AI Magazine Volume 28 Number 2 (2007) (© AAAI)



# Appliance Call Center: A Successful Mixed-Initiative Case Study

*William Cheetham and Kai Goebel*

■ *Customer service* is defined as the ability of a company to afford the service requestor with the expressed need. Due to the increasing importance of service offerings as a revenue source and increasing competition among service providers, it is important for companies to optimize both the customer experience as well as the associated cost of providing the service. For more complex interactions with higher value, mixed-initiative systems provide an avenue that gives a good balance between the two goals. This article describes a mixed-initiative system that was created to improve customer support for problems customers encountered with their appliances. The tool helped call takers solve customers' problems by suggesting questions aiding the diagnosis of these problems. The mixed-initiative system improved the correctness of the diagnostic process, the speed of the process, and user satisfaction. The tool has been in use since 1999 and has provided more than $50 million in financial benefits by increasing the percentage of questions that could be answered without sending a field service technician to the customers' homes. Another mixed-initiative tool, for answering e-mail from customers, was created in 2000.

One of the more frustrating experiences in calling an automated answering system for information or help is being forced to listen to all the menu options and then being directed through the system, sometimes requiring a redial (only to start again at the top menu option). These systems are rather popular with companies because they save money—the companies' money, that is. However, in a competitive service industry, these fully automated systems may not be the best choice for a service provider because customer satisfaction plays an increasingly important role. Therefore, more sophisticated tools are employed that attempt to balance the need of making the customer happy—and perhaps increasing some direct tangible benefit at the same time. This is where mixed-initiative systems play an important role.

General Electric (GE) Consumer & Industrial's Appliances Division manufactures and sells a wide range of home appliances. GE's subsidiary, Advanced Services Inc. (ASI), provides customer service call centers that help solve customer issues over the telephone and schedule field service visits when needed. One of the telephone services ASI provides is a group of more than 300 field service call takers who schedule field service personnel to visit customers' homes. Around 1.4 million home visits are scheduled per year. The call takers' primary goal was to schedule service visits, not diagnose problems with appliances over the telephone. However, they were able to help the customer without sending a field service technician in 3.9 percent of the calls. There was an opportunity to increase this percentage because about 20 percent of the time all the field service representative who visited the home needed to do was to inform the customer about a particular setting or remedial action. If this information could have been obtained over the telephone, it would have saved time for both the customer and the field service representative. However, training the call takers to diagnose appliances was difficult because of the large number of appliances that can be

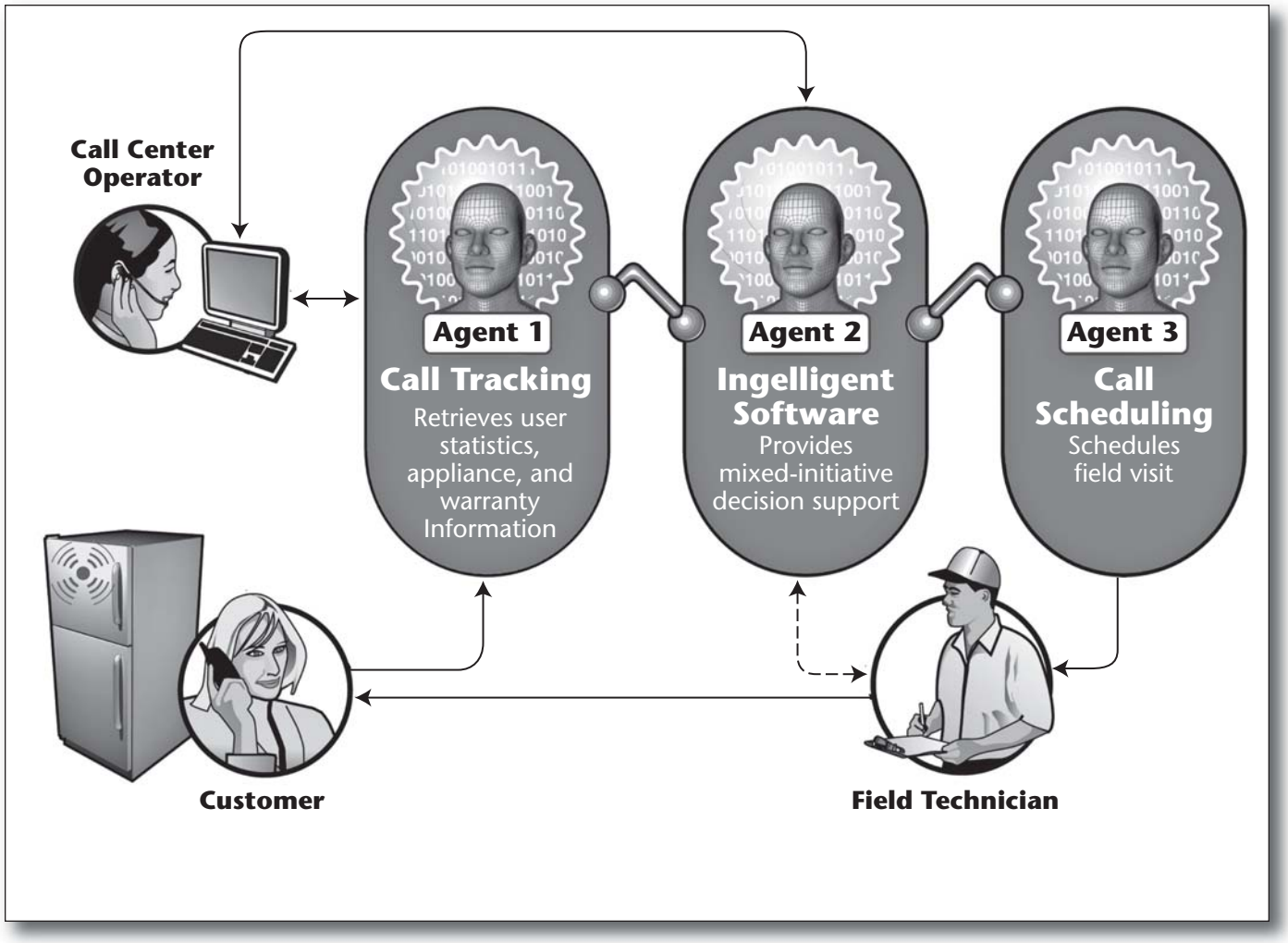


*Figure 1. Customer Support Process.*

serviced, the complexity of modern appliances, and the high turnover in the call takers. The solution for this was to create a software tool that acts as a mixed-initiative assistant (Allen 1999) for the call takers called Support the Customer (STC).

