Glossary

Adaptation. Adjusting a retrieved case to fit the given problem situation by applying *adaptation knowledge*.

Adaptation Knowledge. Adaptation knowledge is the expert domain knowledge about how to adapt a retrieved case from the case base so that it is better suited to the current problem. Adaptation knowledge is not mandatory for CBR applications but can be optionally used to improve the retrieved solutions.

Agents. In the INRECA methodology, agents are humans who enact a *method* in order to perform a certain *process*.

Attribute-Value Pair. A case can be represented as a flat set of attribute-value pairs where each pair encodes the value of a certain attribute for that case.

Awareness. The success of the project depends on all prospective users being made aware both of its implications and of the possibility that "there is something in it for them," thus generating motivation.

Case. A case represents one piece of reusable experience. Typically, a case consists of two parts: a problem description and a solution.

Case Base. This is the collection of all available cases.

Case Base Administrator. She/he has all rights over the system, including the right to update the domain model. Updating should be done occasionally as new experiences arise.

Case-Based Reasoning. Case-Based Reasoning (CBR) tries to model the acting by experience. It maintains a memory of experiences (*case base*) and solves new problems by *retrieving* similar cases from the cases.

Case-Based Reasoning Cycle. The CBR cycle describes the basic steps involved in case-based problem solving: *retrieve*, *reuse*, *revise*, and *retain*.

Case Buffer. This is a temporary location where current cases are stored pending a decision about whether or not to edit them into the case base.

CASUEL. CASUEL (Manago et al. 1994) is the object-oriented case representation language developed by INRECA. It allows the developer to define *domain models*, *cases*, *similarity measures*, and *adaptation knowledge*.

Common Generic Level. This is a level in the experience base of the INRECA methodology at which processes, products, and methods are collected that are common for a very large spectrum of different CBR applications.

Conversational CBR Approach. An approach to CBR in which a case is represented as a flat list of questions and answers, and the list of questions is different for every case; there is no *domain model*.

Cookbook Level. This is a level in the experience base of the INRECA methodology at which processes, products, and methods are tailored for a particular class of applications (e.g., help-desk, technical maintenance, product catalog). For each application class, the cookbook level contains a *recipe* (see below).

Corporate Memory. When corporate knowledge becomes easily retrievable for decision support, one speaks of a "corporate memory."

CQL. CQL is the case query language that is based on CASUEL. It is used to formulate queries to a CBR retrieval engine.

Decision Tree. This is one form of the output of an "induction engine." It displays successive partitions of a case base into subsets differentiated from each other by the values of parameters. The first parameter selected for the subdivision is the one generating the greatest information gain as regards factors leading to the target outcome. The subdivision process is then repeated for all other parameters within each branch. See also *fault tree*.

Deployment. When a system has been developed to the satisfaction of the users, it is usual to generate a "runtime version" of the software, in which all changes made to date are embedded robustly, and system users can use it without help from the developers. The system is then said to be "deployed."

Domain Model. In the *structural CBR approach*, the domain model is a set of attributes, with either defined sets of symbolic values or defined ranges of numerical values sufficient to characterize each unit of knowledge in the knowledge domain. Each case is represented using the attributes, each of which is given one of the allowed values.

Experience Base. The experience base is the collection of software development experience within the experience factory.

Experience Factory. An experience factory is a logical and/or physical organization that supports project development by analyzing and synthesizing all kinds of experience, by acting as a repository for such experience, and by

supplying that experience to various projects on demand. An experience factory packages experience by building informal, formal, or schematized models.

Fat Client. When the CBR system is working in client-server mode, the master-version of the system resides on the server. However, for the system to work at an acceptable level of performance, a client with lots of computation capabilities might be necessary. This is called a "fat client," since it could require significant time to download to the client machine.

Fault Tree. A fault tree is a *decision tree* where the branches are based on the evidence leading to the diagnosis of a fault.

GUI (Graphical User Interface). The graphical user interface is usually customized interactively with the user to meet the user's needs. The information should be presented without overwhelming the user.

