The Knowledge Engineering Review, Vol. 00:0, 1–5. © 2005, Cambridge University Press
DOI: 10.1017/S0000000000000000 Printed in the United Kingdom

Case-based reasoning for diagnosis applications

MEHMET H. GÖKER[1], ROBERT J. HOWLETT[2] and JOSEPH E. PRICE[3]

[1] *PricewaterhouseCoopers L.L.P., Center for Advanced Research, Ten Almaden Blvd., Suite 1600, San Jose, CA 95113, USA*
E-mail: mehmet.goker@us.pwc.com
[2] *University of Brighton, School of Engineering Research Centre, Moulsecoomb, Brighton, BN2 4GJ, UK*
E-mail: R.J.Howlett@brighton.ac.uk
[3] *MITRE, 7515 Colshire Drive, McLean, VA 22102-7508, USA*
E-mail: JPRICE@mitre.org

Abstract

Case-based reasoning (CBR) is widely applicable to the diagnosis of problems and the identification of solutions to them. This review of the literature identifies key papers relating to this use of CBR. The stages in diagnosing and troubleshooting a problem are considered and the elements of a CBR system for achieving this are described. Five systems are discussed that have frequently been cited in the literature and illustrate the use of CBR in diagnosis and troubleshooting applications.

1 Reuse of experience for diagnosis and troubleshooting through case-based reasoning

Diagnosis is the identification of the root cause of abnormal or defective behavior in a system by means of the exposed symptoms, the system state, the system's general specifications and the operating environment. It links the observable system behavior to a disease or a problem condition pertinent in the system and thereby explains it. Troubleshooting, as a next step, takes the result of this diagnosis and provides a remedy to restore normal system operation.

When human beings diagnose systems and troubleshoot problems, they use their experiences with similar, previously solved problems extensively. Rather than deriving new solutions from scratch every time a problem is observed, they prefer to reuse existing experience and adapt it to the new circumstances. As such, diagnosis and troubleshooting are excellent application areas for the development of case-based systems. Existing applications range from IT help-desk support to web self-service, from on-board troubleshooting of technical products to medical diagnosis and treatment planning.

Reusing problem solving experiences to diagnose and troubleshoot new failures allows one to fix faults much faster and more consistently. Since case-based reasoning (CBR) is a learning process, the system fills the gaps in its knowledge over time and enables companies to retain and share experiences across the entire organization. First call resolution (be it in medical diagnosis or help-desks) increases drastically. Case-based diagnostic and troubleshooting applications are also very useful for training new, inexperienced personnel and ensure that the collective knowledge of the experts is instantaneously accessible to whoever needs it.

2 Elements of a case-based diagnosis application

Diagnosing and troubleshooting a problem typically involves three stages:

1. Gathering information about the status of the system (i.e., the symptoms, signs or manifestations of the problem, the specifications and the current condition of the system to be diagnosed, and the characteristics of the operating environment)
2. Generating the diagnosis, which describes the root cause of the problem; and, in troubleshooting systems,

3. Suggesting the remedy, or steps necessary to rectify the fault

Diagnosis and troubleshooting systems can acquire information regarding the system to be diagnosed directly from the device (on-line) or through human or electronic intermediaries (off-line). In the case of an on-line or condition monitoring system, the symptoms and system state are derived, without continuous user intervention, from interfaces and sensors monitoring the system. In the case of an off-line diagnostic system, the descriptions of the symptoms and the system are obtained from a user (e.g., a technician or knowledgeable user) or, after a failure is reported, downloaded electronically. Applications that fall in this category can provide web self-service to end-users, support field technicians and medical personnel, or assist help-desk personnel while they are conversing with the end-users. Both on-line and off-line systems typically augment the information about the system or patient to be diagnosed with data they obtain from additional information sources.

Off-line systems with human interaction acquire the symptoms and system characteristics from the user through three different interaction modalities: (1) free text queries, (2) conversational, guided interaction, and (3) form based information acquisition. Free text queries enable end users to describe their problem by typing it using their own words, and, after parsing the information provided in the query, the system can either ask the users more questions or retrieve a solution. This interaction modality is typically suited for users who are not experienced with the system to be diagnosed or are unfamiliar with the technical vocabulary. While early systems matched the user's queries based on the characters in the text, newer systems allow for the definition of vocabularies, synonyms and semantic rules.

