returned by SMART. It runs on a 386-based PC under the Microsoft Windows environment and has produced a significant increase, from 50% to 87%, in the user problems resolved on the first call [1].

SmartUSA 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 and the self-improving capability. 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, most techniques require a significant amount of effort on the part of the casebase builder and maintenance group. NEC Corporation's customer support system, SQUAD [22] 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 very cumbersome to develop and maintain. There is also a significant overhead and less flexibility in terms of handling free-format (plain) text in SQUAD. Smart USA on the other hand allows the end user to directly input the problem in free-format text and attempts to resolve the problem by presenting a single precedent case. If the case is accepted by the user, solutions for that case will be presented in an appropriate order. The self-improving capability of SmartUSA resides on the continuous improvement of the alias table and the adjustment to the rankings of the potential solutions which will be discussed later.

## 3. REPRESENTATION

Representation is the first and probably the most significant phase of CBR. It deals with the storage and construction of the casebase. It is typically built by organizing the input from a domain expert. A case can be defined in terms of the problem description and the potential solutions for the problem[4]. Furthermore, a case must include all the relevant domain knowledge such as problem

category, application type, the context in which it exists, and its expiration date if any. This kind of domain knowledge is stored as attributes to the cases and primarily used in the process of identifying the precedent cases [15].

If semantic analysis is used as the retrieval approach, cases would have to be stored in some form of semantic networks or grammar trees [26]. 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 casebase builder toward 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*. These systems would certainly be more comprehensive and human-like in their overall reasoning process than a syntactic analysis based system [17].

On the other hand, there is more flexibility in utilizing syntactic analysis as the retrieval approach of a CBR system. Such a system needs only to handle the storage of the problem description and then decide on indexing techniques that an effective search can be facilitated. The key will be to balance between 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]. Syntactic analysis based CBR systems can store case descriptions in free-format text or as database records.

SmartUSA stores the complete problem description of a precedent in a free-format text (provided by the domain expert) and transforms the description into a brief "key words filled" description which can be used by the search algorithm to identify this precedent. The key to a high rate of case retrieval success is the presence of *key words*, which uniquely identify the precedent [9]. SmartUSA generates the brief description by stripping out all non-key words such as "the", "when", and "how", and abbreviating words in the original problem description. The system *learns* the words to be stripped out as well as the words to be abbreviated as cases are entered into the casebase. This information is stored in a alias table which contains REAL_WORD = = > ALIAS_WORD relationships. Once cases have been entered, the alias table will contain a list of all abbreviations and *non-key* words. This allows the system to parse all user problems through the alias table and thus generate a brief description for each problem description. As the number of entries in the alias table increases, the system knowledge increases as well and better brief descriptions can be generated. After a relatively small number of case entries in the knowledge base, the casebase builder will have to edit or enhance the system generated brief description only rarely, and thus user friendliness is achieved. A sample case is illustrated in Figure 1.

SmartUSA learns the aliases and abbreviations from the casebase builder and then utilizes this table globally to convert all past and future problem descriptions into the consistent 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 1 contains key words required to uniquely identify this problem and therefore allows for a successful search and retrieval. In this manner Smart-USA mimics human being's analytical process as most problems are resolved by memory triggers on key words [7].

**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 1. A sample problem entry in the casebase of SmartUSA

## 4. RETRIEVAL

The representation lays the foundation for a good CBR system, however the end product is only as good as the quality of its search and retrieval mechanisms. These mechanisms are highly dependent on the indexing scheme used in the system. Many CBR systems use the technique of preindexing, whereby indices are fixed prior to case entry time [10]. While this method is highly successful when the domain or purpose of the system is well defined, it lacks the flexibility in multiple domains [25]. The principal reason for its popularity is its simplicity. The other approach is the automatic indexing scheme, in which free-format 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.

