# Case-Based Support for Collaborative Business

Ralph Bergmann[1], 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
{ralph.bergmann|andrea.fressmann|kerstin.maximini|rainer.maximini}@wi2.uni-trier.de
[2] rjm business solutions GmbH
68623 Lampertheim, Germany
t_sauer@rjm.de

**Abstract.** This paper describes the development of the generic collaboration support architecture CAKE incorporating case-based reasoning (CBR). 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. Adaptive workflows and collaboration patterns selected by a CBR process are introduced for explicitly describing collaboration among agents. In order to guide the technical design of the architecture, a systematic analysis of the requirements for collaboration support has been performed in various application domains.

## 1 Introduction

While the first case-based reasoning systems have been designed and implemented as stand-alone applications addressing an isolated problem solving task, a current trend can be observed towards the integration of CBR techniques as one component into more complex environments. Within such environments CBR can play different roles in situations in which decisions are to be made based on previous experience. One such complex environment under investigation at the Business Information Systems II research group at the University of Trier is the support of collaborative business.

Since the 1980s *CSCW (Computer supported cooperative work)* [1, 2] is the research field that deals with the analysis and the design of information technology to improve the way human beings are working together in a certain working environment. In such a collaborative working environment, support for *communication*, *coordination*, and *cooperation* must be integrated systematically with the business processes of the organization. Recently, it has been recognized that collaboration support must not be limited to humans but must also cover automated decision support systems and automated knowledge sources in a seamless fashion. This demands for a much stronger link between CSCW and research on decision support technologies like for example CBR.

One main goal of research was to develop a new generic collaboration support architecture that integrates access to automated knowledge sources and whose overall behaviour is guided by previous experience. Previous experience can be useful when knowledge sources are selected or whenever successful ways of collaboration among human and automated collaboration partners must be determined. In order to guide the technical design of such an architecture, a systematic analysis of the requirements for collaboration support and particularly for the support by previous experience is necessary. In various R&D projects such analyses have been performed. It turned out that similar problems occur in quite different application domains: geographical information management, fire services, and software engineering. In all these domains there is a need to coordinate collaborative activities jointly together with the knowledge required for enacting these activities. Furthermore, the flow of these activities is very often quite dynamic and subject of change depending on the current context. Thus there is a strong need for a highly flexible CSCW solution incorporating CBR. The authors have identified the most important requirements in each application domain in detail, generalized them to a more abstract level, and addressed them during the development of a domain-independent system called *Collaborative Agent-based Knowledge Engine (CAKE)* [3–6]. 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. Adaptive workflows and collaboration patterns selected by a CBR process are introduced for explicitly describing collaboration among agents.

This paper summarizes the experiences made during the development of the first prototype of CAKE and its application in three application domains, namely geographical information management, fire services, and software engineering. It starts with a presentation of the three application scenarios and a summary of the most important requirements for CAKE in Section 2. Section 3 shortly presents the CAKE architecture and lays a special focus on the role of CBR. How these technologies have been utilized to build up support systems for the three regarded application scenarios is sketched in Section 4. The paper closes with the presentation of related work in Section 5 and with a short conclusion in Section 6.

## 2 Requirements derived from the Application Scenarios

In the following, three applications scenarios are described, with a special focus on the problems that the people face. After that, requirements for a software solution that can be used as a basis to develop support systems for all the application scenarios are summarized.

### 2.1 Workflow support for geographical information management

The company rjm business solutions GmbH has been assigned by the Monument Protection Agency of Hessen to conduct the long-term eGovernment project *DenkXweb*. DenkXweb provides a freely accessible Internet service to publicly access the monument register[3], that lists each building and site of historic interest together with its exact location and a rationale why it is subject to protection. That is advantageous for landlords, planners and architects, who have to ensure themselves that they comply with all

[3] A demonstration is available under http://www.denkmalpflege-hessen.de/denkxweb.

regulations that apply to the corresponding land parcels, including monument protection. During the development of DenkXWeb, rjm business solutions GmbH is facing several problems and the two most engraving ones are shortly sketched below. A more detailed description of that problems can be found in [6].

**Inconsistency of data.** Geospatial data is maintained by 41 regional authorities, that provide their data on CD once a year using separated files to describe map data in *ALK*[4] format and meta information in *ALB*[5] format. After having combined map data and meta information, the location and dimensions of monuments and protected sites have to be transferred from a printed map, often collected manually by specialists in the fields. Thus, data inconsistencies are quite common, e.g. map and meta information often cannot be merged because of missing or contradicting descriptions or different granularity of the data, or transferring monument related data can fail because of differences between digital and printed map editions. In all cases, inquiries to the authorities are required, and while waiting for answer– that may take up to half a year –, the area in question cannot be processed further.

**Determination of the current processing status and open issues.** Several programs and databases are currently used to document flaws and to store the duration needed for each piece of work: a time recording tool, a Wiki system, and Excel spreadsheets. This heterogeneous data storage makes it difficult to determine the exact status of progress, and it is hard to derive documentation for the project stakeholders. But especially such documentation is required to estimate efforts for future pieces of work.

A flexible support system should tackle these problems: the staff could be supported during data acquisition and publishing by coordinating and documenting inquiries to the authorities and by automatic flaw detection, and the project stakeholders could be supported by automatic documentation and open issue collection.

### 2.2 Support for time-critical processes for fire services

In recent years, the demand for support systems for emergency services has increased significantly in order to optimize methods for all types of protection. Focus is put on training, qualifying, and supporting members of emergency services that can be characterized by time-critical processes on incident locations. In that scope, the AMIRA (Advanced Multi-modal Intelligence for Remote Assistance) project[6] was initiated to address innovative technologies and their combinations leveraged in high safety of time-critical application domains. Particular to fire services, a lot of knowledge is captured in several data sources in different forms, e.g. structured databases and unstructured text.

The overall goal of the AMIRA project is to provide a multi-modal solution that will significantly improve the accessibility of resources available to support urgent and critical decisions that must be taken by mobile workers. The envisaged AMIRA system makes knowledge available by integrating best practices for improving search

---

[4] ALK = “automatisierte Liegenschaftskarte”, graphical representation of real estates

[5] ALB = “automatisiertes Liegenschaftsbuch”, textual information on real estates

[6] AMIRA is funded by the EU. Project partners are Kaidara Software, Fast DataSearch, DaimlerChrysler RIC, University of Trier, Fire Service College, West Midlands Fire Service, and Avon Fire & Rescue.

processes. This project aims at developing a solution that provides a hands-free access (e.g. headset) for operatives whose hands and eyes are otherwise occupied. For achieving this goal the solution is developed basing on two mobile working domains, roadside assistance and fire services.

An envisaged scenario within fire services can be as follows: the *incident commander (IC)* is the only person who is in charge and responsible for the decisions made on location. Therefore, all fire fighters involved provide him/her precise known details of the incident. Consequently, there is a demand for supporting the IC when he/she lacks information necessary for decision-making and in estimating current resources. For example, a fire fighter comes to the IC talking about a burning van loaded with explosives and found on the incident ground. Therefore, the IC uses the mobile AMIRA system in order to receive information about what has been found on the location and how to deal with it, while staying close to the incident and close to the fire fighters. For supporting the information flow between IC and fire control, the incident control centre, and for improving collaboration on the incident the AMIRA systems pro-actively routes the IC's requests and corresponding answers to fire control.

### 2.3 Workflow Support for Agile Software Engineering

In the software engineering domain, workflow systems are an established technology for team coordination and cooperation. That is, workflow support covers overall structuring of activities (e.g. by providing project plans or to-do lists), reuse of task descriptions to efficiently handle situations encountered before, or effective scheduling to ensure that team members always work on the most important tasks. However, software development processes tend to be unknown, unrepeatable, and knowledge intensive, with changes occurring frequently and the team members can be supported by collaborating with other team members (*human agents*) and by usage of *computerized agents* like bug-tracking systems or automated test tools. Nevertheless, in many organizations there are norms [7] shaping the way of working within the organization. Hence, effective collaboration strategies following such norms are necessary to ensure that the efforts of individual team members are all aiming in the same direction.

During the course of defining and enacting business processes, finding appropriate agents is crucial. First, modellers have to make sure that tasks are performed by the most experienced team members. Second, there may be activities that are known to succeed only when an appropriate collaboration strategy is taken. Finally, an agent trying to enact a task might seek assistance from another agent.

An envisaged scenario within an agile development [8] is a follows: the team has proceeded to the first iteration, and an intermediate system has been deployed at the customer site. Negotiations with the customers throughout the first iteration have resulted in the customer preference that an e-commerce system should be added to the system. Now, the team members playing management roles have to settle on the most basic workflow, which will control the very next steps. The team starts to instantiate the very first business process and suitable team members playing the roles of a domain expert and a software developer are assigned. Thereby, organization-specific constraints have to be taken into account (by querying a human resources management information agent providing availability data and a Wiki system containing tentatively planned leaves). These team members can now autonomously decide how to proceed for fulfilling the tasks. E.g., the software developer starts work on the assigned task by adapting the actual business process. Because of not being familiar to the actual project the software developer is searching comprehensive information about this project with respect to similar tasks, to relevant Java references, and to relevant design rationales. That is, relevant information is presented to the developer that he/she can be familiarized to the work at issue. Additionally, the developer gets a link for searching a Java expert for booking systems if more explicit information is necessary. Primarily the idea is to support the developer through computerized agents but if this is not enough there is an option to ask human experts. Therefore, a search facility organizes retrieval on human Java experts, returning information about how to get in contact to such an expert. Surely, the developer can proceed further in adapting the assigned task. Either he/she uses best practices in terms of collaboration which already includes useful domain knowledge or he/she carries out own adaptations on the task for achieving the envisaged task artefact.

### 2.4 Requirements

The authors have identified the most important requirements in each application domain and generalized them to a more abstract level. These requirements have than been tackled during the development of a domain-independent software solution, in the following denoted as *envisaged system*.

**Requirements for Integration.** In all application scenarios the necessary knowledge is scattered over a rich set of heterogeneous information sources within the organization. Thus the envisaged system should provide functional features and services that are able to access different information sources and can be tailored to integrate the particular sources in a given application domain. The achievement of interoperability between existing tools, features, and services and the envisaged system is a general requirement in organizations in order to avoid restricting previous business processes.

In all application scenarios, people are collaboratively working together and a seamless integration of humans and machines is needed. This can be reached by implementing an agent architecture that is able to administrate human as well as machine agents. Such an agent architecture could also allow organizations to efficiently distribute business tasks among organization members with reference to their competencies.

**Requirements for Business Process Support.** All application scenarios are characterized by complex knowledge-intense business processes that are executed over and over again, but always with (slight) variations depending on the actual context and on user interaction. Even worse, in some cases the tasks of a business process cannot be specified at the time of business process modelling, so that late-modelling of processes is required as well. An appropriate workflow technology, allowing representing and executing such highly flexible business processes, must be developed for the envisaged system. It should allow using past experiences done within organizations that may help novices and inexperienced organization members in carrying out their jobs. For alleviating collaboration among humans and machines the envisaged system should also provide means for reusing best practices in terms of agent collaboration. This would allow the improvement of business processes within organizations with regards to time-saving and cost-optimizing purposes.

**Requirements for Search Facilities.** Regarding the two kinds of requirements described above a need for search facilities on business processes and agents becomes obvious. More precisely, search facilities can support the modelling of new business processes through reuse of business process or business task representations and the execution of business processes through reuse of old business processes, a key feature for adaptive workflows and late-modelling. Search facilities for agents, meaning humans and machines, are also needed to enable the execution of flexible, situation-dependent business processes, allowing assigning competent and currently available agents to a business task during runtime.

## 3 The CAKE Approach

The fundamental idea of the CAKE approach [3–6] is to realize CSCW technology through the combination of *workflow*, *agent*, and *CBR technologies*. According to the requirements summarized in Section 2.4, different key features were considered during the development of CAKE. It makes use of *agent technology* in order to integrate IT systems as well as humans as agents. It uses *workflow technology* for representing business processes and for specifying collaboration among agents which are following a common goal. Instead of supporting static and pre-defined processes, CAKE aims at supporting flexible and changing processes through adaptive workflow management. In order to realize the high flexibility, sophisticated search facilities for business processes, single tasks, and agents are required. A *Structural CBR (SCBR)*[9] approach, working on top of a domain-specific data model and considering the semantics and structural aspects, meets all requirements. An overview of CAKE's architecture, showing the interactions of the three key technologies is shown in Figure 1. These key technologies, set up on the underlying common data model, are explained in the following sections.

### 3.1 Workflow Technology

The goal of the workflow system is to model and execute processes to support business processes within organizations. CAKE conceptually distinguishes between the representations of such *business processes* ("real-world processes") and the "internal processes" *collaboration pattern* and *administrative process*. Business processes aim at producing an end-product and collaboration patterns produce a by-product of a final business process product. Furthermore, collaboration patterns focus on agent collaboration based on best practices. Administrative processes are CAKE-specific processes used for internal procedures. For example, sometimes a top-level workflow is needed that waits for initial user interactions. All of these processes are formally represented as *workflow definitions* that can be instantiated and executed in concrete situations at runtime. These workflow instances are in the following shortly denoted as *workflows* and the situations are represented in the so called *context* of the workflow.

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 allows arranging the tasks in sequence, in parallel, or in loops, but does not cover data flow at all; data exchange is realized by using the context.

