Artif Intell Rev
DOI 10.1007/s10462-007-9033-6

Autonomic self healing and recovery informed by environment knowledge

Sa'adah Hassan · David McSherry · David Bustard

Received: 1 October 2006 / Revised: 30 October 2006

Abstract An important goal of autonomic computing is the development of computing systems that are capable of self healing with a minimum of human intervention. Typically, recovery from even a simple fault will require knowledge of the environment in which a computing system operates. To meet this need, we present an approach to self healing and recovery informed by environment knowledge that combines case based reasoning (CBR) and rule based reasoning. Specifically, CBR is used for fault diagnosis and rule based reasoning for fault remediation, recovery, and referral. We also show how automated information gathering from available sources in a computing system's environment can increase problem solving efficiency and help to reduce the occurrence of service failures. Finally, we demonstrate the approach in an intelligent system for fault management in a local printer network.

Keywords Autonomic computing · Self healing · Case based reasoning · Rule based reasoning · Fault diagnosis · Fault management

1 Introduction

Inspired by the autonomic nervous system's ability to control major functions without requiring conscious thought, the goal of IBM's autonomic computing initiative is the development of *self managing* computing systems (Horn 2001; Telford et al. 2003; Kephart 2005; Ganek 2007). The aspect of self management on which we focus in this paper is *self healing* (i.e., fault detection, diagnosis, and remediation with a minimum of human intervention). As noted

S. Hassan · D. McSherry (✉) · D. Bustard
School of Computing and Information Engineering, University of Ulster, Coleraine, Northern Ireland BT52 1SA, UK
e-mail: dmg.mcsherry@ulster.ac.uk

S. Hassan
e-mail: hassan-s@ulster.ac.uk

D. Bustard
e-mail: dw.bustard@ulster.ac.uk

by Bustard et al. (2006), a related issue that has received less attention in autonomic computing is fault recovery (i.e., solving a user's immediate problem when a service failure occurs). Other self-managing properties include *self configuration*, *self optimization*, and *self protection* (Horn 2001). An autonomic computing system must also be aware of the *environment* in which it operates and have the ability to learn from experience.

Autonomic computing is emerging as an important application domain for artificial intelligence in which case based reasoning (CBR) is likely to play a prominent role, not least because of the ability to learn from experience (Anglano and Montani 2005; Kephart 2005; Montani and Anglano 2006). CBR is an approach to problem solving in which previous problems and their solutions are stored in a case base and retrieved in response to a query describing a target problem (López de Mántaras et al. 2005). The solution from the most similar retrieved case is reused as a solution to the target problem, a process that may involve adaptation of the solution to account for differences between the target problem and the most similar case.

In conversational CBR, a description of the target problem is elicited in an interactive problem solving dialogue, often with the aim of minimizing the number of questions required to discriminate between competing cases (Aha et al. 2001; McSherry 2001; Aha et al. 2005). Conversational CBR has been applied with considerable success in domains such as interactive fault diagnosis and customer help desk support where it is often not possible for users to provide a complete description of a target problem (Watson 1997; Aha 1998). In autonomic computing, a similar requirement for user involvement in the problem solving process arises in the case of faults that rely on user feedback for their detection and diagnosis (Crapo et al. 2003; Bustard et al. 2006).

In contrast to problem solving domains such as planning and design, there is usually no adaptation of previous solutions in CBR systems for fault diagnosis. However, a minimum level of similarity is often required for the solution from the most similar case to be applied to the current problem. If no case reaches the required similarity threshold, the problem is referred to a domain expert, a step known as problem *escalation* in help desk applications (Watson 1997). When the problem has been solved by the domain expert, it is stored with its solution in the case base so that similar problems can in future be solved with no need for problem referral (Fig. 1).