The STC system is just one part of the call-taking process. The full process is shown in figure 1. In this process, the customer, in the bottom left, calls an 800 number provided for scheduling home visits of field service technicians. A customer call-tracking system (Agent 1) receives the call, accesses a customer database to retrieve information, such as the customer's address and registered products, then displays this information to the call taker who actually answers the telephone. STC is Agent 2. Before STC was developed, the call takers would answer the questions to the best of their ability using their personal knowledge, training, paper manuals, frequently asked questions lists, and weekly paper flyers on new issues. STC uses a case base, a rule base, and a decision tree to assist the call takers in helping the customers. STC stores cases of problems and their solutions, a decision tree of questions that are used in a diagnostic process to differentiate the actual case from all others, and rules that can automatically answer questions. After the appliance has been diagnosed, the results of the diagnosis are stored in the call-scheduling system, Agent 3. The call-scheduling system creates a planned routing for the field service technician the night before the service call is to take place and suggests parts to stock on the repair person's truck. The call-taking system also sends data about the customer to STC, and the field service technician can provide feedback on the success or failure of the service visit to STC.

The next section discusses STC in more detail. Then the mixed-initiative issues of STC are described. The final section gives results from multiple years of usage for STC.

## Support the Customer Application

The call-center tool is designed to help call takers solve customers' problems by suggesting questions that can be used to diagnose the customers' problems. The primary goals of the call taker are the correct diagnosis of the appliance, speed of diagnostics, and customer satisfaction. The purpose of the software tool is to improve metrics associated with these goals. In order to do this, there were multiple requirements for the software tool. These requirements are similar to the principles of mixed-initiative user interfaces described by Eric Horvitz (1999). The requirements are (1) suggest correct resolutions to customer issues; (2) suggest questions that will diagnose the problem; (3) increase the speed of the diagnosis process; (4) increase user satisfaction with the process; (5) explain to the user why something was suggested; and (6) have the tool learn from its experiences and be able to adapt to a changing environment.

### Related Work and Tool Usage

Case-based reasoning (CBR) (Aamodt and Plaza 1994) has been used to automate customer-support help desks by many companies such as Compaq (Acorn and Walden 1992) and Broderbund (Watson 1997). Using CBR for customer support (Simoudis 1992) has become of interest to many other companies with help desks. We used a CBR tool to assist GE Consumer Products customer-support personnel. STC was created using a CBR tool from Inference Corporation called k-commerce. (Inference was later acquired by eGain.) K-commerce allowed for mixed initiative between the call taker and the automated assistant. It provided standard user interfaces for the knowledge engineer and end user (that is, call taker). The knowledge engineer interface included forms for creating cases and rules. During project development, Inference Corporation also released a tool for creating decision trees. These tools greatly reduced the development time for the project and allowed us to focus on knowledge acquisition instead of tool creation.

The k-commerce form for creating a new case allows the knowledge engineer to enter a title, description, multiple questions, and a resolution for the case. The system selects the most relevant cases using two criteria. The first criterion is the degree of match between the text description of the problem typed in by the call taker and the text in the title and description of the case. The second criterion is the percentage of questions in the case that have been answered correctly. When the diagnostic process is started (before any questions have been answered) the selection of the most relevant cases is based on the text match. Then, as questions from the highest matching cases are suggested and answered, the correct answers play a larger roll in determining the most relevant cases. The appropriate case is selected by answering the questions that differentiate the possible cases. For our application we wanted to be able to suggest one and only one case as the solution. A way to guarantee that these questions can differentiate every case in the case base is to form a tree of questions over the case base where each case is a leaf in the tree and each internal node in the tree is a question. We consider this approach as a hybrid approach that employs both decision tree and CBR methodology where the decision tree is simply the selection mechanism of the CBR system. Some problems can be represented by cases without creating a decision tree (FAQs are an example of these), but for others it is better to have a decision tree created for them (anything with a significant diagnostic process used the decision tree approach). Figure 2 shows a portion of a decision tree where rectangles are questions, the arrows are answers, and ovals are cases.

K-commerce also allows the creation of rules that can automatically answer questions. An example of a rule is, if the answer to the question "What type product is your appliance?" is "Refrigerator" and the third and fourth character in the model number are both "S" then the answer to the question "What type of refrigerator do you have?" is "Side-by-Side."

### User Interface

The current interface for STC is shown in figure 3. The critical information tab at the top has the product line and model number, which are both passed in from the call-taking system. It also has the problem description, which is typed in by the call taker. The model group can be determined by a rule that uses the model number. The *symptom i*s a keyword phrase that is selected by the call taker. The questions tab has a set of questions the call taker can ask the customer to diagnose the problem. The results tab has a set of solutions. Selecting the correct result is the goal of the process.

### Application Development

The development of the STC system was a five-step process. These same five steps can be followed for deploying other applications of AI: first, standardize the process and knowledge; second, digitize the inputs and outputs; third,