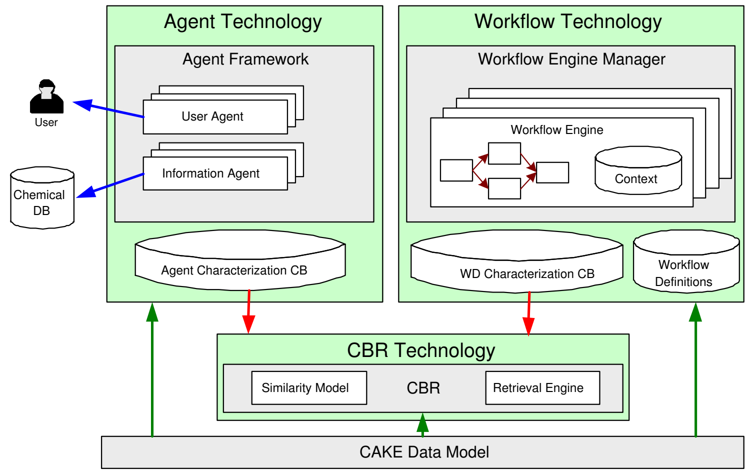


**Fig. 1.** CAKE System Architecture

Workflow characterizations and task descriptions are structural CAKE data objects describing the workflow definitions or tasks. They are stored in a central repository called *WD Characterization CB*. One of the capabilities of a task is to trigger further workflow definitions, so hierarchical decomposition of complex processes is enabled.

Due to the domain requirements a further distinction of the workflows is done. *Short term workflows* are representations of short-lived (time-critical) business processes. The usage of such workflows aims at achieving dynamic changes and flexibility during enactment. *Long term workflows* represent processes without time limitations. As opposed to short term workflows, the support of these workflows requires maintaining the state of workflow executions.

### 3.2 Agent Technology

The CAKE agent framework provides a unified interface to couple external knowledge sources as well as user interfaces with the core system. Technically, the framework distinguishes between *information agents* and *user agents*. Information agents provide and may also change knowledge like search engines, databases, groupware calendars, human experts, data mining tools, classifiers, or dialogue 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 an *agent characterization*. The technology component is the external knowledge system that is made accessible via CAKE. To enable the communication between CAKE and different technology components, wrappers are used as mediators that fulfill two tasks: firstly, to map between the technology components' ontologies and the CAKE domain specific ontology and secondly, to realize the technical interface. The agent characterization is a structural CAKE data object, characterizing the agent's competencies.

The flexibility of the workflows is carried forward to the agent framework by providing a dynamic set of agents whereas agents are able to enter and leave the CAKE *agent society*. Therefore, the agent framework is able to manage heterogeneous agents that have completely different purposes, knowledge, and capabilities [10]. This information can be captured by agent characterizations. When intending to enter the agent society an agent has to register its characterization that is stored into a central repository called *Agent Characterization CB*.

### 3.3 Structural CBR Technology

Within CAKE, CBR technology is applied for two different purposes. First, it is used for the selection of suitable workflows whenever a new workflow must be started. Second, it is used for the selection of appropriate agents, whenever an agent communication is initiated. CAKE uses CBR for these tasks to overcome the inflexibility of static assignments of workflows and agents. Therefore, workflows and agents are treated as cases, whose characterizations are stored in the *WD* and *Agent Characterization Case Base* (see Figure 1). A characterization of a particular workflow describes the situation in which this workflow can be (or has been) applied successfully. When a new similar situation occurs, the workflow is proposed again. A characterization of a particular agent describes the situations for which the agent is competent to answer a query. When a new similar situation occurs, the agent is used again as information source. Hence, two different similarity-based case retrieval tasks occur.

CAKE incorporates a single generic CBR component to implement both retrieval tasks. The requirements analysis strongly supported the need for a structured description of the respective application domains and for the need to construct an overall domain ontology. Its purpose is to add structure to the various information sources, items, and tasks that must be considered. Therefore we decided to implement the CBR component as a standard structural CBR approach [9]. The idea underlying the structural CBR approach is to represent cases according to a common structure called domain model. The domain model specifies a set of typed attributes (also called features) that are used to represent a case. In CAKE the domain model for the case bases are subsets of the overall Cake Data Model (see Figure 1). This data model supports full object-oriented data modelling including the modelling of class hierarchies as well as relational and multi-valued attributes [11]. For a particular application the overall domain ontology is encoded in this data model. Characterizations of workflows and agents are then constructed as instances of this data model. This data model is also used to represent the current working context in which workflows and agents must be selected.

The case retrieval step of the CBR component makes use of an explicitly modelled similarity measure. This similarity measure is a function that compares two instances of the data model (i.e. the current context and a characterization of a case) and assesses their similarity. We do not provide a single standard similarity, but enable the system developer to model the notion of similarity according to the requirements of the domain [11]. Similarity modelling is guided by the traditional local-global principle. For each attribute in the domain model the developer can chose a local similarity measure and for each class in the domain model the developer must assign a global measure that describes the aggregation of the similarity values obtain from the local measures.

