

Weakly-structured Workflows for Knowledge-intensive Tasks: An Experimental Evaluation

L. van Elst, F.-R. Aschoff, A. Bernardi, H. Maus, S. Schwarz

German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
{elst|bernardi|maus|schwarz|aschoff}@dfki.uni-kl.de

Abstract

Knowledge-intensive activities can typically not be modeled sufficiently by classical, static process models and workflows. To enable a process-oriented knowledge management approach under these circumstances, the concept of weakly-structured workflows has been developed. This approach integrates process modeling and workflow enactment and facilitates active information support in dynamically changing process environments. Furthermore, the approach allows process knowledge itself to be used as a valuable resource in knowledge management.

In order to investigate whether this approach is viable and useful for supporting knowledge-intensive activities we designed and performed a comprehensive experimental evaluation. We outline the conceptual approach and realization in an agent-based software framework for organizational memories and describe the experimental settings. The results of the experiment demonstrate the viability of our key concepts.

1. Introduction

Business-process oriented Knowledge Management (BPOKM) is one of the major approaches to support acquisition, utilization and distribution of knowledge in modern enterprises. This approach employs explicit process models as a representation of the application context of any task in the enterprise. This context information is then used to annotate information items (thus structuring acquired information according to their generation context) and serves as a retrieval key (thus facilitating task-specific delivery of relevant information). Process models which are enacted in workflow systems can trigger information delivery on their own, thus realizing various forms of pro-activity in knowledge management systems.

Both process modeling and workflow systems have proved their value in a variety of settings: Explicit process models help in standardization and quality insurance; workflow systems support and guarantee appropriate performance in complex but repetitive tasks. Furthermore, process models are a prerequisite for successful reflection and improvement on business processes: Business Process

Re-engineering is a keystone for efficiency improvements in an enterprise.

While the benefits of process modeling and workflow are well known, their use to support knowledge management is still in its infancy. