Using Cases to Support Divergent Roles in Distributed Collaboration*

David Leake,[1] Steven Bogaerts,[1] Michael Evans,[2] Rick McMullen,[2] Michael Oder,[1] Alejandro Valerio[1]
[1]Computer Science Department, Indiana University, Lindley Hall 215
Bloomington, IN 47405, U.S.A.
[2]Pervasive Technologies Labs, Indiana University, 501 N. Morton, Suite 224
Bloomington, IN 47404 USA
{leake,sbogaert,avalerio,moder}@cs.indiana.edu,{micaevan,mcmullen}@indiana.edu

Abstract

Case-based reasoning has been applied successfully to many diagnostic tasks, and much attention has been directed towards maximizing performance of the case-based diagnostic process. In distributed collaboration contexts, however, high-performance CBR alone may not be sufficient: individual abilities and organizational roles introduce unique constraints on how support should be applied. To maximize the usefulness of a case-based support system, system design must reflect divergent user capabilities and roles. This paper presents a case study of a CBR-based system to support collaborative distributed troubleshooting by ad hoc teams of sailors and shipboard experts. It shows how case sharing between participants can be used to increase confidence and aid situation assessment, "jump-starting" the aid process. It also shows how information from cases can be used to streamline communication between collaborators, and how the communication process needed to handle novel situations can be exploited as a natural vehicle for dialogue-driven generation of new cases to fill gaps in the existing case-base.

Introduction

Case-based reasoning (CBR) applications have been successfully developed to support human problem-solving for a wide range of tasks. The benefits of these systems depends not only on the effectiveness of the systems themselves, but also on their fit with the needs and constraints of the organization that applies them. When CBR is applied in collaborative contexts, organizational roles and policies have profound effects on how cases can be applied, and must be reflected by the system design. This suggests taking a holistic socio-technical view of how to facilitate tasks with CBR.

This paper presents a case study of the design of a case-based support system for distributed collaborative diagnosis of critical naval equipment by shipboard sailors and onshore subject matter experts (SMEs). Here the acceptance and usefulness of the system depends on more than simply the performance of its core CBR process. Beyond that process, the system must be integrated into a framework that reflects individual needs and capabilities, and must also bridge the gaps between different users' knowledge and capabilities, all within the constraints of organizational policies. This study illustrates that designing systems that reflect role-based user needs may be as crucial as addressing classic CBR issues.

In our system, collaboration takes place within a conversational case-based reasoning (CCBR) framework. In CCBR, users build up a problem description by successively answering questions, as the system incrementally ranks candidate cases and questions to ask based on the partial information available (Aha & Breslow 1997). For our task, each partner in the collaborative process has different abilities and a different organizational role. In response, we have designed a core CCBR interface with different functions for each role to support individual duties while aiding shared understanding. This approach illustrates how *cases for communication*, combined with *role-based access to and manipulation of captured cases*, can help provide appropriate support for divergent users in distributed collaboration.

Socio-Technical Approaches to System Development

Our approach to supporting sailor-SME collaboration arises from a socio-technical analysis of their needs and the capabilities of practical technology. The phrase "socio-technical systems" was coined in the 1950's by Eric Trist and Ken Bamforth at the Tavistock Institute of Human Relations in the UK (Trist & Bamforth 1951). The term referred to the interdependent nature of work, structure, technology, and change in organizations. Trist and Bamforth discovered that the efficient use of coal-getting technology was influenced by the social relationships of workers and superiors. In the 1970's and 1980's the basic principles and methodology of socio-technical design for information systems were developed in seminal works including (Cherns 1976; Mumford 1983; Eason 1988).

The advantage of a socio-technical approach to system development is that it avoids potential pitfalls from a piecemeal, deterministic view of technology. The socio-technical approach recognizes that the design, adoption, and use of new technologies are susceptible to organizational con-

* This work is supported in part by the U.S. Department of the Navy, NSWC Crane Division, under contracts N00164-04-C-6514 and N00164-04-C-6515. We thank Navy personnel and lab colleagues for aid and participation in this project. We also thank the anonymous reviewers for helpful comments.

straints. Following this approach, to develop a CBR system for naval technicians we examined three types of organizational features: strategy, structure, and task complexity.