The ability to learn (i.e., acquire new cases) through referral of problems that the system is currently unable to solve is an important advantage of CBR. In applications such as printer

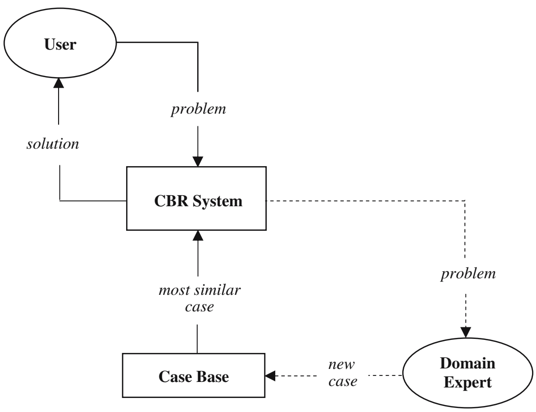

Fig. 1 Learning through problem referral in CBR

fault diagnosis, however, a CBR system's task is usually limited to diagnosing the fault (e.g., out of toner) and perhaps suggesting a standard remedy (e.g., replace toner cartridge). It is typically assumed that a replacement toner cartridge is available, that a qualified person is available to install it, and that the user has time to wait for this task to be completed. In practice, a more convenient recovery strategy for the user might be to redirect her print job to another available printer. However, identifying appropriate *remedial actions* and *recovery strategies* requires detailed knowledge of the computing system's constantly changing environment:

- Who is responsible for maintenance of the printer?
- Is he or she available?
- What alternative printers are available and where are they located?

In this article, we present an approach to self healing and recovery informed by environment knowledge (SHRIEK). The use of CBR to facilitate fault diagnosis is combined in SHRIEK with the use of rule based reasoning to guide fault management tasks such as fault recovery (i.e., solving the user's immediate problem) and fault remediation (i.e., correcting the fault). We demonstrate the approach in an intelligent system called *Shriek*-P for fault management in a local printer network. Environment knowledge in *Shriek*-P (e.g., printer locations and staff responsibilities) is maintained separately from the rules used to apply the knowledge in fault recovery, remediation, and referral. Some elements of environment knowledge required for self healing and recovery are continuously updated in *Shriek*-P by automated information gathering from available sources in the computing environment.

When a user seeks assistance with a printing problem, *Shriek*-P's initial goal is to diagnose the cause of the problem. On completion of a problem solving dialogue, driven by conversational CBR, *Shriek*-P reports its diagnosis, if any, and uses its environment knowledge to:

- Trigger remedial actions (e.g., notifying a technician that the printer is out of toner) or referral of the problem if a diagnosis cannot be reached
- Identify possible recovery strategies that the user may wish to consider (e.g., redirect her print job to another printer, wait for the toner cartridge to be replaced, or tackle the problem herself)

In Sect. 2, we describe the view of fault management on which our approach to autonomic self healing and recovery is based. We also describe the tools and techniques used to implement *Shriek*-P, our intelligent system for fault management in a local printer network, as an interactive Web application. In Sect. 3, we describe our approach to modeling environment knowledge required for self healing and recovery in the printer troubleshooting domain. In Sect. 4, we describe how conversational CBR is seamlessly integrated with rule based reasoning informed by environment knowledge in *Shriek*-P. Related work is discussed in Sect. 5 and our conclusions are presented in Sect. 6.

2 Autonomic self healing and recovery

Autonomic self healing and recovery requires an approach to fault management that is proactive, non stop, and informed by environment knowledge. Elements in the environment of a computing system include users, technical support staff, and resources such as printers, databases, and software. Our approach to modeling the environment knowledge required for fault management in a local printer network is described in Sect. 3. Some elements of environment knowledge are relatively static (e.g., printer locations, staff responsibilities), while others can be dynamically updated by automated information gathering. For example, toner level and status information can be obtained directly from printers in the network, while staff availability can be checked from login records.

2.1 Fault management cycle

We view autonomic fault management as a cycle of five related tasks or goals, all of which may benefit from environment knowledge (Table 1). Of course, not all tasks in the cycle may be necessary for a given fault. Fault referral is needed when a diagnosis cannot be reached, or when an appropriate remedial action or recovery strategy cannot be identified. Potential service failures can sometimes be detected by automated information gathering, and remedial (or *preventive*) action taken before a service failure occurs. For example, when a printer runs out of paper, there is no need for fault diagnosis. Instead, remedial action (e.g., informing a technician) can be taken immediately.