Case retrieval implementation can vary widely in the CBR systems [22]. In the case of semantic analysis, methods such as nearest-neighbor algorithms, decision trees, or connections associative memories are best suited to traverse the semantic networks and grammar trees. Backtracking capability, early recognition of mismatches, handling different flavors of the same problem and exception handling will be some of the issues that must be taken into account in the semantic analysis based applications. Syntactic analysis of cases usually involves a simpler retrieval mechanism. However, the retrieval algorithm must not only search the indexed fields as developed in the representation phase but also apply some domain expert provided rules to capture exceptions or common-sense type information that human beings apply almost unconsciously [6]. The expert provided rules must also include a quantitative measurement which specifies the degree of match between a precedent and the current problem [18].

SmartUSA improves the retrieval time by indexing each word in the brief description and using them to match cases. As stated earlier, the user's problem description is filtered through the alias table to come up with a brief description. The description is compared against all the brief descriptions in the casebase. A case is considered to be a precedent candidate if any of the words in the brief description

of the current problem matches the words of the brief description of the case. The candidate cases are ranked according to the number of words matched and then filtered through some expert provided rules that take into account factors such as problem category, user's job title, 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 its associated solutions are then provided to the user for validation. An example of this type of filtration through expert provided rules is provided in Figure 2. In this example, three cases are found containing some of the key words of the problem description. SmartUSA ranks these cases according to the rules and returns the highest ranked case as the closest-matching precedent.

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

**Brief Description of the Problem**: *(Generated by System)*
slsman commsn rpt err

**Possible precedents retrieved from the casebase**: *(Generated by System)*

| | |
|---|---|
| 1. slsman commsn rpt #2416 err | (Dept. A) |
| 2. slsman rpt #5540 err | (Dept. B) |
| 3. slsman commsn rpt #5520 causing err | (Dept. A) |

**Relevant Expert-provided Rules**:

a) Add two points to the ranking of the case which pertains to the same division or department as the end user.
b) Add one point to the ranking for each key word matched

**Case returned as the 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 been ranked higher even if it did not pertain to the end user's department.

Figure 2. Filtration of matched cases through expert rules in Smart USA

In a typical customer support environment, a problem can be described in different words. For this reason, it is important for a system to be equipped to handle this type of situations [21]. The retrieval mechanism of SmartUSA facilitates the grouping of multiple descriptions of the same problem and proposes the same solution to all of them. 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 thus increasing the system's retrieval success rate. Figure 3 illustrates these concepts through an example. The example shows the structure of a case as stored in the casebase and a problem description as entered by the customer.

**Problem Description of a Case**: *(Entered by domain expert and stored in casebase)*
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 casebase)*
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 precedent)*
trouble car not start turn ignition key not started 6 mos

Figure 3 Details of Retrieval mechanism in SmartUSA

SmartUSA begins the search by first converting the user's current problem description into a brief description using the same alias table that has been used to build the casebase. Each word in this brief description is then used to search the casebase.

## 5. ADAPTATION

The adaptation phase proposes a solution to the current problem. In most systems this phase is implemented in a limited manner in the sense that all solutions of the retrieved "closest matching precedent" are presented to the user [3]. Nevertheless, in theory, significant intelligence can be added to the system at this stage. For example, before proposing a solution, the computer may analyze factors such as the success rate of the solution in the past and the compatibility with the given context and user. If any of these factors returns 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, provides little gain. SmartUSA takes a simplistic approach to adaptation by returning the solutions of the closest matching precedent, one at a time, in decreasing ranking sequence, until a solution is accepted. The solution that has been the most successful historically is assigned the highest ranking and therefore suggested first to the user. The system suggests all documented solutions to the problem before informing the user that outside help is needed.