Strategically, the U.S. Navy intends to leverage expertise by developing advanced information technologies to encourage knowledge generation and distribution. Consequently, our system has been designed to ensure that users are able to communicate and access a rich store of troubleshooting experience to which they may add (Evans 2004). Structurally, the Navy has organized its troubleshooting units in an ad hoc fashion. Consequently, our system must be flexible enough to serve users from a variety of functional contexts and levels of expertise. Finally, the task of troubleshooting at a distance is both unpredictable and hard to codify. This entails that the system supplement codified knowledge with support for use of tacit knowledge (e.g., encouraging remindings and associations), the latter being especially useful in times of improvisation and innovation.

The Collaborative Distributed Troubleshooting Task

In current U.S. Navy practice, non-expert sailors troubleshoot complex electronic systems by relying on standard diagnostic flowcharts and technical manuals to isolate one or more detected faults. This prescribed troubleshooting process is most effective for documented faults. When standardized methods and support materials are exhausted, the attending sailor contacts a remote shore-based SME for assistance. For this interaction, sailor(s) and SME(s) often have only basic media support, in the form of email, chat, and paper documents. In order to resolve a problem successfully: 1) the SME must request a detailed account of all actions taken prior to the request for technical assistance; 2) the sailor must convey any unique aspects of the operating environment, including recent upgrades and climatic conditions, 3) based on data received, the SME must diagnose the problem and prescribe corrective actions; and 4) the sailor must correctly perform the corrective actions and either confirm successful resolution of fault(s) or request further assistance (Evans 2004).

Plans for significant staffing reductions in the U.S. Navy, combined with increasingly complex shipboard systems, loss of experience due to job turnover and retirement, and the difficulty and expense of extensive training, make shipboard automation a major priority. The success of CCBR in diagnostic applications (see (Watson 1997) for samples) and the comparative simplicity of leveraging experience and learning in CBR systems suggested applying these methods to the diagnosis of shipboard systems.

Challenges for Shaping Case-Based Support to Fit Navy Policies

In conversational CBR diagnosis systems, cases for previously-diagnosed problems suggest questions to ask a user, to guide retrieval of similar prior cases suggesting diagnoses for the current problem. This appears to provide a natural replacement for the sailor's flowchart-following process, and initial efforts to build a CCBR system for this task were encouraging, both from the viewpoint of technical promise and for reactions of potential end users.

Unfortunately, when a prototype system was demonstrated, two problems arose. First, the actors in the Navy process are bound to act in accordance to institutional policies. In particular, sailors are *required* to follow standard diagnostic flowcharts; they do not have the authority to skip portions of the standard flowcharts as dictated by the choice of questions in CCBR. Second, learning from a sailor's diagnostic experiences is problematic as well. When the systems being diagnosed play a vital role in the ship's mission, confidence in system advice is crucial. If sailor cases were captured and reused directly, the accuracy of cases could not be assured due to the sailors' lack of expertise in and authority for solution certification. Thus simply providing the sailor with case-based support was impossible.

In the naval process, only the SME is empowered to go beyond the standard troubleshooting procedures. Thus CCBR support can be provided to the SME. We decided to study how to maximize support for both the sailor and SME, *in the context of their interaction*. Although the sailor is not allowed to use case-based reasoning directly, *cases* can still support his or her role by aiding the sailor's capture and communication of information, in addition to the more traditional CBR tasks provided to the SME, as discussed in the following section.

Opportunities for Supporting Collaborative Distributed Troubleshooting

A first step towards developing support for a collaboration is to examine the needs of the individuals involved. Previous studies note that ad hoc teams are susceptible to a number of difficulties arising from interaction problems and other limitations. In our task context, these difficulties include:

Forgetfulness: Diagnostic actions performed by the sailor may be forgotten and not conveyed from sailor to SME.

Insufficient Knowledge: Sailors may need additional support to carry out tasks or describe the problem effectively.

Unfamiliarity With Personnel: The SME may lack a clear picture of the sailor's level of understanding.

Lack Of Confidence: The SME does not know what the sailor has truly done, only what the sailor says he or she has done. Thus, it is often necessary for the SME to ask the sailor to repeat steps.

Communication Breaks: External events may impose delays in the interaction, making it difficult to maintain mutual understanding.