### 3.4 Combination of the three key technologies

The three key technologies can be combined to realize adaptive workflows and dynamic agent societies.

**Adaptive Workflow Support.** According to the requirements different kinds of workflow adaptations are required depending on changing contexts. Regarding time-critical situations a quick reaction on user actions is demanded, which is supported by short term workflows. As opposed to this, long term workflows are often subject to change when adaptations to ongoing business processes become necessary. From a more technical point of view, adaptations to workflows can be realized through both, instantiations of new workflows and insertion/deletion/modification of tasks during runtime. Therefore, CAKE integrates the following features:

1. Ability to search workflow definitions
2. Sub workflow instantiation during enactment
3. Ability to search task descriptions
4. Inserting tasks during workflow enactment

Those features have different impact on short or long term workflows. Due to short-lived constraints, short term workflows require automated adaptations based on the first two features. Automated adaptations can be viewed as situational workflow instantiation for a concrete context that may be changed after user interaction. Based on the current context information, CAKE starts a search for an appropriate business workflow or collaboration pattern. Then the most suitable workflow definition is instantiated as sub workflow. Different context constellations during enactment influence the selection of the next sub workflow to be enacted as depicted in Figure 2. Both features lead to an automated form of late modelling [12] of workflows which is a key feature for adaptive workflow management and it is well-known for long term workflows. Changes to such workflows are intended to be done manually, which means that the decision what kind of adaptation has to be done is left to users.

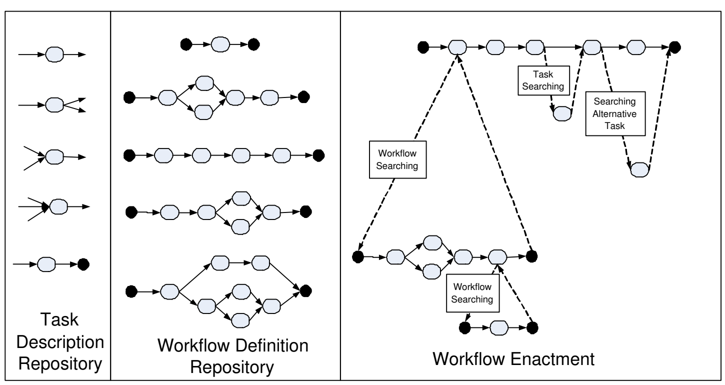


**Fig. 2.** Adaptive Workflows by CAKE

**Dynamic Agent Societies.** The second application field of CBR technology in CAKE is the retrieval on agent characterizations. Agent characterizations consist of attributes like the agent's role, the quality of data the agent provides, the area of expertise, the type of service it offers, and the data format. In order to support a dynamic agent society where agents can be registered and deregistered, CAKE allows dynamically agent allocation during workflow enactment. For exploiting these potentials, task descriptions only contain agent roles in order to specify what kind of agent should carry out the corresponding task. Based on these roles an appropriate agent characterization can be retrieved and the corresponding agent can be allocated to the task in an ad-hoc manner. Thus, user agents can be integrated in workflows for carrying out a particular job according to their competencies and information agents can be selected according to their knowledge and quality. In time-critical situations only registered agents, which are definitively available, are retrieved.

A difference to other agent-based approaches is that CAKE agents do not negotiate with each other. Instead, collaboration patterns are especially developed for organizations acting as loose contract among agents and capturing best practices about efficient cooperation. Best practices often occur as organizational norms or past experiences, e.g. about what agents are working together efficiently.

## 4 Solutions for the Application Scenarios

In the following a description of how CAKE was put into practice in the three application scenarios presented in Section 2 is given. Due to space limitations, only the geographical information management scenario is described in detail. For the other scenarios, the solutions are only sketched in brief.

### 4.1 Workflow support for geographical information management

A group of five students from the University of Trier have used CAKE to develop a support system called *GIS-DOKU* [13, 6] that improves the data acquisition and publishing processes and provides automated documentation and flaw detection. The first step during the development of GIS-DOKU was modelling the workflow definitions to describe the data acquisition and publishing processes for the closed area of a commune. As such a business process takes a while, long-term workflows are used. Because CAKE supports sub workflow modelling and execution, the students started with the very general top-level workflow definition shown in Figure 3, and refined each of the three process steps by further workflow definitions of its own.