Hierarchical Model. Hierarchical models breakdown a complex system into a hierarchy of submodels, each of which has its own case base. A rule of thumb for the scale of a submodel is that it should have about 15 or so attributes.

Induction. Induction is the generation of rules or decision trees for achieving a desired outcome. The rules or decision trees are generated automatically from the analysis of cases in a case base. This induction process abstracts from the experience of many expert decisions stored as cases in the case base.

Initial Domain. This is a subset of the total target domain that is used in initial trials of the CBR approach. It should be selected so that positively perceived results are obtained rapidly, thus creating *awareness*.

Integration. Integration means bringing together the various parts of the CBR application: The search engine, GUIs, case base, related database, and so on.

Knowledge Base. Knowledge base is a generic word for an assembly of chunks of formally represented, distilled knowledge, some or all of which may be in the form of *cases*.

Knowledge Container. Knowledge container model, introduced by Michael Richter (see Richter 1998), describes the knowledge that a CBR system uses. The containers are the *vocabulary*, i.e., the *domain model*, the *case base*, the *similarity measure*, and the *adaptation knowledge*. In principle, each container could be used to represent most of the knowledge, but for efficient application development it must be carefully decided which knowledge to put into which container.

Managerial Process. The primary goal of managerial processes is to provide an environment and services so that software that meets the product requirements and project goals can be developed. The services enact the technical and the organizational processes.

Method. A method is a particular way of achieving a specific goal. A method can be simple or complex. In the latter case, it can embody a number of subprocesses and intermediate products.

Methodology. A methodology is a collection of methods and guidelines that enables a person to work effectively and efficiently in the domain for which the methodology has been developed.

Nearest Neighbor Retrieval. This is a search approach that selects experience based on some geometrical distance computed in the attribute space. The search engine evaluates the n-dimensional "distance" between the query and all cases in the case base, taking into account the weights. The results are presented in order of n-dimensional "proximity."

Organizational Process. Organizational processes cover those parts of the business process that need to be changed in order to make best use of a new software system. They address those parts of the user organization's business process in which the software system will be embedded. New processes have to be introduced into an existing business process, such as the training of end-users or the technical maintenance of the system. Existing processes may need to be changed or reorganized in order to make best use of the new software system.

Precision. This is the proportion of retrieved cases that turn out to be relevant to a user who needs specific knowledge.

Process. A process is an activity that has the goal of transforming some input product(s) into some output product(s). It is a clearly defined step in a development project.

Process Input Product. A process input product is a product that is consumed by a process in order to generate the desired output product.

Process Model. Process models identify and document the *processes*, *products*, and *methods* in a clear and understandable way. In the INRECA methodology, process models are used to document the experience stored in *cookbook-level recipes* and on the *common generic leve*l.

Product. A product is an object that is either consumed as input, modified, or created as output of a *process*. A "modified product" is a product that is changed during the enactment of a process. Typically, it existed before the process was executed. An output product is a product that is created as a result of the enactment of a process. This product did not exist before the process was executed.

Project Plan. The project plan is the temporal and logical sequence of processes that have to be executed and the products that have to be consumed, modified, and created to achieve the goal of the overall project.

Recall. This is the proportion of relevant cases from the cases base (in the context of the user's current knowledge need) that where retrieved by the retrieval engine.

Recipe. A recipe describes the processes used to build a CBR application for a particular application class. All recipes are collected in the cookbook level of the *experience base* of the INRECA methodology.

Resources. Resources are objects that might be required to achieve a project goal. They can be financial, temporal, or material, as well as human resources.

Retain. The retain phase is the fourth step in the *CBR cycle*. Retain means storing new experience in the case base.

Retrieval. The retrieval phase is the first step in the *CBR cycle*. Retrieval means selecting a relevant case from the case base. There are different techniques for retrieval, like traversing an induction tree or nearest neighbor retrieval.

Retrieval Engine. This is a software component that performs the retrieval, i.e., it selects a case from the case base.

Reuse. The reuse phase is the second step in the *CBR cycle*. Reuse is a synonym for "adaptation." It means modifying the retrieved case to fit the given problem situation.