These difficulties can be summarized as reflecting a *communication gap* between the sailor and SME. For example, in own studies a shore-based SME of seventeen years' service reported:

> *"I'll be honest: [remote troubleshooting is] really difficult. But, the key is the technician on the other end, you know... I mean you can have all the documentation in the world and if you don't have an experienced tech on the other end of the line, you can do Distance Support*

until you're blue in the face. But, if he can't describe the problem to you [then it will inevitably remain unresolved]."

Thus bridging the communication gap between sailor and SME is crucial. We established the goal of providing an integrated CCBR interface, following institutionally-sanctioned practice, to meet this need.

Our approach centers on using cases as a shared basis for collaboration, with information added to or accessed from cases in light of individual participants' roles. A key contribution is that the use of cases is not restricted to a single centralized problem-solving system. Instead, cases are used as a vehicle for knowledge sharing between users, with aspects of CBR processing exploited when helpful, and case information used in different ways at different parts of the process.

System assistance centers on four phases of the collaborative distributed troubleshooting process, each one supported by different methods, as listed below. For each, we summarize the goals of the assistance and the methods we selected to address them. The specific communication difficulties that these address are listed in parentheses. In the following section we describe the resulting process and support in more detail.

During initial sailor troubleshooting: Aid the sailor following standard procedures (insufficient knowledge) and automatically log actions (sailor forgetfulness, lack of confidence).

Method: Present diagnostic questions in a CCBR-like interface, capturing responses to form an initial case.

During establishment of sailor-SME contact: Aid initial communication(forgetfulness), SME situation assessment and confidence in the result (lack of confidence), and continuity over multiple episodes (communications breaks)

Method: When a sailor initiates contact, automatically transfer accumulated case to the SME, giving immediate and accurate knowledge about the steps the sailor traced. This case remains accessible as a reference whenever contact is reinitiated after an interruption.

During collaborative troubleshooting:

- Aid the SME's diagnostic task.
  Method: Use data from the sailor's case as retrieval cues to suggest possible cases for the SME to pursue.
- Streamline further communication.
  Method: Allow the SME to select questions from existing cases to dispatch to the sailor, for presentation in the sailor's conversational interface, and transmit responses while recording them as additions to the sailor case.

During knowledge update: Aid the SME in capturing useful knowledge.

Method: At the end of the diagnostic process, present the new case, including SME-selected additional questions, to be vetted by the SME.

A Detailed Walkthrough of System Use

In the implemented system, the sailor and SME interact with the system according to the following process:

Standard Troubleshooting: When a fault occurs for a piece of equipment, the sailor is dispatched and performs standard troubleshooting as guided by the system.

- The system selects questions to present to the sailor one at a time based on standard troubleshooting flowcharts.
- For each question, the sailor is presented with related resources to assist in answering the question (e.g. schematics, additional directions).
- The sailor provides answers to these questions. These answers are added automatically to the initial problem description of the case under development for this troubleshooting session, for later storage as a new case.
- Often, after a series of questions, the flowchart will dictate a diagnosis and corrective action.

Establishment of Contact with SME: If the sailor exhausts standard procedures, the SME is contacted.

- The preliminary case (problem description), based on the actions of the sailor so far, is presented to the SME. Similar cases are automatically retrieved from the case base.
- The SME may optionally suggest a diagnosis and corrective action. These could come from either a similar case or personal experience.
- If the SME is not prepared to suggest a diagnosis and corrective action, additional questions may be prepared for the sailor in order to further assess the problem. These questions can come from similar cases, the question collection at large, or any new questions the SME wishes to define based on personal experience.
- The SME sends a response to the sailor, containing either the diagnosis and corrective action, or further questions.

Sailor Response To SME: The sailor receives the SME's advice.

- If the SME sent additional questions, the sailor answers them, within the same interface that the standard troubleshooting procedures were done. These answers are added to the current case.
- If the SME sent a diagnosis and corrective action, the sailor performs the corrective action and checks if the fault has been resolved. If the fault has not been resolved, the failed diagnosis and corrective action are added to the case's description. If the fault has been resolved, this result is prepared for communication to the SME.
- The sailor's answers, or the results of the corrective action, are sent back to the SME in the form of the case.

