CBR-based Execution and Planning Support for Collaborative Workflows

Andrea Freßmann[1], Kerstin Maximini[1], Rainer Maximini[1], and Thomas Sauer[2]

[1] University of Trier
Department of Business Information Systems II
54286 Trier, Germany
{andrea.fressmann|kerstin.maximini|rainer.maximini}@wi2.uni-trier.de
[2] rjm Business Solutions GmbH
68623 Lampertheim, Germany
t_sauer@rjm.de

Abstract. In many application domains, people are faced with processes that are unpredictable, unrepeatable, and subject to change as they are enacted. In this paper, we present the Collaborative Agent-based Knowledge Engine (CAKE) which offers domain-independent support for such highly flexible processes. This approach is based on the well-known techniques of workflows, agents, and CBR-based search, combining them innovatively: Agent technology is used for integrating various services, e.g. different data sources, whereas workflow technology is used for coordinating collaboration among agents and for supporting context-based information delivery. CBR-based search is used for the retrieval of suitable agents and workflows in concrete situations.

1 Introduction

In many domains, people are faced with processes that are mostly unpredictable, unrepeatable and subject to change as they are enacted. Thus managing projects incorporating such processes is a challenging task. Recently, agile development methodologies have been developed to alleviate such management [1, 2]. When performing agile projects, effective planning skills are required, e.g. in order to ensure that the agents involved are carrying out the most important tasks at any given time [3]. However, managing such projects is a creative and knowledge-intense task, requiring identifying, creating and sharing of knowledge across the organization. This knowledge is often scattered over many different information sources the involved agents may not even be aware of, provided in different representations and levels of abstraction. In time or business critical situations, this may lead to unnecessary or even unacceptable delays.

In this paper, the presented *Collaborative Agent-based Knowledge Engine* (*CAKE*) [4] provides the infrastructure for mastering any kind of processes. CAKE provides unified access to knowledge available within an organization, and CBR technology is used throughout the system to distribute this knowledge to agents as required. Agile workflows, which offer a light-weight yet formal representation of processes, and collaboration patterns for explicitly describing collaboration among agents are introduced. CBR is used throughout the system in order to guarantee effective search facilities on workflow and agent characterizations. Therefore, a unified data model is provided which enables to represent workflows and agent competencies using structured data, allowing to use CBR efficiently to retrieve appropriate workflow definitions and most competent agents as required.

CAKE has been designed domain-independently and aims to, but is not limited to, cover the field of fire services, medical services, geographical information systems management, and software development which have been proven to share many aspects regarding the occuring processes. For the sake of brevity, this paper focuses on the fire services scenario within the the AMIRA (Advanced Multi-modal Intelligence for Remote Assistance) project. The overall goal of the AMIRA project is to provide a multi-modal solution that will significantly improve the accessibility and resources available to support urgent and critical decisions that must be taken by mobile workers. Fire services demand highly flexible processes and collaborative working in the field: Fire fighters may encounter rare problems and want to perform questions to a system or experts, or they need pro-active information support for optimizing their collaboratively working procedures. Therefore, a hands-free access (e.g. headset and natural language interface) for operatives whose hands and eyes are otherwise occupied must be provided. In addition, performing tasks may be time critical, e.g. while extinguishing a fire, a fire fighter requires workflow descriptions or background information as soon as possible.

In the following section, the CAKE overall architecture is sketched, followed by the idea of collaboration patterns in Section 3. Section 4 discusses the support of agile workflows, especially the planning aspects. Related and adjacent research is discussed in section 5. Finally, a short conclusion and outlook to future work closes this paper.

2 The CAKE Architecture

CAKE is a general domain-independent architecture that can be used for developing domain-specific applications [4]. It provides a very general data and process model, that can be expanded by domain-specific data classes and processes, leading to domain specific ontologies. For coping with knowledge intensive tasks required for context-based information support CAKE uses *workflow technology*. It additionally makes use of *agent technology* to enable and mediate arbitrary access and communication to different agent-based services. For combination of workflow and agent technology a need for an appropriate search facility is encountered. Therefore, a CBR-based search engine is integrated in CAKE for searching on workflow and agent characterizations. The underlying technology can be denoted as third key technology provided by CAKE. An overview of CAKE's architecture, showing the interactions of the different key technologies, is provided in Figure 1. All used technologies are explained more detailed in the remainder of this section.

