Collaborative Multi-Expert-Systems
Realizing Knowledge-Lines with Case Factories and Distributed Learning Systems

Klaus-Dieter Althoff, Kerstin Bach, Jan-Oliver Deutsch, Alexandre Hanft, Jens Mänz, Thomas Müller, Régis Newo, Meike Reichle, Martin Schaaf & Karl-Heinz Weis

University of Hildesheim
Institute of Computer Science
Intelligent Information Systems Lab
Marienburger Platz 22, 31141 Hildesheim, Germany
second-name@iis.uni-hildesheim.de

Abstract. We describe a new research effort for developing knowledge-based systems using a combination of methods from Software Engineering and Artificial Intelligence: software product-lines, experience factory, case-based reasoning, multi-agent-systems, and semantic web technology. We motivate our approach, shortly describe three different application scenarios, and provide our current ideas of how to implement our approach, which we call “collaborative multi-expert-systems” (CoMES).

1 Introduction

The notion of “Knowledge-Intensive Services” and especially “Knowledge Work” [1, 2] becomes a more and more important part of the service sector. Thereby, “Knowledge-intensive work” refers to activities that require intensive education and experience on a specific subject that has been accumulated over many years [3, 4]. “Knowledge-intensive services” need the resource knowledge as their most important input factor for delivering the respective service [6]. Problem solving strategies as part of “Knowledge work” activities must not be limited to use static knowledge acquired once. Instead, they require frequent revision, improvement, and update of the knowledge asset [3, 4]. Hence, experience represents the success-critical knowledge for knowledge-intensive services and knowledge work [7].

Within this paper we suggest a methodology of how to develop intelligent information systems for supporting knowledge work and knowledge-intensive services with a specific focus on the use of experience [8, 9]. Our approach especially includes computer-based knowledge work that is partially or fully enabled by software agents. Besides its potential application to service economics (for a lot of success stories see [10, 11]), our research also contributes to achieve ambitious goals as being formulated by the European Union (e.g., the so-called ambient intelligence scenarios [12], also [13, 40] or scenarios described in reports on “converging technologies” or similar [14], also [15]). Fully or partially automating knowledge work has the additional advantage that the knowledge asset is not only accessible by the user but – with an increasing degree of formality – also by the computer. This enables automated processing of knowledge and offers a unique added-value if compared with more traditional approaches.

Nowadays, software systems offering intelligent services require lots of knowledge and have to fulfill many requirements making development and maintaining extremely expensive. In addition, usually many such systems are developed to meet the various requirements. Our approach here is to organize such knowledge as a knowledge-line. By "knowledge-line" we denote the systematic application of the software product-line approach [16] to the knowledge of knowledge-based systems. This enables the necessary "knowledge level modularization" for building potential variants in the sense of software product-lines. This can be achieved by making use of multi-agent systems [17, 18] as a basic approach for knowledge-based systems. An intelligent agent – as a first approximation – is implemented as a case-based reasoning (CBR) system [19], which, besides case-specific knowledge, can also include other kinds of knowledge. Each CBR system agent is embedded in a case factory that is responsible for all necessary knowledge processes like knowledge inflow, knowledge outflow as well as knowledge analysis.

A "case factory" (CF) [20] is an organizational unit that emulates the well-known experience factory approach [21] from software engineering (SE). For each role within an experience factory, software agents are introduced for modelling (carrying out) those tasks that can be automated more and more. Like the CBR system agent(s), the associated respective CF agents are expected to learn from experience. For example, they could implement machine learning techniques for analyzing, evaluating, and maintaining the case base of the CBR system agent[1]. In principle each software agent has a human coach, namely the one being responsible for the role (in the sense of the experience factory approach), jointly taking over the respective assigned tasks). The human coach provides case-specific knowledge for the case base of her CF agent(s) as well as feedback to suggested decisions. Her motivation for providing knowledge is to make her CF agents as competent as possible to transfer more and more routine tasks to these CF agents. Of course, overall responsibility and control remain with the human decision maker.

Using the more general knowledge-line approach, a knowledge-based system can be developed using different knowledge modules. The latter would be associated with the respective variabilities and requirements they fulfil. One module could be realized by one or more software agents. Developing knowledge-based systems would be more flexible because of the various possibilities for combining these different software agent groups and the learning abilities of the respective software agents.

