Case-Based Reasoning

Introduction and Recent Developments

Ralph Bergmann, Klaus-Dieter Althoff, Mirjam Minor, Meike Reichle, Kerstin Bach

Case-based reasoning (CBR) is a sub-field of Artificial Intelligence that deals with experience-based problem solving. CBR has its roots in different disciplines such as cognitive science, machine learning, and knowledge-based systems. Today, it is a well established research field of its own, which produced a rich variety of specific methods, as well as applications implementing those methods for particular tasks and domains. This paper gives a compact overview of CBR in general and further discusses recent advancements in selected topics.

1 Introduction

Case-based Reasoning (CBR) is a well established research field in Artificial Intelligence that involves the investigation of theoretical foundations [27], system development, and practical application building [10] of experience-based problem solving. The core of every case-based problem solver is the *case-base*, which is a collection of previously made and stored experience items, called *cases*. A case-based problem solver solves new problems primarily by reuse of solutions from the cases in the case base. For this purpose, one or several relevant cases are selected. This selection process is guided by one of the core assumptions behind CBR, namely that *similar problems have similar solutions*. Once similar cases are selected, the solution(s) from the case(s) are adapted to become a solution of the current problem. Finally, when a new (successful) solution to the new problem is found, a new experience is made, which can be stored in the case-base to increase its competence, thus implementing a learning behavior.

CBR has its roots in different disciplines, particularly cognitive science, machine learning, and knowledge based-systems, including knowledge representation and reasoning. CBR research is also related to certain topics in information retrieval, data bases, semantic web, and knowledge management. CBR is to a large degree characterized by the fact that it combines methods from different areas in AI in a particular manner for the purpose of experience-based problem solving. Hence, there is a strong focus on developing frameworks for certain types of problems (e.g. diagnosis, planning, product recommendation, experience management) as well as on implementing case-based systems for certain application domains (e.g. for medicine). However, there are also several CBR-related tasks for which other AI disciplines do not already provide a solid methodological foundation. For example, CBR research made significant original contributions to the field of similarity modeling, similarity-based retrieval, and adaptation. As several reviews of CBR exist [28, 32, 1], this paper provides only a compact overview on CBR in general and is then focussed on recent advancements in selected topics.

2 Knowledge and Reasoning in CBR

We now briefly describe CBR from the perspective of knowledge representation and reasoning as this view shows the similarities and differences to traditional methods used in AI.

2.1 Types of CBR

There are three main types of CBR that differ significantly from one another concerning case representation and reasoning: *structural*, *textual*, and *conversational* CBR [10]. The idea underlying the *structural CBR* approach is to represent cases according to a common structured vocabulary, i.e. an ontology. Once this vocabulary is defined, all cases are restricted to represent experience that can be expressed with this vocabulary. In the various structural case representations, the describing features of a case may be organized as flat attribute-value tables, in an object-oriented manner, as graph structures, or by sets of atomic formulas of a predicate logic language. The structural CBR approach is most widely used and will therefore be the focus of the remainder of this survey.

In *textual CBR*, there is no common case structure, but cases are represented as free text, i.e. strings [31]. This is very useful in domains where large collections of know-how text documents already exist and the intended user is able to make use of the experience contained in the respective documents immediately.

In *conversational CBR* [5] cases aim at capturing the knowledge contained in customer/agent conversations. A case is represented through a list of questions that varies from one case to another. There is no ontology and no standardized structure for all the cases.

Today, structural CBR approaches make also use of features from textual and conversational CBR. For example, textual cases can be mapped to a structural representation by information extraction and by methods for automating ontological annotations. Also, dialog components are used as part of CBR systems to implement specific strategies for user interaction.

2.2 CBR Cycle

Despite the many different appearances of CBR systems the essentials of CBR are captured in a surprisingly simple and uniform process model (see Fig. 1), the CBR cycle proposed by Aamodt and Plaza [3]. Several refinements have been proposed for different purposes, e.g. for better addressing maintenance issues in CBR. The CBR cycle consists of 4 sequential steps organized around the *knowledge of the CBR system*. Problem solving starts when a new problem (also called new case or query) must be solved. First, the *retrieve* phase, selects one or several similar cases from the case base. In the subsequent *reuse* phase the solutions contained in those cases are adapted according to the query. In the *revise* phase, the solution determined so far is verified in the real world and possibly corrected or improved, e.g. by a domain expert. Finally, the *retain* phase takes the feedback from the revise phase and updates the knowledge, particularly the case base.

