Diagnostic Technology for Customer Service Organizations

Charles E. Matthews
cmatt@ibm.net

Abstract

This paper is a brief discussion of the diagnostic tool technology that is currently available for customer service organizations. Not only will this paper address the range and capabilities of different systems that are available, it will also compare the impacts to the operation of the customer service organization that are caused by these tools. Because different companies have different needs, one single solution for everyone does not exist. Therefore, I include a discussion of tradeoffs among the different types of available tools.

Introduction

The characteristics and needs of different companies vary as widely as the characteristics and needs of different individuals. Therefore, what I will do within this paper is identify the characteristics of a generic company and discuss how these tools affect this hypothetical company. My hypothetical company, *FGS, Ltd.*, is a medium-sized company (annual sales of $10M) with multiple product lines. FGS, Ltd. has a telephone support center that receives calls from its customers who experience a problem with a product and from FGS service technicians who are unable to diagnose an existing problem with one of the products.

This type of support center must be able to log incoming calls (***incident reports***) and to determine how to handle the incoming call. If the incoming call concerns a question of how to use the product properly (a ***how-to*** question), then the support center operator provides the appropriate instruction and the session ends. The product is working correctly, and no defect exists. If, however, the incoming call concerns a potential defect of the product, the responsibility of the support center operator is to record the error symptoms, to identify any operational characteristics that existed when the error occurred, and to make an initial diagnosis if possible. If an initial diagnosis is not possible, the call record is referred to someone who possesses greater product knowledge than the support center operator. For this type of call to be resolved, someone must make a determination whether an actual defect exists or whether the product is being used incorrectly (***user error***). If a defect exists, enough diagnosis of the error must occur in order to determine how to repair or to replace the product.

Within call support organizations, the support center operators are known as ***first-level support*** because they are often the first point of contact by the customer. When the operator is unable to resolve the problem, the call is passed to the ***second-level support*** – either an individual or a department that possesses a higher level of product knowledge than the support center operator. In some cases ***third-level support*** may be an actual member of the engineering staff responsible for the design of the product.

This system of multiple levels of product support is quite among companies. It has three primary characteristics:

1. *Product knowledge* – The higher levels of support are characterized by increasing knowledge about the product. Second-level support personnel know more about the operation of the product than first-level support personnel, and third-level support personnel are the most knowledgeable of all.
2. *Call volume* – Typically, first-level support handles more calls than second-level support. Likewise, for third-level support. If the company notices that the volume of calls that are referred to the next level is **not** significantly lower, then this situation is usually a sign of inadequate product knowledge or training for the lower support level.
3. *Support cost* – Knowledge increases with training and experience. Companies spend money to give individuals training and experience. Likewise, those people who possess higher levels of training and experience are usually paid more than those with less knowledge. Therefore, the cost to the corporation

of handling a problem in first-level support is less than the cost of handling it in second-level support. Likewise, for third-level support.

Numerous companies provide software systems for managing incident reports and tracking them throughout the customer service organization. Among the most common call-tracking companies are Remedy, Vantive, Quintus, and Clarify. Although the base product from each of these companies provides some capabilities for keyword searching within the incident logs, higher levels of automated diagnosis are not provided within their product. Instead, these companies provide software interfaces for other vendors to link their diagnostic tools to the call-tracking system. In this fashion, information is exchanged between the call-tracking system and a diagnostic system. The remainder of this paper discusses the capabilities and features of the three most common types of diagnostic systems – decision tree systems, case-based reasoning systems, and expert systems.

Decision Tree Systems

Decision trees are bodies of knowledge that are organized in a strictly hierarchical structure. The purpose of the decision tree is to provide a structured set of questions that organize the set of results that a user wants to find quickly. By answering successive questions, a user is led to an appropriate result. A decision tree is typically implemented by writing a large number or IF-THEN statements that propose the questions and define the flow of control through the program based upon the answers.

Each node of the tree is a question. Each branch away from the node represents an answer to the question. The branch then specifies either the next question to ask or a result – in which case the question-answer session ends.

Hierarchical decomposition of a problem is one of the simplest problem-solving methods that people use when they are faced with a complex problem. When faced with a complex problem, break it into smaller pieces. Treat each piece separately. If it, also, is a complex problem, continue to break the problem down into smaller and smaller pieces until the individual pieces can be solved. This decomposition is the basis for decision trees and was a focus of early research in artificial intelligence.