We already mentioned (more or less directly) several approaches from SE (experience factory, software product-lines) and artificial intelligence (AI) (CBR, machine learning, knowledge engineering, distributed AI, multi-agent systems) that support our approach. Enabling software agents to make use of learning techniques effec-

[1] CF agents could also be realized as hybrid systems including some CBR/learning subcomponent.

tively requires dealing with knowledge in different degrees of formalization for acquisition, transformation, and processing. For example, reasoning would require the integration of empirical and causal rules, constraints, and analogies. Combining, integrating, reusing the respective knowledge modules (agents) could be supported by knowledge-based configuration approaches [22, 23]. Intelligent user interfaces [24] as well as adaptivity and user modelling [25] could improve the communication between software agent and human coach. For deepening the understanding of learning, research on cognitive architectures is also of interest [26, 27]. Considering the various interfaces between human and software agents many commonalities and analogies with the area of Web2.0/social software [28] can be identified. For example, both approaches have a solution that consists of a combination of some kind of collaborative work involving experts and other users who make their contributions from different locations – potentially distributed over the whole world – at different points of time.

We call our approach "Collaborative Multi-Expert-Systems" (CoMES), which denotes a new research approach that is both a continuation of the well-known expert system approach[2] and a research direction basing on the ideas of case factory and knowledge-line as mentioned above. Thus, not everything is new with CoMES; however, we believe that treating these issues altogether as one research issue is definitely new. To make this clear, CoMES are still more a vision, a set of ideas, and some just started project activities. However, we also believe that not only one group can achieve research progress on such a research topic all alone.

One basic feature of CoMES is that they base on rather simple techniques looking more intensively on the possibilities of how to combine such techniques. While many early (and also modern) expert systems had the problem of acquiring and maintaining its knowledge, the underlying idea in CoMES is to develop "CoMES there where knowledge is produced": that is, into the knowledge communities already available in the Web. CoMES not necessarily try to integrate knowledge from different sources/ experts but base on many experts. These experts are viewed like agents and can be human and/or software agents. As a consequence, CoMES do not have the immediate goal to formally represent the whole knowledge necessary to solve problems associated with a given task. Instead they base on the idea of learning step by step based on made experiences, or case-specific knowledge made available by cooperative authors. Another idea is to keep the resulting learning scenarios/tasks as simple as possible. Thus, having more agents and having each one learning in a rather simple way. Here we are thinking of computer science techniques focusing on experience like case-based reasoning, experience management, case factory, machine and human learning, cognitive architectures, etc.

Summarizing the above arguments we currently associate with CoMES (among other issues) the following:

- Decomposing of an application into different learning scenarios for which case-specific knowledge can be learnt ("experience be made").
  - Research issues here include the identification of different types of experience and having different case bases that may be related.

[2] We do not differentiate between the terms "expert system" and "knowledge-based system" within this paper.

– Processes for filling, evaluating, and maintaining case bases
  - These processes are necessary for each case base (learning scenario)
– Defining and analyzing a "knowledge culture"
  - This addresses what can be learned from a practical point of view.
  - This includes the learning of learning strategies by instantiating them through (discovered) examples.
– Knowledge advancement
  - Could be viewed as using multiple any-time algorithms in parallel.
  - Using generic machine learning approaches of which the associated learning strategy can be instantiated via examples.
  - Introducing a goal-orientation by use of a learning strategy or the combination of different learning strategies.

In the following chapter we will describe some application scenarios we currently work with to give the reader some more concrete ideas what we expect to achieve. In chapter three we give a rough overview how we think to implement CoMES. Finally we finish with an outlook on future research activities.

2 Application scenarios

In the following section we describe several application scenarios we are currently working on to instantiate our ideas of knowledge-lines and case factories. Of course, these specific applications have their own context and requirements and reflect only some aspects of the above mentioned concepts. In particular we explain our work on an information system on free and open source software, a simulation model for effec-tive coping strategies within the simulation of cognitive processes, and on an informa-tion system on travel medicine. All three intelligent information systems have in common, that they utilize different kinds of knowledge sources, integrate distributed experts, and will be realized following a modular structure using software agents.

