TEKA Kom. Mot. i Energ. Roln. – OL PAN, 2011, 11A, 266-275

CASE-BASED REASONING METHOD FOR DIAGNOSTIC DECISION SUPPORT SYSTEM OF BRIDGE CRANES

Vitaly Ulshin, Sergey Klimchuk

Volodymyr Dahl East-Ukrainian National University, Lugansk, Ukraine

Summary. The elements of diagnosic system of bridge cranes is analysed. The stages in diagnosing faults is considered. The decomposition of bridge cranes faults retrieval and modified case-based reasoning cycle are offered. Diagnostic decision support system of bridge cranes has been developed.

Keywords: case, case-based reasoning, diagnosis, bridge crane, decision support system.

INTRODUCTION

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 [1]. As such, diagnosis and troubleshooting are excellent application areas for the development of case-based systems [2-3].

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. 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.

CONCEPT OF CBR

In most CBR systems, the case-based reasoning mechanism has an internal structure divided into two major parts: the case retriever and the case reasoner (fig. 1). The case retriever's task is to find the appropriate cases in the case base, while the case reasoner uses the cases retrieved to find a solution to the problem description given.

Case-based reasoning has been formalized for purposes of computer reasoning as a fourstep process [4]:

1. Retrieve: Given a target problem, retrieve cases from memory that is relevant for solving it. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived.

2. Reuse: Map the solution from the previous case to the target problem. This may involve adapting the solution as needed to fit the new situation.

3. Revise: Having mapped the previous solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise.

4. Retain: After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory.

These steps are part of the CBR cycle, which represents the process-oriented view of the descriptive framework presented by Aamodt and Plaza. The process is supported by supplying the cases with general knowledge about bridge cranes.

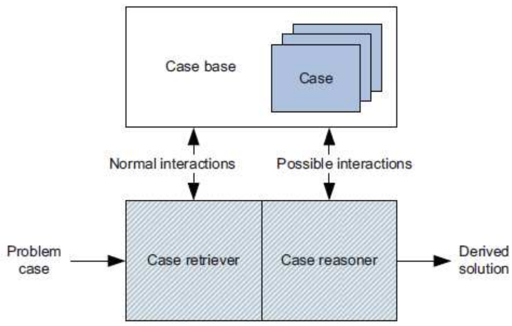

Fig. 1. Two major components of a CBR system

ELEMENTS OF A CASE-BASED DIAGNOSIS APPLICATION

Diagnosing and troubleshooting of bridge cranes typically involves three stages [5]:

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;

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 selfservice to end-users, support field technicians and medical personnel, or assist help-desk personnel while they are conversing with the end-users [6, 7].

While the process-oriented view provides a global and external view of the CBR process, the task-oriented view [8] decompose and describe the four top-level steps, where each step is viewed as a task that the CBR reasoner has to achieve (fig. 2). In the figure, tasks are named in bold letters, while methods are written in italics. The links between task nodes appears as plain lines and indicates task decompositions. The top-level task is problem solving and learning from experience and the method to accomplish this task is case-based reasoning (indicated in a special way by the stippled rectangle). The top-level task is split into the four major CBR tasks corresponding to the four processes: retrieve, reuse, revise, and retain. All the four tasks are necessary in order to perform the top-level task.