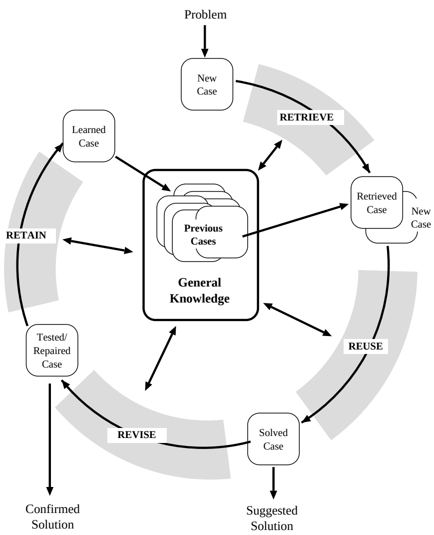

Figure 1: The CBR Cycle according to Aamodt and Plaza.

Michael M. Richter proposed a unified view on the knowledge contained in a structural CBR application by introducing different *knowledge containers* [46] thereby providing some additional structure to the general knowledge box in Figure 1. The knowledge containers are the *vocabulary*, the *case base*, the *similarity measure*, and the *adaptation knowledge*.

2.3 Vocabulary and Case Representation

The vocabulary (which we call ontology today) is the basis of all knowledge and experience representation in CBR. The vocabulary defines the information entities and structures (e.g. classes, relations, attributes, data types) that can be used to represent cases, similarity measures, and adaptation knowledge.

The case base is the primary form of knowledge in CBR. Traditionally, a case is considered as instance in the representation space defined by the vocabulary, e.g. a vector in an attribute-value representation. However, more advanced representations have been studied recently, such as hierarchical and generalized cases using case-specific constraints between attributes [54]. From a knowledge representation point of view, vocabulary and case-base representations are a quite standard applications of AI and database methods.

2.4 Similarity

The notion of similarity plays an important role in CBR since cases are selected based on their similarity to the current problem. While early CBR approaches were usually restricted to standard similarity measures (such as inverse Euclidean or Hamming distances), our current view is that the similarity measure encodes important knowledge of the domain. Consequently, it must be modeled as part of the knowledge acquisition process during application development. Similarity is usually formalized as a function $sim : P \times P \to [0, 1]$, which compares two problem descriptions from P and produces a similarity assessment as a real value from $[0, 1]$. A high value indicates a high similarity. The semantics of such similarity measures can be defined via the notions of preference and utility [9, 45]. For a new problem p a case $c_1 = (p_1, s_1)$ is preferred over a case $c_2 = (p_2, s_2)$ (we write $c_1 \succ_p c_2$) iff $sim(p, p_1) > sim(p, p_2)$, since the similarity-based retrieval lists c_1 before c_2. The exact similarity values are usually not important, but just the *preference order* $\succ_p$ that the similarity function induces on the case base for the problem p. The second observation is that this preference order should be in line with the assessment of the utility of the solution of the cases for solving the problem p during the reuse phase. Case c_1 should be preferred over c_2 ($c_1 \succ_p c_2$) if the utility of s_1 for solving p is higher than the *utility* of s_2 for solving p.

As a means for practical modeling of similarity functions, the local-global-principle, first proposed by Richter (for details see: [9, 45]) is widely used. Modeling similarity means decomposing the similarity function according to the vocabulary. This decomposition is done in such a way that *local similarity functions* for individual attributes model the preference according to the attribute only. Local similarities are then aggregated into the *global similarity* by an appropriate combination of the local similarity values. This aggregation also takes the different weights of attributes into account. This approach has been further developed to support complex representations of vocabularies, including object-oriented representations [9].

2.5 Retrieve

In the retrieve phase of the CBR cycle, one or several cases from the case base are selected, based on the modeled similarity. In a nutshell, the retrieval task is defined as finding a small number of cases from the case base with the highest similarity to the query. Hence, this is a k-nearest-neighbor retrieval task considering a specific similarity function. However, when the case base grows, the efficiency of retrieval decreases, because an increasing number of cases must be taken into account to find the most similar case. Thus one sub-branch of CBR research deals with methods that improve retrieval efficiency, e.g. by using specific index structures such as kd-trees, case-retrieval nets, or discrimination networks (an overview is given in [9]).

2.6 Reuse

When one or several similar cases have been retrieved, the solution (or other problem solving information) contained in these cases is reused to solve the current problem. Reusing a retrieved solution can be quite simple if the solution is returned unchanged as the proposed solution for the new problem. This is mostly appropriate for classification tasks with a limited number of solutions (classes) and a large number of cases. In such scenarios every potential solution is contained in the case base and hence adaptation is usually not required. On the other hand, for synthetic tasks (such as configuration or planning) solution adaptation is necessary since the solution space clearly exceeds the number of available cases or is even infinite.

