Integrated Maintenance with Case Factories for distributed Case-Based Reasoning Systems

Pascal Reuss[1,2]

[1] German Research Center for Artificial Intelligence
Kaiserslautern, Germany
`http://www.dfki.de`
[2] Institute of Computer Science, Intelligent Information Systems Lab
University of Hildesheim, Germany
`http://www.uni-hildesheim.de`

1 Integrated maintenance with Case Factories

Maintenance approaches for Case-Based Reasoning (CBR) systems focus on a single CBR system and mainly on a single knowledge container like the case base, the adaptation knowledge or the similarity. There are a few approaches that deal with shifting knowledge between knowledge containers, especially between the case base and adaptation knowledge. These approaches have been successfully applied to CBR systems in the past. In many systems, especially multi-agent systems, the knowledge is distributed among several homogenous or heterogeneous knowledge sources. Therefore, a maintenance approach has to consider the dependencies between these knowledge sources as well as high-level maintenance goals. An example is removing one or more cases from a case base. Cases in other CBR systems could depend on one of the removed cases, so they may become inconsistent (to some degree). The system should suggest an appropriate maintenance action like removing the depending cases to keep the system's correctness/consistency. To address these dependencies between CBR systems and their knowledge containers, a new maintenance approach for distributed CBR systems is required. In the following, I will describe the idea of an integrated maintenance approach based on so-called Case Factories, that will be capable of generating a maintenance plan for multiple CBR systems, considering dependencies, and providing explanations for the generated maintenance actions. The approach is designed for multi-agent systems based on the SEASALT architecture ([2]), which supports distributed knowledge sources. To develop this approach, four research goals have to be reached:

- Extend the original Case Factory approach from single system support to multi system support.
- Integrate maintenance explanation capabilities into the new approach
- Develop a methodology to apply the new approach to existing multi-agent systems as well as integrating it into the development of new multi-agent systems
- Evaluate the new approach and the methodology within an industrial use case for a decision support system

1.1 Case Factory and Case Factory Organization

The original idea of the Case Factory (CF) is from Althoff, Hanft and Schaaf ([1]) and is based on the Experience Factory approach from software engineering. A CF consists of several software agents for different tasks like evaluating incoherence or modifying the case base. Each knowledge source, in this context CBR systems, has its own CF that is responsible for maintaining the dedicated knowledge. The original approach has to be extended to support the maintenance of the other three knowledge containers, namely vocabulary, similarity and adaptation knowledge, considering intra-system dependencies between the knowledge containers. The original approach contains several software agents to monitor the case-base and one agent to do the necessary maintenance actions. To support all knowledge containers reasonably some more agents are required to monitor these containers and the maintenance tasks should be split on several agents. An own maintenance agent per knowledge container is required to support parallel maintenance of the knowledge. Additionally, a supervising agent is required to coordinate the maintenance actions. This agent is also responsible for the communication between the multiple CFs. Monitoring a knowledge container means to notify the supervising agent about changes of the knowledge inside the container, e.g. removing a case, renaming a concept, or adding a rule. Evaluation of a knowledge container means to check a knowledge container for inconsistencies, problem solving competence or efficiency. Therefore, different evaluation strategies could be used, for example computing the coverage and reachability of cases ([9]) or inconsistency checking ([8], [7] and [6]).

To store information about dependencies, maintenance goals and evaluation criteria, a so-called Maintenance Map is used. The Maintenance Map is based on the Knowledge Map from Davenport and Prusak ([4]), which was adapted to multi-agent-systems by Bach et al. ([3]). In contrast to a Knowledge Map, the Maintenance Map is a bidirectional graph. The vertices represent the knowledge sources in a distributed knowledge-based system and the edges represent the dependencies between the single sources. The weights of the edges could be used to describe the importance of the dependency. Additional information like maintenance goals, metrics, thresholds, or constraints could be defined, too. The Maintenance Map can also contain information about the priority of maintenance actions or about associated maintenance actions. This may be helpful, when a combination of maintenance actions is necessary to preserve the competence or efficiency of a single CBR system or the MAS ([5]).

While a CF is able to maintain a single CBR system, a high-level Case Factory Organization (CFO) is required to coordinate the actions of all CFs and take the dependencies between the single CBR systems into account. This CFO consists of several additional software agents to supervise the communication between the CFs and the adherence of high level maintenance goals. Additionally, agents collect the maintenance suggestions from the CFs and derive a maintenance plan from all single maintenance suggestions. The agents are also responsible for checking constraints or solving conflicts between individual maintenance suggestions. In addition, a maintenance suggestion may trigger follow-up maintenance actions based on the dependencies between the CBR systems. The concept of the CFO allows to realize as many CFs and layers of CFOs as required. This way a hierarchy of CFOs can be established that is scalable and supports multi-agent systems with many agents and layers.

To support the knowledge engineer, the CF and CFO should explain, why a certain maintenance action was suggested. To give a CBR system explanation capabilities a lot of knowledge is necessary. The underlying research assumption here is that the minimal knowledge required for the explanation of maintenance actions is the same knowledge that is required to suggest a maintenance action. It follows, that the minimal knowledge for explanations already exists in the system, if the system is able to (reasonably) suggest maintenance actions.

1.2 Current state of research and research plan

Currently, the concepts of the CF and the CFO are defined. An initial version of a CF is implemented within a research system called docQuery in the travel medicine domain. This implementation has to be extended with a CFO to test the integrated maintenance approach. The next steps on conceptual level are to formalize the concept and advance the maintenance planning and the explanation strategies on a more detailed level and implement them, too. It is planned to complete the conceptual development of the integrated maintenance approach this year and implement the complete approach within the docQuery system. Next year, the approach will be applied to an industrial use case in the domain of aircraft diagnosis to test the approach within an industrial environment. The evaluation of the approach and methodology will be completed till the third quarter of 2016 and the dissertation will be written until the beginning of 2017.

References

1. Althoff, K.D., Hanft, A., Schaaf, M.: Case factory - maintaining experience to learn. Advances in Case-Based Reasoning Lecture Notes in Computer Science 4106/2006, 429–442 (2006)
2. Bach, K.: Knowledge Acquisition for Case-Based Reasoning Systems. Ph.D. thesis, University of Hildesheim (2013), dr. Hut Verlag Mnchen
3. Bach, K., Reichle, M., Reichle-Schmehl, A., Althoff, K.D.: Implementing a coordination agent for modularised case bases. In: Proceedings of the 13th UK Workschop on Case-Based Reasoning. pp. 1–12 (2008)
4. Davenport, T.H., Prusak, L.: Working Knowledge: How Organizations Manage What they Know. Havard Business School Press (2000)
5. Leake, D., Kinley, A., Wilson, D.: Learning to integrate multiple knowledge sources for case-based reasoning. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. pp. 246–251. Morgan Kaufmann (1997)
6. McSherry, D.: An adaptation heuristic for case-based estimation. In: 4th European Workshop, EWCBR-98. pp. 184–195 (1998)
7. Racine, K., Yang, Q.: Maintaining unstructured case bases. In: Second International Conference on Case-Based Reasoning. pp. 553–564 (1997)
8. Smyth, B.: Case-base maintenance. In: 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA-98-AIE. pp. 507–516 (1998)
9. Smyth, B., Keane, M.: Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence. pp. 377–382 (1995)