2.1 Information system on free and open source software

Free/Libre and open source software (FLOSS) has produced a large and diverse range of software that often offers numerous and high quality alternatives to almost all commercial software applications. Popular FLOSS like Firefox[3], Thunderbird[4], OpenOffice.org[5], vlc[6], or the GNU/Linux operating system are steadily gaining users both in the private and commercial sector. However, the FLOSS community is a complex social and technical network that consists of tens of thousands of individual groups and projects that produce software in all degrees of quality. Research [29, 30] shows ex-

[3] http://www.mozilla-europe.org/en/products/firefox/
[4] http://www.mozilla-europe.org/en/products/thunderbird/
[5] http://www.openoffice.org/
[6] http://www.videolan.org/vlc

isting FLOSS directories are mostly used by expert users and FLOSS insiders, while less experienced users prefer general search engines and the advice of friends when choosing software. The success of choosing software in this way is limited though. A general web search for software for a specific purpose will most likely yield the most popular software (or rather that with the most popular website) but not necessarily the one that is best suited for the given purpose. Asking friends or colleagues for software advice is also of only limited use, since those usually have the same level of knowledge as the seeking person and can thus not offer qualified advice.

The aim of this project is to use the broad supply of already existing information about FLOSS that is offered by conventional FLOSS directories and Linux distributions such as the Debian Project and use them for a more intelligent retrieval of FLOSS. In order to achieve this we combine textual descriptions of software with qualitative information like the software's popularity and vitality at different FLOSS directories, meta information as provided by the Debian distribution and collaboratively maintained tags covering different aspects of software. The added value of the combination of these different information sources and the advantages of CBR in combination with the case factory approach and an industrial strength tool such as *e:IAS* (cf. Section 3) promise considerably better retrieval results than the simple full text search used by most conventional FLOSS directories or the website oriented general web search. [29] has already developed a model to represent FLOSS as a case in a CBR case base and the planned system has been successfully presented at different conferences.

2.2 Simulating cognitive processes: effective coping strategies[7]

Cognitive processes have been studied in the field of psychology. There already exist many simulations used in order to support the developed theories in that area. The main purpose of those simulations is to evaluate the behaviour of people when they face different situations. Our simulation is also based on psychological research, namely a theory developed by Werner Greve. Our goal is to analyse how human beings act in critical situations by emphasizing which coping strategies they use.

An example would be to know how someone reacts after her/his boyfriend/girlfriend dropped her/him. The idea here consists in representing each human being as a hierarchically organized multi-agent system called "holon" (i.e., an agent that can contain further (sub)agents). Though the social abilities (communication) of the human – simulated via an agent – plays a role, the design of the holons (respectively its characteris-tics) as well as their interaction are even more important.

Fig. 1 shows an internal representation of a simulated person including the characteristics and the possible strategies that can be applied in critical situations. More details can be found in [27]. We plan to implement each holon as a knowledge-based system. Since we think that experience is an important factor (e.g. for the choice of the adequate coping strategy), we will use case-based resoning as the main technique for the knowledge-based systems.

[7] This project is a cooperation with Werner Greve (Institute of Psychology, University of Hildesheim).