Often in practice, fault remediation and recovery are very different goals. Though not a requirement often made explicit in autonomic computing, fault recovery is particularly important in a computing *service* such as that provided by a local printer network (Bustard et al. 2006). In contrast to fault remediation, fault recovery may often be possible even if a fault cannot be diagnosed. For example, most problems that users are likely to experience in a local printer network can be solved at least temporarily by using another printer.

Examples of the types of environment knowledge required by tasks in the fault management cycle are shown in Table 1. The potential benefits of an approach to self healing and recovery informed by environment knowledge are most apparent in fault remediation, recovery, and referral. For example, a possible recovery strategy for a user with a printing problem may be to wait for the person responsible for maintenance of the printer, if available and aware of the problem, to correct the fault. However, some of the elements of environment knowledge (D) that are dynamically updated by automated information gathering (e.g., printer status and toner level) are also relevant in fault detection and diagnosis, and so these aspects of fault management also stand to benefit from the environment awareness provided by automated information gathering.

Table 1 Fault management tasks and related elements of environment knowledge (D = dynamic, S = static) in the printer troubleshooting domain

	Environment knowledge
Fault detection (detecting system errors and service failures)	Printer status (D) Toner level (D)
Fault diagnosis (identifying the cause of the problem)	Printer status (D) Toner level (D)
Fault remediation (taking action to correct the fault)	Printer name (S) Printer location (S) Staff responsibilities (S) Staff availability (D)
Fault recovery (solving a user's immediate problem)	printer locations (S) Staff responsibilities (S) Staff availability (D)
Fault referral (referring an unresolved problem to a human expert)	Printer name (S) Printer location (S) Staff responsibilities (S)

2.2 Printer troubleshooting

Our approach to printer troubleshooting in *Shriek*-P recognizes that many problems cannot be solved entirely without human intervention, such as faults that rely on user feedback (e.g., print quality) for their detection and diagnosis, or previously unseen faults that can be diagnosed only by a human expert. There are also many faults that cannot be corrected until remedial action is taken by a computer engineer or technician. In the case of faults that rely on user feedback for their detection and diagnosis, a problem solving dialogue driven by conversational CBR is initiated in *Shriek*-P when a user seeks assistance with a printing problem. As described more fully in Sects. 3 and 4, diagnosis of the fault is followed in our approach by the identification of remedial actions and recovery strategies using rule based reasoning informed by environment knowledge. If a diagnosis cannot be reached, the fault is referred to the technician responsible for maintenance of the printer, and the user may be given the option of redirecting her work to another available printer.

Figure 2 shows the basic architecture of *Shriek*-P, our interactive Web application for printer troubleshooting in a local printer network. PHP (Hypertext Preprocessor) and MySQL are used to create and maintain the case base for fault diagnosis and the environment knowledge base (e.g., printer locations, staff responsibilities). Rules for application of environment knowledge in fault remediation, recovery, and referral (Sect. 3) are embedded in the PHP code. Automated information gathering from printers in the network is based on SNMP (*Simple Network Management Protocol*), while WSH (*Windows Script Host*) is used to access information from the client computer (e.g., installed printers, default printer).

Figure 3 shows a snapshot of a printer troubleshooting dialogue in *Shriek*-P. Having obtained the printer's current status (*ready*) and toner level (*okay*) by automated information gathering, *Shriek*-P is able to infer that the printer is switched on and is now asking the user if the printer is working. If the user answers *unknown* to any question, the dialogue moves on to the next most useful question. A detailed account of our approach to conversational CBR in *Shriek*-P (e.g., case structure, similarity measure, dialogue inferencing, question selection strategy) is provided in Sect. 4.