Several techniques for adaptation in CBR have been proposed so far (for a review on adaptation see [1]). The most basic distinction between adaptation methods is whether transformational adaptation or generative adaptation is applied. *Transformational adaptation* relies on a set of adaptation rules or operators that describe how differences in the problem lead to required modifications in the solution. The present differences between the new case and the retrieved case are analyzed, a set of applicable transformations are selected, and the proposed modifications to the solution are performed. On the other hand, *generative adaptation* methods require a complete generative problem solver that is able to solve problems based on general knowledge, i.e., without using any cases at all. Not the solution but the problem solving traces from previous cases are then reused to guide the generative problem solver to find a solution to the new problem. This approach has its origin in derivational analogy.

In any case, the acquisition of such explicit adaptation knowledge is a very difficult and time consuming task. Hence, most practical CBR applications today try to avoid extensive adaptation for pragmatic reasons.

2.7 Revise

In this phase feedback related to the solution constructed so far is obtained. This feedback can be given in the form of a correctness rating of the result or in the form of a manually corrected *revised case*. The revised case or any other form of feedback enters the CBR system for its use in the subsequent retain phase.

2.8 Retain

The retain phase is the learning phase of a CBR system. The typical form of learning that occurs in a CBR system is learning by adding a revised case to the case base. Thereby, the new problem solving experience becomes available for reuse in future problem solving episodes. However, the continuous increase of the case base causes a utility problem as it continuously decreases retrieval efficiency. Explicit competence models [47] have been developed that enable the selective retention of cases.

In another branch of research, specific methods for learning similarity measures have been developed. While early approaches are restricted to learning feature weights [59] recent methods address the more difficult problem of learning local and global similarity functions [49, 17]. Very few approaches so far addressed the problem of learning adaptation knowledge [25, 19], which is probably an issue of future interest, for example in the context of the computer cooking contest [24].

3 Architectures, Frameworks, Tools

CBR can be found in both, architectures that are covering only the CBR methodology or architectures in which CBR is one methodology among others. The following paragraph presents recent work on CBR architectures, frameworks, or tools focusing solely on the CBR cycle (or on steps of it) within a knowledge-intensive application domain. The second paragraph describes multi-agent-system architectures in which CBR and its underlying methodology play a significant role.

CREEK[1] is an architecture for knowledge-intensive CBR, which has been developed at the Norwegian University of Science and Technology and focuses on open and weak-theory domains. Within CREEK, the case-based interpreter uses a three-step process to execute the retrieve, reuse and retain step. Each step consists of activating relevant parts of the semantic network (vocabulary), generating and explaining derived information, and selecting a conclusion that conforms with the goal [2]. The (Java-based) implementation of the CREEK architecture is the tool *TrollCreek*, which has been used to implement applications in the petroleum industry. *jCOLIBRI*[2] is an object-oriented framework for building CBR systems that has been developed by the Department of Software Engineering and Artificial Intelligence at Complutense University of Madrid. The underlying architecture of jCOLIBRI consists of two layers: The design layer and the developer layer. The design layer provides tools guiding users through the configuration process and explaining the components' behaviour within the application. The developer layer provides basic Java components, which are required to create a CBR application from a developer's point of view. The architecture includes connectors to import different kinds of data into a case base. Additionally, jCOLIBRI provides several extensions, among them components required for textual CBR [44]. jCOLIBRI is used for teaching and research purposes at several European universities and institutes. *myCBR*[3] [50] is an open source CBR tool developed at the DFKI, which focuses on domain and similarity modeling for case retrieval. Its intended use is in research and education, as well as for rapid prototyping of applications. Since recently, models created with myCBR can be used within applications built using the jCOLIBRI framework. *IUCBRF*[4] is also an open source framework for CBR system development, which is implemented in Java and developed at the Indiana University. The framework is designed to facilitate fast and modular development of CBR systems as well as providing a foundation for code sharing by those who are developing CBR systems. On the commercial side the probably most widely used tool in Europe is the *Information Access Suite* (named *e:IAS*) from empolis[5]. e:IAS uses CBR as its underlying methodology and provides a client-server based knowledge provision component, the Knowledge Server, and a knowledge modelling component, the Creator, for developing knowledge management systems for a variety of commercial application domains.

Another use of the CBR methodology is its inclusion in multi-agent-systems. The combination of CBR and multi-agent-systems can either happen in the form of agent-enhanced CBR, where intelligent software agents autonomously carry out tasks of the CBR process [7] or in the form of CBR-enhanced agents, where CBR is used to provide software agents with a reasoning capability [42]. Moreover, CBR can be combined with further technologies such as workflows [11]. The *CAKE* (Collaborative Agent-based Knowledge Engine) architecture, for instance, combines workflow technology, agent technology, and