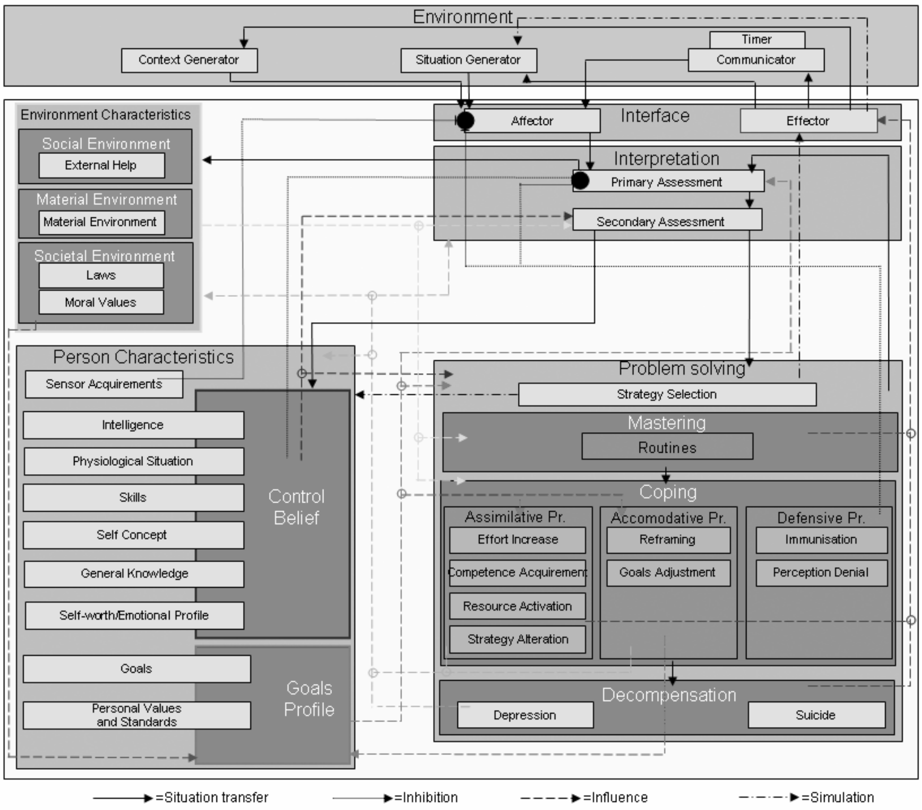

Fig. 1. SIMOCOSTS model

2.3 Information system on travel medicine[8]

It has never been easier than today traveling to different places, experience new cultures and get to know new people. In preparation for a healthy journey it is important to acquire high quality of reliable information about travel medicine prevention. Currently the World Wide Web offers many websites, discussion forums and services where a traveler can gather information. Usually those websites do not contain all medical information a traveler might need and the editors are mostly unknown. Furthermore the information is spread over hundreds of websites and it is challenging and time-consuming to find the appropriate one(s). Together with certified doctors of medicine with a strong background of tourism related medicine we will use the CoMES approach to provide travel information based on key data of their journey (travel period, destination, age(s) of traveler(s), activities, etc.).

[8] This project is a cooperation with Thomas Schmidt (mediScon).

The application will deal with a variety of knowledge sources (diseases, medications, outbreaks, guidelines, etc.) which are provided in different knowledge bases (cf. Fig. 2). As described in [31] for each specific issue a knowledge base will be created to ensure a high quality of information. To access and process data for each knowledge base a domain model depending on the structure of information has to be defined [32; 33]. Information for travelers should contain pre-travel advices, guidelines while abroad and past-travel advices. Medical advices are complex and we always have to consider contraindication and interdependences. In addition to the medical description, we will provide different kinds of explanation ensuring travelers understand the given advices.

Travelers who are using the CoMES will enter their key data and depending on the input the application will proceed. The user will be asked for more information to capture the entire circumstances. Based on the information given by the traveler software agents will collaborate and determine answers from the knowledge base answering the request. Therefore we will use software agents following all kinds of tasks like creating an appropriate question structure avoiding redundant questions or agents extracting information given in the knowledge bases and creating responses. We aim extending the set of agents step by step furthering the application logic and covering more and more issues. The approach will be supervised by experts, although we will integrate the experiences of travelers using the given advices to improve the quality and identify new issues early and providing information about it.