Data Representation. All components in the CAKE system are using the same data representation. As CAKE should be applicable in several very different application scenarios, a flexible and unified data representation concept for all components is paramount. This is put into practice with a data model and corresponding data objects.

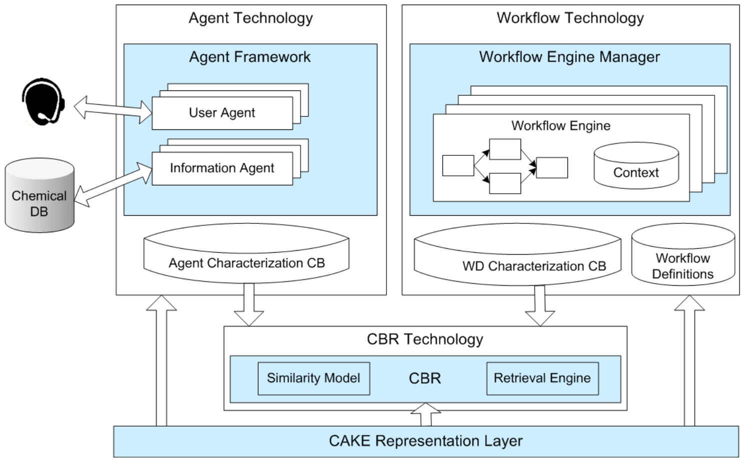

Fig. 1. CAKE System Architecture

The *CAKE Data Model* describes all kinds of data that can occur in the system. It is an object-oriented model using inheritance and aggregation to define the data classes. Available data classes are atomic classes like integer or double, as well as compound data classes like aggregates or intervals. These *system classes* can be used to define an application specific data model. Each specialization of a system class is called *user class* that again can be specialized or used like any other class. Using the data classes of the data model, *CAKE data objects* can be instantiated, and each data object has exactly one corresponding data class that defines the possible value(s) of the object. The object hierachy is the same as defined in the data model.

CBR Technology. A similarity model is defined on top of the data model. The similarity model already contains about 30 similarity measures like Hamming Distance, Simple Matching Coefficient, or sigmoid function. More information about the similarity measures available is provided by Bergmann [5]. All similarity measures may be extended easily. As of this writing, the retrieval engine features a linear CBR retrieval only. The architecture, however, has been designed with more sophisticated retrieval strategies in mind which are expected to be available in the near future.

Workflow Technology. The goal of each workflow system is to model and execute processes of the real world. In CAKE, processes are described using *workflow definitions* that are instantiated and executed in concrete situations. These situations are abstractly represented in the so called *context* of the process.

Each workflow definition consists of initial context values, a *workflow characterization*, a set of *task descriptions*, and the control flow relationships between the task descriptions. The latter allow to arrange the tasks in sequence, in parallel, or in loops, but does not cover data flow at all. The workflow characterization is a semantic description of the workflow definition represented with CAKE data objects. Thus, it enables a semantic CBR search for an appropriate workflow definition in a concrete situation. A task description mainly consists of a task name, the name of the Java class that implements the task's functionality, and a set of parameters that are required for the execution, e.g. the name of the context keys to access and store information.

As tasks are technically realized as Java classes, their possible functionality ranges from simple like adding two numbers to very complex like sending emails or merging retrieval result sets. Another capability of a complex task is its ability to trigger further workflow definitions, so hierarchical decomposition of a complex activity description like "extinguish fire" is achieved by following the way a human would solve this instead of following a fixed process model. The selection of a suitable workflow definition in a concrete situation bases either on utilizing a unique ID associated with a workflow definition or on the workflow characterization by using the CBR retrieval.

The modelling of workflow definitions is very time consuming. To make modelling easier, *task templates* are established. Each task template contains a description of the functionality, the name of the executable Java class, a list of required and optional parameters, and the name of input and output ports that are used for defining the control flow. Based on this information a concrete workflow definition is defined by substantiating the task templates as task descriptions which will be graphically depicted.

At runtime, the *workflow engine manager* initializes a workflow instance of a workflow definition. Each workflow instance is executed in an own *workflow engine (WE)* that controls the interpretation of the workflow definition by executing the tasks in the specified order. Separating workflow definition and execution enables modifications on the workflow instances without doing changes on their workflow definitions. Beyond controlling the life cycle and control flow of tasks, each WE administrates the local context assigned to its workflow instance and facilitates the capturing, storing, and changing of context-based data. Due to the possibility of nested (sub)workflow instances local contexts can be nested as well. Assigned to the higher-level workflow engine manager itself only one global context exists that is accessible by all workflow engines under control.