**Fig. 3.** Top-Level workflow definition represented as an EPC

Additionally, workflow definitions for open issue resp. flaw handling have been modelled. They are not part of each workflow, but the corresponding process steps are dynamically inserted in case of occurring failures, e.g., an inconsistency of ALK and ALB data. As the next step, 426 workflows have been instantiated, representing how each municipality in Hessen is processed. Thereby, each workflow indicates whether for a commune data acquisition could have been completed, data acquisition has not been started yet or the data acquisition is already running.

When preparing data for the DenkXweb Internet service, each commune requires an individual strategy in order to overcome ALK and ALB handling problems and issues arising from preprocessing and transferring monument related data. In addition, external events have to be handled, like communal authorities voting against publishing data for certain areas. Using CAKE, handling these deviations is supported well, since workflows can be altered using late modelling. Using CBR retrieval, CAKE helps to find appropriate workflow definitions which can be incorporated as exception handlers. That is, employees can tailor workflows easily as CAKE suggests appropriate workflow definitions for a given situation. Furthermore, the CBR engine helps to find appropriate agents to put these workflows into practice. That is, the numerous information agents can be leveraged best, and human team members can be selected based on their characterization profile as well.

**Example.** The sub workflow of the first process step "Import ALK and ALK data" of the Top-Level workflow definition includes a task called "Assure granularity of ALK and ALB data" that can be started after ALK and ALB data are available in electronical form. During the execution of this task, several problems may occur. They are arranged in a taxonomy, together with other problems that can occur during the whole data acquisition and publishing process. Figure 4 shows an excerpt of the whole taxonomy.

- ⊟ Problem
  - ⊟ ALK
    - ALK-CD Reading Error
    - Wrong granularity of ALK
    - …
  - ⊟ ALB
    - ALB-CD Reading Error
    - Wrong granularity of ALB
    - …
  - …

**Fig. 4.** Problem Taxonomy

Assume that an employee has problems in assuring the granularity. Then CAKE can find appropriate workflow definitions by the help of their characterizations. For the sake of brevity, the following example limits the latter to contain only two attributes: (1) problem description (an element of the Problem Taxonomy) and (2) an Integer type attribute describing how often the task has been executed for the same commune. In table 1, three characterizations for such workflow definitions (i.e. three cases) are given. The corresponding "solutions" are texts displayed to the employee. The words in brackets are representatives for real data: when applying the case, information agents retrieve the corresponding data from data bases already deployed at rjm business solutions GmbH and replace the representatives with the correct values.

| ID | Characterization | Solution |
|---|---|---|
| 1 | Problem=*ALB*<br>NumberOfTries=*[0,1]* | Contact <responsiblePerson> of commune <commune>. His telephone number is <telephoneNumber>. Ask him to re-send ALB data with correct granularity. |
| 2 | Problem=*ALB*<br>NumberOfTries=*[3,8]* | Escalate problem to <superiorContact>, phone number is <telephoneNumber>. Explain what <responsiblePerson> has already tried in order to solve the problem and ask for assistance. |
| 3 | Problem=*ALB-CD Reading Error*<br>NumberOfTries=*[0,∞]* | Contact <responsiblePerson> of commune <commune>. His telephone number is <telephoneNumber>. Ask him to re-send the CD because of CD reading errors. |

**Table 1.** Three Cases

Assume that the employee wants to start the acquisition of a new commune but finds that ALB does not match the corresponding ALK. His/her query would have the values Problem = “Wrong granularity of ALB” and NumberOfTries=0. By usage of a taxonomy similarity measure, Case 1 is retrieved with a similarity of 100% and the employee inserts it as a sub workflow within the workflow of the commune. Finally, he/she would call the responsible person as suggested. Two weeks later, a new CD is delivered, but ALK and ALB still won’t match. Again, Case 1 is found, inserted and executed. However, the problem persists, and his/her next query would have the values Problem = “Wrong granularity of ALB” and NumberOfTries=2. As there is no perfect matching case in the case base, CAKE would propose Case 1 and Case 2, both with a similarity of 90%. Depending on the current situation (e.g. the deadline for publishing the commune is coming soon), the employee would insert Case 1 or Case 2.

### 4.2 Support for time-critical processes for fire services

In order to tackle the envisaged scenario sketched in Section 2.2, CAKE supports the user agent in requesting information. First, CAKE assigns an initial workflow to the IC. This workflow captures the IC’s incoming query about “explosives in a van” in the context and starts a retrieval for an appropriate collaboration pattern that describes collaboration among information agents in order to answer this particular query. In this case, this pattern proposes to ask two different information agents that work on high quality structured data in terms of explosives in vehicles. If no relevant results can be found, one other information source is requested capturing more general and unstructured data about explosives. When the query cannot be satisfied again, an information agent is requested searching contact information of an appropriate human expert. This collaboration pattern is instantiated as sub workflow. Before the sub workflow is completed it passes the response as context information to the corresponding superior workflow. This response is sent back to the user agent, which transforms the result into natural language for reading it to the IC. Furthermore, the initial workflow starts a context analysis. E.g., if hazardous materials like explosives are found there is a demand to inform the fire control. Another sub workflow is started that pro-actively notifies fire control about explosives that have been found. Additionally, the response is also added to the control's notification.