Many problems exist for which decision trees represent an acceptable method of searching for a solution. When knowledge about a problem is static and unchanging, decision trees may represent an appropriate means for navigating through a well-structured problem domain. Early examples of diagnostic maintenance lists were primarily organized as decision trees. Even today decision trees represent a solution for searching FAQ lists (Frequently Asked Questions) for a product because such knowledge is usually represented in a hierarchical structure.

The primary problem with decision trees is that they become confusing and unmaintainable as the amount of knowledge represented by them increases. As the size of the decision tree approaches 100 nodes, the task of maintaining the tree structure becomes unmanageable. Complex problem solving knowledge cannot be represented in a strictly hierarchical structure. Current research investigations of problem solving activities focus upon recognizing patterns of knowledge relationships rather than hierarchical decomposition of a problem space. Thus, while decision trees represent an early form of automating diagnostic intelligence, they are inadequate for complex problem domains. This limitation, however, should not prohibit their use when the appropriate conditions exist.

Right Now Technologies, Inc. of Bozeman, MT. (http://www.rightnowtech.com) markets a decision tree system for searching product FAQs from the Internet (or an Intranet) called PowerFAQ. They also provide a full-featured product (Right Now Web) that incorporates decision tree searching into a web-based call tracking system that manages customer support contacts from the Internet.

Case-Based Reasoning Systems

"Case-based reasoning means reasoning based on previous cases or experiences. A case-based reasoner uses remembered cases to suggest a means of solving a new problem, to suggest how to adapt a solution that doesn't quite work, to warn of possible failures, to interpret a new situation, to critique a solution in progress, or to focus attention on some part of a situation or problem."

[KOLO96, p. 31] Janet L. Kolodner and David B. Leake

Case-based reasoning (CBR) is an area of artificial intelligence research that (like the following section on expert systems) investigates problem-solving activity. CBR, as a mechanism for managing, sharing and accessing knowledge, grew out of the academic research in expert systems. It is a commercially viable technology for fault diagnosis in complex problem domains.

In essence, *a **case** is a piece of knowledge that represents an experience that teaches a lesson*. What does this mean? Well, the knowledge that is encapsulated within a typical case can consist of its title, a free-form text description, and a collection of questions with answers that represent pre-conditions for the case. The lesson that the case teaches is usually a result or a conclusion. In the case of a diagnostic system, the result can be the identity of a failing device.

Although the content of a case in a CBR system is similar to the content of a case in a decision tree system, a fundamental difference exists in their algorithms for determining a solution. Within a decision tree, the ability to reach a given result is completely dependent upon the sequence of asking and answering questions within the tree. In contrast a case within a CBR system represents an independent unit of knowledge. Therefore, the process of adding knowledge to a CBR system can be done in an incremental fashion - up to a certain point.

The algorithm for searching the knowledge base of cases employs a nearest neighbor type of search to find those cases with the greatest similarity to the problem that is currently being analyzed. Because the set of matching results can contain cases that are "similar but different" from the problem being analyzed, CBR systems are known as "fuzzy" matching systems. The ability to use fuzzy matching to identify cases that are similar but not exactly the same allows the user to interact with the CBR tool in a complementary, symbiotic fashion. The end result is that CBR systems are much more reliable and stable than decision tree systems.

All of the commercial CBR systems today are more accurately characterized as **Case-Based Retrieval** systems rather than **Case-Based Reasoning** systems. This distinction is important because the commercial systems are quite efficient at finding similarities between an existing problem situation and previously seen cases that are stored in its knowledge base. Therefore, what these systems really do is *reasoning by remembering* rather than strict reasoning by logic or inference.

Although about a half-dozen commercial CBR vendors exist, the two companies of primary importance are Inference Corporation (http://www.inference.com) and The Haley Enterprise (http://www.haley.com). Inference marketed the first commercial CBR system in 1991 under the name CBR Express. Their current system (Content Navigator) is a product line that includes stand-alone, networked, and web-based systems. Although Inference defined the commercial CBR marketplace and remains the predominant market leader for CBR systems, the company failed to maintain its technical leadership with high quality products and services. Specifically, their CBR Express product earned some notoriety by its lack of software quality and Inference's inability to enhance the tool's usability significantly. However, within the past 18 months Inference has announced some significant enhancements to their product line, and there are signals that their development staff is addressing their quality problems. With the addition of the k-Commerce and Interactive Voice Response (IVR) functionality, Inference is once again pushing the technology boundaries.