Systems with conversational or guided interaction prompt the users to provide information by asking questions and guiding them towards a solution. Depending on the underlying architecture or knowledge representation, the questions are either hard-coded in the cases themselves or dynamically generated from the cases using methods such as information gain computations. Depending on the architecture, this approach either supports users with varying levels of experience by asking questions until the query is sufficiently constrained, or, in case of the interactions being coded in the cases directly, is only able to support users for whom the system was created.

Form based interaction allows users to specify the symptoms and system characteristics in a structured form. This is typically only available in structural CBR systems and only suitable for experienced users. Novices might be tempted to specify values for attributes that are irrelevant to the problem at hand, thereby reducing the overall similarity of cases and potentially causing relevant solutions to fall below a similarity cut-off.

Diagnosis and troubleshooting experience can be stored in case-based systems in multiple ways (Bergmann *et al.*, this issue). The choice of representation has an impact on the maintainability of the system in the long term and the interaction modalities the system supports. While *structural CBR systems* require an up-front effort to create a vocabulary or domain model, they allow individual cases to be entered without having an impact on existing cases (Kriegsmann & Barletta, 1993; Göker & Roth-Berghofer, 1999). Some *conversational CBR systems* store the questions and their respective answers in the cases and do not require a domain model (Acorn & Walden, 1992). This approach allows faster initial deployment, but maintenance of the application becomes cumbersome with a growing number or cases. *Textual CBR systems* use existing text files as cases and index these to perform retrieval (Lenz, 1996; Lenz *et al.*, 1999). Depending on the complexity of the vocabulary used to index the text files, the initial effort to set up the domain model for these systems can become comparable with structural CBR systems. On the other hand, since they will allow for reuse of existing documentation, initial set-up of the case base itself is typically very easy. However, the quality of the content in existing documentation and its suitability for use in a CBR system needs to be verified.

Diagnosis and troubleshooting systems do not exist in a vacuum. Typically, they are provided or utilized in a larger organization and contain solutions for a specific system type and for a specific operating environment. Changes in the system, the operating environment or the organization will require the application and the knowledge containers (cases, vocabulary, similarity metrics, adaptation knowledge) to be maintained. The processes for case acquisition, utilization and maintenance have to be put in place in an organization to ensure an application can be successful in the long term (Bergmann *et al.*, 2003).

The initial knowledge in a diagnosis and troubleshooting application can be acquired through interviews with experts, or converted from existing documentation. Documents that are suitable for conversion include FAQ's, troubleshooting and diagnosis manuals, technical service bulletins and the like. These documents have been verified in terms of their correctness and typically contain enough information for creating cases that are helpful in diagnosing and troubleshooting failures. Even so, extracting information from this information may not be a straightforward or easy process. While the idea of reusing information coming from trouble-ticket systems is tempting, these documents are created by help-desk personnel whose primary goal is to close a ticket and not to document the diagnosis process in a detailed manner. Therefore they are only of limited value.

Depending on the application area, case-based diagnosis and troubleshooting systems will utilize a combination of reasoning methods. While some systems will only use cases to generate solutions, especially in situations where adapting an existing solution to a new problem is required, systems will use a combination of CBR and model-based reasoning (Simoudis & Miller, 1991; Portinale & Torasso, 1995), rule-based reasoning, induction, planning, or a mixture of these methods.

3 Sample case-based diagnosis applications

In this section, we discuss five systems that have frequently been cited in the literature and illustrate how CBR can be used in diagnosis and troubleshooting applications.