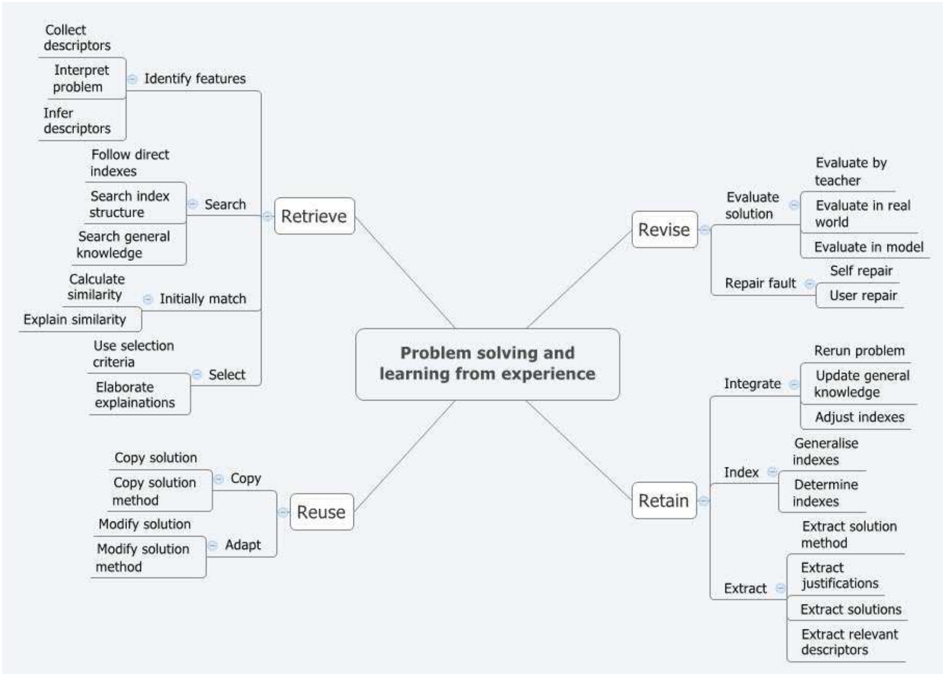

Fig. 2. Task-method decomposition of CBR (adopted from [5])

Diagnosis and troubleshooting experience can be stored in case-based systems in multiple ways (Bergmann et al.). The choice of representation has an impact on the maintainability of the system in the long term and the interaction modalities the system supports [9].While structural CBR systems require an up-front effort to create a vocabulary or domainmodel, they allow individual cases to be entered without having an impact on existing cases (Kriegsmann & Barletta, 1993; Goker & 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 domainmodel for these systems can become comparable with structural CBR systems [10]. 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 [11]. 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 [12]. 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 fromexisting documentation. Documents that are suitable for conversion include FAQ's, troubleshooting and diagnosis manuals, technical service bulletins and the like [13]. 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.

REFINING THE CBR CYCLE

Then the system must be able to execute the learning task more or less independently from its actual tasks. Such a learning functionality is often called introspective reasoning (Fox and Leake, 1995) or introspective learning (Zhang and Yang, 1999), respectively.

To integrate the desired learning functionality into the traditional CBR cycle consisting of the four well-known phases - retrieve, reuse, revise, retain - two basic possibilities can be distinguished [14]:

1. The extension of the existing process model by introducing an additional phase.

2. The refinement of one or several phases to integrate the new functionality into the already established phases.

When reviewing the original interpretation of the traditional CBR cycle it becomes clear that the second possibility seems to be more accurate. Aamodt and Plaza [4] have already discussed that the retain phase could be used to update general knowledge of the CBR system. Concerning the update of similarity measures the possibility to refine case indexes has been mentioned. This can be interpreted, for example, as an adjustment of feature weights.

Basically, the retain phase is not the only phase of the CBR cycle responsible for the capability to learn new knowledge [15-17]. Before memorising a new case, the correctness of this new knowledge item has to be validated during the revise phase. So, the revise phase has a significant influence when learning new case knowledge, because it selects cases considered to be candidates for extending the knowledge base. In the following we show that this holds as well when learning similarity measures.
Fig. 3 illustrates how the traditional CBR cycle can be modified to integrate the possibility to learn similarity measures [18]. These modifications are discussed in more detail in the following sections.