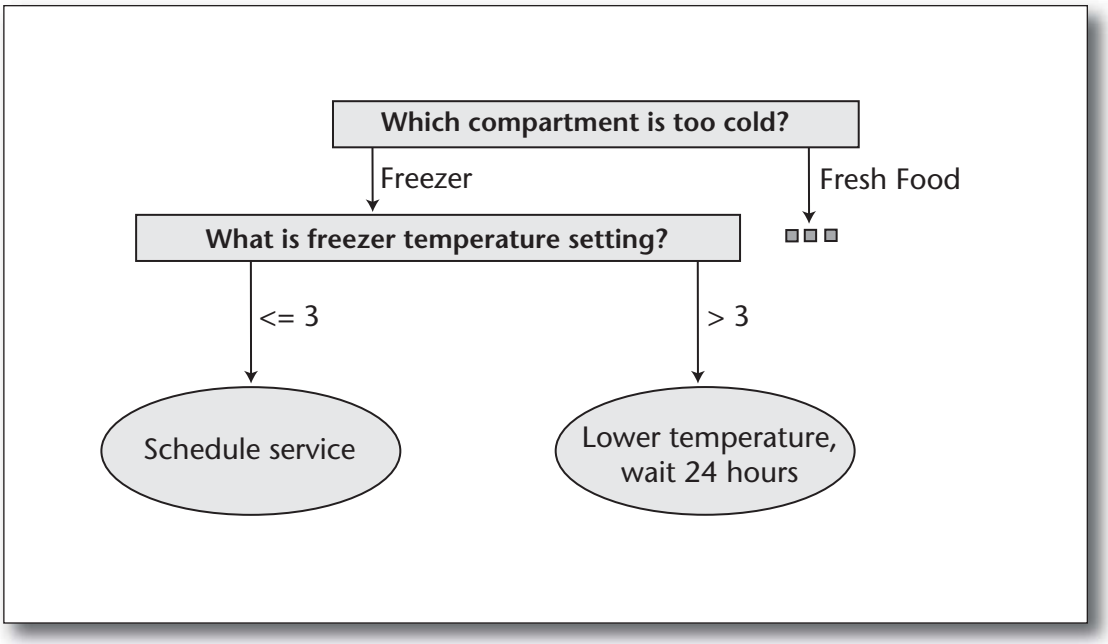


*Figure 2. Decision Tree for "Refrigerator too Cold."*

automate the process as much as possible; fourth, control the quality of the system; and fifth, leverage the system and knowledge for improved impact.

Standardizing the call-taking process involved identifying all cases that can and cannot be solved on the telephone and determining the correct questions to ask and the correct order in which to ask them. Much of this knowledge was tacit knowledge (that is, personal experience of the call takers and engineers). Since different call takers would ask different questions to diagnose the same problem, we formed teams of call takers and engineers to determine what should be the correct cases, questions, and order of questions. The visualizations of the decision trees were very useful in reviewing and optimizing this knowledge. The knowledge was entered into the case base only after the engineers and call takers created these visualizations and safety, legal, quality, and consumer-service personnel validated them. This took the majority of the time needed to create the STC system.

Digitizing the inputs and outputs of the STC system was the next step. The call-taking system was modified to start up the STC system and send relevant data (for example, model number) for every telephone call. K-commerce was modified slightly so that it could accept the data. The call-scheduling agent, from figure 1, was modified to accept a large amount of data from the STC system in addition to the data it was receiving from the call taker. The STC data was captured every time the case base was used to answer a call. The data included start time, end time, customer telephone number, call-taker ID, type of appliance, a short text description of the issue, all questions asked by CBR, all answers given to these questions, the case suggested, and whether this answer was accepted by the customer.

Automating the entire telephone conversation was not possible with the state of natural language processing technology, but a team of a call taker working with the STC system can effectively automate the application of the standardized and optimized process. An attempt was made to automate the application as a web-based customer self-service tool. This was discussed in Cheetham (2003).

Controlling the quality of the system after it was created was more expensive than the initial creation but also continued to improve the success rate. Initially, each week a case author would analyze the data in the call-scheduling system for every call that was taken for the week using standardized database queries. The author was looking for any trends and especially any times a caller would not be satisfied by a suggestion. Any trends or outstanding items would be discussed in a weekly feedback meeting with the call takers. The result of the feedback meetings would be a few changes in the case base, decision trees, and rules. These changes would be made immediately and reviewed in the next week's meeting. The frequency of these reviews decreased as the case base, decision trees, and rules stabilized.

Leveraging the STC system, knowledge in

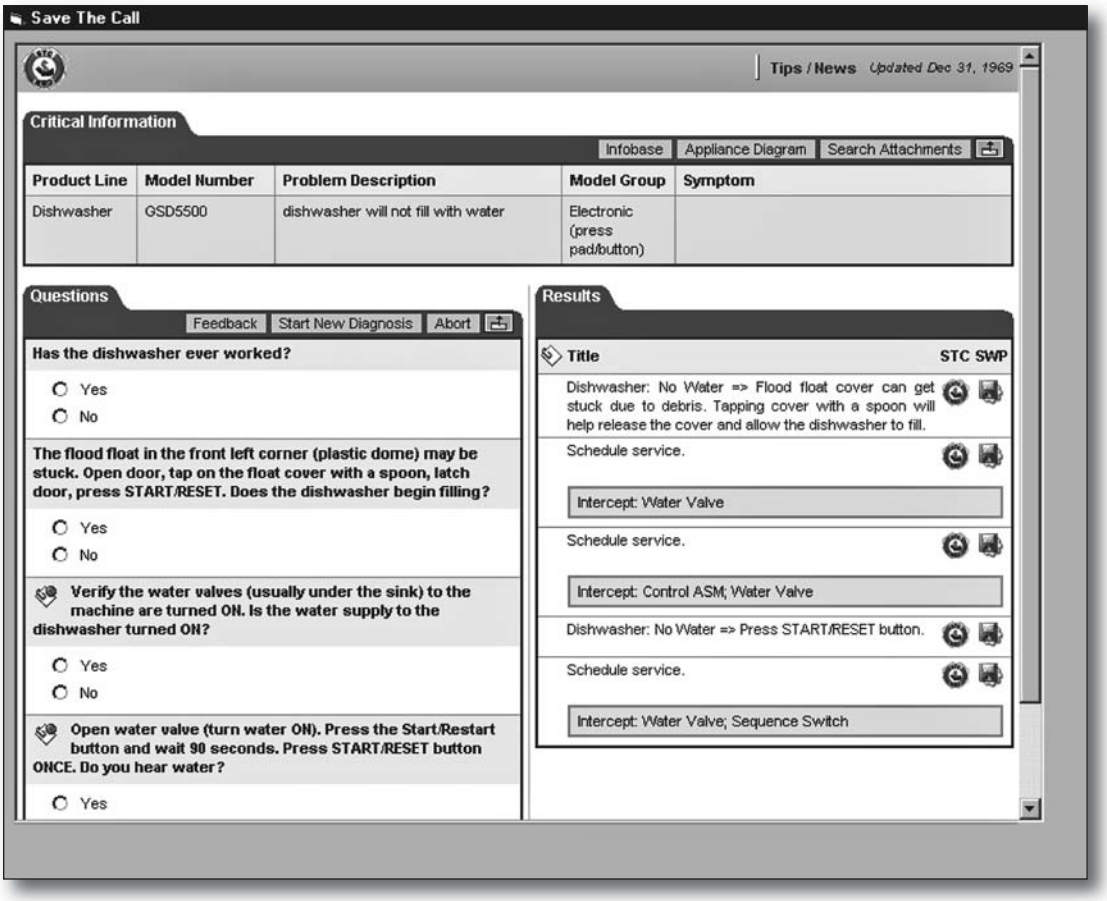


*Figure 3. Call Taker User Interface.*

the case base, and data in the call-record database provided additional benefits that were not all foreseen when the first version of STC was created. One benefit was the ability to automate the response to e-mail. It takes longer for a person at the call center to answer an e-mail message than a telephone message, and in 2000 GE's e-mail volume was growing rapidly.