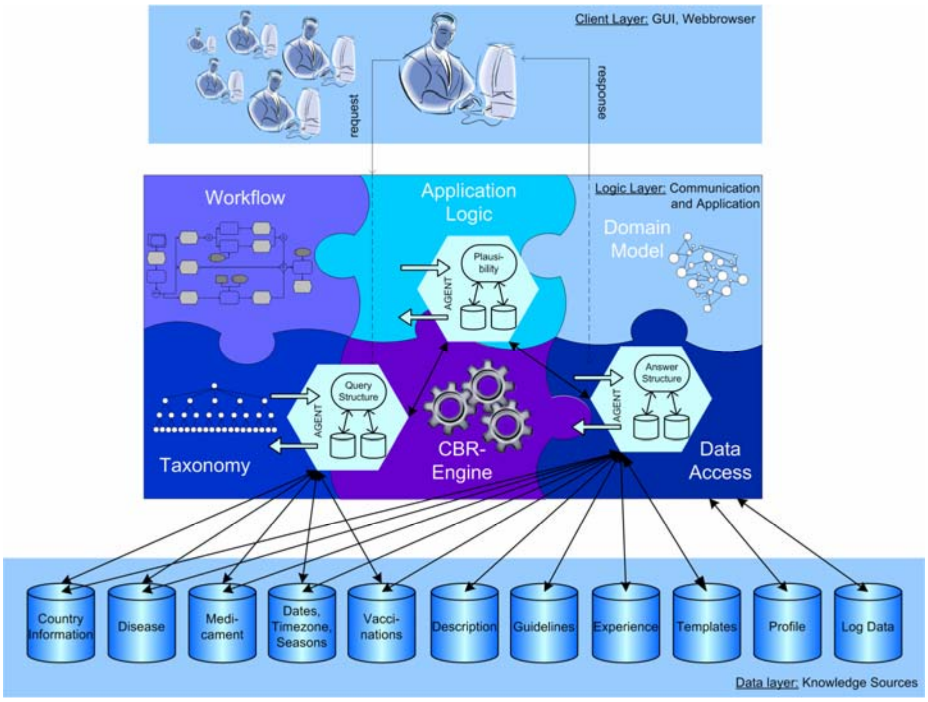

Fig. 2. docQuery architecture

Figure 2 shows the architecture of docQuery which is based on three layers. The Data Layer (bottom) contains all the knowledge sources docQuery will need to provide information. It contains a knowledge source for each topic what makes it easy to follow a modularized structure and enables defining maintenance processes for the contained information. According to the well-known three tier architecture the middle tier will process questions answered of users passed through the client layer. The Logic Layer will use CBR techniques to retrieve cases, has rules and templates to creates answers and is possesses information, derived from workflows, how to solve conflicts creating answers and forward unanswerable questions to experts. The client layer contains forms which will be used of travels to ask questions and interact with docQuery.

3 Implementation Details

The technical implementation of CoMES is being done in cooperation with *empolis*[9]. We base on the *empolis information access* suite (*e:IAS*) [34; see also Section 3.1] for realizing CBR systems as well as knowledge-based systems that include some CBR component. For realizing a CoMES we (plan to) use many *e:IAS* instances in parallel as necessary. Depending on the size of the respective knowledge base there could run several *e:IAS* instances on separate virtual machines on one computer. The communication of the respective instances would be in a similar way when integrating *e:IAS* within certain business processes of some organization, namely via web services. The organization of the respective *e:IAS* instances ("agents") would be in some sense similar to the electronic air freight market scenario described by Bodendorf [35]: several tasks would be explicitly coordinated by a responsible *e:IAS* instance. However, since every instance has in principle the same capabilities also direct communication between *e:IAS* instances would be possible.

Thus, our approach can be viewed conceptually somehow between the multi-agent system approach in the narrow sense on one side and the distributed problem solving approach e.g. [36, 37] on the other side.

Our approach has several advantages. We can use known methods for building one *e:IAS* application e.g. [11] with (hopefully) slight modifications. In principle each instance can grow to a very large knowledge-based system. Also the integration into certain business environments can be done as usual (with *e:IAS*).

The collaboration aspect of CoMES is realized with the normal user interaction processes of *e:IAS*. For example, the text mining and textual CBR capabilities of *e:IAS* allow even textual input to be reasonably processed by an *e:IAS* instance, assumed this instance has enough knowledge to react.

A short description of *e:IAS* is given in the next section.

[9] http://www.empolis.com

3.1 *Empolis Information Access Suite* (*e:IAS*)

e:IAS is an information and knowledge management suite developed by *empolis*, a subsidiary of *Bertelsmann arvato*. *e:IAS* consists of several different components for information and knowledge processing and management. Among these is a rules engine, that can be used to model and apply rules and conditions, for example, for business rules or classification and a text miner that can be used for analyzing documents as well as free text user queries and can further be combined with a downstream pattern matching component. Apart from these, *e:IAS* also includes a powerful CBR engine and the so called creator module, which is based on the free IDE eclipse[10].