Revise. The revise phase is the third step in the *CBR cycle*. During revision the proposed solution case is applied and evaluated in the business environment. If necessary, the proposed solution can be improved.

Second-Level Support. People in a company who have a greater depth of expertise than those in the "front office," who possess routine knowledge. Front-office people may have to refer problems to these experts.

Similarity Measure. A similarity measure is a computational function that computes the similarity between a case and a query. The similarity measure contains expert knowledge that evaluates whether a case contains information that is reusable in the current context defined by the query.

Software Process Modeling. Software process modeling defines what *processes* must be enacted and what *products* must consumed, modified, or created within a software project.

Specific Project Level. The specific project level describes experience in the context of a single, particular project. It contains project-specific information, such as a description of the particular *processes* that were carried out.

Structural CBR Approach. This is a CBR approach that relies on cases that are described with a set of predefined attributes. These attributes are described in a *domain model*.

Technical Process. Technical processes transform product information from the problem description to the final (software) system. They cover the development of the system and the required documentation itself.

Textual CBR Approach. In this CBR approach, cases are represented in free-text form. Keyword matching techniques are used for retrieval. There is no *domain model*.

Tool. A tool is a piece of software, or a hardware-software combination, used by an *agent* to enact a *process* according to a *method*.

Vertical Platform. A vertical platform is a collection of several preconfigured software components together with development guidelines (a *recipe* on the *cookbook level*) particularly tailored for a restricted application domain, such as a help-desk. It is often implemented using a general purpose CBR engine but incorporates domain-specific modifications.

References

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

Aha, D., & Breslow, L. (1997). Refining conversational case libraries. In: Leake, B. & Plaza, E. (Eds.), *Case-Based Reasoning Research and Development* (ICCBR-97), Lecture Notes in Artificial Intelligence. Springer, Berlin, Heidelberg.

Aha, D., & Weber, R. (Eds.) (2000). *Proceedings of the Workshop on Intelligent Lessons Learned Systems at 17th National Conference on AI (AAAI-00).* American Association for Artificial Intelligence.

Aha, D., Breslow, L. & Munoz-Avila, H. (2001). Conversational case-based reasoning. *Applied Intelligence*.

Akkermans, H., Spiel, P.-H. & Ratcliffe, A. (1999). Problem, opportunity, and feasibility analysis for knowledge management: An industrial case study. In: *Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modelling, and Management (KAW '99), Track on "Knowledge Management and Knowledge Distribution throughout the Internet*." Banff, Canada. URL: http://sern.ucalgary.ca/KAW/KAW99/papers/Akkermans1/kaw99-jma.pdf.

Althoff, K.-D. (2001). Case-Based Reasoning. In: Chang, S.K. (Ed.), *Handbook on Software Engineering and Knowledge Engineering. Vol. 1 'Fundamentals* ," pp 549–588. World Scientific.

Althoff, K.-D., Auriol, E., Barletta, R. & Manago, M. (1995*). A Review of Industrial Case-Based Reasoning Tools*. AI Perspectives Report, AI Intelligence, Oxford, United Kingdom.

Althoff, K.-D., Becker-Kornstaedt, U., Decker, B., Klotz, A., Leopold, E., Rech, J. & Voss, A. (2002). The indiGo Project: Enhancement of Experience Management and Process Learning with Moderated Discourses. In: Perner, P. (Ed.), *Data Mining in E-Commerce, Medicine and Knowledge Management*, Lecture Notes in Computer Science. Springer.

Althoff, K.-D., Birk, A., Gresse von Wangenheim, C. & Tautz, C. (1998). Case-based reasoning for experimental software engineering. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D. & Wess, S. (Eds.), *Case-Based Reasoning Technology from Foundations to Applications*, Lecture Notes in Artificial Intelligence 1400, Chapter 9, pp 235–253. Springer.

Althoff, K.-D., Birk, A., Hartkopf, S., Müller, W., Nick, M., Surmann, D. & Tautz, C. (1999). Managing Software Engineering Experience for Comprehensive Reuse. *Proceedings of the Eleventh Conference on Software Engineering and Knowledge Engineering*, Kaiserslautern, Germany, June 1999. Knowledge Systems Institute, Skokie, Illinois, USA.

Althoff, K.-D., Bomarius, F., Müller, W. & Nick, M. (1999a). Using Case-Based Reasoning for Supporting Continuous Improvement Processes in Hospitals. In: *Wrobel et al. (1999)*.

Althoff, K.-D., Bomarius, F. & Tautz, C. (2000). Knowledge Management for Building Learning Software Organizations. *Information Systems Frontiers* 2(3/4):349–367. Kluwer Academic Publishers.

Althoff, K.-D., Decker, B., Hartkopf, S., Jedlitschka, A., Nick, M. & Rech, J. (2001). Experience Management: The Fraunhofer IESE Experience Factory. In: Perner, P. (Ed.), *Proc. Industrial Conference Data Mining*, 24–25 July 2001. Institute for Image Processing and Applied Computer Sciences, Leipzig, Germany.

Althoff, K.-D., Nick, M. & Tautz, C. (1999b). Improving organizational memories through user feedback. In: Bomarius, F. (Ed.), *Workshop on Learning Software Organisations at SEKE'99*, pp 27–44, Kaiserslautern, Germany, June 1999. Fraunhofer IESE, Kaiserslautern, Germany.

Althoff, K.-D. & Pfahl, D. (2002). Integrating Experience-Based Knowledge Management with Sustained Competence Development. In: Aurum, A., Jeffery, R., Wohlin, C. & Handzic, M. (Eds.), *Managing Software Engineering Knowledge*. Springer (forthcoming).

Althoff, K.-D. & Wilke, W. (1997). Potential uses of case-based reasoning in the experience-based construction of software systems. In: Bergmann, R. & Wilke, W. (Eds.), *Proceedings of the 5th German Workshop in Case-Based Reasoning (GWCBR'97)*, LSA-97-01E, Center for Learning Systems and Applications (LSA), University of Kaiserslautern, Germany.

Arango, G. (1994). Domain analysis methods. In: Schäfer, W., Prieto-Díaz, R. & Matsumoto, M. (Eds.), *Software Reusability*, pp 17–49, Ellis Horwood Ltd., New York, USA.

Bailin, S., Simos, M., Levine, L. & Creps, D. (1996). *Learning and inquiry-based reuse adoption (LIBRA): A field guide to reuse adoption through organizational learning, version 1.1*. STARS Asset A1098, Lockheed Martin Tactical Defense Systems, February 1996. URL: http://direct.asset.com/.

Bartlmae, K. (1999). An Experience Factory Approach for Data Mining. In: *Wrobel et al. (1999)*, pp 5–14.

Bartsch-Spörl, B. (1999). Keys of success for CBR applications in practice. In: *Proceedings of the Industry Day at ICCBR '99*, Monastery Seeon, Germany, July 1999.

Basili, V.R. (1985). Quantitative evaluation of software methodology. In: *Proceedings of the First Pan-Pacific Computer Conference, Melbourne, Australia.*

Basili, V.R., Caldiera, G. & Cantone, G. (1992). A reference architecture for the component factory. *ACM Transactions on Software Engineering and Methodology*, 1(1).

Basili, V.R., Caldiera, G. & Rombach, H.D. (1994). The experience factory. In: Marciniak, J. J. (Ed.), *Encyclopedia of Software Engineering*, Vol. 1, pp 469–476. Wiley, New York, USA.

Basili, V.R. & Rombach, H.D. (1988). The TAME Project: Towards improvement-oriented software environments. *IEEE Transactions on Software Engineering* SE-14(6):758–773.

Basili, V.R. & Rombach, H.D. (1991). Support for comprehensive reuse. *IEEE Software Engineering Journal* 6(5):303–316.

Benjamins, V.R., Fensel, D. & Gómez-Pérez, A. (1998). Knowledge management through ontologies. In: *Proceedings of the Second International Conference on Practical Aspects of Knowledge Management (PAKM '98*), Basel, Switzerland.

Bergmann, R. (2001). Highlights of the European INRECA projects. In: Aha, D. & Watson, I. (Eds.), *Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR-2001)*, Lecture Notes in Artificial Intelligence. Springer.

Bergmann, R. (2002). *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*, Lecture Notes in Artificial Intelligence, Vol. 2432. Springer.

Bergmann, R. & Göker, M. (1999). Developing industrial case-based reasoning applications using the INRECA methodology. In: Anand, S., Aamodt, A. & Aha, D. (Eds.), IJCAI-99 Workshop ML-5 on the Automatic Construction of Case-Based Reasoners.

Bergmann, R. & Althoff, K.D. (1998). Methodology for building CBR applications. In: Lenz et al.

Bergmann, R., Breen, S., Fayol, E., Göker, M., Manago, M., Schumacher, J., Schmitt, S., Stahl, A., Wess, S. & Wilke, W. (1998). Collecting experience on the systematic development of CBR applications using the INRECA-II Methodology. In: Smyth, B. & Cunningham, P. (Eds.), *Advances in Case-Based Reasoning (EWCBR'98)*, Lecture Notes in Artificial Intelligence, pp 460–470.Springer, Berlin, Heidelberg.

Bergmann, R. & Stahl, A. (1998). Similarity measures for object-oriented case representations. In:Smyth, B. & Cunningham, P. (Eds.), *Advances in Case-Based Reasoning (EWCBR'98)*, Lecture Notes in Artificial Intelligence, pp 25–36. Springer, Berlin, Heidelberg.

Bergmann, R., Wilke, W., Althoff, K.-D., Breen, S. & Johnston, R. (1997). Ingredients for developing a case-based reasoning methodology. In: Bergmann, R. & Wilke, W. (Eds.), *Proceedings of the 5th German Workshop in Case-Based Reasoning (GWCBR'97),* LSA-97-01E, Centre for Learning Systems and Applications (LSA), pp 49–58, University of Kaiserslautern, Germany.

Bergmann, R., Wilke, W. & Schumacher, J. (1997a). Using software process modeling for building a case-based reasoning methodology. In: Leake, B. & Plaza, E. (Eds.), *Case-Based Reasoning Research and Development (ICCBR-97)*, Lecture Notes in Artificial Intelligence, pp 553–564. Springer, Berlin, Heidelberg.

Birk, A. & Tautz, C. (1998). Knowledge Management of Software Engineering Lessons Learned. *Proceedings of the Tenth Conference on Software Engineering and Knowledge Engineering*, San Francisco Bay, USA. Skokie, Illinois, USA. Knowledge Systems Institute.

Blázquez, M., Fernández, M., García-Pinar, J.M. & Gómez-Pérez, A. (1998). Building ontologies at the knowledge level using the ontology design environment. In: *Proceedings of the Eleventh Workshop on Knowledge Acquisition, Modelling, and Management (KAW '98), 1998*. URL: http://ksi.cpsc.ucalgary.ca:80/KAW/KAW98/blazquez/.

Booch, G. (1994). *Object-oriented analysis and design with applications*. Second Edition. Benjamin /Cummings Publishing Company.

Chandrasekaran, B., Josephson, J.R. & Benjamins, V.R. (1999). What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1):20–26.

Collier, B., DeMarco, T. & Fearey, P. (1996). A Defined Process for Project Postmortem Review. *IEEE Software* 13(4):65–72.

Davenport, T.H. & Prusak, L. (1998). *Working Knowledge. How Organizations Manage What They Know*. Harvard Business School Press, Boston, USA.

Dellen, B., Maurer, F., Münch, J. & Verlage, M. (1997). Enriching software process support by knowledge-based techniques. In: *Int. Journal of Software Engineering and Knowledge Engineering*, 7(2):185–215.

Göker, M. & Roth-Berghofer, T. (1999). Workshop on "The Integration of Case-Based Reasoning in Business Processes." In: *Proceedings of the ICCBR '99 Workshops*. Technical Report, LSA-99-03E, Department of Computer Science, Centre for Learning Systems and Applications, University of Kaiserslautern, Germany.