## 6. VALIDATION

The proposed solution is validated through feedback from the user or the environment in this phase. This is part of the *learning* process of the system as consistently unsuccessful solutions must be weeded out and good solutions must be rewarded so that they will be on the top of the solution list. Using this approach, the historically most successful solution is given the highest ranking. When a

precedent is identified and a solution is proposed, the user is asked to provide feedback by answering questions like:

1. Is the identified precedent 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.

## 7. UPDATE

After receiving user's responses to the validation queries, the CBR system must determine whether or not this case is sufficiently different from others to warrant its addition into the casebase [23]. If so, it must make any enhancements or corrections so that the next occurrence of the same problem may be served by more accurate solutions. While new information is added to the casebase, outdated information is weeded out. Learning by parameter adjustment, which adjusts the ranking of each solution according to its success rate [6], is implemented in SmartUSA. For example, assume a precedent has three solutions. Solution A fails to solve this particular instance of the problem whereas Solution B succeeds in solving the problem, the rankings of these solutions are adjusted in the following manner:

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

In this manner, the next time the same case is retrieved, 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 ranking (as determined by the domain expert) when they are entered into the casebase. Thereafter, SmartUSA maintains the success rate of all suggested solutions and, by adjusting the associated rankings, the system *learns* the better solutions after each processing iteration. Examples on adaptation, validation, and update will be given in section 9.

## 8. SmartUSA IMPLEMENTATION SUMMARY

In practice, a CBR system may implement these five steps in varying degrees of completeness and depth [14]. CBR systems, which base their intelligence and inference on known cases rather than rules, are considered to be more effective than rule-based or other types of expert systems. Researchers argue that 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. In addition, record-like representation of cases in a CBR system allows for easy storage and update in a relational database [19]. The presence of validation and update phases based on the feedback from the user improves the machine's *experience* and incorporates knowledge acquisition into the day-to-day use of a CBR system [11]. Figure 4 provides a step-by-step walk through of the entire reasoning process of SmartUSA.

# File Contains Data for PostScript Printers Only

SmartUSA was developed in Microsoft FoxPro, a PC-based database product, and runs in a client-server environment under the MS-Windows and Novell Netware operating environments. Its primary benefit is to eliminate or drastically reduce the number of calls placed to the helpdesk personnel. Union Camp Corporation, where this system was deployed, is providing this self-assisting tool to the end users in an attempt to funnel helpdesk resources into other much needed areas. The system has been used by the helpdesk personnel for some time 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 SmartUSA. While not comprehensive, in the sense that not all queries may be resolved successfully by the system, SmartUSA is certainly a practical, user-friendly, and intelligent solution to handling large volume of repetitive cases. 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.

## 9. EXAMPLES

### 9.1 Alias Table Construction

When SmartUSA is used for the first time, i.e. before the first case has been entered into the casebase, the alias table is empty. Entries are added as illustrated in Figure 5.

**Problem Description**: *(Entered by the casebase builder)*
error message about security violation of the system when running report BSR600

**Brief Description**: *(Generated by the system. Note: alias table is still empty)*
error message about security violation of the system when running report BSR600

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

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

| | | |
|---|---|---|
| error :: = err | message :: = msg | about :: = "" |
| of :: = "" | the :: = "" | when :: = "" |
| running :: = "" | report :: = rpt | |

Figure 5 Alias table and casebase creation in SmartUSA

As the casebase builder deletes and abbreviates words, all the words in the problem description are processed and appropriate entries are added to the alias table. Note that "security", "violation", "system", and "BSR600" do not create entries in the alias table since they remain unmodified. Following the same approach, more cases can be entered and the alias table will continue to grow. Thus, as more entries are made into the alias table, a better brief description can be generated. Figure 6 shows two more cases with thier original problem descriptions and brief descriptions. Once the cases have been entered into the casebase, SmartUSA is ready for use. The user is provided with a data entry screen to describe their problem in free-format text. This full description is then parsed through the alias table to generate a brief description.

### 9.2 Searching for the Closest Matching Precedent

Let us consider the examples in Figure 2. Since all three cases contain some of the keywords of the given problem, they are considered as precedent candidates. Note that although "#2416" of case 1, "#5540" of case 2, and "causing" of case 3 are not found in the problem's brief description, the system does not discard them from being a precedent. It is important to note that not all words found in a case's brief description have to appear in the user's description, rather all cases containing matched

words are compiled into a list for further processing.