Koton's CASEY (Koton, 1988) is a medical diagnosis program that integrates case-based and causal reasoning techniques with a model-based expert system (the Heart Failure Program) for managing patients with cardiac disease. Patients or abstractions of similar patients, their condition and the diagnosis are described using a structured set of around 40 features (attributes plus associated values) as cases. The system maintains a set of probabilistic principles that explain the relationship between certain conditions and their causes, a set of rules that are used to measure the utility of a given case in the new situation, and a set of repair strategies to modify the cases that are retrieved from memory to match the needs. This enables CASEY to evaluate (justify) and adapt cases in its memory to new problems. If CASEY is unable to find and adapt a suitable solution from memory, it uses the Heart Failure Program (Long *et al.*, 1987) to generate a solution from scratch. Koton reports that CASEY produced a solution that was either successful or satisfactory for 86% of the situations where there was an acceptable solution in its memory and that the system was up to three orders of magnitude more efficient than the Heart Failure Program.

Another medical diagnosis application, López and Plaza's BOLERO (Lopez & Plaza, 1997) is a CBR system that diagnoses pneumonia patients. BOLERO is a case-based planning system that learns both plans and goal states. Its goal is to improve the performance of a rule-based system by adapting its behavior to the most recent information known about the patient, enabling it to generate new hypotheses about a patient's condition and to derive a set of necessary verification steps. Since the actual goal state (i.e., the diagnosis) is not known beforehand, the system must recognize a goal state when it has been achieved. BOLERO's cases are plans that are a set of such verification and decision steps. It is able to retrieve partial plans from memory, generate new plans from scratch if needed, and can learn from experience from either successful or failed plans. BOLERO is integrated with a rule-based system that executes the plans it generates. The result of the execution is fed back into BOLERO and is used in a revision cycle.

Acorn and Walden won the Innovative Applications of Artificial Intelligence Award at IAAI'92 (Acorn & Walden, 1992). In their paper, they describe SMART, a case-based help-desk support system developed for Compaq. SMART is integrated with Compaq's call logging system and enables help-desk operators working at the customer support center to resolve customer complaints efficiently. Operators can enter the problem description in the call tracking system in free text. This information is passed to SMART which parses it using tri-grams matching and, based on the cases that are closest to the entered description, engages in a dialog with the operator. Since this is a conversational CBR system, the dialog is driven mostly by the ordered questions stored in the cases and how many cases match the questions answered so far. Cases that arrive at a similarity that is higher than the cut-off are presented to the help-desk operator as cases that are similar to the user's problem. The system was one of the first deployed CBR help-desk systems and Compaq was able to show a return on investment after one year. The system also changed the way the help-desk operates and enhanced the work-flow.

While SMART is a conversational CBR system, the HOMER system (Göker & Roth-Berghofer, 1999) uses a structural approach. HOMER was developed in the course of the INRECA-II research project (Bergmann *et al.*, 2003) as a help-desk support system for the CAD/CAM hotline at Mercedes car development at DaimlerChrysler. Cases were entered using structured forms and users had the choice of either interacting with the system by answering questions for the attributes with the highest information gain, or by entering the problem description in a structured form. Since one of the goals of the INRECA projects was to develop a methodology for developing industrial CBR systems, Göker and Roth-Berghofer analyze the managerial, organizational and technical processes that take place during the development of a CBR system. They also point out the importance of case-base maintenance and extend the standard CBR cycle of Aamodt and Plaza (Aamodt & Plaza, 1994) with an additional cycle for maintenance operations. During this cycle, new cases are checked for correctness and their impact on the case base.

CASSIOPÉE (Heider, 1996; Auriol *et al.*, 1996) is a troubleshooting system for CFM 56-3 aircraft engines. It is a structural CBR application that uses induction and CBR to arrive at solutions. The developers initially tried to seed the case-base from a reliability and maintenance database containing 30,000 records. However, as is typically the case with trouble-tickets, the records did not contain any information that could be useful for diagnosing the actual fault and the developers reverted back to analyzing the records and generating cases manually. During run-time, the system uses the cases in the case base to generate a fault-tree using induction. The advantage of this lazy approach is that the tree can take the most recent, new cases into account. The system acquires information from the user by asking the questions with the highest information gain and retrieves solutions using CBR. Results are presented as soon as there are no more question to ask or a similarity threshold is met. The system is also integrated in a documentation system and enables technicians to view graphics and documentation related to the repair they need to perform.