Agent Technology. The CAKE agent framework provides a unified interface to couple external knowledge sources as well as user interfaces into the system. In opposite to classical multi agent systems the agents are managed from the central workflow component and are therefore less autonomously. Technically, the framework distinguishes between *information agents* and *user agents*. Information agents provide and may also change knowledge and can be simple ones like search engines, databases, groupware calendars, or human experts or complex ones like data mining tools, classifiers, or dialog strategy tools. User agents request knowledge and are in general interfaces to human system users, e.g. graphical user interfaces or natural language interfaces, but can also be other computer systems. Each agent is composed of a *technology component*, a *wrapper*, and a *competence profile*. The technology component is the external knowledge system, that should be made available via CAKE. To enable the communication between CAKE and different technology components, the wrappers are used as mediators that fulfil two tasks: firstly, to map between the technology components' ontologies and the CAKE domain specific ontology and secondly, to realize the technical interface. The competence profile is a CAKE data object characterizing the agents' competencies and is stored in the agent characterization CB for retrieval.

A very simple but quite typical application scenario for this framework is a system user who communicates with his or her user agent via a natural language interface. The system user queries for information and CAKE routes the request to an appropriate information agent, e.g. a database agent, which performs a query. The results are routed back accordingly.

By combining the workflow engine manager and the agent framework, CAKE enables pro-active and context-based information support of user agents. Each user agent is in general assigned to a workflow instance, and the WE monitors all interactions (e.g. requests) of the user agent with this particular workflow instance. Consequently, based on the monitored interactions the context-based data already collected is enriched, which ultimately allows the WE to build a context.

3 Collaboration Patterns

By utilizing agent technology for integrating heterogeneous data sources there is a need for organizing the collaboration among various agents. Simply integrating each information source is not sufficient because in this case it is up to end users to delegate their own requests to the various data sources. For preventing end users from specifying such details CAKE incorporates *collaboration patterns*. Collaboration patterns are workflow definitions based on best practices and coordinate agents pursuing a common goal. For example, search strategies can be realized through collaboration patterns for delegating requests in consideration of all available information about the current user request. Collaboration patterns allow both general search strategies independent of domain-specific knowledge and domain-dependent search strategies. General search strategies can comprise conventional meta search facilities that are developed in the area of information retrieval [6, 7]. Additionally, the latter search strategies integrate domain-specific information for proceeding current requests, e.g., a strategy could specify exactly which information agent should be requested when the query refers to chemical materials.

By utilizing workflow technology CAKE makes use of a widespread technology for representing collaboration patterns and enables a variable arrangement of machine-executable tasks for realizing different search strategies.

Retrieval of Collaboration Patterns. The retrieval of collaboration patterns means to retrieve workflow definitions based on their characterizations, which are stored in the corresponding characterization case base. The underlying domain data model allows to search collaboration patterns in consideration of semantic classifications. Each collaboration pattern contains for each task at least one *agent role* that describes the information an agent should provide, e.g., descriptions of services or database contents. These agent roles are realized as data objects and are stored in the context where the tasks can access it. To retrieve appropriate agents the agent roles are used as queries to perform a CBR retrieval on the agent characterization case base. Collaboration patterns can be retrieved during workflow execution, even without involving the user. As nesting of workflow is supported by CAKE, collaboration patterns can be applied at any time.

Agent Allocation. For providing a dynamic agent pool where agents can be registered and deregistered during runtime CAKE integrates collaboration patterns that can be specified independently from agents. Consequently, collaboration patterns can

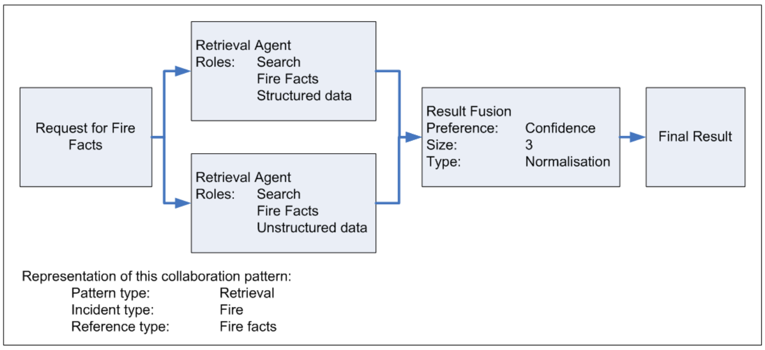

Fig. 2. Collaboration Pattern Example

consist of task descriptions in which only agent roles are specified independently from concrete agents. This leads to the need of retrieval facilities based on agent roles as requests for agent competence profiles. Again the CBR-based search engine is utilized for searching these profiles.