We used Cisco's e-mail manager tool to store replies to frequently asked questions so that they can be reused on future similar questions. In order to select the appropriate reply, a set of decision trees was created where each node in the tree is a rule about metadata regarding the e-mail. Each leaf node in the decision tree specifies a set of replies that can be used to answer that specific type of e-mail message. These trees are similar but not identical to the trees used by STC. The e-mail handler does not automatically answer e-mail, it suggests previous replies that can be reused for a person who then just selects the correct response from a list of suggestions. The person can edit the suggested reply, create a new one, or send the suggested reply unchanged with one button click.

Figure 4 shows a case from the e-mail manager. A customer e-mailed GE saying the customer had a 50-year-old refrigerator in perfect working condition and asked if we would like to purchase it. The reply to this e-mail was stored in the case base and is now used about two times per week. Customer-support personnel either send this exact reply or edit it if the request was slightly different.

Understanding a free text e-mail message is difficult for an automated system, so we had the decision tree use only metadata that was sent with the e-mail. The metadata was created by not allowing customers to send e-mail directly to us; we made them enter the information on a form on our web site. All e-mail addresses on our products were replaced with web pages. In addition to the free text of the e-mail, customers needed to classify their messages based on a few pull-down menus on the web site and provide a telephone number. This self-classification of the e-mail created the metadata that made it much eas-

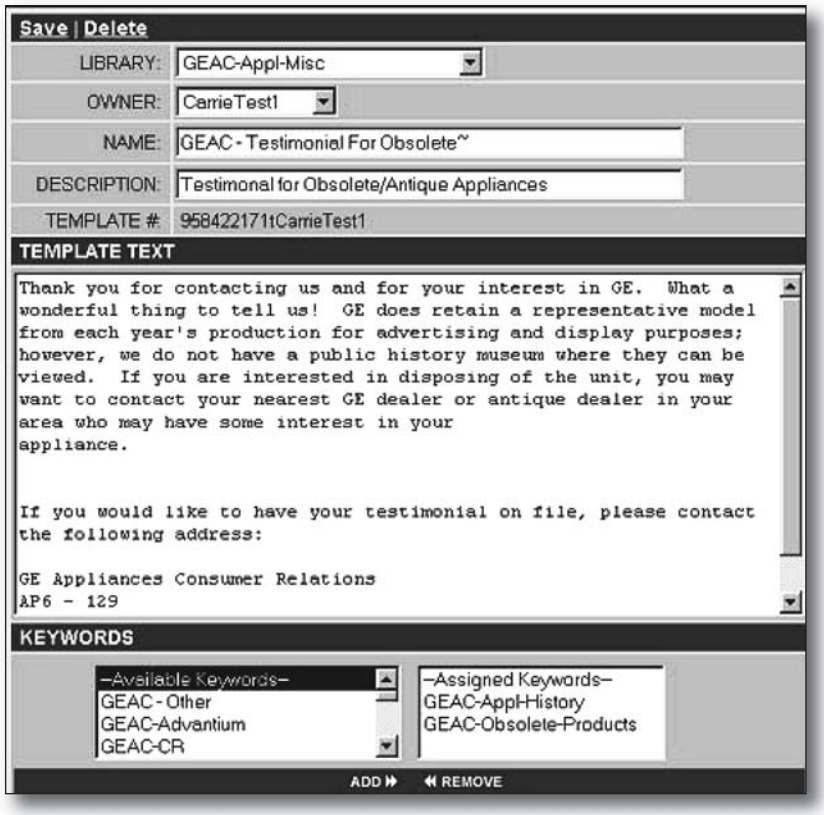


*Figure 4. E-mail Example Case.*

ier for an automated system to select similar cases.

The e-mail handling and STC tool work together. If we determine an e-mail would be better answered with a telephone call then we can use STC to help the customer. A new group of customer-support personnel was created. They do not answer calls. They call the customers themselves. We keep track of the success rate for each product/nature-of-problem combination so we know the predicted chance of solving the problem over the telephone. If the customer does not answer the telephone at our first attempt, we can repeatedly try to contact customers with a high chance of success using callback before replying with an e-mail.

## Mixed-Initiative Issues of STC

STC is a mixed-initiative conversational CBR system that acts as an intelligent assistant for the call taker. Together, STC and the call taker form a team in which each provides a needed strength where the other has a weakness. This section describes how each mixed-initiative issue (Tecuci, Boicu, and Cox 2007) was addressed in the STC application.

### Task Issue

The task issue addresses the division of responsibility between the human and the agents. To decide which tasks humans should do and which intelligent agents should do, one should determine where humans are more appropriate than intelligent agents and where agents have advantages over humans. Humans are hard to beat at common sense, interacting with other humans, and creativity. Intelligent agents, on the other hand, are good at many things that humans find cumbersome or tedious, such as following a precise process, storing and retrieving large quantities of information, and mathematical computations. While humans have a moderate cost to train and a moderate to high cost to perform a task, agents have a high cost to program for a task but a low cost to implement that program.

Tasks can be performed by a human (without an agent), an agent (without a human), or a human and an agent. Before computers, all tasks were performed without an agent. In the recent past, AI has been used to create systems in which an agent operates without a human. However, there are many tasks that can benefit from a human performing a portion of the task and an agent performing a different portion of the task. Tasks like this are good candidates for mixed-initiative systems.

In a call center, the task of helping customers requires interaction with the customer and common sense but also adherence to a process and accessing a large amount of information in a timely fashion. The call takers are good at the natural language processing that is needed to interact with the customers, but it is difficult for them to store and correctly retrieve the large amount of technical information that is needed to help the customer. Luckily, STC is very good at storing and retrieving this information even though it cannot do any natural language processing. The user (call taker) performs the interactions with the customer, and the agent (STC) stores and retrieves the standardized knowledge about diagnosing appliances. Before this system was fielded, much of the information needed to diagnose a problem was in manuals and tacit knowledge of a few call takers. Customers would often be placed on hold while the call takers looked for the correct manual and then searched for the pertinent information. Having an agent that could store this information, provide it automatically, and guide the call taker in the diagnosis removed many frustrating delays.