4 Conclusion

The systems described above, along with many other operational case-based diagnosis systems, demonstrate the applicability of case-based reasoning to problem diagnosis and troubleshooting. By distilling information relating to the cause of a problem from example problem cases, the need for defining the link between symptoms and causes necessary in purely rule-based diagnosis systems is removed. By providing different options for the represention of cases and methods of user interaction, case-based reasoning provides a flexible foundation for the efficient and accurate generation of solutions in these challenging problem domains.

References

Aamodt, A & Plaza, E, 1994. “Case-based reasoning: Foundational issues, methodological variations, and system approaches,” *AI Communications* 7(1):39–59.

Acorn, TL & Walden, SH, 1992. “Smart: Support management automated reasoning technology for Compaq customer service” in *The Fourth Conference on Innovative Applications of Artificial Intelligence (IAAI-92)*, Menlo Park, California: AAAI Press, pp. 3–18.

Auriol, E, Guiot-Dorel, J & Manago, M, 1996. “Cassiopée: Fehlerdiagnose von CFM 56-3 triebwerken für Boeing 737 flugzeuge,” *KI - Künstliche Intelligenz* 1:47–53.

Bergmann, R, Althoff, K, Breen, S, Göker, MH, Manago, M, Traphöner, R & Wess, S, 2003. *Developing Industrial Case-Based Reasoning Applications, The INRECA Methodology*, 2nd edition, Berlin: Springer.

Bergmann, R, Kolodner, J & Plaza, E, 2006. “Representation in case-based reasoning,” *The Knowledge Engineering Review*, this issue.

Göker, MH & Roth-Berghofer, T, 1999. “The development and utilization of the case-based help-desk support system Homer,” *Engineering Applications of Artificial Intelligence* 12(6):665–680.

Heider, R, 1996. “Troubleshooting CFM56-3 engines for the Boeing 737 using CBR and data-mining” in Smith, I & Faltings, B (eds.) *Advances in Case-Based Reasoning, Proceedings of the Third European Workshop on Case-Based Reasoning (EWCBR-96)*, Berlin: Springer, pp. 512–518.

Koton, P, 1988. “Reasoning about evidence in causal explanations” in *Proceedings of the Seventh National Conference on Artificial Intelligence*, Menlo Park, California: AAAI Press, pp. 256–263.

Kriegsman, M, and Barletta, R, 1993. "Building a case-based help desk application," *IEEE Expert* 8(6)18–26.

Lenz, M, Busch, K, Hübner, A & Wess, S, 1999. The Simatic knowledge manager in Aha, DW (ed.), *Exploring Synergies of Knowledge Management & Case-Based Reasoning: Proceedings from the AAAI 1999 Workshop*, Menlo Park, California: AAAI Press, pp. 40–45.

Lenz, M, 1996. "Defining knowledge layers for textual case-based reasoning" in Smyth, B & Cunningham, P (eds.), *Advances in Case-Based Reasoning, Proceedings of the Fourth European Workshop on Case-Based Reasoning (EWCBR-98)*, Berlin: Springer, pp. 298–309.

Long, WJ, Naimi, S, Criscitello, M & James, R, 1987. "The development and use of a causal model for reasoning about heart failure" in *Symposium on Computer Applications in Medical Care*, IEEE, pp 30–36.

López, B & Plaza, E, 1997. "Case-based learning of plans and goal states in medical diagnosis," *Artificial Intelligence in Medicine* 9(1):29–60.

Portinale, L, and Torasso, P, 1995. "Adapter: An integrated diagnostic system combining case-based and abductive reasoning" in Veloso, M & Aamodt, A (eds.), *Case-Based Reasoning Research and Development: Proceedings of the First International Conference on Case-Based Reasoning (ICCBR-95)*, Berlin: Springer, pp. 277–288.

Simoudis, E & Miller, JS, 1991. "The application of CBR to help-desk applications" in *Proceedings, Case-Based Reasoning Workshop*, Palo Alto, California: Defense Advanced Research Projects Agency, pp. 25–36.