When specifying a task description where a service is demanded the domain modeler can utilize domain ontologies for assigning agent roles like “chemical expert” or “expert for fire facts”. These roles describe which kind of agent is able to carry out the demanded service. The CBR-based search engine utilizes the ontologies for searching the most competent agent. CAKE allocates this agent to the task description before the task can be executed.

Collaboration Pattern Scenario. In order to illustrate the functionalities of collaboration patterns a scenario is presented that gives a brief overview about the combination of workflow characterizations, retrieval on collaboration patterns, and agent allocation. Here, the CAKE approach comprises the following processes:

– Analysis of the user’s input
– Information extraction for collaboration pattern retrieval
– Collaboration pattern retrieval
– Agent allocation
– Execution of the collaboration pattern

By using the CAKE approach in a fire fighter scenario a fire fighter can perform a request like “There are plastic crates burning”. A speech dialogue agent analyzes the utterance determining the incident type “fire”, reference type “fire facts”, and the material “polypropylene” inferred from “plastic” and sends it to CAKE. According to the categorizations of incident and reference type CAKE retrieves a suitable collaboration pattern that realizes a retrieval on corresponding data sources. Here, both the retrieval on structured fire facts and the retrieval on unstructured fire facts extracted from written reports are recommended; the respective collaboration pattern is pictured in Figure 2.

The collaboration pattern consists of two task descriptions, which contain references on retrieval agents and agent roles. When such tasks are executed CAKE initiates a retrieval for suitable agents. In order to allocate appropriate agents CAKE performs a retrieval process that searches agents based on the roles specified within the collaboration pattern: one agent that offers retrieval on structured data and bases on fire facts, one agent that provides retrieval on unstructured data and also bases on fire facts. The agent retrieval is applied on specific agent competence profiles. These profiles capture the agents' service, expertise, and the underlying technologies. Based on these profiles a CBR-based retrieval is performed for finding suitable agents that can be allocated to the collaboration pattern. Assume three information agents are currently available and experts for fire facts. First, one agent bases on Wiki technology and searches the web for fire facts. Second, one agent works on text documents stored in a database. The last agent bases on CBR technology. By executing CBR-based search for agent allocation CAKE selects the Wiki agent for retrieval on unstructured data because the Wiki agent has likely access to newsworthy information in contrast to the database agent. As information agents that searches on structured data the CBR agent is chosen. Consequently, the agents are allocated to the task descriptions, which can be now executed by CAKE.

By performing two agent requests CAKE also receives two different result lists. Here, results of information agents are preferred that have a high degree of confidence specified in the competence profile. According to the result fusion the first three results are taken from both result lists. In order to be able to compare between the new result sets CAKE conducts a normalization process for merging both sets to one final set. The first three final results are transferred to the end user.

4 CBR-driven Planning Support

The CAKE workflow definitions can be used for planning activities which are expected to be performed during project enactment. Each workflow definition describes an individual activity, which may either be refined by introducing tasks (including subworkflows) and a control flow among them, or which may be defined abstractly in terms of a "black box". This allows coarse-grained planning as required by many application domains in order to represent capricious situations. These situations occur in many application domains incorporating creative or knowledge-intense processes which make it impossible to lay out detailed procedures beforehand (e.g. because selecting a concrete procedure depends on context parameters: a fire fighter has to know about the type of fire before an applicable procedure to extinguish it can be chosen.) Moreover, considering the individual strengths and weaknesses of user agents in order to choose the most competent agent for enacting a workflow can be a daunting task.