SME Receives Response: The SME receives the sailor's response to the previously sent advice.

- If questions were answered, or a corrective action failed, the SME continues troubleshooting as in the *Initial Contact with SME* step above.

- If the corrective action succeeded, the SME has the option of indicating the diagnosis and action as the case solution. The case may be further edited by the SME, removing features deemed irrelevant for the problem description. The case can then be added to the case base for use in future problem-solving episodes.

Interface Design

To support the above processes, sailor and SME interfaces must enable multiple types of interactions and use of multiple types of information. Both the sailor and SME interfaces support system question-asking and answer capture, following the standard CCBR model, but how these are used and the additional capabilities they provide reflect specific user roles.

Sailor Interface The sailor interface contains the following components:

- Conversational flowchart guidance (see figure 1a.): The sailor answers questions, either from standard procedures or from those suggested by the SME. The sailor can view the problem description up to this point as well as select resources related to the current question to be displayed.
- Conversational CBR: We have implemented CCBR in a version of this interface, to demonstrate that if direct CCBR support were allowed for this "first-line" troubleshooter, it could be seamlessly integrated with the flowchart process by retrieving cases, based on accumulated answers to questions, if the end of a flowchart is reached without a diagnosis. This would enable deflecting SME contacts for situations in which CCBR is sufficient. However, we expect institutional constraints to preclude its inclusion in the fielded system.
- Resource Search: Flowchart steps may be annotated with links to resources, and the sailor may do a free-text search through on-line resources (e.g., technical documents and "business cards" for relevant personnel) for additional assistance.

SME Interface The SME interface is divided into the following:

- Current Tech Assist: The view of the current problem (tech assist), any notes entered by the sailor or SME, and the solution, if one has been reached.
- Related Tech Assists: (see figure 1b.) A ranked list of cases similar to the current problem.

  Details of any case may be viewed, with a color-coded comparison of features matching and mismatching the current problem. Any question answered in a case may be queued to be sent to the sailor.

  The bottom portion of this screen lists each feature in a specific case selected from the list, color-coded—Green features have the same values as their counterparts in the current case, while features marked in red have values that do not match. The white features represent questions that have not yet been asked in the current troubleshooting session. The user may then select one of these features and add it to a queue of questions to be sent to the sailor.

  For questions that the SME would like to ask that are not part of prior cases, the SME may browse the collection of questions, and also define entirely new ones. Additional unstructured communications become annotations of the case, potentially available for future examination or analysis.
- Continue Troubleshooting: A viewer for available questions and questions queued to be sent to the sailor. Additional questions may be added to the queue.
- Suggest Resolution: An interface to allow the SME to suggest a diagnosis and corrective action for the current problem.

After problem resolution, the SME can access an additional interface to vet cases, reviewing questions answered by the sailor, selecting questions to retain/discard by clicking (e.g., to delete unfruitful paths), and choosing whether to store the cases.

Reflecting Divergent Needs

The implemented approach addresses the sailor and SME's divergent needs in many ways:

Sailor diagnostic support: The system directly supports the sailor's routine tasks by automatically guiding the sailor through a flowchart and making technical documents available as needed.

Knowledge capture for future sailor diagnostic support: The system supports sailor organization/grouping of accessed documents, to automatically make useful collections available in the future.

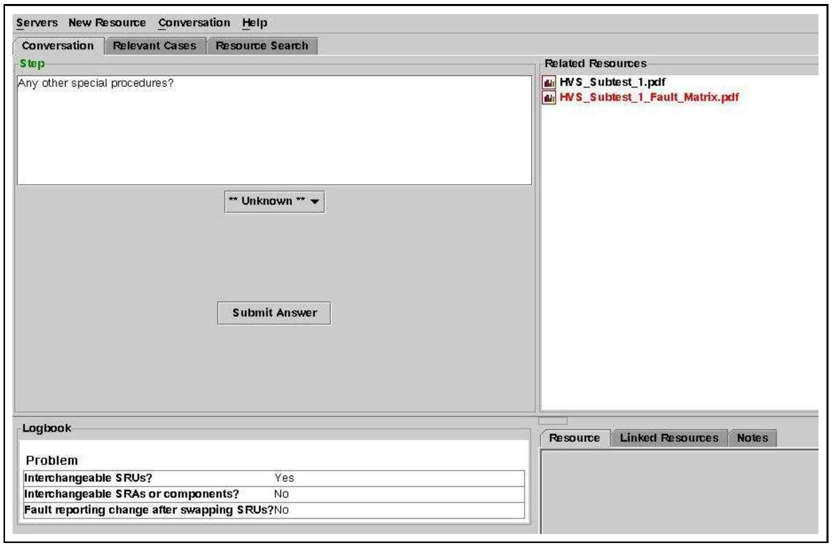
Information capture from sailor, for use by SME: The system captures each sailor response, in order to provide a trustworthy record of the sailor's task context. In this way, case generation relieves the sailor of the need to record the troubleshooting sequence and aids the sailor's role of being the "eyes and ears" for the SME.