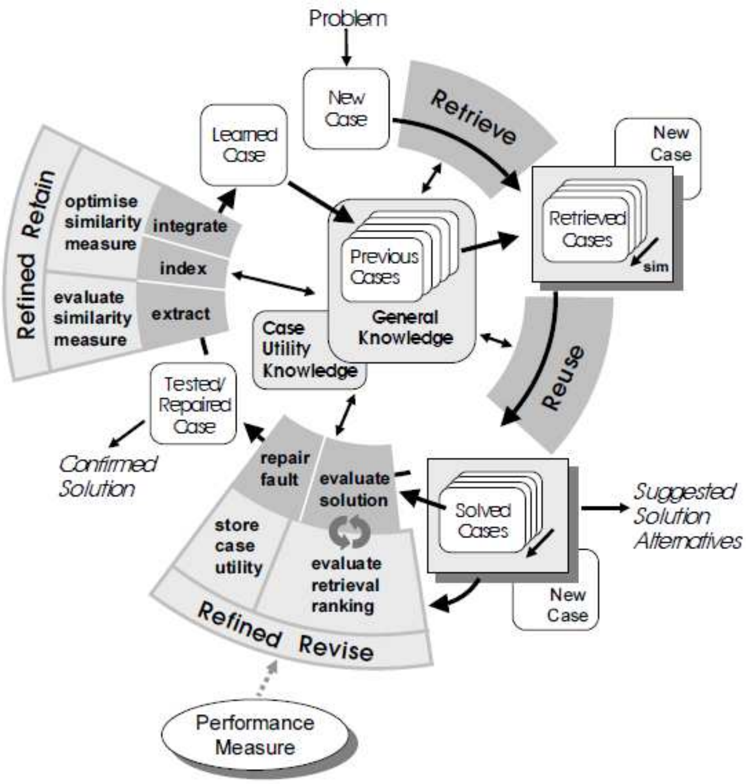

Fig. 3. Refining the CBR cycle for learning similarity measures

EXTENDED USE OF RETRIEVED CASES

In the traditional view of CBR, the retrieve phase provides one or several cases used to generate exactly one solution during the reuse phase. This solution is then proposed for solving the current problem and has to be evaluated during the revise phase. However, in many application domains where CBR has been employed successfully this traditional view is not always suitable. Here, it is not desired that the CBR system generates exactly one solution, but several independent alternatives for solving the given problem.

The retrieval phase always should provide a list of retrieved cases ordered by the computed similarities [19]. If case adaptation is supported, this list is processed during the reuse phase where several solution proposals might be generated by adapting several retrieved solutions independently from each other. Basically, two ways to generate solution alternatives can be distinguished:

- Ad hoc: If it is feasible with respect to computation time, the reuse phase might perform adaptation for a fixed number of cases immediately. The resulting list of solution proposals, still ordered as determined in the retrieval phase, is then directly passed to the revise phase.
- On demand: If case adaptation is computational expensive, only the most similar case may be adapted first. The generated solution is then passed to the revise phase where it has to be evaluated. If the evaluation fails, because the solution cannot be applied to solve the current problem or due to poor solution quality, two ways for proceeding are possible. On the one hand, the faulty solution might be repaired during the revise phase. On the other hand, the revise phase could trigger the adaptation of the next similar case in anew execution of the reuse phase to obtain an alternative solution proposal.

Both approaches lead to the suggestion of several solution alternatives - when applying the on demand approach, at least if the most similar case could not be reused successfully - after the reuse phase [20]. In the following we only assume the possible existence of such a list of suggested solution alternatives but we do not care about the approach used to generate it. It is only assumed that solution alternatives are ordered according to the similarity of the underlying cases.

REFINING THE REVISE PHASE

According to the original process model that assumes the existence of only one solved case after the reuse phase, the revise phase can be subdivided into two subsequent tasks [14]:

1. Solution evaluation: In a first step the proposed solution, i.e. the outcome of the reuse phase has to be evaluated. This evaluation might be based on feedback from a teacher, on the results obtained through application in the real world, or on the outcome of a model-based simulation.

2. Fault repair: When recognising faults in the suggested solution during evaluation, the solution has to be repaired to obtain a valid solution. Basically, it might be repaired manually by the user or it might be repaired by the system based on additional general knowledge.

To enable a CBR system to learn similarity measures we propose a refinement of the revise phase. Besides the two described traditional tasks that ensure the generation of a valid solution, we introduce two additional tasks [21]:

1. Evaluate retrieval ranking: This task can be characterised as a superior control process for the common solution evaluation task. It initiates the evaluation of several solution alternatives and processes the obtained evaluation results. The foundation of the evaluation might be internal general knowledge or an external performance measure in form of a teacher, the real world, or a model.

2. Store case utility: This task is responsible for storing the results of the retrieval ranking evaluation for further processing. Basically, these results represent knowledge about the utility of cases with respect to the given query.

Generally, one could also argue that storing of evaluation results belongs more to the retain phase of the CBR cycle. However, we decided to assign this task to the revise phase. On the one hand, the decision whether to store particular results or not might be influenced by the performance measure, for example, by a human teacher. On the other hand, the retained knowledge is not directly used by the phases of the CBR cycle that are relevant for problem-solving. It is more an intermediate knowledge buffer that collects knowledge to be used only during the retain phase and thus it does not directly contribute to solving problems.

Basically, the refined revise phase consists of two parallel processes. On the one hand, the traditional revision process that only evaluates and repairs a single solution. On the other hand, a parallel process that evaluates the outcome of the retrieval phase based on the results obtained during several solution evaluations. While the evaluation of the retrieval ranking relies on the solution evaluation process, the traditional revision of a single solution can be initiated independently. This means, the retrieval evaluation can be interpreted as an optional process to be performed if desired.