**Case#2 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.

**Brief Description** *(created by SmartUSA)*
err msg security violated system happening rpt BSR600

**Case #3 Description**: *(in free-format text)*
Report BSR600 is giving a security violation error message when run from the menu.

**Brief Description**: *(Generated by the system)*
rpt BSR600 security violation err msg

Figure 6 Two sample cases

Once a list of precedent candidates has been compiled, rules are applied to rank the precedents. 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. 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 database, since their solutions, expiration date, and other 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 cases as possible precedents. The rules may require the system to increase the ranking for the case related to the Billing application by a fixed number. If the number of words matched were the same or the difference was less than the fixed number between these two cases, the Billing case would be returned as the closest-matching precedent. The rules as well as the fixed number involved are provided by the domain expert. Note that the rules would have effect only when the difference in the number of word matches were less than the fixed number. In the examples of Figure 2, the chosen precedent is case 2, i.e. the case is related to Department B.

### 9.3 Presenting Solutions Based on the Precedent

Once the closest-matching precedent has been identified, all possible solutions are proposed to the end user in decreasing order of ranking, one at a time. If one of the solutions solves the user's problem, then the technique of learning by parameter adjustment is applied to modify the rankings of all proposed solutions. That is, the rankings of the solutions which were proposed to the end user but failed to solve the user's problem are decreased by a fixed number and the solution that succeeded is given an increase in ranking. The rankings of the solutions that had not been proposed to the end user remain unchanged. In this manner, SmartUSA maintains a list of solutions for each case ranked by their historical success rates. 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 7 illustrates the process of proposing multiple solutions, and updating the rankings after receiving feedback from the user. All solutions are assigned an initial ranking which is in bold and enclosed in parentheses. Figures 8 and 9 provide two complete sample runs of SmartUSA. In these examples, the ranking adjustment value is 2.

**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 ranking, 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 preferably 12M of memory in your PC.
NOTE: Make sure the memory sims are compatible with your hardware.
Call the MIS dept at 912-9xx-x286 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 7 Proposal of multiple solutions and updates in SmartUSA

**Problem Description**: *(Entered by a 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 for 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 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 912-9xx-x288.

**User's Feedback and Updated Rankings for the Solutions**:

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

Figure 8 First sample run of SmartUSA

**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 for 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 912-9xx-x286.

Figure 9 Second sample run of SmartUSA

## 10. CONCLUSION

Questionnaires have been distributed among the users of SmartUSA, i.e. both information system staff as well as the end user community in the Union Camp Corporation. The summary of the findings follows. :

* There are currently about 500 (as March of 1995) cases in SmartUSA's casebase and more are being added every day.

* Workload for the database builders has drastically decreased since the system has learned all the common abbreviations and aliases and therefore generates an accurate key-word-filled brief description automatically.

* 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, and the freedom to type in any of the key words for a problem in order to retrieve a precedent from the database as some of the benefits of using the SmartUSA system.

* Although a user is supposed to type in full problem description, there seems to be a trend in typing in only the words that would identify the problem. This works well as there is less typing for the user and less processing and thus results in greater speed for SmartUSA.

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

These findings clearly indicate that for a CBR-based system to be practical and useful, it must provide flexibility, easiness, and capability. The flexibility allows its diverse users to use similar but different descriptions for problems; while the easiness prevents the strict format requirement on its users' input. However, the convenience provided to the user still cannot replace the solution finding capability of the system. It must not only find appropriate solutions for its users but also continuously improve its own case base. These features should be the general design requirements for future CBR-based systems.

