Towards Collaborative Agent-based Knowledge Support for Agile Projects

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

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

Abstract. Using agile methodologies for enacting projects has been subject to intense research recently. In this paper, we present the Collaborative Agent-based Knowledge Engine approach for supporting mobile workers performing time-critical or business critical tasks. By a combination of sophisticated knowledge management and light-weight workflow model, this approach provides guidance and knowledge as required. Moreover, aspects for maintaining project history is discusses, as well as possibilities for integrating tools regarding computer-supported collaborative work already deployed in organizations.

1 Introduction

In recent years, agile development methodologies have been the subject of intense research [1]. Although agile methods have been put in opposition to more traditional approaches on the planning spectrum [2], agile projects still demand effective planning skills. As explained in [3], planning is required for arranging tasks effectively, i.e. in order to ensure that the agents involved are carrying out the most important tasks. Planning fulfills a documentation purpose as well, e.g. in order to make progress available to the project stakeholders.

However, planning and documentation is not sufficient to support projects including mission-critical or literally life-critical processes. While planning helps to identify *when* to perform these processes, agents working in knowledge-intense domains will require additional support for leveraging the tacit knowledge available in the organization in order to find out *how* to perform these processes effectively.

In addition, the lessons learned are valuable sources of information for future knowledge intense and creative tasks, which clearly includes management activities like designing an initial release plan. As pointed out in [4], this idea of iterative enhancement of the overall methodology is shared among agile approaches and more traditional planning methodologies found in domains like software engineering.

In this paper, we present the Collaborative Agent-based Knowledge Engine (CAKE) approach, which aims at supporting empirical processes [5], i.e. processes that are mostly unpredictable and unrepeatable. The CAKE concept provides an infrastructure

for constant measurement and control through intelligent and light-weight workflow modeling, leading to the idea of planning sketched above. Furthermore, knowledge-intense tasks are supported by sophisticated knowledge management, which allows to present context-dependent information to agents carrying out unknown or unexpectedly difficult tasks.

In the following section, we will discuss background architecture for CAKE, and the CAKE approach is presented in brief. Finally, related and adjacent research will be discussed in section 4.

2 Business and Time Critical Processes in the AMIRA Context

The presented approach of CAKE is developed domain independently but is motivated by the fire service domain within the AMIRA (Advanced Multi-modal Intelligence for Remote Assistance) project[3]. This domain addresses both business critical and time-critical situations for mobile workers. While wearing operational kits or gloves, accessing information written on paper or stored on laptops is very cumbersome. Hence, the envisaged mobile workers wear head-sets to access diagnoses support by speech. The fire services demand highly flexible processes and collaborative working in the field. Different representative processes are worked out: First, operatives encounter rare problems and want to perform questions to a system or experts. Second, they need pro-active information support for optimizing their collaboratively working procedures. Third, they require support in report activities and review procedures.

Single Person Request while collaboratively working. A fire fighter extinguishes a fire in collaboration with colleagues and encounters a cylinder with unknown abbreviations of chemicals. While collaboratively working he sends a request to the system for getting information about which chemicals are in the cylinders. The response is only sent to the single mobile worker and the headquarters. The others could get this information from the headquarters if necessary.

Pro-active Context-based Information Support. In time critical situations it is a demand for headquarters becoming aware of the activities of officers or fire fighters who work under the headquarters' control for making correct diagnoses. Hence, all interactions of the fire fighters with the system are monitored and logged by the system. Based on these logs context-based information is extracted, so that the headquarters are supported in getting corresponding guidelines, important information, and possible instructions for the mobile workers.

Collaborative A-posteriori Analysis. Collaborative a-posteriori analysis of the operations should be managed. This encompasses pro-actively asking the involved persons, headquarters and/or fire fighters, for information about their last actions concerning possible modifications to guidelines or other information used. Furthermore, it is possible to support methods for capturing information about the incident itself, e.g. in order to alleviate handover procedures. For achieving reliable information sources new or additional information is integrated into the databases.

[3] funded by the EU. Project partners are Kaidara, Fast Search & Transfer, DaimlerChrysler AG, and the Fire Service College

3 The CAKE Approach

Motivated by the AMIRA context a concept of the collaborative working system CAKE is developed that acts as moderator between several services (e.g. search engines) and user interfaces for providing their communication and context-based information support. For example, information may be retrieved from other search services, and the user interface contains a speech service for converting speech-based requests into machine-processable requests.

For coping with knowledge intensive tasks required for context-based information support CAKE comprises a workflow engine manager. Furthermore, an agent framework enables arbitrary access and communication to different agent-based services mediated by CAKE. At last, for providing highly flexible tasks of mobile workers the collaborative working system integrates a planning component for modifying workflows at runtime.

Workflow Engine Manager. The CAKE approach describes collaboration using *workflow definitions*. Each workflow definition consists of a set of tasks, as well as a control flow relationship between them. The latter allows to arrange the tasks in sequence, in parallel, or by using splits and branches, but does not cover data flow at all. The interchangeability of one task through a workflow definition enables nesting of workflow definitions resulting in reusing such definitions in different application scenarios.

A *task* is either a machine-executable program (e.g. “send notification”) or a descriptor for real-world processes (e.g. “write report”). Triggering a CAKE workflow definition is covered by the task definition, so hierarchical decomposition of a complex activity description like “design component” is achieved by following the way a human agent would solve this instead of following a fixed process model.

In order to enact a workflow definition, an *agent* has to be assigned. Agents may describe human actors, or information agents, with the latter being connections to information providers. Depending on the service used, information may be accessible read only or with write permissions: While a search engine like Google may be queried as an read-only source, a groupware calendar application deployed in an organization would be available for writing operations, too.

At runtime, the *Workflow Engine* (WE) initializes an instance of a suitable workflow definition. Separating these levels enables modifications on the workflow instances without doing changes on workflow definitions. Beyond controlling tasks the WE contains the local context assigned to a single workflow instance shortly denoted as workflow that 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 instances under control.

Agents Framework. From a more technical point of view agents are represented as a combination of a *technology component*, a *competence profile*, and a *wrapper*. The technology component enables the service provided by the agent (e.g. speech recognition, search, information delivery), while the competence profile includes characterizations about the agent’s competencies. The wrapper makes sure that communication to CAKE is based on a *unified data model*. Hence, wrappers act as interfaces between

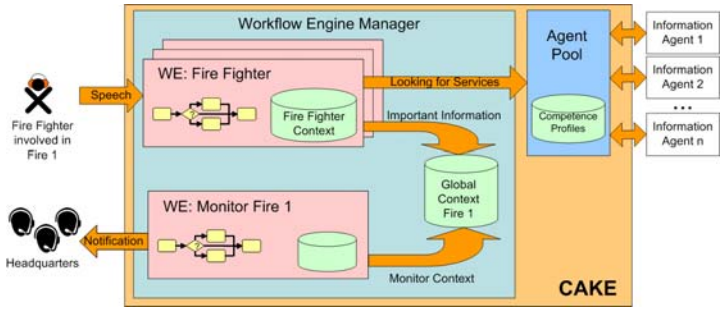

Fig. 1. The CAKE Approach

agents and CAKE for converting data. For being manageable, arbitrary agents are able to register in the *Agent Pool* by publishing their competence profiles. Due to dynamical registrations the Agent Pool works highly flexible in allocating agents. For instance, based on these competence profiles the most suitable information agent may be found for providing answers to requests performed by user agents.

Beyond simple mediations among agents, strategies based on best practice structures the schedule of agent communication. This information is stored in *collaboration patterns* which specify what to do when the agent firstly contacted is not able to support the user agent. Consequently, more than one information agent can be requested before sending responses back to user agents. The challenge is to develop strategies that are domain independent for preserving the universality of CAKE.

Pro-active and Context-based Information Support. By combining the workflow engine manager and the agent framework, CAKE enables pro-active and context-based information support of user agents. One workflow instance is assigned to one user agent, 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.

The underlying model for building contexts is a domain specific ontology. According to this ontology context-based data can be semantically interpreted and enables retrievals for context-based information. A crucial issue is to develop a quality threshold when having achieved enough context-based data for performing the retrieval. Nevertheless, user agents can be pro-actively supported by context-based information. Otherwise, user agents can get notifications about context-based information of other user agents as shown in Figure 1.

CBR-driven Planning Support. By providing weakly-structured workflows [6], CAKE offers the options of either modeling single workflow tasks in detail or to model abstractly, supporting late modeling as well. Combining late modeling and previously recorded planning activities leads to the idea of adaptive workflows supported by Case Based Reasoning (CBR) [7]. CBR technology enables a similarity-based retrieval by incorporating further experience: When proceeding to an abstractly planned task the

WE allows the corresponding user agent to retrieve a suitable workflow definition in a special workflow database. In that scope, ad-hoc planning is facilitated during runtime. Concerning the workflow retrieval the user agent specifies his or her intended further procedures, whereon CAKE activates a retrieval mechanism for searching the most similar available subworkflow based on experience. Further modifications on the subworkflow instance can be done by the user agent for adapting. The CBR-driven retrieval mechanism bases on similarity measures derived from the context-based data that annotates the workflow models. In particular, the domain ontology is fundamental for the similarity measure. Though it will be a challenge to develop the similarity measures in detail.