REFINING THE RETAIN PHASE

The aim of the retain phase is to select knowledge entities to be integrated into the knowledge resources of the CBR system in order to improve its problem-solving competence and/or efficiency during future usage. Therefore, the traditional retain phase identifies the following three tasks:

1. Extract: This task is responsible for the extraction of relevant knowledge entities from the current problem-solving episode to be retained for future usage. Such knowledge entities might be represented by found solutions, solution methods, justifications, etc.

2. Index: The objective of this task is to determine indexes to be used for retrieving the learned case. This may be interpreted as the selection of an accurate vocabulary used to characterise the case but it might also be interpreted as the determination of accurate attribute weights.

3. Integrate: During the final task the extracted knowledge has to be integrated into the knowledge base of the system. This process might comprehend an update of the case base, the index structure, and of other general knowledge.

Although this traditional interpretation of the retain phase, in principle, already considers the modification of general knowledge and even an adjustment of attribute weights, it seems to be necessary to introduce two additional tasks [14]:

1. Evaluate similarity measure: Here, the quality of the currently used similarity measure is estimated based on the case utility knowledge acquired in the previous revise phase.

2. Optimise similarity measure: This task can be seen as a specialisation of the index and integrate task of the traditional retain phase but with focus on learning similarity measures. During this task, machine learning or optimisation methods, respectively, are being used to optimise the current similarity measure regarding the available case utility knowledge. This optimisation might be triggered by the outcome of the prior evaluation of the current similarity measure.

Similar to the refined revise phase, the tasks additionally introduced in the refined retain phase have not necessarily to be executed during every pass of the cycle. Instead, in certain application scenarios all described extensions of the traditional CBR cycle might only be relevant during explicit knowledge acquisition or maintenance phases [22]. For example, if the performance measure is supplied by a human domain expert playing the role of a teacher, the refined revision phase can only be executed in situations where this expert is available. During problem-solving situations where the system is used by a “standard user” who does not possess the required expertise, the introduced retrieval ranking evaluation might be skipped.

CBR SYSTEM FOR DIAGNOSIS OF BRIDGE CRANES

The bridge cranes diagnosis DSS has been delevoped. The main window of this system is shown on a fig. 4. As an initial set of cases the data of observations of bridge cranes made by the reports of technical diagnostics "The Engineering center of industrial safety" LLC (Lugansk, Ukraine) and Expert-diagnostic research laboratory "Lifting machines and industrial building" of Volodymyr Dal East-Ukrainian National University (Lugansk, Ukraine) is used.

Fig. 4. The CBR DSS main window

The DSS allows to set the local similarity for every diagnostic parameter, weight of parameters and global similarity for a whole case. After setting of all necessary of similarity parameters the search of cases and their conclusion are carried out in order of diminishing of relevance with pointing of degree of similarity of every case is made.

Since a corresponding case is selected, its adaptation can be executed is modification of present in it decision with the purpose of its accordance to the parameters of current situation. In the case of absence of necessity for adaptation maintenance of the chosen case is executed without the change of diagnostic parameters.

CONCLUSION

The research described above, along with many other operational case-based diagnostic systems, demonstrate the applicability of case-based reasoning to diagnosis and troubleshooting of bridge cranes.

The conducted research show that diagnostics on the basis of cases allows to decide the weak formalized tasks of diagnostics of bridge cranes, simplify the aquisition knowledge from experts, shorten time of search of decision and implement self-training.

The bridge cranes diagnosis decision support system is developed. Using of this DSS assists diminishing of the informative loading on decision-making person in the process of troubleshooting, decline of influence of factors of subjectivity at the analysis of current situation, reduction of time, necessary for a decision-making.

REFERENCES

1. Bergmann R., Althoff K., Breen S., Göker M.H., Manago M., Traphöner R., Wess S., 2003.: Developing Industrial Case-Based Reasoning Applications, The INRECA Methodology, 2nd ed., Springer, Berlin: 236.
2. Göker M.H., Roth-Berghofer T., 1999.: The development and utilization of the case-based help-desk support system in a corporate environment // Engineering Applications of Artificial Intelligence, Vol. 12, Issue 6: 665-680.
3. Eremeev A., Varshavsky P., 2007.: Application of Case-based reasoning for Intelligent Decision Support Systems // Proceedings of the XIII-th International Conference “Knowledge-Dialogue-Solution” – Varna, Vol. 1: 163-169.
4. Aamodt A., Plaza E., 1994.: Case-based reasoning: Foundational issues, methodological variations, and system approaches // AI Communications, Vol. 7, Issue 1: 39-59.
5. Göker M.H., Howlett R.J., Price J.E., 2005.: Case-based reasoning for diagnosis applications // The Knowledge Engineering Review, Vol. 20: 277-281.
6. Aamodt A., 2004.: Knowledge-intensive case-based reasoning in Creek. In Procs. of the 7th ECCBR, Springer, Berlin/Heidelberg: 1-15.
7. Anguel F., Sellami M., 2009.: Knowledge Management for Fault Diagnosis of Gas Turbines Using Case Based Reasoning // Communications of the IBIMA, Vol. 10, number 24: 186-190.
8. Indahl C., Rud K.M., 2007.: Arbitration and Planning of Workflow Processes in a Context-Rich Cooperative Environment // Master of Science thesis, Norwegian University of Science and Technology, Department of Computer and Information Science.