Another task that the agent can perform is automatically answering the questions that it is confident it can answer correctly. Some of

these questions include information that can be obtained from other sources than the customer. Information about the customer, such as models owned and previous diagnostics sessions, can be stored in a database. When questions require this information, it can be answered automatically. One future goal for high-end appliances is to include a telephone or Ethernet connection so additional diagnostic data, such as measurements from a temperature sensor, can be sent directly to the diagnostic tool, allowing it to take initiative on a wider set of questions.

## Control Issue

The control issue addresses the shift of initiative and control between the human and the agents. This primarily focuses on how the agents should show initiative, including proactive behavior. Humans often show initiative and can generally strike a good balance in judging when initiative will be useful and when it will not be useful. To that end, humans consider a range of guiding principles including: (1) Have an understanding of the goals and priorities. (2) Have an understanding of the current situation. (3) Identify a task from the current situation that can help goals. (4) Identify potential benefits and problems from doing a task. (5) Determine the confidence at which a task should be performed (high—do task, low—do not do task, and medium—ask if task should be done). (6) Have the ability to perform a task. (7) Inform others that a task has been done.

Items one, two, and three are needed to correctly identify opportunities for taking initiative. Item four gathers information about the advantages and disadvantages of taking the initiative. Care must be taken to identify the disadvantages. Possible disadvantages include bothering others, doing an unwanted task, doing a low-priority task instead of a high-priority one, and being destructive or wasteful.

The fifth item is weighing the advantages and disadvantages. If the advantages greatly outweigh the disadvantages then the initiative should most likely be taken. If the disadvantages outweigh the advantages then the initiative should not be taken. If neither the advantages nor the disadvantages outweigh the other then a human can ask another person (for example, a supervisor) if the initiative should be taken. The human needs the ability to perform the task and should inform others affected that the task has been completed.

Software agents can use the same set of guidelines when they attempt to show initiative. They should understand the goals, priorities, current situation, and set of possible initiative actions. They should be able to identify advantages and disadvantages. Disadvantages of an automated assistant taking inappropriate initiative include bothering the user, locking the user out (stealing cycles), acting in an unknown way, undoing the user's desired actions, making the user undo the agent's actions, and keeping the user from doing tasks.

The agent should be able to determine the confidence that the initiative should be taken (Cheetham and Price 2004). Then, the agent executes the action with highest confidence and informs the user of the action taken.

Since speed of call is a requirement in STC, we do not want to have the call taker ever wait for the agent. All of the agent's actions take place so quickly that they appear to be instantaneous to the call taker. This is also important for the customer's satisfaction, because we do not want to have the customer wait for the agent.

Finally, the action suggested after diagnosis is completed can be automated. The call taker could step the user through a precreated repair process or e-mail or fax the process to the user. If parts are required, the agent can place an order for the parts to be sent to the address of the customer. If a service technician is required, the time for that visit can be scheduled and a description of the problem automatically sent to the call-scheduling system.

## Awareness Issue

The awareness issue addresses the maintenance of a shared awareness with respect to the current state of the human and agents. This can be difficult. The humans and agents have completely difference senses. Humans have sight, hearing, touch, taste, and smell. Any information gathered with these senses will need to be relayed to the agent. Agents have access to digitized data sources such as sensors or databases. Any pertinent information discovered by the agent needs to be relayed to the human. Humans have experiences and knowledge that can be used to make conclusions from existing information. Agents may include an expert system that also makes conclusions, most likely in a different way from the human.

One good way to keep a shared awareness is to share three types of information: facts and beliefs, reasoning, and conclusions. To obtain a clear understanding of an agent's awareness, all three information types are needed by the human. Showing facts and beliefs allows the human to check for any that are not correct, current, or consistent with the human's facts and beliefs. The conclusions are the value that

the agent produces and need to be shown to the human. The reasoning is the bridge between the facts and beliefs and the conclusion. It can be difficult to share the reasoning of black-box techniques, such as neural nets. Therefore, black-box techniques should be used in mixed-initiative systems only when their reasoning does not need to be inspected (for example, this process does not need to be audited or is guaranteed to be correct). Two other rules of thumb on having a shared awareness are (1) do not make the agents guess at the situation, as incorrect assumptions can cause many problems in a mixed-initiative system, and (2) allow the human to inspect and change the agent's facts and beliefs.

For STC, in order for the agent to be able to make valid suggestions, it needs to have awareness of all the information that the call taker has about the problem. This requires the STC system to have all possible questions that the call taker can ask and all possible answers to those questions. We quickly found that each question needed an answer of "unknown" and that the agent needed to be able to correctly deal with these answers. For instance, the customer may not be able to answer the question "What is the model number of your appliance?" This question also led to the creation of a model number locator agent that can be used to locate the model number on any appliance.

The user must also have awareness about what the agent is doing. Since the agent can take the initiative to answer questions, the user must be able to inspect the conclusions that the agent has made. In STC, all questions that the agent has answered automatically are listed for the agent so they can be easily reviewed, but answering them does not interrupt the user. It is also important that the call taker be able to change the answers when a question is answered automatically. The user can click on the question that the agent has answered and then select a different answer. A system administrator can also inspect the facts and rules that caused a question to be automatically answered.

## Communication Issue

The communication issue addresses the protocols that facilitate the exchange of knowledge and information between the human and the agents. These protocols include mixed-initiative dialog and multimodal interfaces.

The communications from human to agent in a mixed-initiative system need to be as efficient as possible for the human and as complete as possible for the agents. These are often competing goals. The human has limited time to update the awareness of the agents. One goal of a mixed-initiative system is often to save time for the human and having the human spend a large amount of time or effort updating the agents' awareness defeats this purpose. However, the agents need as complete an awareness of the situation as possible in order to provide assistance to the human.

The communication from the agents to the human needs to be easy and unobtrusive for the human, but the human also needs to have access to all information that could be valuable. If the communication is not easy and unobtrusive for humans, they will be unlikely to use the system because the benefits to them do not outweigh these costs. One way to make the communications less obtrusive is to reduce the false alarm rate (that is, the percentage of communications when the agent needlessly communicates to the human). These needless communications just bother the human. One real deployed example of possibly bothersome communication is a voice-based warning system for the F-16 combat aircraft. The female voice tells the pilot to "pull up" from a dive or "warning, warning" when the pilot needs to pay attention to various panel displays (the nagging female voice got nicknamed "Bitching Betty" by pilots).