Although impressive success has been achieved by SmartUSA, more sophisticated approaches to case representation, storage, and indexing are still needed for CBR based systems. Development methods and tools that utilize a combination of cases, rules, and models in their dynamic memory are also in demand [27]. As companies begin to think and operate on a global basis, it will become increasingly crucial for adequate information to be available to personnel throughout the organization and for personnel to find more efficient approaches to problem solving. CBR based systems facilitate a smooth transfer of knowledge throughout the company and ensure that solutions are not re-generated 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 [22].

## 11. REFERENCES


[1] Acorn, T., and Walden, S., "SMART: Support Management AutomatedReasoning 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] Raman, P., "Case-Based Reasoning: An Implementation Methodology," Technical Report 95-04, Department of Computer Science and Engineering, Auburn University, 1995.

[17] 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.

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

[19] Riesbeck, C.K., "An interface for case-based knowledge acquisition,{{breve}}" Proceedings of

the DARPA Workshop on Case-Based Reasoning, DARPA, Washington, D.C., 1988.
[20] Schank, R.C., "Dynamic Memory: A Theory of Reminding and Learning in Computers and People," Cambridge: Cambridge University Press, 1982.

[21] 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.
[22] Simoudis, E., "Using case-based retrieval for customer technical support," IEEE Expert, Vol. 7, No. 5, October 1992.
[23] Stanfill, C., and Waltz, D., "Toward Memory-Based Reasoning," Communications of the ACM, Vol. 29, No. 12, December 1986, pp. 1213-1228.
[24] 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.
[25] Waltz, D., "Is Indexing Used for Retrieval?" Proceedings of the DARPA Case-Based Reasoning Workshop, Morgan Kaufmann, San Mateo, California, 1989, pp. 41-45.
[26] Wilkins, D.R., "Domain-independent planning: Representation and plan generation," Artificial Intelligence, Vol. 22, 1984, pp. 269-301.
[27] 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.

## SHORT BIOGRAPHIES ABOUT AUTHORS

**Pradeep Raman** is a Senior Programmer/Analyst at Blockbuster Entertainment Group in Ft. Lauderdale, Florida. His responsibility involves developing 16 and 32-bit client-server applications that service some critical business needs. Mr. Raman received a Master of Computer Science and Engineering degree from Auburn University and a BS degree in Computer Science from Armstrong State College. He won the 1991 NCAA Division II Tennis Singles Championship and still enjoy playing the game today.

**Kai H. Chang** is an associate professor of the Computer Science and Engineering Department, Auburn University. Dr. Chang's primary research interests are in artificial intelligence, expert expert systems, software quality assurance, testing, metrics, and computer supported cooperative work environments. He received a diploma in electrical engineering from the Taipei Institute of Technology, and MS and PHD degrees in electrical and computer engineering from the University of Cincinnati. He is a member of IEEE and ACM.

**W. Homer Carlisle** is an associate professor of the Computer Science and Engineering Department, Auburn University. Dr. Carlisle's primary research interests are in languages and environments for software design, and testing. He received his BS, MA, and PHD degrees from Emory University. He is a member of IEEE/CS and ACM.

**James H. Cross II** is an associate professor of the Computer Science and Engineering Department, Auburn University. Dr. Cross's primary research interests are in software engineering, software methodology, quality assurance, testing, metrics, and reverse engineering. He received a BS degree from the University of Houston, an MS degree from Sam Houston State University, and a PHD from Texas A&M University. He is a member of IEEE and ACM.

**FIGURE CAPTIONS**

(Please note that figures are placed in the main text body)

Figure 1 A sample problem entry in the casebase of SmartUSA

Figure 2 Filtration of matched cases through expert rules in SmartUSA

Figure 3 Details of retrival mechanism in SmartUSA

Figure 4 Process steps of SmartUSA

Figure 5 Alias table and casebase creation in SmartUSA

Figure 6 Two sample cases

Figure 7 Proposal of multiple solutions and updates in SmartUSA

Figure 8 First sample run of SmartUSA

Figure 9 Second sample run of SmartUSA