Communication of context sailor → SME: When standard flowchart-based diagnostic procedures are completed without problem resolution, the system provides the complete case record to the SME. In this way, the case is used to alleviate the need for the sailor to act as a reporter.

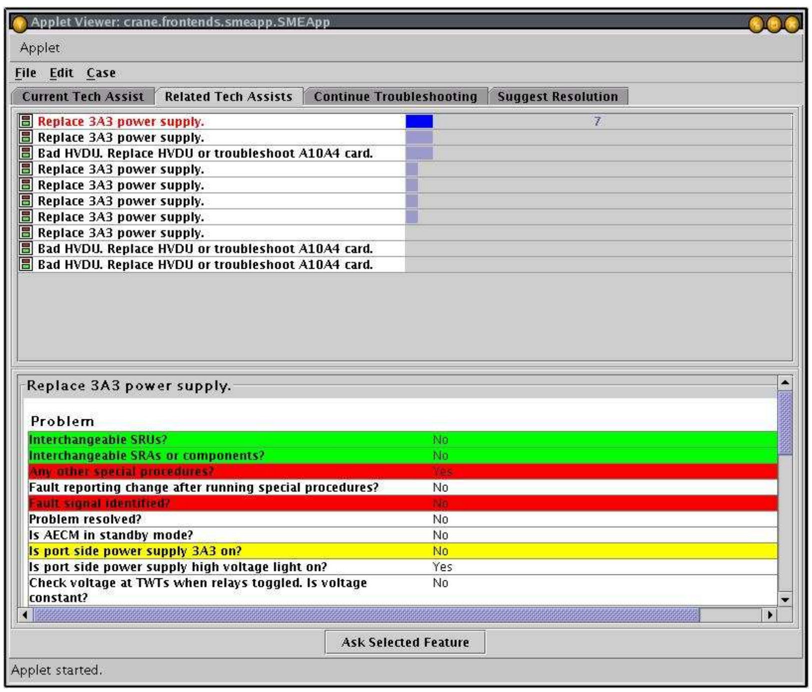
SME diagnostic support: The system provides the SME with standard CCBR support for diagnosis.

Communication SME → sailor: The system streamlines SME information-gathering by enabling the SME to select questions from existing cases—either from those highly ranked by the CBR system, or from others the SME may choose due to his or her own experience and knowledge—to send to the sailor.

Support for knowledge capture for SME use: After problem resolution, the system supports analysis of the case record by the SME, to select key diagnostic features to add to the case base, distilling and distributing knowledge.



(a) Portion of interface for supporting sailor.



(b) Interface for supporting SME.

Figure 1: . Interfaces for supporting sailor and SME reasoning and communication.

Perspective

Conversational case-based reasoning has been extensively applied to diagnostic help desks. Heider (1996) describes the application of CBR and induction in a system that constructs fault trees, similar to the fault trees taken as the starting point for our diagnosis task. The APPLICUS project applied CBR to robot maintenance (Bartsch-Spörl 1997), in circumstances similar to those of our project, including distance support under organizational constraints. Both of the above systems stem from the INRECA projects (Bergmann

2001), which illuminated (among many other points) the importance of knowledge refinement by an expert. In our project, refinement is accomplished by the SME's verification and final selection of attributes to be stored with a case.