For STC, the primary communication from the agent to the user is a suggested question. It would be bad if there was only one question suggested and it was not correct for the given situation. Multiple methods can be used to reduce the cost of an incorrect suggested question. Some of these are providing multiple suggestions from which the user can pick, not forcing the user to act on the suggestion, and unobtrusively presenting the suggested questions to the call taker.

The primary communications from the user to the agent are the answers selected for the questions. The set of answers to a question is created so that each answer is appropriate for a different case. For instance, we do not ask for answers that are continuous variables like temperature. We do ask for ranges of temperatures where each range would have a meaning associated with it. This way no postprocessing is needed to turn the answer into a diagnostic fact.

Call takers and customers often wonder why the system is suggesting a specific question. User trust is enhanced if there is a clear explanation for why the system is taking some action. When the questions are created, we often also create an explanation for why the question should be asked. This explanation can be displayed for the call taker by clicking on

the question in the user interface. Creating these explanations in CBR systems can be dangerous. One project to create automated explanations produced explanations that were not always appropriate for the customer. For example, the explanation for why we are asking the question "Is the power cord plugged in?" was "We are trying to confirm the problem cause 'Idiot User.'"

## Evaluation Issue

The evaluation issue addresses the human and agent contribution to the overall system's performance. Evaluation of a mixed-initiative system involves evaluation of the users, agents, and the agent/user interaction. One field of study for evaluating user (that is, employee) performance is called human resource management (HRM). The basic premise of HRM is that humans should not be treated like machines. In the evaluation of an agent, it would also be good to evaluate the agent less like a machine and more like a person. In a company setting, HRM evaluates employees on two major areas: their individual abilities and how they work with others. The agent should be evaluated on both its individual abilities and how it interacts with the user. An agent will not be as effective as possible unless it is competent in both of these areas. The individual effectiveness of the agent can be measured with quantitative indicators such as speed (seconds), coverage (number of tasks the agent can complete), false alarm rate, and accuracy of results. However, the quality of its user interactions will probably need to be evaluated with qualitative indicators such as user satisfaction, ease of interaction, trustworthiness, friendliness, manners, and value provided to the user.

A mixed-initiative system can be evaluated on its speed, cost, and correctness in performing its tasks. Different systems will place different importance on each of these issues. In a mixed-initiative system there is a question of who will do each task, the user or the agent. The simple answer to this is to have the one with the lowest cost, highest speed, and highest correctness perform the task. The agent almost always has the highest speed, so it would be good to focus on how to have the agent also have a low cost and high correctness. The cost of the agent performing a task for the user can be thought of as the sum of the costs of the following five activities: (1) creating the agent; (2) the user telling the agent to do the task; (3) the agent doing the task; (4) the agent communicating the result to the user; and (5) the user receiving and validating the result of the agent.

All of these costs should be reduced to make the agent as useful as possible. A simple way to have the agent be as correct as possible is to have it only do tasks (or specific instances of a given task) where it is likely to be correct.

Figure 5 shows a generalized flow of control through a mixed-initiative system. The process can start and go to either a user or an agent. Then users and agents will work the task until either the user or the agent ends the task. Some mixed-initiative systems may require either the user or agent to start and end the task. One extreme of this flow of control is to have the user start the task, work on the task without help from the agent, and then end the task. This is the way many tasks are done before the mixed-initiative system is created. The other extreme is to have an agent start the task, work on the task without help from the user, and then end the task. This is how many traditional AI systems operate. This would usually be the lowest cost and highest speed approach, but there are many domains where the correctness of such a system would not be acceptable. So, the proper trade-off for a mixed-initiative system is to have the agent do all of the tasks where it can have an acceptable correctness at a cost that is lower than the cost of the user performing the task. This would move as much of the task as possible away from the user and to the agent.

One way to evaluate job performance from people is with a three-level scale: (1) Poor—does not finish tasks correctly or on time. (2) Good—finishes tasks correctly and on time. (3) Excellent—finishes tasks correctly and on time plus shows initiative to suggest or do other tasks.

Agents could be evaluated on a similar scale. This would require initiative from the agent in order for it to be an excellent agent.

With STC, we found that call takers quickly lost trust in the system if it gave poor suggested questions or initiative. That is why we went though the additional effort of creating a decision tree on top of the case base. K-commerce per se does not require a decision tree; it can automatically suggest questions from the most likely cases. However, at the start of the diagnosis process, it is often unclear which cases are the most likely ones. We created diagnostic trees for each possible symptom of each type of appliance to improve the correctness of the system and retain the users' trust. Figure 4 shows the tree for a side-by-side refrigerator where a compartment is too cold.

The customers' satisfaction is primarily based on the correctness, speed, and professionalism of the help provided by the call tak-

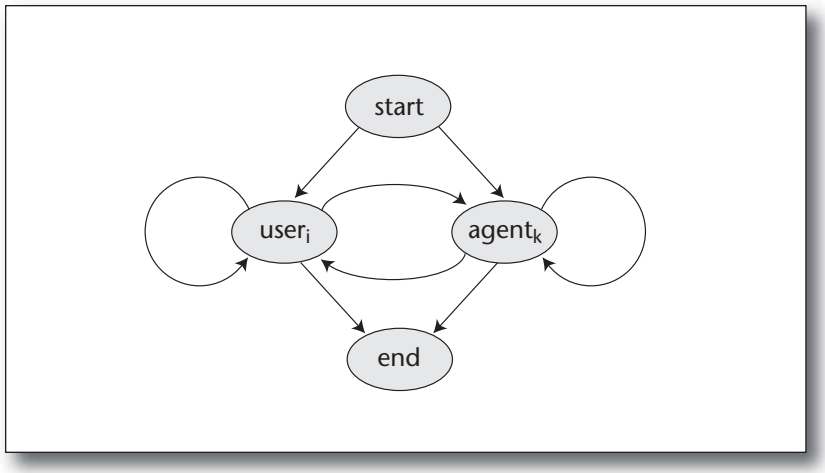


*Figure 5. User-Agent Interaction.*

er. We have already described how the system helps the correctness and speed of the process. The system also helps the professionalism of the call taker by providing the exact wording that should be used to ask a question of the customer. Call takers are given two weeks of training before they are able to take calls from a customer. This training used to involve education about the appliances and how to professionally interact with a customer. Now, training focuses mainly on professional interaction, and the system acts as "on the job" training about the appliances.