3 Modeling environment knowledge

Figure 4 shows how relevant environment knowledge in the printer troubleshooting domain is represented at a conceptual level in our approach to autonomic self healing and recovery. For example, Mary's default printer is printer-1 which is located in Room 14. Its toner level is *okay* and its current status is *ready*. The technician responsible for maintenance of

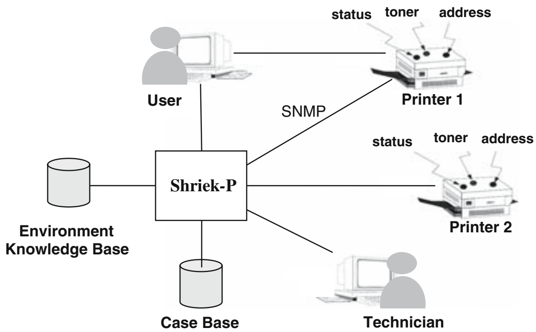

Fig. 2 Fault management in a local printer network

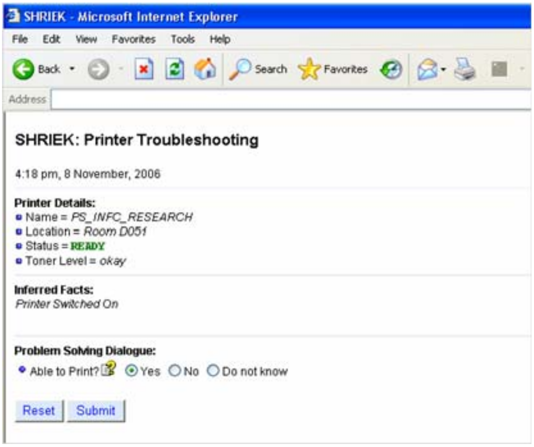

Fig. 3 A printer troubleshooting dialogue driven by conversational CBR

printer-1 is John, and he is currently available. Another printer called printer-2 is available in Room 11.

Some of rules that *Shriek*-P uses to apply its environment knowledge in fault remediation, recovery, and referral are shown in Fig. 5. For example, Rules 1 and 2 trigger remedial actions when a printer fault has been diagnosed or a potential service failure has been detected. According to Rule 1, if the printer is out of toner, a request to replace the toner cartridge should be sent to the person responsible for maintenance of the printer (e.g., John in the case of printer-1).

Rules 3 and 4 are two of the rules used by *Shriek*-P to identify possible recovery strategies following the diagnosis of a printer fault. If the printer is out of toner, a possible recovery strategy according to Rule 3 is to wait for the technician—if known to be available—to replace the toner cartridge. In principle, the user should not have long to wait as the technician will have been informed as a result of the remedial action triggered by Rule 1. In practice, continuous monitoring of the situation (e.g., informed by temporal reasoning) may be necessary to ensure that the problem is dealt with in reasonable time. According to Rule 4, another option available to the user is for her print job to be redirected to another printer. All the recovery strategies that apply in a given situation are presented as options to the user by *Shriek*-P.

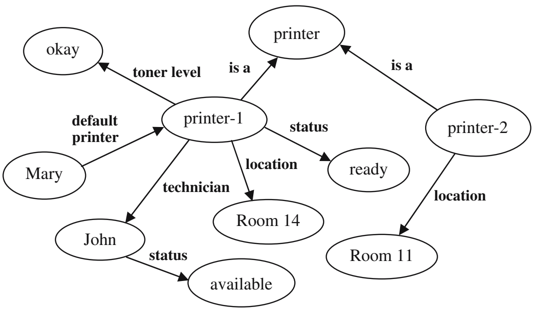

Fig. 4 Environment knowledge base

Fault Remediation

Rule 1: **if** diagnosis = out of toner
and target printer = P
and technician responsible for P = T
then notify T to replace toner cartridge in P

Rule 2: **if** P is a printer
and status of P = tray empty
and technician responsible for P = T
then notify T that P is out of paper

Fault Recovery

Rule 3: **if** diagnosis = out of toner
and target printer = P
and technician responsible for P = T
and status of T = available
then recovery strategy = wait for toner cartridge to be replaced by T

Rule 4: **if** diagnosis = out of toner
and target printer = P1
and P2 is another printer
and location of P2 = room X
then recovery strategy = redirect print job to P2 in room X

Fault Referral

Rule 5: **if** diagnosis = unrecognized fault
and target printer = P
and technician responsible for P = T
then refer problem to T

Fig. 5 Rules for fault remediation, recovery, and referral

Finally, Rule 5 triggers a referral of a problem that *Shriek*-P is currently unable to solve to the technician responsible for maintenance of the printer.