Gómez-Pérez, A. (1998). Knowledge sharing and reuse. In: Liebowitz, J. (Ed.), *The Handbook of Applied Expert Systems*. CRC Press.

Gómez-Pérez, A. & Benjamins, V.R. (1999). Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods. In: Benjamins, V. R., Chandrasekaran, B., Gomez-Perez, A., Guarino, N. & Uschold, M. (Eds.), *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5)*, Stockholm, Sweden, August 1999. URL: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/.

Gresse von Wangenheim, C., Althoff, K.-D. & Barcia, R.M. (1999). Intelligent retrieval of software engineering experienceware. In: *Proceedings of the Eleventh Conference on Software Engineering and Knowledge Engineering*, pp128–135, Kaiserslautern, Germany. Knowledge Systems Institute, Skokie, Illinois, USA.

Gresse von Wangenheim, C. & Tautz, C. (1999). Workshop on "Practical Case-Based Reasoning Strategies for Building and Maintaining Corporate Memories." In: *Proceedings of the ICCBR '99 Workshops*, Technical Report, LSA-99-03E, Department of Computer Science, Centre for Learning Systems and Applications, University of Kaiserslautern, Germany.

Gruber, T.R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal on Human-Computer Studies,* 43:907–928.

Guarino, N. (1997). Understanding, building, and using ontologies. *International Journal on Human-Computer Studies*, 46(2/3):293–310.

Haley, T.J. (1996). Software process improvement at Raytheon. *IEEE Software* 13(6):33–41.

Heister, F. & Wilke, W. (1998). An architecture for maintaining case-based reasoning systems. In: Smyth, B. & Cunningham, P. (Eds.), *Advances in Case-Based Reasoning (EWCBR'98)*, Lecture Notes in Artificial Intelligence pp 221–232. Springer, Berlin, Heidelberg.

Henninger, S. (1995). Developing domain knowledge through the reuse of project experiences. In: Samadzadeh, M. (Ed.), *Proc. of the Symposium of Software Reusability (SSR'95)*, pp 186–195.

Houdek, F., Schneider, K. & Wieser, E. (1998). Establishing experience factories at Daimler-Benz: An experience report. *Proc. 20th Internat. Conf. on Software Engineering (ICSE'98).*

Humphrey, W.S., Snyder, T.R. & Willis, R.R. (1991). Software process improvement at Hughes Aircraft. *IEEE Software 8*, pp 11–23.

Jedlitschka, A. & Althoff, K.-D. (2003). Using Case-Based Reasoning for User Modeling in an Experience Management System. Forthcomming.

Kitano, H. & Shimazu, H. (1996). The experience-sharing architecture: A case study in corporate-wide case-based software quality control. In: Leake, D. (Ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, pp 235–268. AAAI Press/The MIT Press, Menlo Park, California, USA, 1996.

Leake, D. (Ed.) (1996). *Case-Based Reasoning: Experiences, Lessons, & Future Directions.* AAAI-Press.

Leake, D. & Wilson, D. (1998). Categorizing case-base maintenance: Dimensions and directions. In: Smyth, B. & Cunningham, P. (Eds.), *Advances in Case-Based Reasoning (EWCBR'98)*, Lecture Notes in Artificial Intelligence, pp 196–207. Springer, Berlin, Heidelberg.

Lenz, M., Bartsch-Spörl, B., Burkhardt, H.D. & Wess, S. (Eds.) (1998). *Case-Based Reasoning Technology, From Foundations to Applications*, Lecture Notes in Artificial Intelligence, Vol. 1400. Springer, Berlin, Heidelberg.

Lenz, M., Hübner, A., & Kunze, M. (1998b). Textual CBR. In: Lenz et al.

Lenz, M. & Ashley, K. (1998). *Proceedings of the AAAI98 Workshop on Textual Case-Based Reasoning*. AAAI Press.