## Architecture Issue

The architecture issue addresses the design principles, methodologies, and technologies for different types of mixed-initiative roles and behaviors. Horvitz (1999) described 12 critical factors for mixed-initiative systems. These factors include considering uncertainty about a user's goals, employing dialogue to resolve key uncertainties, employing socially appropriate behaviors for agent-user interactions, maintaining working memory of recent interactions, and continuing to learn by observing.

These are excellent design principles for a mixed-initiative system. However, even with great architecture principles it is still difficult to write a good mixed-initiative agent. For example, many people do not like the Microsoft Office assistant, sometimes called Clippy.

Clippy did have many good properties, but it was removed from Microsoft Office because the value of its help was not greater than the cost of the incorrect interruptions. Many users considered a small number of incorrect interruptions or even a single incorrect interruption to be too much. It was very difficult for Clippy to give correct and timely advice at all times.

However, there is another form of mixed-initiative interaction taking place in Microsoft Office. The spelling and grammar checkers act as a simple mixed-initiative system. Microsoft Word puts a wavy line under words that may not be correctly spelled or may have inappropriate grammar. The goal of this is to help the user write with proper spelling and grammar. The user should usually want to have proper spelling and grammar, so there is little uncertainty about the user's goals. There is a dialogue available if the user clicks on the underlined word. The dialogue is a pop up that includes a guess at the correct word, the ability to "ignore" the suggestion, which removes the line, the ability to "add" the word to a personal dictionary, and a description of the grammatical rule that caused this to be underlined if it was a grammar issue. Underlining a word is socially appropriate interaction since it does not interrupt a user. Memory of the usage of the "ignore" and "add" options can allow the grammar and spelling agents to continue to learn the users' preferences.

The STC architecture puts extra emphasis on the maintainability of the system. Every time the case base was used to answer a call, a description of the call was written to a reporting database. The description included start time, end time, caller telephone number, call-taker ID, type of appliance, a short text description of the issue, all questions asked by the system, all answers given to these questions, the final diagnosis suggested, and whether this answer was accepted by the caller. Each week a case author would analyze this information in the reporting database for every call that was taken for the week. The call takers would help with this analysis and were a valuable part of the development team.

## Personalization Issue

The personalization issue addresses the adaptation of the agent's behavior to the preferences of its user. Agents can be more useful if they are personalized for the user. All users are not the same, so it can be difficult for one agent to make all users happy all of the time. One common way to personalize the actions of an agent for a specific user is to store information about the user and use this information to determine the type and timing for help the agent provides. One other way for the agent and user to work together more effectively is to have common shared experiences. This means the agent needs to remember past experiences with the user and use these to benefit the user. One simple example of this is when the tool Matlab (or other software development environments)

remembers past user commands and allows the user to quickly select and reissue these commands. This has the effect of allowing the user to quickly say, "Do it again." The past experiences can be both short-term and long-term experiences.

There can be problems with personalization. An agent should be careful about making assumptions about a user and assuming these preferences do not change. Preferences can change over time or be misunderstood by the agent, so the agent should allow users to edit their personal preferences. For example, two years ago one of the authors purchased two DVDs on potty training for his daughter from amazon.com. Amazon was still annoyingly recommending potty-training books and videos two years later. Fortunately, Amazon now does allow users to improve their recommendations by editing the list of items they have purchased or previously rated. Another example is from the TiVo digital video recorder, which suggests and takes initiative to record shows it thinks its user would enjoy. Many people have commented that from recording shows like *Will & Grace* and *Queer Eye for the Straight Guy* their TiVo now thinks they are gay (Zaslow 2002). One person reportedly tried to teach his TiVo that he was not gay by recording many things a straight man would watch like World War II movies and documentaries. Now the TiVo thinks he is a Nazi.

STC has some limited personalization capability. There is no personalization for the call taker and some personalization for the customer. The ideal personalization scenario is when customers initialize their personal information by registering the products they purchase. Alternatively, call takers will update the information for the customer when the customer calls ASI. Information about customers and the appliances they own is stored in a database. This information is used by STC when it suggests questions and solutions. In addition, past experiences of telephone conversations (questions asked and answers given) are also stored. This is beneficial if, for example, a customer is forwarded from a first-level call taker to a supervisor. In that case, all answered questions are visible to the supervisor. Having this personalized information readily available reduces customers' frustration that comes from repeating the same information. The same principle allows personalized information to be shared between different call takers. If a customer calls multiple times, then previous sessions can be reviewed to get the background on the current call. All questions and answers are stored for 30 days in one database (short-term memory) and a brief summary of the call is stored permanently in another database (long-term memory).