4 Integrating CBR and rule based reasoning

In this section, we describe how fault diagnosis driven by conversational CBR is seamlessly integrated with a rule based reasoning approach to fault remediation, recovery, and referral informed by environment knowledge in *Shriek*-P. A detailed account of our approach to conversational CBR in fault diagnosis is followed by a discussion of the complementary role of rule based reasoning in the fault management cycle. Finally, we use an example problem solving dialogue in the printer troubleshooting domain to demonstrate our overall approach to autonomic self healing and recovery.

4.1 Conversational CBR

Table 2 shows an example case base for printer fault diagnosis that we use to illustrate our conversational CBR approach to fault diagnosis. Each case consists of a problem description (i.e., a set of question-answer pairs) and a solution. As often in conversational CBR, the case structure is *heterogeneous* (i.e., not all questions are answered in every case).

Table 2 Example case base for printer fault diagnosis

	Able to Print?	Print quality?	Toner level?	Printer status?	Printer switched on?	Solution
Case 1		White streaks	Low			Out of toner
Case 2	Y	Spots				Damaged cartridge
Case 3	N			No response	N	Printer switched off
Case 4	Y	Blank	Okay	Off line		Cartridge sealed
Case 5				Tray empty	Y	Out of paper
Case 6		Blank	Low	Ready		Out of toner

4.1.1 Similarity measure

Our approach to assessing the similarity of each case to the current query is based on the measure proposed by Aha et al. (2001) to take account of the heterogeneity of cases that is common in conversational CBR. We define the similarity of a given case C to a query Q to be:

$$similarity(C, Q) = \frac{|matches(C, Q)| - |mismatches(C, Q)|}{|description(C)|}$$

where: $description(C)$ is the set of question-answer pairs in C; $matches(C, Q)$ is the set of questions in C that also appear in Q and have the same answer in C and Q; $mismatches(C, Q)$ is the set of questions in C that also appear in Q but have different answers in C and Q.

4.1.2 Information gathering

Often in conversational CBR, the user is the only source of problem solving information, though some approaches support information gathering from other sources (Carrick et al. 1999; Giampapa and Sycara 2001). In *Shriek*-P, relevant information is also obtained by automated information gathering from available sources in the computing environment (e.g., printers). Thus when a user seeks assistance with a printing problem, some of the relevant information (e.g., toner level, printer status) is already known.

4.1.3 Question selection

Question selection in *Shriek*-P is based on the approach proposed by Aha et al. (2001). The question selected at each stage of the problem solving dialogue is the unanswered question that appears most frequently in the most similar cases (i.e., those whose similarity exceeds a predefined threshold).

4.1.4 Unknown information

In practice, some of the information requested by *Shriek*-P may be unknown to the user (e.g., print quality cannot be assessed if a printer is not working). If the user answers *unknown* to any question, the dialogue moves on to the next most useful question. Often in conversational CBR, a solution may be possible without a complete description of the target problem (McSherry 2001; Aha et al. 2005).

Rule 6: **if** diagnosis = unrecognized fault
and target printer = P1
and P2 is another printer
and location of P2 = room X
then recovery strategy = redirect print job to P2 in room X

Fig. 6 A general rule for recovery from an unrecognized fault in the printer troubleshooting domain

4.1.5 Dialogue inferencing

As in some CBR tools for customer help desk support (Watson 1997), *Shriek*-P uses simple rules to infer the answers to certain questions from information obtained earlier in the problem solving process. For example, if the problem reported by the user is that there are white streaks on printed pages, then there is no need to ask the user if the printer is switched on. Often in conversational CBR, dialogue inferencing is important to avoid question choices that needlessly extend problem solving dialogues and may reduce user confidence in the system (Aha et al. 2001; Aha et al. 2005).