The creator is used to model the cases for the CBR engine. A case is modeled as an aggregate of attributes. The creator is used to model the required attributes, their domains and also underlying concepts and taxonomies and the respective attributes' similarity functions. The model is stored using RDF respectively OWL, all further information is stored in an XML format.

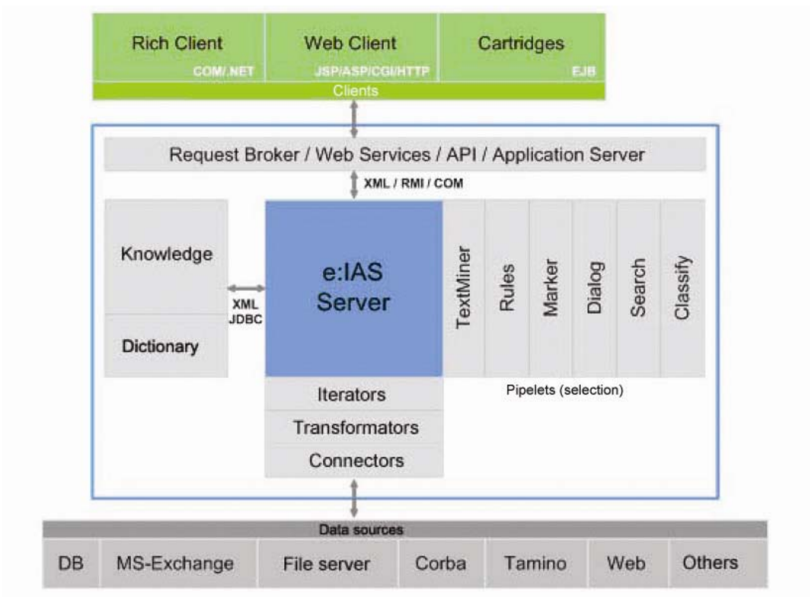

Fig. 4. *empolis Information Access Suite* architecture

Figure 4 shows the general setup of an *e:IAS* knowledge management system. The case models can be filled with imported data from a multitude of sources. The data import and their further processing is done using a modular pipeline system in which the different functionalities can be freely combined using individual pipelets. These pipelets offer, for example, the import from simple text files, documents, data bases, or websites using an integrated crawler, and their subsequent processing such as breaking the input data into single values and assigning them to their respective at-

[10] http://www.eclipse.org

tributes, analyzing texts with a text miner or spell checker or stripping input of html or xml elements. Once the data are imported and processed the system's knowledge base is ready and can be used by the *e:IAS* knowledge server.

Pipelines are, however, not only used for importing data, but also for integrating the aforementioned functionalities and making them available to the Knowledge Server. *e:IAS* offers pipelets for text mining, the creation and application of rule sets, searching, automated classification and the generation of dialogs. Additionally the Knowledge Server is able to use external knowledge sources such as already existing dictionaries. Different types of clients including simple web clients but also rich clients or Java Bean applications can be used to access the Knowledge Server. Communication between the server and its clients can be implemented using various languages such as XML, COM, or RMI.

4 Outlook

We currently work on the integration of the various requirements from the different application scenarios – as well as further ones – to come up with a kernel architecture and implementation of collaborative multi-expert-systems (CoMES). While we base on a cooperation with *empolis* and base our realization efforts on the *empolis Information Access Suite* (*e:IAS*), we, in addition, work on the integration of intelligent services in the Semantic Web [38, 39], with the goal of also integrating these research efforts.

5 Acknowledgement

We would like to thank all our cooperation partners for achieving progress within the respective application scenarios: Werner Greve for providing the idea and the psychological knowledge for the SIMOCOSTS project; Thomas Schmidt for providing idea, knowledge, and platform for the docQuery project; the Debian community members providing us with feedback for the FLOSSWALD project. We thank *empolis GmbH* for providing us with their *empolis Information Access Suite* (*e:IAS*) and Ralph Traphöner for exchanging ideas on using *e:IAS* for implementing distributed learning systems. We thank Ralph Bergmann for various discussions on “collaborative multi-expert systems” and all the related issues.