### 4.3 Workflow Support for Agile Software Engineering

Software development using a Scrum-like agile method is supported by CAKE in the following way: The development team starts by instantiating a baseline workflow definition (long-term workflow) for the first iteration. The team adds "pre-game", "sprint" and "post-game" sub workflow definitions to it, with each triggering a CAKE sub workflow of their own in sequence. The pre-game workflow definition covers the negotiations with the customer like "meetings on site" or "writing recommendation reports". Using SCBR, workflows outlining best practice learned from previous projects or iterations can be identified to ease workflow modelling. The "sprint" workflow definition acts as the Scrum backlog: While executing the tasks included within pre-game, the team will add tasks to the sprint workflow required to actually produce the system. For instance, after having executed pre-game, the sprint workflow would include tasks like "design database schema" or "create input forms". For each task, risk is estimated, and by placing control flow relationships between the tasks, the most risky tasks will be executed first. The risk estimation itself is knowledge-intense, thus CAKE supports this as well: By using an information agent which connects to a bug database, information about the reliability of a specific component may be retrieved.

## 5 Related Work

Although the CAKE combines agent, workflow, and CBR technology in an innovative manner some parallels in state-of-the-art approaches can be found. Workflow technology is mostly assigned to CSCW technology because of being the core technology of systems, which manage activities to be done by different persons within organizations. Jennings et al. [14] propose an approach to use autonomous agents for business process management, which is similar to agent-based workflow management. Here, responsibilities are split for enacting various components of business processes to agents leading to more flexibility, agility, and adaptability. Instead of supporting agent-based workflow management CAKE enables agent enhanced workflows [15] where an agent can be assigned to a task for carrying out this task. Thereby, CAKE is able to provide agents that are conform to the established agent definition [14]. Beyond that, in CAKE non-autonomous services can be integrated as agents as well.

A further parallel to CAKE is the CBRFlow approach [16], which combines workflow technology with conversational CBR (CCBR) in order to cope with changing and unpredictable processes. The CBR technology is used for acquiring new knowledge when exceptions within processes are encountered. Combining CBR and modelling/planning is not new: the workbrain approach [17] uses CBR for workflow modelling prior enactment whereas the HICAP approach [18] make use of CCBR for acquiring information from users and for adapting plans based on alternative proposals. As parallel for CAKE's agent retrieval, a process-based knowledge management approach [19] incorporates a search for appropriate information sources backed by CBR retrieval and a CCBR approach [20] is available where agents are characterized by questions, which they are able to answer. This could lead to a cumbersome task when many questions express one agent's suitability.

## 6 Conclusion

In this paper, the CAKE system has been presented and how CBR techniques are used by it to create a generic collaboration support architecture. This architecture has been already proven useful within three different application domains, namely geographical information management, fire services, and software engineering, where there is a need to coordinate collaborative activities and to automatically provide the knowledge required to enact them. By using the concepts of information and user agents, the CAKE architecture provides unified access to knowledge available within the organization, integrating already established information systems seamlessly. Collaboration and coordination between agents are described explicitly by using adaptive workflows, allowing tailoring whenever required.

As of this writing, a first prototype of CAKE was already implemented and successfully applied in the three presented application scenarios. The next steps in these scenarios will be extensive evaluations to assess the benefit that can be achieved by the usage of the newly developed supporting systems. First impressions revealed weaknesses in the usability of these systems. Thus, focus will be put on UI enhancements and on improvements regarding long-term workflow support. Two additional application domains are currently evaluated: CAKE will be used to support the design of nanoelectronic chips, aiming to improve design efficiency and to reduce design errors, and it will be used in a medical context, where it will optimize the diagnosis and therapy processes for stroke patients.

Further research interests include aspects of workflow and collaboration pattern evolution. For instance, by capturing information on real-world workflow enactment, workflow definitions may be tailored automatically without obliging a user. The latter may be considered a workflow definition itself, and CAKE could make use of CBR to retrieve a well-working tailoring strategy.