As explained above, by separating workflow definitions from workflow instances, CAKE supports late planning by allowing to apply changes to workflows even during their execution. In addition, abstract tasks allow to specify workflow definitions at any level of detail, which in combination with late planning leads to support of weakly structured workflows [8]. These concepts overcome limitations of "classic" process models and workflow enactment control known from traditional workflow management systems, which are unsuitable in agile environments. During workflow execution, a user agent may do late planning in order to further refine the situation within the current context when approaching an abstractly defined workflow definition. However, in time and business-critical situations this is insufficient: For instance, while extinguishing a fire, a fire fighter cannot wait until late planning has been completed by the headquarters. In order to overcome this, late planning may be backed by previously recorded planning activities, leading to agile workflows supported by CBR.

Agile workflows. For workflow definitions, goals and metrics may be defined with respect to the underlying domain ontology (e.g. “workflow goal is to produce a report”) to form a workflow characterization. Workflow goals may be one of the following (with “workflow” denoting the workflow instance based on the workflow definition the respective goal has been assigned to):

1. Artifact goals: The workflow is expected to produce a specific artifact, e.g. a report. The artifacts may be specified further by additional metrics (e.g. length, maximum time allowed to create the artifact etc.)
2. Time goals: The workflow is expected to be executed within specific time constraints, e.g. maximum allowed duration.
3. Structural goals: The workflow is expected to contain specific structural elements, e.g. placeholder tasks. These goals may be specified further by additional metrics, too, such as the minimum number of alternatives available to perform this task, or a maximum allowed number of placeholders.

When proceeding to an abstractly planned task, the WE queries for a suitable workflow definition from the WD characterization CB. However, retrieval from the local CAKE instance may not yield to ideal results, e.g. if there is only a small number of workflow definitions. So, procedures how to retrieve the actual replacement for the abstract task are described using collaboration patterns. Thus, further information agents, as well as domain or organization-specific constraints can be respected. For example, when reaching the placeholder task “estinguish fire” while enacting a workflow, CAKE can offer the replacement candidate “estinguish fire using the water mist method” by taking orginaization-specific experience into account.

Since collaboration patterns are workflows too, goals, metrics, and competence profiles may be assigned to them. For artifact goals, workflow definitions, and agent descriptions may be accessed as artifacts. Competence profiles for collaboration patterns may be used in distributed environments for optimization purposes, e.g. to save bandwidth when querying information agents.

Support for Interleaved Planning and Enactment. When instantiating a workflow definition, the user agent assigned to it may use it as guidance to fulfill a particular activity. For instance, the workflow definition may provide a formal specification of a standard procedure how to extinguish a dangerous fire. Deviations from this initial definition have to be expected, and CAKE embraces change by allowing ad-hoc modification in the sense of a Plan-Do-Check-Act cycle allowing to transform tacit knowledge of the project participants (e.g. experience of a senior fire fighter) into an explicit workflow definition. Notably, domain-specific knowledge is kept aside from the workflow definitions and accessed through the information agents, hence the workflow definitions may be shared across the organization (e.g. between different fire departments). This enables to capture knowledge in the sense of an organizational memory [9].

5 Related Work

In terms of workflow technology combined with CBR technology, various approaches have been developed that utilize CBR technology for workflow retrieval and planning facilities. Usually, the approaches make use of CBR during build-time. Focusing traditional workflow management systems Workbrain merges Organizational Memory and workflow technology [10]. For structural planning a CBR application is integrated and can be utilized by a human workflow planner before the respective workflow will be enacted. By retrieval on workflow characterizations the CBR application allows to describe a problem and retrieves similar solutions, allowing to construct new workflows.

Case-Oriented Design Assistant for Workflow Modeling (CODAW) includes a CBR-approach to workflow model management [11] that focuses on design of processes including ordering relevant business tasks in consideration of pre-conditions and post-conditions. Therefore, appropriate structured workflow case representations, similarity-based retrieval algorithms, and procedures for composing workflow definitions were developed. Instead of searching on workflow characterizations the CBR application works on workflow definitions regarding pre-conditions, post-conditions, and procedural conditions.

Similar to CAKE, CBRFlow [12] combines workflow technology with conversational CBR for coping with changing and unpredictable environments. For realizing the concept of workflow modification during runtime workflow instances can be annotated with cases that consist of a set of question-answer pairs and one action. After the annotation these cases can be directly used as assistance for decision-making processes because of representing additional knowledge, maybe in form of instructions, which can be used during workflow execution. Modifications on the workflow definitions is done by workflow modelers and no search facility on workflow definitions is supported.