The Help!CPR system from The Haley Enterprise (THE) has many of the same technical features of Inference's earlier CBR Express product. However, in addition to the Help!CPR product line, THE also markets a full-blown rule-based inference engine (RETE++ and ECLIPSE). The ability to produce

integrated solutions with Help!CPR and RETE++ allows THE customers to develop diagnostic systems for problem domains that are substantially more complex than allowed by Inference's Content Navigator.

Expert Systems

Expert systems are computer programs that attempt to solve problems in some specialized application domain with the same level of competence as that of a human expert in that problem domain. Although one could argue that the goal of any computer program is to solve problems in particular domains, conventional programs are usually limited (by their design) to a narrow and restricted ability to process data in strict algorithmic and logical steps. Expert systems, on the other hand, approach the problem solving process in a fashion similar to how psychologists believe that humans solve problems.

People organize their knowledge in a separate entity (our brain) and access it to reason out specific problems. The human solving process involves at least three types of knowledge: **domain knowledge**, **heuristics**, and **problem solving strategy**. [MATT91, p. 21] Within an expert system, the program knowledge is collected into an entity called the **knowledge base** (KB). This knowledge, also, is usually a collection of three types: data, rules, and control (corresponding to the domain knowledge, heuristics, and problem solving strategy of humans). Additionally, since most tasks that are performed by experts use significant amounts of knowledge, the expert system needs mechanisms to accommodate large amounts of knowledge in order to permit finding the relevant knowledge easily when it is needed.

Regard the data within an expert system as a collection of assertions of facts about the problem and its environment. This data (or **working memory)** is often in a dynamic state of change. Some facts will be created and others discarded as the system merrily works along on a problem. How these facts are created, used, and discarded is determined primarily by the **rules** within the expert system. A rule can be viewed as an independent unit of knowledge that looks for the existence of some pattern of facts and infers some conclusion.

For example, assume that the following facts exist:
f-1 [Bob is-the-parent-of John]
f-2 [Bob is-a MAN]

and the following rule exists:

```
r-1 {IF
  (?x is-the-parent-of ?y) AND
  (?x is-a ?z)
THEN
  (infer (?y is-a ?z))
}
```

This simple rule looks at the existing list of facts (f-1 and f-2) and infers that a new fact exists:
f-3 [John is-a MAN]

This new fact depends upon the state of the existing list of facts (which can change) and the rules that exist within the knowledge base. If fact f-2, instead, stated that [Bob is-a DOG], then the new fact f-3 would be [John is-a DOG]. The rule r-1 is looking for the existence of a specific pattern of data within the working memory. When the pattern is found, the rule executes. The result of that rule may conclude new facts that are then used by other rules, etc. Generally speaking, rules look for relationships between facts and exploit those relationships (when found) to infer new facts or knowledge.

Since the late 1960's researchers have constructed a number of expert systems that established the ability of these computer programs to solve many complex problems effectively. These systems ranged from interpreting mass spectrograms for complex molecules (DENDRAL) to a system that expertly diagnosed bacterial infections (MYCIN). In 1981, Digital Equipment Corporation began using a knowledge base written for the OPS5 system to automate the task of configuring VAX computers for DEC's sales and

marketing staff. XCON was the first commercial success of expert systems, and it performed its task better than a panel of DEC's most expert configuration technicians. When it became clear that XCON was saving DEC millions of dollars annually, other computer companies soon followed with configuration systems of their own.

Beginning in the early 1980's, the commercial market for expert systems was established by a number of companies. IntelliCorp, Inference, Teknowledge, and Carnegie-Group (dubbed the Gang of Four by Business Week) were the major players in the early expert systems marketplace. Later IBM entered with its own line of inference engines (ESE, KnowledgeTool, and TIRS). Unfortunately, many of these companies failed to maintain viable commercial product lines for a variety of reasons.

As of today three main expert systems engines exist – ART-Enterprise from Brightware, ECLIPSE from The Haley Enterprise, and CLIPS (a public domain inference engine developed for NASA). Interestingly enough, these three systems all grew from a common intellectual source. As one of the early members of the AI marketplace, Inference Corporation marketed a lisp-based system known as ART. At this time, NASA began work on an ART clone (using the C language) to run on IBM PCs. This result became CLIPS (C Language Integrated Production System) and entered the public domain. Dr. Paul Haley, one of the individuals involved in developing CLIPS for NASA, later became Chief Scientist for Inference Corporation when it began its own port of ART from lisp to the C language. This product became ART-IM. Later, Dr. Haley left Inference to found his own company under the name (can you guess it?) The Haley Enterprise and developed ECLIPSE.