López, M.F. (1999). Overview of the methodologies for building ontologies. In: Benjamins, V.R., Chandrasekaran, B., Gómez-Pérez, A., Guarino, N. & Uschold, M. (Eds.), *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5*), Stockholm, Sweden.

Manago, M., Bergmann, R., Wess, S. & Traphöner, R. (1994). *CASUEL: A Common Case Representation Language —Version 2.0* . ESPRIT-Project INRECA, Deliverable D1.

McGarry, F. & Pajersky, R. (1990). Towards Understanding Software — 15 Years in the SEL. *Proc. of the 15th Annual Software Engineering Workshop*, NASA Goddard Space Flight Center, Greenbelt, Maryland, USA, Software Engineering Laboratory Series, SEL-90–006.

Neal, M. (1997). *Parametric search: Evolving information retrieval for the Web*. CADIS Inc. IDM 2(5).

Nick, M., Althoff, K.-D. & Tautz, C. (2001). Systematic Maintenance for Corporate Experience Repositories. *Computational Intelligence*, 17(2):364–386.

Racine, K. & Yang, Q. (1997). Maintaining unstructured case-bases. In: Leake, B. & Plaza, E. (Eds.), *Case-Based Reasoning Research and Development* (ICCBR-97), Lecture Notes in Artificial Intelligence, pp 553–564. Springer, Berlin, Heidelberg,.

Rech, J., Decker, B. & Althoff, K.-D. (2001). Using Knowledge Discovery Technology in Experience Management Systems. *Proc. 13th Workshop of the German Special Interest Group on Machine Learning, Knowledge Discovery, and Data Mining, Workshop Days of the German Society of Computer Science on 'Learning, Teaching, Knowledge, and Adaptivity ,"* University of Dortmund, Germany.

Rech, J., Althoff, K.-D., Decker, B., Klotz, A., Leopold, E. & Voss, A. (2002). Thoughts on Text Mining in Organizational Process Learning. *Proc. 14th Workshop of the German Special Interest Group on Machine Learning, Knowledge Discovery, and Data Mining, Workshop Days of the German Society of Computer Science on 'Learning, Teaching, and Adaptivity,"* Learning Lab Lower Saxony (L3S), Hannover.

Reinartz, T., Iglezakis, I. & Roth-Berghofer, T. (2001). On quality measures for case-base maintenance. *Computational Intelligence* 17(2).

Richter, M. (1998). Introduction. Chapter 1 in Lenz, M., Bartsch-Spörl, B., Burkhardt, H. D. & Wess, S. (Eds.), *Case-Based Reasoning Technology, From Foundations to Applications*, Lecture Notes in Artificial Intelligence, Vol. 1400, pp 1–15. Springer, Berlin, Heidelberg.

Richter, M. (2000). *Personal communication*.

Rombach, H.D. (1996). New institute for applied software engineering research. *Software Process Newsletter No 7*, pp 12–14.

Rombach, H.D. & Ulery, B.D. (1989). Establishing a measurement-based maintenance improvement program: Lessons learned in the SEL. *Proc. of the Conference on Software Maintenance,* pp 50–57. IEEE Computer Society Press.

Rombach, H.D. & Verlage, M. (1995). Directions in software process research. In: Zelkowitz, M. V. (Ed.), *Advances in Computers*, Vol. 41, pp 1–61. Academic Press.

Schreiber, G., Wielinga, B., de Hoog, R., Akkermans, H. & Van de Velde, W. (1994). CommonKADS: A comprehensive methodology for KBS development. *IEEE Expert*, 6(6):28–37.

Seshagiri, G. (1996). Continuous process improvement: Why wait till level 5? *Proc. 29th Hawaii Internat. Conf. on System Sciences*, pp 681–692. IEEE Computer Society Press.

Shimazu, H. (1998). A textual case-based reasoning system using XML on the world-wide web. In: Smyth, B. & Cunningham, P. (Eds.), *Advances in Case-Based Reasoning (EWCBR'98)*, Lecture Notes in Artificial Intelligence. Springer, Berlin, Heidelberg.