[1] http://creek.idi.ntnu.no/
[2] http://gaia.fdi.ucm.es/projects/jcolibri/
[3] www.mycbr-project.net
[4] www.cs.indiana.edu/ sbogaert/CBR/
[5] www.empolis.com

structural CBR to select appropriate agents and workflows in knowledge-intensive application domains using CBR [11]. Plaza and Ontañón present the *AMAL* (Argumentation-based Multi-Agent Learning) framework [40] for multi-agent learning, in which CBR-agents increase prediction quality, efficiency, the range of solvable problems and the range of accessible resources by arguing proposed solutions (learning to generate arguments and counterarguments). Within AMAL, software agents are able to advance their reasoning capabilities by learning knowledge and preference relations from experience. The distribution of tasks and knowledge by means of a Collaborative Multi-Expert-System has been introduced by Bach et al. [8] presenting the *SEASALT* (Sharing Experience using an Agent-based System Architecture LayouT) architecture. This architecture comprises, among other components, the so-called Knowledge Line, a number of intelligent CBR-based agents, which enable distributed knowledge management. Further on, these agents' case bases are again maintained by agents (Case Factory), thus combining the approaches of CBR-enhanced agents and agent-enhanced CBR.

4 CBR and the Future Internet

The development of the future internet is affected by two major factors: semantics and collaboration. Since the amount of information available on the internet is rapidly increasing, a more meaningful, semantic description of the available content is necessary in order to tackle the problem of information overflow. This need is addressed by the Semantic Web. At the same time the internet's content no longer mainly consists of editorial content and static private homepages, but has also become a social medium. Today the internet is not only used for information gathering, but also for social networking and sharing personal views, experiences and media. This development is usually subsumed by the term Web 2.0. Both these new fields, the Semantic Web as well as the Web 2.0, benefit from the application of case-based reasoning.

Two of the most influencing developments of the Semantic Web are the resource description language RDF (*Resource Description Framework*) and the knowledge representation language OWL (*Web Ontology Language*), which is based on RDF. Already before the development of RDF and OWL, XML has been used as a case representation within the case-based reasoning community. There is a notable similarity between the ontologies developed within semantic applications and the representation of cases in structural case-based reasoning. Due to this similarity RDF and OWL both lend themselves to be used as case representation languages and thus expand the possibilities of case-based reasoning within the general WWW. Bergmann and Schaaf [12] illustrate the technological and methodological similarities between ontologies and structured case-based reasoning and describe the synergies that can be reached by merging both approaches. A similar approach is presented by Chen and Wu [16], who describe an RDF based Case Markup Language called CaseML. CaseML offers a domain-independent case ontology and also aims to make case-based reasoning available within the Semantic Web. Aktas et al. [6] also use RDF for case representation in their system SERVOGrid. Here it is embedded in a conversational case-based reasoning system that aids scientists in finding resources such as program code or data that are needed to solve a specific task by assisting them in describing the necessary resources using meta data. Another case-based reasoning architecture that is making heavy use of Semantic Web technologies is the jCOLIBRI framework. Here OWL is being used as the case interchange language and it is planned to advance the already distributed framework towards an architecture consisting of Semantic Web Services (SWS) where problem solving methods are represented as Web Services [43]. In order to use these services the whole case-based reasoning process is decomposed into single tasks, which are then carried out by according Web Services. D'Aquin et al. [20] also aim to integrate case-based reasoning with Semantic Web technologies, but this time using OWL for representing adaptation knowledge and applying OWL reasoning in order to carry out case-based reasoning's retrieval and adaptation steps.

Within Web 2.0 the area of research being most related to case-based reasoning is the collaborative filtering approach. Already in 2001 Hayes et al. [26] described the close relation between collaborative filtering and CBR and how these can benefit from each other. Since then combinations of CBR and collaborative filtering have been presented in several papers. O'Sullivan et al. [53] use collaborative filtering profiles as cases in a TV recommender system. There are also other combinations possible such as in the work by Chedrawy et al. [15], where CBR adaptation techniques are used to refine the results of collaborative filtering or in the work by Aguzzoli et al. [4], in which collaborative filtering is used to assess the similarity between songs in a CBR system creating custom music compilations (CoCoA). Guo et al. [23] further extend the application of CBR on collaborative filtering data – this time within a recommender system in e-commerce – by additionally adding a social trust model to improve prediction accuracy. A slightly different approach has been presented in the recent paper by Briggs and Smyth [14], which illustrates the idea of a community based web search that uses the results of previous web searches of similar users in order to improve web search results.

Most of the approaches presented above rely on already structured or otherwise prepared cases. Given the assumption that the cases used in CBR systems are usually representation of experiential knowledge, the next logic step would be to try to directly access and (re)use the experiential knowledge enclosed in Web 2.0 platforms. Plaza [41] introduces the EDIR cycle for this purpose, which describes four processes (*Express - Discover - Interpret - Reuse*) for reusing experiential knowledge directly from the web.

5 Procedural Knowledge in CBR

Procedural knowledge plays an important role in many companies. Experience in the form of procedures has found its way into CBR mainly in two fields: *case-based planning* (CBP) and *workflow-oriented CBR*.