Process-based knowledge management has also been addressed by Holz [13], who introduces a formalism for describing information needs required to perform processes, as well as for describing information sources where to satisfy these needs. The search for apppropriate information sources is backed by CBR retrieval.

In summary, the four approaches sketched above support CBR-driven planning facilities with respect to workflow technology like CAKE, but these approaches lack techniques for accessing external information sources. Particularly, agent technology is not supported that facilitates methods like collaboration patterns for addressing agents and organizing collaboration among them.

6 Conclusion and Outlook

CAKE is a general infrastructure for dealing with any kind of processes in very different application domains. It provides a flexible, object-oriented representation for domain specific data and innovatively combines the three key technologies workflows, agents, and CBR-based search. In the current release, the domain ontology, the workflow definitions as well as the task descriptions, and the agents' competence profiles may be specified using the appropriate XML (eXtensible markup language) schemas, and the CAKE Editor is currently developed to provide a user-friendly modelling environment.

We further plan to improve the CBR-based component: For applying CAKE in the application scenarios regarded so far, linear retrieval was sufficient for all CBR-based searches. But in future application scenarios, advanced retrieval strategies like index-based retrieval may be required. As the architecture is designed openly, these enhancements can be added easily. Other topics of interest cover similarity measures, as current approaches from the area of process mining will be integrated and specified as special similarity measures to allow retrieval of workflow definitions not on basis of their characterizations, but by comparing the workflow definitions themselves. Therefore, structural and temporal aspects of the workflows will have to be taken into account.

Acknowledgements. The authors acknowledge the European Commission for funding AMIRA under grant number FP6, project IST-2003-511740.

References

1. Boehm, B.: Get ready for the agile methods, with care. IEEE Computer **35** (2002) 64–69
2. Rising, L., Janoff, N.S.: The Scrum software development process for small teams. IEEE Software **17** (2000) 2–8
3. Beck, K., Fowler, M.: Planning Extreme Programming. Addison-Wesley (2000)
4. Freßmann, A., Maximini, R., Sauer, T.: Towards collaborative agent-based knowledge support for agile projects. In Althoff, K.D., Dengel, A., Bergmann, R., Roth-Berghofer, T., eds.: WM2005: Professional Knowledge Management Experiences and Visions, Kaiserslautern, Germany, DFKI GmbH (German Research Center for Artificial Intelligence) (2005) 383–388
5. Bergmann, R.: Experience Management. LNAI 2432. Springer (2002) ISBN 3540441913.
6. Fox, E.: Combination of multiple searches. In Harmon, D., ed.: The Second Text REtrieval Conference (TREC-2), Raithersburg, MD (1994) 243–252
7. Lee, J.: Analysis of multiple evidence combination. In: ACM SIGIR'97. (1997) 267–275
8. van Elst, L., Aschoff, F.R., Bernardi, A., Maus, H., Schwarz, S.: Weakly-structured workflows for knowledge-intensive tasks: An experimental evaluation. In: Proceedings of the 18th International Workshops on Enabeling Technologies: Infrastructures for collaborative enterprises, Linz, Austria, IEEE Computer Society Press (2003) 340–345
9. Wargitsch, C., Wewers, T., Theisinger, F.: An organizational-memory-based approach for an evolutionary workflow management system - concepts and implementation. In: HICSS '98: Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences-Volume 1, Washington, DC, USA, IEEE Computer Society (1998) 174
10. Wargitsch, C., Wewers, T., Theisinger, F.: Workbrain: merging organizational memory and workflow management systems. In: Workshop on Knowledge-Based Systems for Knowledge Management in Enterprises, Freiburg, Baden-Wuerttemberg, Germany, 21st Annual German Conference on AI'97 (1997)
11. Madhusudan, T., Leon Zhao, J., Marshall, B.: A case-based reasoning framework for workflow model management. Data & Knowledge Engineering: Special Issue on Business Process Management **50(1)** (2004) 87–115
12. Weber, B., Wild, W.: Towards the agile management of business processes. In Althoff, K.D., Dengel, A., Bergmann, R., Roth-Berghofer, T., eds.: WM2005: Professional Knowledge Management Experiences and Visions, Kaiserslautern, Germany, DFKI GmbH (German Research Center for Artificial Intelligence) (2005) 375–382
13. Holz, H.: Process-Based Knowledge Management Support for Software Engineering. PhD thesis, Department of Computer Science, University of Kaiserslautern (2003)