In 1994, Inference spun-off its expert systems group into a new company called Brightware (http://www.brightware.com) that evolved ART-IM into ART-Enterprise. Inference, itself, chose to market its CBR tool (CBR Express) that was discussed in the previous section. Today, ART-Enterprise, CLIPS, and ECLIPSE have strong similarities and have each evolved with similar strategies.

Comparison Factors

Construction / Deployment:

The construction and deployment resources required for decision trees and case bases are more similar than those required by expert systems. Because decision trees and case bases rely upon relatively simple forms for their representation of knowledge, it is possible to construct the knowledge base using template-like forms. In fact, the authoring tools for Content Navigator and Help!CPR are based upon template forms. Although it is strongly recommended that an experienced knowledge engineer be used to create the initial case base architecture, local client personnel who are experts in the product knowledge (but not necessarily in case base technology) are often used to create the majority of the case base knowledge. Once the knowledge base is designed, anyone trained in the use of the authoring tools can assist in the implementation. Because the structure of decision trees is rigidly defined, the actual implementation is more difficult than case based systems. Often, decision tree implementation requires the use of specific key words and duplicated questions to complete the tree.

In contrast, the knowledge bases for expert systems usually require a knowledge engineer to construct the entire knowledge base. Expert systems are more complex because their reasoning abilities are much more complex than case based systems or decision trees. Therefore, the construction of their knowledge base is also more complex. Knowledge engineers require specialized training to learn how to architect the knowledge base for a particular expert system. More importantly, knowledge engineers are trained in methods of acquiring knowledge from product domain experts through interviewing techniques, modelling, constructing user scenarios, and other techniques for constructing the knowledge base.

Once the knowledge base for any of the three technologies is complete, it can be deployed in a variety of ways. Depending upon the features of the particular chosen tool, deployment could be as a single user system, as a networked knowledge base, or as a web-based system (Internet or Intranet).

Maintenance:

Maintenance operations are necessitated by two reasons – to fix incorrect information in the knowledge base and to add new knowledge to the knowledge base. Case based systems are the simplest systems to maintain. Correcting incorrect knowledge consists of editing the case that is defective. New knowledge can usually be added in simple increments to the knowledge base. Decision trees, however, are quite brittle in their ability to grow. Often the tree structure changes significantly with the addition of small increments of new information. The knowledge bases for expert systems can often tolerate the addition of new knowledge without substantial change to their architecture as long as the new knowledge is similar to existing information. The following list ranks the difficulty of maintenance by increasing difficulty:

1. Case-based systems
2. Expert systems
3. Decision tree systems

Cost:

Cost is a quantitative number that often has extremely fuzzy sources. Depending upon the specific tool that is chosen, there may be different costs for the authoring tools used to construct the knowledge base and the run-time tools used to access the completed knowledge base. Other typical expenditures are:

- The cost of consultant knowledge engineers for designing the knowledge base (because this expertise often does not exist within the client organization).
- The labor cost of any client personnel involved in constructing and maintaining the knowledge base.
- The training cost for authors.
- The training cost for the end-users.

To offset this cost, one must calculate the cost savings to the company in the reduced time to diagnose faults, the use of lower cost personnel to diagnose problems, the reduction of calls to the product support center by its customers, and the increase in customer satisfaction.

Solution Effectiveness:

The solution effectiveness depends upon the complexity of the problem domain. If the diagnostic task requires complex reasoning, then expert systems are the most appropriate solution. However, few diagnostic tasks in the support center environment fall into this category. In general substantial diagnosis can be done with case-based systems. Decision trees are appropriate in relatively few situations where the knowledge doesn't change significantly over time.

Staff Considerations:

A characteristic of most product support centers (especially in larger companies) is that the support staff undergoes substantial employee turnover. Within some companies the annual staff turnover can be as high as 75%. As a result, training and education of new personnel becomes a substantial cost for running the support center. Automated diagnostic tools help to reduce the training time for new personnel. Without the diagnostic tools, the support personnel must be trained as to which questions to ask the caller and what information is necessary to gather during the initial call session. In most cases of complex devices, the number of potential failures is so large that it is not possible to compile a generic list of questions for the Level 1 support personnel that adequately diagnoses the problem without the assistance of a diagnostic tool. This factor necessitates either more training time for the Level 1 support personnel or more calls being referred to the Level 2 personnel. Either situation results in increased cost to the company.