In CBP, a plan describes procedural knowledge per se. The single steps of a plan as well as the sequential or hierarchical order of the steps contain a lot of experience that can be addressed by CBR. Instead of planning from scratch, CBP focusses mainly on retrieval and adaptation of plans. CBP has a history of nearly 20 years, which has been summarized in the literature [48, 1]. As it is still a vivid research area, we will concentrate here only on some recent trends that have emerged in the past three years. The *utility of cases* for solving planning problems has been investigated in order to improve the performance of planning [21]. Some work has been done on *planning under real-time constraints* [39, 52]. By considering *personalized plan execution* [29], an existing knowledge planning approach (plans are search procedures) has been extended in order to rank the results of applying the plan (search results) according to user preferences.

Workflows are "*the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules*" [18]. In a medical domain, for instance, such tasks might be the diagnosis and treatment steps for a patient, while the procedural rules determine the control flow for the execution of the tasks. *Workflow-oriented CBR* deals with knowledge on how to model, apply, and adapt procedures in the form of workflows. The reuse of workflow templates is widely spread in recent commercial systems. Before a workflow is enacted, a new workflow instance is derived from the workflow template. CBR is a means to go beyond this kind of assistance. First case-based approaches exist that give sophisticated *modeling support* for workflows. Madhusudan et al. [33] support the incremental modeling of workflows by a similarity-based reuse of the workflow templates. A process repository with workflows and workflow snippets is used by an HTN planner that employs an inexact, graph-based matching. Leake and Morwick [30] evaluate the execution paths of past workflows in order to support user extension of workflows that are under construction. The *application of workflows* is supported by CBR as well. Case-based retrieval functionality is employed to select suitable workflows for a certain situation [11] or for knowledge-intensive tasks of a knowledge worker [56]. The *adaptation of workflows* at execution time is also a recent research topic in CBR. It can not yet be performed automatically but it can benefit from CBR approaches. Conversational CBR has been applied in the tool CBRFlow [58] to guide the user in adapting a workflow to changing circumstances. The changes between revisions of past workflows have been reused in the tool CAKE [34] in order to generate suggestions for workflow adaptations to the user.

6 CBR Applications

During the past twenty years, many CBR applications have been developed, ranging from prototypical applications build in research labs to large-scale fielded applications developed by commercial companies. Application areas of CBR include help-desk and customer service, recommender systems in electronic commerce, knowledge and experience management, medical applications and applications in image processing, applications in law, technical diagnosis, design, planning, as well as applications in the computer games and music domain. CBR applications are documented in part in the conference proceedings of the regular ICCBR/ECCBR conferences as well as on the Industry Day included in these conferences.

Watson [57] reports that more than 130 major companies had been fielding CBR applications worldwide till 1997. Early prominent fielded applications in engineering and technical domains included CLAVIER (configuration of autoclave loadings), Cassiopee (troubleshooting of CFM56-3 engines in the Boeing 737) and ICARUS (locomotive diagnostics). Help-desks and customer support are two of the most obvious applications of CBR and many such applications had already been implemented by 1999 [51].

An important application area that also specifically addresses the method-oriented view of CBR is experience management or, if considered from a more general perspective, knowledge management. A well-known application example is SQUAD, which deals with corporate knowledge management for supporting software quality control. While Tautz [55] describes how CBR is used as part of a more general method for developing experience management applications, Nick [37] provides details on a number of real-life applications of CBR for experience management tasks. Interesting real-life applications on experience management include the variation reduction adviser [36], and the connection machine [22], an approach for collective expertise development.

Bichindaritz [13] gives an overview of the use of CBR in the health sciences. Bergmann at al. [10] describe more than two dozens of real-life applications of CBR developed using the INRECA methodology and tools. Recent applications on autonomous systems include applications of CBR for ambient intelligence [38] and self-healing [35]. Mantaras et al. [1] give an overview on emergent applications of CBR in domains like music (e.g., generating expressive musical performances), poetry generation, computer games (script adaptation, character control, planning in strategy games), molecular biology (speed-up of crystal growth process for proteins) and support for spatial reasoning in geographical information systems.

7 Conclusion

By now it is difficult to exhaustively report on a few pages on the current state of the art in CBR, because the number of CBR approaches and applications developed up to now has become quite large. Apart from the broad applicability of CBR, one reason for this is that there is meanwhile a significant number of CBR research groups and commercial companies, which develop CBR methods, software components, and applications on a regular basis. Another reason is that already in the 1990s various CBR researchers pointed out that CBR is not only a technology but also a (process oriented) method. Since CBR researchers are experts in processing experience, detailing, adapting, and improving the CBR method for various innovative and challenging application fields and tasks has been an important research area in the CBR community from its very beginning. This also explains why the combination of CBR with various other technologies within a great bandwidth of applications has become increasingly attractive for researchers as well as business professionals. Especially for these integrated approaches it is not always made explicit that there is indeed CBR inside.