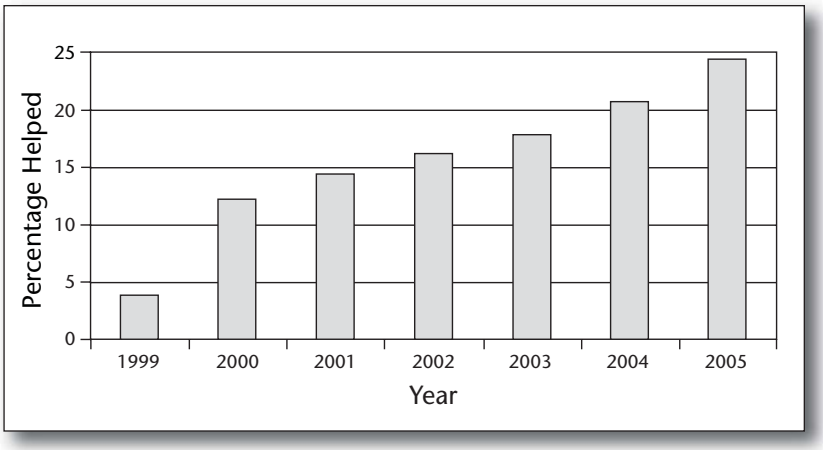


*Figure 6. STC Success Rate.*

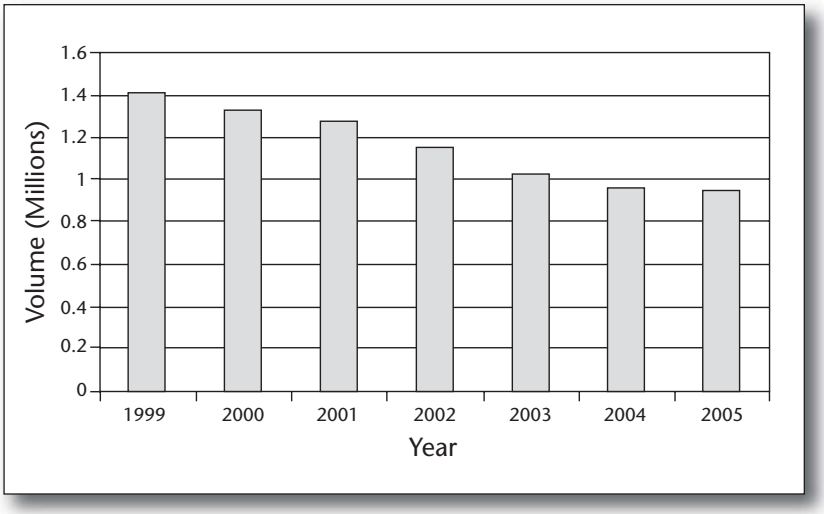


*Figure 7. Call Volume.*

## Results

STC has been in constant use since the end of 1999. More than 300 call takers at multiple locations in the United States use the system. The percentage of calls that are correctly answered over the telephone has increased each year. Figure 6 shows the success rate for each year. Frequent customer surveys show customer satisfaction is higher with cases that are answered using STC than with calls that are not.

This system has also been a financial success for GE. The initial development in 1999 cost $1 million for the software tool, Inference Corporation professional services, two person years of effort by GE personnel, and the cost of hard-

ware to deploy the system. The maintenance in each of the next six years has averaged $0.5 million for ongoing maintenance of the cases and a major upgrade to a new version of the eGain software in 2004. The benefit of not sending a field service technician to a customer's home when the product is in warranty is $50 for GE. The dollar savings for GE can be calculated by the following formula

Savings = increase in success rate * call volume * $50

Figure 7 shows the call volume per year. The sum of the savings for each year from 2000 to 2005 is $44.5 million. This is a project that can both provide better service for customers and reduce the cost of this service.

Other benefits include the following: (1) Higher first-call success rate. This increases customer satisfaction and decreases the number of calls GE needs to handle. (2) Early identification of new types of customer problems. This feedback can be sent to design teams, who can fix the problem in future releases and reduce future customer problems. (3) The ability to enforce policies such as "If the model is sold by a particular retailer *S,* then ask if it was purchased at *S." S* pays for this service, but call takers often forgot to ask, and GE ended up shouldering the cost. (4) The increased ability to mail parts out with instructions and avoid a service call for items that cannot be fixed over the telephone but are an easy fix and the customers want to do it themselves. (5) STC identifies parts needed when a field service technician is sent to the home so more fixes can be made on the first trip, saving time for the customer and field technician (the part needed is referred to as "intercept" in figure 3). (6) The increased consistency of call takers has reduced repeat calls from customers "fishing" (a practice in which customers call repeatedly in hopes to get different answers).

Our implementation of Cisco's e-mail tool went into use in June of 2000. The tool reduced the average time a person needs to answer an e-mail message by 40 percent. Since our yearly projected cost to answer e-mails was $1.5 million, the savings of this tool is $600,000 per year.

## Conclusion

GE Consumer Products has been successful using the STC mixed-initiative system to provide customer support. The system contains exhaustive knowledge about appliance problems and the questions needed to diagnose them. The human call taker provides some knowledge about appliances and a professional interface between our customers and the diagnostic knowledge stored in the tool. The team of computer system and human call taker performs better than the call takers had before the system was created.

## References


Aamodt, A., and Plaza, E. 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *AICOM* 7(1).

Acorn, T., Walden, S. 1992. SMART: Support Management Cultivated Reasoning Technology for Compaq Customer Service. In *Innovative Applications of Artificial Intelligence 4,* ed. C. Scott and P. Klahr. Cambridge, MA: AAAI Press/The MIT Press.

Allen, J. 1999. Mixed-Initiative Interaction. *IEEE Intelligent Systems* 14(6): 14–16.

Cheetham, W., 2003. Lessons Learned Using CBR for Customer Support. In *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference.* Menlo Park, CA: AAAI Press.

Cheetham, W., Price, J. 2004. Measures of Solution Accuracy in Case-Based Reasoning Systems. Paper presented at the Seventh European Conference on Case-Based Reasoning, Madrid, August 30–September 2.

Horvitz, E. 1999. Principles of Mixed-Initiative User Interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems*, 159–166. New York: Association for Computing Machinery.

Simoudis, E. 1992. Using Case-Based Reasoning for Customer Technical Support. *IEEE Expert* 7(5): 7–13.

Tecuci, G.; Boicu M.; and Cox, M. 2007. Seven Aspects of Mixed Initiative Reasoning: An Introduction to this Special Issue on Mixed Initiative Assistants. *AI Magazine* 28(2).

Watson, I. 1997. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. San Francisco: Morgan Kaufmann Publishers.

Zaslow, J. 2002. What to Do When Your TiVo Thinks You're Gay. *The Wall Street Journal,* November 26.





**William Cheetham** is a senior researcher in the artificial intelligence laboratory of the General Electric Global Research Center in upstate New York, where he has worked since 1985. His job is to invent or improve knowledge-based products and services. This often involves the application of artificial intelligence and related techniques. He has led the development of more than a dozen intelligent systems that are in use throughout the General Electric Company. Since 1998, he has been an adjunct professor at Rensselaer Polytechnic Institute, where he now teaches the class Applied Intelligent Reasoning Systems.



**Kai Goebel** is a senior scientist at RIACS, where he is coordinator of the Prognostics Center of Excellence for NASA Ames Research Center. Prior to that, he worked at General Electric's Global Research Center in Niskayuna, New York, from 1997 to 2006 as a senior research scientist. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion. His research interest lies in advancing these techniques for real-time monitoring, diagnostics, and prognostics. He has fielded numerous applications for aircraft engines, transportation systems, medical systems, and manufacturing systems. He holds half a dozen patents and has published more than 75 papers. Since 1998, Goebel has been an adjunct professor of the Computer Science Department at Rensselaer Polytechnic Institute (RPI), Troy, New York, where he has been coteaching classes in soft computing and applied intelligent reasoning systems with William Cheetham.