Furthermore, the planning skills controlled by workflow engine manager enable documentation support for user agents. By logging changes to workflow definitions and contexts, the engine enables user agents to query this data to create documentation as required.

4 Related Work

Providing workflow management support for agile methodologies has been discussed before [8]. This approach follows the idea of "heavy agile" [9] methods by suggesting to augment the specific methodology of Extreme Programming with additional documentation, formality, and tools, in order to support larger undertakings like distributed-team projects. However, because of limitation to a specific methodology, scope of application is limited.

In order to model workflows, various concepts have been proposed in the past decades, however none gained broad acceptance. Most rely on process description languages focusing on task dependencies. For instance, formal languages like MVP-L [10] have been designed specifically for expressing relationships between the various aspects of a software project, and to provide a formal execution model. Other efforts propose state and activity charts as means of workflow specification and execution [11]. While these efforts have their advantages to detect infeasible or suboptimal configurations, they require complex tool support, because of their rather non-intuitive model representation.

Workflow management systems have been discussed before as a valuable source of information for supporting knowledge-intense tasks [12]. In [13], the author suggests a framework for explicitly expressing information needs and souces, and how to support team members by proactively providing them.

Finally, deriving documentation by tracing workflow execution has been suggested in [14]. The idea is to look for patterns within a project history in order to create agile documentation [15] as required, and without obliging users to enter additional information.

5 Conclusion

Motivated by mission-critical or literally life-critical domains from the AMIRA project, the CAKE approach for coping with business and time-critical processes has been

devised. As described in this paper, CAKE rejoins several approaches for supporting application-driven scenarios based on requirements derived from two domains. Besides workflow knowledge and agent mediation, the CAKE approach aims at integrating planning and agile documentation, too. As of this writing, CAKE is getting implemented for the application domains of roadside assistance and fire services. In future work, further research will discuss other domains involving knowledge-intense and creative processes, like software engineering and the medical domain.

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

References

1. Abrahamson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile software development methods: Review and anaylsis. Number 478. VTT Publications (2002)
2. Boehm, B.: Get ready for the agile methods, with care. IEEE Computer **35** (2002) 64–69
3. Beck, K., Fowler, M.: Planning Extreme Programming. Addison-Wesley (2000)
4. Glazer, H.: Dispelling the process myth: Having a process does not mean sacrificing agility or creativity. Crosstalk: The Journal on Defense Software Engineering **14** (2001) 27–30
5. Advanced Development Methods, Inc.: Control chaos: Living on the edge. the origins of scrum, http://www.controlchaos.com (1996)
6. 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
7. Weber, B., Wild, W., Breu, R.: Cbrflow: Enabling adaptive workflow management through conversational case-based reasoning. In Funk, P., Calero, P.A.G., eds.: Advances in Case-Based Reasoning, Proceedings of 7th European Conference, ECCBR 2004. LNAI3155, Madrid, Spain, Springer Verlag, Berlin-Heidelberg (2004) 434–448
8. Maurer, F., Mertel, S.: Process support for distributed extreme programming teams. In: Proceedings of the International Workshop on Global Software Development (ICSE'02 GSD), Orlando, FL (2002)
9. Highsmith, J.: Agile Software Development Ecosystems. Addison-Wesley (2002)
10. Bröckers, A., Lott, C.M., Rombach, H.D., Verlage, M.: MVP-L language report version 2 (1997)
11. Wodtke, D., Weißenfels, J., Weikum, G., Dittrich, A.K., Muth, P.: The mentor workbench for enterprise-wide workflow management. In Peckham, J., ed.: Proceedings of the ACM SIGMOD International Conference on Management of Data, Tucson, AZ, ACM Press (1997) 576–579
12. Maus, H.: Workflow context as a means for intelligent information support. Lecture Notes in Computer Science **2116** (2001) 261–274
13. Holz, H.: An incremental approach to task-specific information delivery in se processes. In: Proceedings of the 18th IEEE International Conference on Automated Software Engineering. (1997) 295–298
14. Sauer, T.: Using design rationales as agile documentation. In: Proceedings of the 18th International Workshops on Enabeling Technologies: Infrastructures for collaborative enterprises, Linz, Austria, IEEE Computer Society Press (2003) 326–331
15. Ambler, S.: Agile documentation. Essay, http://www.agilemodeling.com/essays/ agileDocumentation.htm (2001)