References

[1] Case-based reasoning commentaries. *The Knowledge Engineering Review*, 20(3), 2006.

[2] A. Aamodt. Knowledge-intensive case-based reasoning in creek. In *Advances in Case-Based Reasoning, ECCBR 2004*, LNAI 3155, pages 1–15. Springer, 2004.

[3] A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, 1994.

[4] S. Aguzzoli, P. Avesani, and P. Massa. Collaborative case-based recommender system. In S. Craw and A. Preece, editors, *Advances in Case Based Reasoning, ECCBR 2002*, pages 460–474. Springer, 2002.

[5] D. Aha, L. A. Breslow, and H. Munoz-Avila. Conversational case-based reasoning. *Applied Intelligence*, 2001.

[6] M. S. Aktas, M. Pierce, G. C. Fox, and D. Leake. A web based conversational case-based recommender system for ontology aided metadata discovery. In *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, GRID2004*, pages 69–75. IEEE Computer Society, 2004.

[7] K.-D. Althoff, K. Bach, J.-O. Deutsch, A. Hanft, J. Mänz, T. Müller, R. Newo, M. Reichle, M. Schaaf, and K.-H. Weis. Collaborative multi-expert-systems – realizing knowlegde-product-lines with case factories and distributed learning systems. In *3rd Workshop on Knowledge Engineering and Software Engineering (KESE 2007)*, 2007.

[8] K. Bach, M. Reichle, and K.-D. Althoff. A domain independent system architecture for sharing experience. In A. Hinneburg, editor, *Proceedings of LWA 2007, Workshop Wissens- und Erfahrungsmanagement*, pages 296–303, 2007.

[9] R. Bergmann. *Experience Management - Foundations, Development Methodology, and Internet-Based Applications*. LNAI 2432. Springer, 2002.

[10] R. Bergmann, K.-D. Althoff, S. Breen, M. Göker, M. Manago, R. Traphöner, and S. Wess. *Developing industrial case-based reasoning applications: The INRECA methodology*. LNAI 1612. Springer, 2nd Edition, 2003.

[11] R. Bergmann, A. Fressmann, K. Maximini, R. Maximini, and T. Sauer. Case-based support for collaborative business. In *Advances in Case-Based Reasoning, ECCBR 2006*, LNAI 4106, pages 106–120. Springer, 2006.

[12] R. Bergmann and M. Schaaf. Structural case-based reasoning and ontology-based knowledge management: A perfect match? *Journal of Universal Computer Science*, 9(7):608–626, 2003.

[13] I. Bichindaritz. Role and significance of case-based reasoning in the health sciences. *This Volume*.

[14] P. Briggs and B. Smyth. Provenance, trust, and sharing in peer-to-peer case-based web search. In *Advances in Case-Based Reasoning, ECCBR 2008*, LNAI 5239, pages 89–103. Springer, 2008.

[15] Z. Chedrawy, S. Sibte, and R. Abidi. Intelligent knowledge sharing strategy featuring item-based collaborative filtering and case based reasoning. In *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications, ISDA2005*, pages 67–72. IEEE Computer Society, 2005.

[16] H. Chen and Z. Wu. Caseml: A rdf-based case markup language for case-based reasoning in semantic web. Technical Report 4/2003, Norwegian University of Science and Technology, Department of Computer and Information Science, 2003.

[17] W. Cheng and E. Hüllermeier. Learning similarity functions from qualitative feedback. In *Advances in Case-Based Reasoning, ECCBR 2008*, LNAI 5239. Springer, 2008.

[18] Workflow Management Coalition. Workflow management coalition glossary & terminology. Retrieved on Dec 16, 2008 from http://www.wfmc.org/View-document-details/WFMC-TC-1011-Ver-3-Terminology-and-Glossary-English.html, 1999.

[19] S. Craw, J. Jarmulak, and R. Rowe. Learning and applying case-based adaptation knowledge. In *Case-Based Reasoning Research and Development, ICCBR 2001*. Springer, 2001.

[20] M. d'Aquin, J. Lieber, and A. Napoli. Case-based reasoning within semantic web technologies. In *Proceedings of the 12th International Conference on Artificial Intelligence: Methodology, Systems, and Applications, AIMSA2006*, pages 190–200. Springer, 2006.

[21] T. de la Rosa, A. García Olaya, and D. Borrajo. Using cases utility for heuristic planning improvement. In *Case-Based Reasoning Research and Development, ICCBR 2007*, LNAI 4726, pages 137–148. Springer, 2007.

[22] M. H. Göker, C. Thompson, S. Arajärvi, and K. Hua. Connecting people with questions to people with answers. *Künstliche Intelligenz 4/07*, pages 23–27, 2007.