6 References

1. Bullinger, H.-J., Ilg, R.: Leben und Arbeiten in einer vernetzten, mobilen Welt. In Uhr, W., Esswein, W., Schoop, E. (eds.): Wirtschaftsinformatik 2003 Band I, Physica Verlag, pp. 1–8, (2003)
2. Bundesministerium für Forschung und Technologie, Bekanntmachung über die Förderung von Forschungsvorhaben auf dem Gebiet „Wissensintensive Dienstleistungen", 14.1.2000 (2000)
3. Willke, H.: Organisierte Wissensarbeit. In: Zeitschrift für Soziologie, Vol 3, 1998, pp. 161–177 (1998)
4. Hermann, S.: Produktive Wissensarbeit – Eine Herausforderung. In [5], pp. 204–224 (2003)
5. Hermann, S. (eds.): Integrierter Schlussbericht – Verbundprojekt SIAM „Strategien, Instrumente und arbeitsorganisatorische Gestaltungsmodelle zur Förderung der Dienstleistungskompetenz in Unternehmen, (2003).
6. Cramer, J.: Management wissensintensiver Dienstleistungen. In [5], pp. 179–203 (2003)
7. Brasse, C., Uhlmann, M.: Integration von Erfahrungswissen. In [5], pp. 121–132 (2003)
8. Decker, B., Althoff, K.-D.: Prozesslernen und Erfahrungsmanagement: Ergebnisse aus dem indiGo-Projekt. In: Proc. Lernen – Wissen – Adaptivität (LWA) 2004, pp. 138–145 (2004)
9. Althoff, K.-D., Mänz, J., Nick, M.: Integrating Case-Based Reasoning and Experience Factory: Case Studies and Implications. In: Proc. 28th German Conference on Artificial Intelligence Workshop on Knowledge Engineering and Software Engineering, Technical Report 8/2005, University of Koblenz-Landau, ISSN 1860-4471, pp. 1–12 (2005)
10. Manago, M.: Knowledge Management in the Service Center. Presentation at empolis User Conference, Berlin, 15.–16. November 2006, (2006)
11. Bergmann, R., Althoff, K.-D., Breen, S., Göker, M., Manago, M., Traphöner, R., Wess, S.: Developing Industrial Case-Based Reasoning Applications. LNAI 1612, Springer Verlag, Berlin, (2003)
12. Ducatel, K., Bogdanowicz, M., Scapolo, F., Lejten, J., Burgelman, J.-C. :Scenarios of Ambient Intelligence in 2010. IST Advisory Group (ISTAG), European Commission Community Research, (2001).
13. Schmid, K. (ed.): Ambient Intelligence. Special Issue of the German Journal on Artificial Intelligence, KI(2)07 (2007)
14. Bibel, W., Andler, D., Da Costa, O., Küppers, G., Pearson, I. D.: Converging Technologies and the Natural, Social and Cultural World. Report of the EU High Level Expert Group on "Forsighting the New Technology Wave" (FoNTWave), 30.6.2004 (2004)
15. Rech, J.,, Althoff, K.-D.: Artificial Intelligence and Software Engineering - Status and Future Trends. Special Issue on Artificial Intelligence and Software Engineering, KI (3) 2004, pp. 5–11 (2004)
16. Van der Linden, F., Schmid, K., Rommes, E. (eds.): Software Product Lines in Action – The Best Industrial Practice in Product Line Engineering. Springer Verlag (2007)
17. Burkhard, H.-D.: Software-Agenten. In Görz, G., Rollinger, C.-R., Schneeberger, J., Handbuch der Künstlichen Intelligenz, 4. Auflage, Oldenbourg Verlag, pp. 943–1020 (2003)
18. Gerhard Weiß (ed): Multiagent Systems A Modern Approach to Distributed Artificial Intelligence. The MIT Press, (1999)
19. Althoff, K.-D.: Case-Based Reasoning. In: Chang, S. K. (ed.): Handbook on Software Engineering and Knowledge Engineering. Vol.1, World Scientific, pp. 549-587 (2001)
20. Althoff, K.-D., Hanft, A., Schaaf, M.:Case Factory – Maintaining Experience to Learn. In: M. Göker, T. Roth-Berghofer (eds.): Proc. 8th European Conference on Case-Based Reasoning (ECCBR'06), Ölüdeniz/Fethiye (Turkey), Springer Verlag (2006)