4.1.6 Dialogue termination

A problem solving dialogue ends in *Shriek*-P when any case reaches a similarity of one, or when all questions have been answered. The most similar case must reach a similarity of 0.5 for its solution to be applied to the user's problem. If so, *Shriek*-P presents the solution from the most similar case as its diagnosis. The user is also shown the most similar case and its matching (+) and mismatching (−) features. If no case reaches the required similarity threshold, the problem is treated as an *unrecognized* fault for the purposes of fault recovery and referral (Sect. 4.2).

4.2 Rule based reasoning

On completion of a problem solving dialogue driven by conversational CBR, *Shriek*-P initiates a fault remediation and recovery process driven by rule based reasoning and informed by environment knowledge (e.g., staff responsibilities and availability). In this process, any of the rules for fault remediation and recovery (Fig. 5) whose conditions are satisfied are used to trigger remedial actions (e.g., notifying a technician that a printer is out of toner) and identify recovery strategies that the user may wish to consider (e.g., redirect her print job to another printer, wait for the technician to deal with the problem, or tackle the problem herself).

Some of the rules used to trigger remedial actions and identify recovery strategies in *Shriek*-P were discussed in Sect. 3. As noted in Sect. 2, recovery from a printer fault may be possible even if a definitive diagnosis cannot be reached. Figure 6 shows a rule used by *Shriek*-P to guide recovery from an unrecognized fault in the printer troubleshooting domain.

Fault referral in *Shriek*-P is also driven by rule based reasoning and informed by environment knowledge. The rule used by *Shriek*-P to trigger a referral of an unrecognized fault to the technician responsible for maintenance of the printer is shown in Fig. 5. As noted in Sect. 2, potential service failures can sometimes be detected by automated information gathering and remedial action taken with no need for a problem solving dialogue (or CBR). For example, Rule 2 in Fig. 5 immediately triggers a remedial action (i.e., notifying a technician) when a printer runs out of paper.

4.3 Problem solving dialogue

The version of *Shriek*-P that we use to demonstrate our overall approach to autonomic self healing and recovery is an off line version developed as a test bed for evaluating the system's responses to simulated faults in the printer troubleshooting domain. In this version, functions requiring connection to the computing environment (e.g., information gathering, notifying technicians) are simulated.

Figure 7 shows a problem solving dialogue based on our example case base (Table 2), environment knowledge base (Fig. 4), and rules for fault remediation and recovery (Fig. 5). In the example dialogue, the user's answer to a single question is enough to provide *Shriek*-P with a complete description of the problem. The solution from the most similar case (i.e., Case 4) is that the pages are blank because the seal was not removed from the toner cartridge when it was installed in the printer. The problem description that the user is now shown, together with the most similar case, includes facts inferred by *Shriek*-P from its domain knowledge (e.g., printer switched on = yes) and information obtained, in principle, directly from the printer (e.g., toner level = okay).

With three matching features (+) and one mismatching feature (−), the similarity of Case 4 to the problem description is $\frac{3-1}{4} = 0.5$, the minimum required for the solution of a retrieved case to be reused by *Shriek*-P as a solution to the current problem. The user is now informed that the solution from the most similar case is *Cartridge Sealed* and that the technician

Printer Details: name = printer-1, location = Room 14, toner level = okay, status = ready

Inferred fact: printer switched on = yes

Problem Solving Dialogue:

Print quality? blank

Inferred fact: able to print = yes

Problem Description: toner level = okay, printer status = ready, printer switched on = yes, print quality = blank, able to print = yes

Most Similar Case:

Case 4: Cartridge Sealed

able to print = yes (+)
print quality = blank (+)
toner level = okay (+)
printer status = off line (-)

Similarity: 0.5

Fault Diagnosis: Cartridge Sealed

Remedial Action: Technician (John) notified that toner cartridge in printer-1 is sealed

Recovery Strategies:

- Wait for technician to remove seal from toner cartridge
- Redirect your print job to printer-2 in Room 11
- Remove seal from toner cartridge and replace cartridge in printer

Fig. 7 A printer troubleshooting dialogue showing the diagnosis reached and remedial action and recovery strategies identified by *Shriek*-P responsible for maintenance of the printer has been notified. Finally, the user is offered three alternative recovery strategies, such as redirecting her print job to another available printer.