[23] Y. Guo, G. Deng, G. Zhang, and C. Luo. Using case-based reasoning and social trust to improve the performance of recommender system in e-commerce. In *Proceedings of the Second International Conference on Innovative Computing, Informatio and Control, ICICIC 2007*, page 484. IEEE Computer Society, 2007.

[24] A. Hanft, N. Ihle, K. Bach, and R. Newo. Cookiis – competing in the first computer cooking contest. *This Volume*.

[25] K. Hanney and M. Keane. Learning adaptation rules from a case base. In *Proceedings of the 3th European Workshop on Case-Based Reasoning*, LNAI 1168, pages 179–192. Springer, 1996.

[26] C. Hayes, P. Cunningham, and B. Smyth. A case-based reasoning view of automated collaborative filtering. In D. Aha and I. Watson, editors, *Case-Based Reasoning Research and Development, ICCBR 2001*, pages 243–248. Springer, 2001.

[27] E. Hüllermeyer. *Case-Based Approximate Reasoning*. Springer, 2007. See book review in this volume.

[28] J. Kolodner. Case based reasoning, 1993.

[29] D. Leake and J. Powell. Knowledge planning and learned personalization for web-based case adaptation. In *Advances in Case-Based Reasoning, ECCBR 2008*, LNAI 5239, pages 284–298. Springer, 2008.

[30] D. B. Leake and J. Kendall-Morwick. Towards case-based support for e-science workflow generation by mining provenance. In *Advances in Case-Based Reasoning, ECCBR 2008*, LNAI 5239, pages 269–283. Springer, 2008.

[31] M. Lenz and K. Ashley, editors. *Proceedings of the AAAI98 Workshop on Textual Case-Based Reasoning*. AAAI Press, 1998.

[32] M. Lenz, Bartsch-Spörl B., H.-D. Burkhard, and S. Wess, editors. *Case-Based Reasoning Technology: From Foundations to Applications*. Springer, 1998.

[33] T. Madhusudan, J. L. Zhao, and B. Marshall. A case-based reasoning framework for workflow model management. *Data and Knowledge Engineering*, 50:87–115, 2004.

[34] M. Minor, A. Tartakovski, and R. Bergmann. Representation and structure-based similarity assessment for agile workflows. In *Case-Based Reasoning Research and Development, ICCBR 2007*, LNCS 4626. Springer, 2007.

[35] S. Montani and C. Anglano. Case-based reasoning for autonomous service failure diagnosis and remediation in software systems. In *Advances in Case-Based Reasoning, ECCBR 2006*, LNAI 4106, pages 489–503, 2006.

[36] A. P. Morgan, J. A. Cafeo, K. Godden, R. M. Lesperance, A. M. Simon, D. L. McGuinness, and J. L. Benedict. The general motors variation – reduction adviser: A deployed system for experience management. *Künstliche Intelligenz 4/07*, pages 27–29, 2007.

[37] M. Nick. *Experience Maintenance Through Closed-Loop Feedback*. PhD thesis, University of Kaiserslautern, 2005.

[38] M. Nick, S. Schneickert, J. Grotepaß, and I. Heine. Checkmate. *Künstliche Intelligenz 4/07*, pages 34–37, 2007.

[39] S. Ontañón, K. Mishra, N. Sugandh, and A. Ram. Case-based planning and execution for real-time strategy games. In *Case-Based Reasoning Research and Development, ICCBR 2007*, LNAI 4726, pages 164–178. Springer, 2007.

[40] S. Ontañón and E. Plaza. Learning and joint deliberation through argumentation in multi-agent systems. In *Proc. AAMAS-2007*, volume 13 N. 3, pages 971–978, 2007.

[41] E. Plaza. On reusing other people's experiences. *This Volume*.

[42] E. Plaza and S. Ontañón. Learning collaboration strategies for committees of learning agents. *Journal of Autonomous Agents and Multi-Agent Systems*, 13 N. 3:429–461, 2006.

[43] J. A. Recio-García, B. Díaz-Agudo, and P. A. González-Calero. A distributed cbr framework through semantic web services. In *Research and Development in Intelligent Systems XXII*, pages 88–101. Springer, 2005.

[44] J. A. Recio-García, B. Díaz-Agudo, A. Sánchez-Ruiz, and P. A. González-Calero. Lessons learnt in the development of a cbr framework. In *Proccedings of the 11th UK Workshop on Case Based Reasoning*, pages 60–71. CMS Press,University of Greenwich, 2006.

[45] M. Richter. Foundations of similarity and utility. In *Proceedings of the 20th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2007)*. AAAI Press, 2007.

[46] M. M. Richter. The knowledge contained in similarity measures. Invited Talk on the ICCBR-95, 1995.