21. Basili, V.R., Caldiera, G., Rombach, H. D.: Experience Factory. In Marciniak, J.J (ed.): Encyclopedia of SE, vol 1, John Wiley & Sons; pp. 469–476 (1994)
22. Hotz, L., Günter, A.: Using knowledge-based configuration for configuring software. Proceedings of ECAI '02, July 2002 (2002)
23. Sauer, J., Edelkamp, S. (eds.): 20 Jahre Planen und Konfigurieren. Special Issue of the German Journal on Artificial Intelligence, KI(1)07 (2007)
24. Wahlster, W. (ed.): SmartKom: Foundations of Multimodal Dialogue Systems. Springer Verlag (2006)
25. Henze, N. (ed.): Adaptivität und Benutzermodellierung. Special Issue of the German Journal on Artificial Intelligence, KI(3)02 (2002)
26. Bach, J., Bauer, C., Vuine, R.: MicroPsi – Contributions to a Broad Architecture of Cognition. In Proc. 29th Ann. German Conf. on Artificial Intelligence (KI 2006) (2006)
27. Müller, T.. Simulation kognitiver Prozesse mit Multiagentensystemen – Modellierung effektiver Bewältigungsstrategien als konkurrierende Problemlöseagenten. Bachelor Thesis, Institute of Computer Science, University of Hildesheim (2006)
28. C. Schmitz, A. Hotho, R. Jäschke, G. Stumme: Kollaboratives Wissensmanagement. In T. Pellegrini, A. Blumauer, (eds.): Semantic Web – Wege zur vernetzten Wissensgesellschaft, pp. 273–290, Springer Verlag (2006)
29. Reichle, M.: Entwicklung und Implementierung eines Modells zum Retrieval von Free/ Libre Open Source Software unter Verwendung eines Case-Based Reasoning Systems. Master Thesis (Magisterarbeit), University of Hildesheim (2007)
30. Hanft, A., Reichle, M. The FLOSSWALD Information System on Free/Libre Open Source Software. In Gronau, N. (ed.), Proc. 4th Conference on Professional Knowledge Management (WM 2007), 4th German Workshop on Experience Management (GWEM'07) (2007)
31. Bach, K.: docQuery – Reisemedizinisches Informationssystem. Internal project report (2007)
32. Bach, K.: Domänenmodellierung im Textuellen Fallbasierten Schließen. Master Thesis, Institute of Computer Science, University of Hildesheim (2007)
33. Bach, K., Hanft, A.: Domain Modeling in TCBR Systems: How to Understand a New Application Domain. Accepted for Workshop Proc. of 7th International Conference on Case-Based Reasoning (ICCBR) 2007, August 2007, Workshop on Knowledge Discovery and Similarity (2007)
34. empolis GmbH: Technisches White Paper e:Information Access Suite. Technical Rport, empolis GmbH, http://www.empolis.de/executive-forum/Downloads/m_Technisches-WP-IAS_V1.1.1_050906.pdf; last access on 3rd of July 2007 (2005)
35. Bodendorf, F.: Wirtschaftsinformatik im Dienstleistungsbereich. Springer (1999)
36. Bamberger, S.: Verteiltes Problemlösen mit wissensbasierten Diagnosesystemen. DISKI Band 203, infix, Akademische Verlagsgesellschaft Aka Berlin, (1999)
37. Fluck, W.: Anwendungsmöglichkeiten Verteilter Künstlicher Intelligenz in Expertensystemen zur Bestimmung von Laufkäfern. Diploma Thesis, Department of Computer Science, University of Kaiserslautern (1992)
38. Schaaf, M.: Integrating Intelligent Services within the Semantic Web. PhD Thesis, Institute of Computer Science, University of Hildesheim (forthcoming) (2007)
39. Deutsch, J.-O.: Entwicklung einer Infrastruktur zur regelbasierten Verarbeitung von RDF/ OWL Spezifikationen. Master Thesis, Institute of Computer Science, University of Hildesheim (2007)
40. Althoff, K.-D.: Experience Technology for Developing Ambient Intelligent Systems. Invited talk at the 4th Conference on Professional Knowledge Management, 4th German Workshop on Experience Management and the 9th International Workshop on Learning Software Organizations, Potsdam, 28th of March 2007 (2007)