9. Vachtsevanos G., Lewis F.L., Roemer M., Hess A., Wu B., 2006.: Intelligent Fault Diagnosis and Prognosis for Engineering Systems. John Wiley & Sons, Inc.: 434.
10. Olsson E., Funk P., Xiong N., 2004.: Fault diagnosis in industry using sensor reading and case based reasoning // Journal of Intelligent & Fuzzy Systems, Vol. 15, Issue 1: 41-46.
11. Eremeev A., Varshavsky P., 2007.: Methods and Tools for Reasoning by Analogy in Intelligent Decision Support Systems // Proc. of the International Conference on Dependability of Computer Systems. Szklarska Poreba, Poland, 14-16 June, IEEE: 161-168.
12. Pal S., Shiu S., 2004.: Foundations of soft case-based reasoning. John Wiley & Sons, Inc.: 274.
13. Ulshin V.A., Klimchuk S.A., 2009.: The diagnosis model of bridge cranes // Praci Lugansk Branch of International Informatization Academy, №2(20) part 2: 61-71 (Russian).
14. Stahl A., 2003.: Learning of Knowledge-Intensive Similarity Measures in Case-Based Reasoning // PhD thesis, University of Kaiserslautern: 237.
15. Mantaras R.L., McSherry D., Bridge D., Leake D. and others, 2005: Retrieval, reuse, revision and retention in case-based reasoning // The Knowledge Engineering Review, Vol. 20, Issue 3: 215-240.
16. Recio-García J.A., 2008.: jCOLIBRI: A multi-level platform for building and generating CBR systems // Phd Thesis, Complutense University of Madrid: 335.
17. Recio-García J.A., Diaz-Agudo B., Sanchez A., Gonzalez-Calero P.A., 2006.: Lessons learnt in the development of a CBR framework // Proceedings of the 11th UK Workshop on Case Based Reasoning: 60-71.
18. Li M., Chen X., Li X., Ma B., Vitanyi P., 2003.: The similarity metric // In Procs. of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, Maryland: 863-872.
19. Eremeev A., Varshavsky P., 2008.: Case-based Reasoning Method for Real-time Expert Diagnostics Systems // International Journal Information Theories & Applications, Vol. 15, Issue 2: 119-125.
20. Portinale L., Magro D., Torasso P., 2004.: Multi-modal diagnosis combining case-based and model-based reasoning: a formal and experimental analysis // Artificial Intelligence, Vol. 158, Issue 2: 109-153.
21. Sánchez-Ruiz-Granados A.A., González-Calero P.A., Díaz-Agudo B., 2009.: Abstraction in Knowledge-Rich Models for Case-Based Planning // Lecture Notes in Computer Science, Vol. 5650/2009: 313-327.
22. Hanemann A., 2006.: A Hybrid Rule-Based/Case-Based Reasoning Approach for Service Fault Diagnosis // In Procs. of 20th International Conference on Advanced Information Networking and Applications (AINA'06), Vol. 2: 734-740.

МЕТОД ДИАГНОСТИКИ КРАНОВ МОСТОВОГО ТИПА НА ОСНОВЕ ПРЕЦЕДЕНТОВ ДЛЯ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Ульшин В.А., Климчук С.А.

Аннотация. Проанализированы элементы системы технической диагностики мостовых кранов. Рассмотрены этапы диагностики неисправностей. Предложена декомпозиция поиска неисправностей мостовых кранов и модифицированный цикл вывода на основе прецедентов. Разработана СППР диагностирования мостовых кранов.

Ключевые слова: прецедент, рассуждение на основе прецедентов, диагностика, кран мостового типа, система поддержки принятия решений.