Our work emphasizes the role of cases in supporting a broader process, in light of institutional factors that constrain the application of CCBR technology. The lessons of this project relate to a number of topics of interest for CBR:

- The possibility of, potential need for, and value of integrating limited facets of CBR into parts of the work process for which full CBR is not allowed, as done in the sailor's CCBR-style interface and sailor experience capture.
- That case building by automatic user tracking can aid a natural transfer of problems between collaborating participants, as shown here aiding situation assessment by the SME and increasing SME confidence.
- That the roles of the expert and the CCBR system should be cooperative, as illustrated here with the SME provided support to easily merge case-provided knowledge with his or her own knowledge, supported by capture and editing for these new cases.
- That a natural approach to case acquisition is to track questions from a human diagnostic dialogue—and that the use of stored questions from prior cases to support the dialogue can facilitate that process. This can make building of new cases an extension/revision process starting from near-miss cases.

A number of issues remain for future study. For example, the current design relies on the SME to vet cases, but does not address possible conflicts between different SMEs' judgments of important features, etc. Likewise, as SMEs add features, tools are needed to help standardize their feature choices, to make sure that related cases remain comparable. Such consistency issues have been previously identified in case-base maintenance research (e.g., (Racine & Yang 1996)), as have additional issues in managing the development of a case base and CBR system as a whole (Leake *et al.* 2001). We are examining the application and extension of methods to address such problems.

Another class of problems that we identified remains to be addressed as well: The problems which may arise due to lack of familiarity between participants in an interaction. We anticipate that by various methods, such as examining stored training records or assessing the sailor's needs to consult additional resources, it will be possible for the system to present the SME with not only a problem case, but also with information about the sailor with whom the SME is communicating. How to generate this information, and how to help the SME tailor responses and suggest appropriate resources to aid the sailor, is a future area for our research.

Conclusion

This paper has described a system to support knowledge capture, transfer, and sharing as it provides a vehicle for communication between sailors and SMEs. The system captures and conveys information about previous diagnoses, supporting the SME with specific case information from the sailor, and with a case-based reasoning (CBR) system to provide suggestions. When existing cases are on-point, the SME can satisfy institutional confidence requirements by simply certifying the solution; when existing cases are not, they nevertheless may suggest useful questions. This facilitates the SME's own question-asking process—a process that is automatically used to build new cases for the future. Informal feedback from potential users has been positive, and we have developed plans for a formal evaluation to quantitatively assess the system benefits. We believe that systems aimed at case-based support for collaboration, reflecting overarching institutional constraints, are a promising avenue for added acceptance for real-world CBR.

References

Aha, D., and Breslow, L. 1997. Refining conversational case libraries. In *Proceedings of the Second International Conference on Case-Based Reasoning*, 267–278. Berlin: Springer Verlag.

Bartsch-Spörl, B. 1997. How to introduce CBR applications in customer support. In *Proceedings of the 5th German CBR Workshop*.

Bergmann, R. 2001. Highlights of the european INRECA projects. In *Proceedings of the Fourth International Conference on Case-Based Reasoning*, volume LNCS 2080, 1–15. Berlin: Springer-Verlag.

Cherns, A. 1976. The principles of socio-technical design. *Human Relations* 28(8):783–792.

Eason, K. 1988. *Information technology and organisational change*. London: Taylor and Francis.

Evans, M. 2004. *Knowledge and Work in Context: A Case of Distributed Troubleshooting Across Ship and Shore*. Ph.D. Dissertation, Indiana University.

Heider, R. 1996. Troubleshooting cfm 56-3 engines for the boeing 737 using cbr and data-mining. In *Proceedings of the Third European Workshop on Case-Based Reasoning*, volume LNCS 1168, 512–523. Berlin: Springer-Verlag.

Leake, D.; Smyth, B.; Wilson, D.; and Yang, Q., eds. 2001. *Maintaining Case-Based Reasoning Systems*. Blackwell. Special issue of *Computational Intelligence*, 17(2), 2001.

Mumford, E. 1983. *Designing Human Systems*. Manchester, UK: Manchester Business School Publications.

Racine, K., and Yang, Q. 1996. On the consistency management of large case bases: The case for validation. In *Proceedings of the AAAI-96 workshop on Verification and Validation*. Menlo Park, CA: AAAI Press.

Trist, E., and Bamforth, K. 1951. Social and psychological consequences of the Longwall method of coal-getting. *Human Relations* 4:3–38.

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