5 Related work

Anglano and Montani (2005) propose a CBR approach to service failure diagnosis and remediation in autonomic computing. When a service failure (e.g., an email delivery failure) is detected, the repair plan from the most similar case is passed to a repair module that executes the plan. The authors focus on service failures that can be detected automatically and solved without any human intervention, though acknowledge that this may not always be possible. As there is no user interaction, efficiency of problem solving dialogues is not an issue, and there is no need for identification of alternative recovery strategies that may be preferable to the user.

In contrast, our emphasis on self healing and recovery with a *minimum* of human intervention reflects our view that fully autonomous self healing is achievable only for a limited range of faults. As noted by Bustard et al. (2006), it is important to distinguish between faults that can be detected automatically and those whose detection relies on feedback from users on service quality. For example, an autonomic computing system may be aware that a print job has been completed, but has no way of knowing if the print quality is acceptable to the user. Equally, there are many faults for which remediation and recovery may not be possible without human intervention.

An important feature of our approach to autonomic self healing is that potential service failures can sometimes be detected by proactive information gathering, and preventive action taken before a service failure occurs (Sect. 2). In contrast, remedial actions are triggered only when a service failure has been detected in the approach proposed by Anglano and Montani (2005). Integration of CBR with rule-based reasoning is another feature that distinguishes our work from most previous contributions, although the potential role of muli-modal reasoning in autonomic diagnosis and remediation is briefly discussed by Montani and Anglano (2006).

Combining CBR with other reasoning modalities is a topic of increasing research interest driven by user expectations (e.g., interoperability) and potential benefits such as improved solution quality and explanatory power (Marling et al. 2005). Integration of CBR with rule based reasoning in our approach to autonomic self healing and recovery addresses the requirement for knowledge of a computing system's dynamically changing environment that historical cases cannot provide. Separation of environment knowledge from the rules that apply the knowledge reduces maintenance effort in response to changes in the environment (e.g., staff responsibilities and locations). For example, no change in the rule that triggers referral of an unrecognized printer fault (Fig. 5) is needed when a new member of staff assumes responsibility for the printer.

In health care applications of CBR, rule based reasoning can also play an important role in extending the problem solving process beyond an initial diagnosis or treatment decision. For example, Marling and Whitehouse (2001) present a CBR approach to the decision to prescribe neuroleptic drugs for patients with Alzheimer's disease. Instead of simply using the drug prescribed in the most similar case, which may no longer be the most effective, their system uses rule based reasoning to select the most suitable drug. Additional recommendations may also be generated, such as the best time of day to administer the drug. However, queries (i.e., patient descriptions) are not elicited interactively as in conversational CBR, and there may be no benefit in doing so if the data is already available in electronic form.

Giampapa and Sycara (2001) demonstrate the ability of NaCoDAE, a generic shell for conversational CBR (Aha et al. 2001), to proactively dialogue with other agents in a multi-agent approach to information gathering in a command and control scenario. In contrast to our work, the CBR system's role is to converse directly with other agents and not with a human user. Information gathering from a variety of sources is also a feature that our approach shares with Carrick et al.'s (1999) CaseAdvisor.

6 Conclusions

Resent research has highlighted the potential benefits of CBR as an approach to fault diagnosis and remediation in autonomic computing, not least the ability to learn from experience (Crapo et al. 2003; Anglano and Montani 2005; Bustard et al. 2006). However, we have argued in this paper that autonomic self healing also requires knowledge of the computing environment that historical cases cannot provide. We have also argued that in contrast to CBR applications that focus on fault diagnosis and remediation, an effective approach to fault management in autonomic computing must address all aspects of fault management (i.e., fault detection, diagnosis, remediation, recovery, and referral).

As a first step towards achieving these goals, we have presented an approach to autonomic self healing and recovery in which conversational CBR is combined with rule based reasoning informed by environment knowledge. Dynamic elements of environment knowledge required for fault remediation, recovery, and referral are continuously updated in our approach by automated information gathering from available sources in the computing environment. As we have shown in the printer troubleshooting domain, fault detection and diagnosis also stand to benefit from the environment awareness provided by automated information gathering.