## References


1. Grudin, J.: Computer-supported cooperative work: Its history and participation. IEEE Computer **27(5)** (1994) 19–26
2. Carstensen, P.H., Schmidt, K.: Computer supported cooperative work: New challenges to systems design. In Itoh, K., ed.: Handbook of Human Factors/Ergonomics, Asakura Publishing, Tokyo (2003) 619–636
3. Freßmann, A., Maximini, R., Sauer, T.: Towards collaborative agent-based knowledge support for time-critical and business-critical processes. In Althoff, K.D., Dengel, A., Bergmann, R., Nick, M., Roth-Berghofer, T., eds.: Professional Knowledge Management. Volume 3782 of LNAI., Kaiserslautern, Germany, Springer-Verlag (2005) 421–430
4. Freßmann, A., Maximini, K., Maximini, R., Sauer, T.: CBR-based execution and planning support for collaborative workflows. In: Workshop "Similarities - Processes - Workflows" on the 6th Int. Conference on Case-Based Reasoning (ICCBR 2005), Chicago, USA (2005)

5. Freßmann, A., Sauer, T., Bergmann, R.: Collaboration patterns for adaptive software engineering processes. In Czap, H., Unland, R., Branki, C., Tianfield, H., eds.: Self-Organization and Autonomic Informatics (I). Volume 135., Amsterdam, Netherlands, Frontiers in Artificial Intelligence and Applications (FAIA), IOS Press (2005) 304–312
6. Sauer, T., Maximini, K., Maximini, R., Bergmann, R.: Supporting collaborative business through integration of knowledge distribution and agile process management. In: Multikonferenz Wirtschaftsinformatik (MKWI 2006), Teilkonferenz "Collaborative Business". (2006)
7. Liu, K., Sun, L., Dix, A., Narasipuram, M.: Norm-based agency for designing collaborative information systems. Information Systems Journal **11** (2001) 229–247
8. Boehm, B.: Get ready for the agile methods, with care. IEEE Computer **35** (2002) 64–69
9. Bergmann, R., Breen, S., Göker, M., Manago, M., Wess, S.: Developing Industrial Case-Based Reasoning Applications: The INRECA Methodology. LNAI 1612. Springer (1999)
10. van Elst, L., Dignum, V., Abecker, A.: Towards agent-mediated knowledge management. In: Agent-Mediated Knowledge Management, International Symposium AMKM 2003, March 24-26, 2003, Revised and Invited Papers, Stanford CA, USA, Springer-Verlag (2004) 1–30
11. Bergmann, R.: Experience Management - Foundations, Development Methodology, and Internet-Based Applications. Volume 2432 of LNAI. Springer Berlin, Heidelberg, New York, Hong Kong, London, Milan, Paris, Tokyo (2002)
12. Heinl, P., Horn, S., Jablonski, S., Neeb, J., Stein, K., Teschke, M.: A comprehensive approach to flexibility in workflow management systems. In: Proceedings of the International Joint Conference on Work Activities Coordination and Collaboration. (1999) 79–88
13. Birkner, J., Ognyanova, M., Pütz, J., Rücker, A., Yao, Y.: Workflow-Unterstützung für Geoinformation. Technical report, University of Trier (2005) Endbericht.
14. Jennings, N.R., Norman, T.J., Faratin, P., O'Brien, P., Odger, B.: Autonomous agents for business process management. International Journal of Applied Artificial Intelligence **14(2)** (2000) 145–189
15. Judge, D., Odgers, B., Shepherdson, J., Cui, Z.: Agent enhanced workflows. BT Technology Jounal **16** (1998) 79–85
16. Weber, B., Wild, W.: Towards the agile management of business processes. In Althoff, K.D., Dengel, A., Bergmann, R., Nick, M., Roth-Berghofer, T., eds.: WM2005: Professional Knowledge Management Experiences and Visions, Kaiserslautern, Germany, German Research Center for Artificial Intelligence (DFKI GmbH) (2005) 375–382
17. Wargitsch, C., Wewers, T., Theisinger, F.: Workbrain: merging organizational memory and workflow management systems. In: Proceedings of KI'97: Workshop on Knowledge-Based Systems for Knowledge Management in Enterprises, Freiburg, Germany (1997)
18. Muñoz Avila, H., McFarlane, D.C., Aha, D.W., Breslow, L., Ballas, J.A., Nau, D.: Using guidelines to constrain interactive case-based HTN planning. In: Proceedings of the 3rd International Conference on Case-Based Reasoning (ICCBR 99), Springer (1999) 288–302
19. Holz, H.: Process-Based Knowledge Management Support for Software Engineering. PhD thesis, Department of Computer Science, University of Kaiserslautern (2003)
20. Giampapa, J.A., Sycara, K.: Conversational case-based planning for agent team coordination. In: Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR 2001). Volume 2080 of LNAI., Springer (2001) 189–203