[47] B. Smyth and E. McKenna. Competence models and the maintenance problem. *Computational Intelligence*, 17(2):235–249, 2001.

[48] L. Spalazzi. A survey on Case-Based Planning. *Artificial intelligence review*, 16:3–36, 2001.

[49] A. Stahl. Learning similarity measures: A formal view based on a generalized cbr model. In *Case-Based Reasoning Research and Development, ICCBR 2005*, LNAI 3620. Springer, 2005.

[50] A. Stahl and T. Roth-Berghofer. Rapid prototyping of cbr applications with the open source tool mycbr. *This Volume*.

[51] M. Stolpmann and S. Wess. *Optimierung der Kundenbeziehungen mit CBR-Systemen – Intelligente Systeme für E-Commerce und Support*. Addison Wesley, 1999.

[52] N. Sugandh, S. Ontañón, and A. Ram. Real-time plan adaptation for case-based planning in real-time strategy games. In *Advances in Case-Based Reasoning, ECCBR 2008*, LNAI 5239, pages 533–547. Springer, 2008.

[53] D. O' Sullivan, D. Wilson, and B. Smyth. Improving case-based recommendation: A collaborative filtering approach. In *Proceedings of the 6th European Conference on Advances in Case-Based Reasoning, ECCBR 2002*, pages 278–291. Springer, 2002.

[54] A. Tartakovski. Reasoning with generalized cases. *This Volume*.

[55] C. Tautz. *Customizing Software Engineering Experience Management Systems to Organizational Needs*. PhD thesis, University of Kaiserslautern, 2001.

[56] L. van Elst, F. R. Aschoff, A. Bernardi, H. Maus, and S. Schwarz. Weakly-structured Workflows for Knowledge-intensive Tasks: An Experimental Evaluation. In *12th IEEE International Workshops on Enabling Technologies, WETICE 2003*, pages 340–345. IEEE Computer Society, 2003.

[57] I. Watson. *Applying case-based reasoning: techniques for enterprise systems*. Morgan Kaufmann Publishers Inc., 1998.

[58] B. Weber, W. Wild, and R. Breu. CBRFlow: Enabling Adaptive Workflow Management Through Conversational Case-Based Reasoning. In *Advances in Case-Based Reasoning, ECCBR 2004*, LNCS 3155, pages 434 – 448. Springer, 2004.

[59] D. Wettschereck and D.W. Aha. Weighting features. In *Case-Based Reasoning Research and Development, Proc. ICCBR-95*, LNAI 1010, pages 347–358. Springer, 1995.

Contact

Prof. Dr. Ralph Bergmann
University of Trier
Department of Business Information Systems II
54286 Trier, Germany
Tel.: +49 (0)651 201 3876
Email: bergmann@uni-trier.de

Prof. Dr. Klaus-Dieter Althoff
University of Hildesheim
Intelligent Information Systems Lab
Marienburger Platz 22, 31141 Hildesheim
Tel.: + 49 (0)5121-883 751
Email: althoff@iis.uni-hildesheim.de

Bild

Ralph Bergmann is full professor at the University of Trier and is directing a research group on business information systems. He studied computer science at the University of Kaiserslautern and finished his doctoral dissertation on hierarchical case-based planning in 1996. In 2001 he received the postdoctoral degree (Habilitation) with a thesis on experience management. His main interests include case-based reasoning, experience management, machine learning, semantic technologies including semantic workflow reasoning.

Bild

Klaus-Dieter Althoff is full professor at the University of Hildesheim and is directing the Intelligent Information Systems Lab. He wrote his Phd thesis on an architecture for knowledge-based technical diagnosis at the University of Kaiserslautern, where he also received the postdoctoral degree (Habilitation) with a thesis on the evaluation of case-based reasoning systems. His main interests include knowledge-based systems, with a focus on case-based reasoning, agent technology, experience management, and machine learning.

Bild

Mirjam Minor is a postdoc at the University of Trier, Germany, in the research group on business information systems. She studied computer science at the Humboldt Berlin, and received her PhD degree there in 2006. Her main research interests are in agile workflow technology, case-based reasoning, experience & knowledge management, and multiagent systems.

Bild

Meike Reichle studied International Information Management with Computer Science and History as minors at the University of Hildesheim. She graduated in 2007 and is since working in a PhD position at the University of Hildesheim's Intelligent Information Systems Lab. Her main research interests include the combination of case-based reasoning and machine learning techniques, multi-agent systems, knowledge and experience maintenance, and natural language processing.

Bild

Kerstin Bach studied Information Management and Information Technology at the University of Hildesheim from 2001 to 2007. She graduated in 2007 and is since working in a PhD position at the University of Hildesheim's Intelligent Information Systems Lab. Her main research interests include intelligent information systems, agent-based systems, travel medicine, case-based reasoning, textual case-based reasoning, knowledge discovery and similarity.