Simos, M., Creps, D., Klingler, C., Levine, L. & Allemang, D. (1996). *Organization Domain Modeling (ODM) guidebook, version 2.0*. Informal Technical Report STARS-VC-A025/001/00, Lockheed Martin Tactical Defense Systems, Manassas, Virginia, USA.

Smyth, B. & McKenna, E. (1998). Modeling the competence of case bases. In: Smyth, B. & Cunningham, P. (Eds.), *Advances in Case-Based Reasoning (EWCBR'98)*, Lecture Notes in Artificial Intelligence, pp 208–220. Springer, Berlin, Heidelberg.

Stolpmann, M. & Wess, S. (1999). *Optimierung der Kundenbeziehungen mit CBR systemen-Intelligente Systeme für E-Commerce und Support* . Addison Wesley Longmann (Business & Computing), Bonn, Germany.

Studer, R. (1999). Knowledge engineering: Survey and future directions. In: *Proceedings of the Workshop on Knowledge Management, Organizational Memory and Knowledge Reuse during Expert Systems '99 (XPS-99*), pp 1–23, Würzburg, Germany.

Surma, J. & Tyburcy, J. (1998). A Study on competence-preserving case replacing strategies in case-based reasoning. In: Smyth, B. & Cunningham, P. (Eds.), *Advances in Case-Based Reasoning (EWCBR'98)*, Lecture Notes in Artificial Intelligence, pp 233–238. Springer, Berlin, Heidelberg.

Tautz, C. (2000). *Customizing Software Engineering Experience Management Systems to Organizational Needs*. Ph. D. thesis, Department of Computer Science, University of Kaiserslautern, Germany. Fraunhofer IRB Verlag, Stuttgart.

Tautz, C. & Althoff, K.-D. (1997). Using Case-Based Reasoning for Reusing Software Knowledge. In: Leake, D. & Plaza, E. (Eds.), *Case-Based Reasoning Research and Development, Second International Conference on Case-Based Reasoning (ICCBR'97)*, pp 156–165, Springer.

Tautz, C. & Althoff, K.-D. (2000). A Case Study on Engineering Ontologies and Related Processes for Sharing Software Engineering Experience. In: *Proc. 12th International Conference on Software Engineering and Knowledge Engineering (SEKE'00).*

Tautz, C., Althoff, K.-D. & Nick, M. (2000). A Case-Based Reasoning Approach for Managing Qualitative Experience. In: *Aha & Weber (2000).*

Thomas, H., Foil, R. & Dacus, J. (1997). New technology bliss and pain in a large customer service center. In: Leake, B. & Plaza, E. (Eds.), *Case-Based Reasoning Research and Development (ICCBR'97)*, Lecture Notes in Artificial Intelligence, pp 166–177. Springer, Berlin, Heidelberg.

Uschold, M. & Gruninger, M. (1996). Ontologies: Principles, methods, and applications. *The Knowledge Engineering Review*, 11(2):93–136.

Vollrath, I., Wilke, W. & Bergmann, R. (1998). Case-based reasoning support for online catalog sales. *IEEE Internet Computing*, **2** (4):47–54.

Wartik, S. & Davis, T. (1999). A phased reuse adoption model. *Journal of Systems and Software* 46(1):13–23.

Weibelzahl, S. & Weber, G. (1999). Benutzermodellierung von Kundenwünschen durch Fallbasiertes Schließen. In: *Wrobel et al. (1999)*, pp 295–300.

Wiig, K.M. (1995). *Knowledge Management Methods: Practical Approaches to Managing Knowledge*. Schema Press, Arlington, Texas, USA.

Wilson, D. & Leake, D. (2001). Maintaining case-based reasoners: Dimensions and Directions. *Computational Intelligence* 17(2).

Wrobel, S., Henze, N. & Perner, P. (Eds.) (1999). *Proc. 11th Workshop of the German Special Interest Group on Machine Learning, Knowledge Discovery, and Data Mining, Workshop Days of the German Society of Computer Science on 'Learning, Knowledge Discovery, and Adaptivity,"* University of Magdeburg.

Index