The work completed so far has demonstrated the feasibility of the approach. At the next stage of evaluation *Shriek*-P will be assessed through its practical use in a local area network. This should help to identify how users react to faults in practice and bring out the operational strengths and weaknesses of the system, leading to refinements in its design.

Acknowlegement Sa'adah Hassan's research is supported by the Ministry of Higher Education, Malaysia.

References

Aha DW (1998) The omnipresence of case-based reasoning in science and application. Knowledge-Based Syst 11:261–273

Aha DW, Breslow LA, Muñoz-Avila H (2001) Conversational case-based reasoning. Appl Intell 14:9–32

Aha DW, McSherry D, Yang Q (2005) Advances in conversational case-based reasoning. Knowl Eng Rev 20:247–254

Anglano C, Montani S (2005) Achieving self-healing in autonomic software systems: a case-based reasoning approach. In: Proceedings of the international conference on self organization and adaptation of multi-agent and grid systems, IOS Press, Amsterdam, pp 267–281

Bustard D, Hassan S, McSherry D, Walmsley S (2006) Standard exemplars for autonomic computing concepts. In: Proceedings of the 3rd IEEE workshop on engineering of autonomic and autonomous computing systems, pp 267–281

Carrick C, Yang Q, Abi-Zeid I, Lamontagne L (1999) Activating CBR systems through autonomous information gathering. In: Althoff K-D, Bergmann R, Branting LK (eds) Case-based reasoning research and development. LNAI, vol 1650. Springer, Berlin, Heidelberg, New York, pp 74–88

Crapo AW, Aragones AV, Price JE, Varma A (2003) Towards autonomic systems for lifecycle support of complex equipment. In: Proceedings of the international conference on information reuse and integration. IEEE, pp 322–329

Ganek A (2007) Overview of autonomic computing: Origins, evaluation, direction. In: Parashaw M, Hariri S (eds) Autonomic computing: concepts, infrastructure, and applications. CRC Press, Boca Raton, FL, pp 3–18

Giampapa J, Sycara K (2001) Conversational case-based planning for agent team coordination. In: Aha DW, Watson I (eds) Case-based reasoning research and development. LNAI, vol 2080. Springer, Berlin, Heidelberg, New York, pp 189–203

Horn P (2001) Autonomic computing: IBM's perspective on the state of information technology. AGENDA 2001, Scottsdale, AR, IBM TJ Watson Labs

Kephart JO (2005) Research challenges of autonomic computing. In: Proceedings of the 27th international conference on software engineering, ACM Press, New York, USA, pp 15–22

López de Mántaras R, McSherry D, Bridge D, Leake D, Smyth B, Craw S, Faltings B, Maher ML, Cox MT, Forbus K, Keane M, Aamodt A, Watson I (2005) Retrieval, reuse, revision, and retention in case-based reasoning. Knowl Eng Rev 20:215–240

Marling C, Rissland E, Aamodt A (2005) Integrations with case-based reasoning. Knowl Eng Rev 20:241–245

Marling C, Whitehouse P (2001) Case-based reasoning in the care of Alzheimer's disease patients. In: Aha DW, Watson I (eds) Case-based reasoning research and development. LNAI, vol 2080. Springer, Berlin, pp 703–715

McSherry D (2001) Interactive case-based reasoning in sequential diagnosis. Appl Intell 14:65–76

Montani S, Anglano C (2006) Case-based reasoning for autonomic service failure diagnosis and remediation in software systems. In: Roth-Berhgofer TR, Göker MH, Güvenir HA (eds) Advances in case-based reasoning. LNAI, vol 4106. Springer, Berlin, pp 489–503

Telford R, Horman R, Lightstone S, Markov N, O'Connell S, Lohman G (2003) Usability and design considerations for an autonomic relational database management system. IBM Syst J 42:568–581

Watson I (1997) Applying case-based reasoning: techniques for enterprise systems. Morgan Kaufmann, San Francisco, CA