Internet / Intranet Access:

At the core of a company is the group of people who contain the most knowledge about a company's products. On the technological side of the company, these people are the product architects and designers. Outside of this group is another group of personnel (service technicians) who know how the product works but not necessarily all of the side-effects that can be caused by faults or by changes to the design. Outside of this group is another group (support center personnel) who are responsible for responding to customer calls for assistance when the product doesn't operate according to the customer expectations. And, finally, the outermost group is the company's customers.

Each of these groups possesses specialized knowledge that the groups outside of it do not possess. An effective principle of knowledge management is to move knowledge in an outward direction to those

people who need it. The right diagnostic tools can accomplish this movement by moving diagnostic knowledge from technicians to Level 1 Support personnel and from them directly to the customers. The most of effective way to provide customers with diagnostic knowledge appears to be by Internet or Intranet access to the knowledge bases. All of the major tool vendors have some type of Internet access, and this method is quickly growing in popularity.

Characteristics of the Ideal Company for Each Technology

How does a company know which technology to choose? Although there is no list of decisive criteria that selects case-based systems for this company versus expert systems for that one, there are some general rules of thumb. In general the cost of the tools for decision tree systems tend to be the least expensive. Therefore, companies with a minimal budget may choose them because of their lower cost. Another reason for choosing decision trees is when the knowledge is fairly static and will not be changing significantly over time.

If the knowledge is expected to change, then either case-based systems or expert systems are more appropriate. The primary distinction between these two technologies relates to the complexity of the diagnostic knowledge. When the diagnostic expertise cannot be represented as cases of previous faults (*reasoning by remembering*), then the expert system approach should be used. Otherwise, a case-based system is probably workable.

Continuing Development

Case-based reasoning systems rely upon previous experience to construct the case base. If a problem hasn't been seen yet, then a case for it doesn't exist. Likewise, most expert systems that are developed for troubleshooting devices use heuristic rules based upon experience. Therefore, with typical product lifecycles, a lag period exists between the time when a product is released to customers and the time when a diagnostic tool is available for use. This lag period (typically at least six months) is necessary in order for the product service technicians to gain the necessary experience in diagnosing product defects so that they can construct a diagnostic knowledge base. With the trend toward shorter product lifecycles, any lag period between the release of a product and an available knowledge base significantly lessens the value of the knowledge base.

Prior work by the author [JOHN89] investigated the applicability of constructing diagnostic knowledge as a by-product of the design process instead of as a result of experience with diagnosing actual product failures. The value with this approach is that a diagnostic knowledge base can be available immediately when the product is released – thereby removing the normal post-release lag period. Although this early work focused upon hardware products, recent work by the author has replicated similar results for software products with the use of meta-comments in software source code.

Strategic Direction

The trend for a customer service organization should almost always be to move product knowledge closer to its customers. Moving the detailed design knowledge of third-level support engineers to the second-level support personnel and from the second-level support to first-level support replaces the time spent with a company's most expensive personnel by time spent with less expensive personnel. When the quality of the diagnostic knowledge is the same, then the company saves money and the customer is happier because the solution to his problem occurs sooner.

But, this movement of knowledge should not stop at the company boundary. It should extend beyond the company to make diagnostic expertise directly available to the customer. A number of commercial companies have developed systems that are accessible from their Internet web sites to handle diagnostic problems. Broderbund Software is one of a growing number of software manufacturers that allow direct customer access to their product knowledge bases. When the customer is able to diagnose and resolve his

problems directly without contacting the customer service organization, the **call avoidance** savings for the company can be significant. However, much more important than the financial savings is the increase in customer satisfaction due to faster problem resolution and the customer's sense of controlling his own needs. A customer who believes that he can resolve problems with Product A faster than with Product B will prefer Product A.

Bibliography

[JOHN90] Paul E. Johnson, Dmitry Volovik, Imran A. Zualkernan, and Charles E. Matthews, "Design Knowledge for Discovering Troubleshooting Heuristics", IASTED International Symposium Expert Systems Theory and Applications, Zurich, Switzerland, June 26-28, 1989.

KOLO96 – Janet L. Kolodner and David B. Leake, "A Tutorial Introduction to Case-Based Reasoning," in Case-Based Reasoning, ed. David B. Leake, MIT Press, 1996.

MATT91 – N. M. Mattos, An Approach to Knowledge Base Management, Springer-Verlag, 1991.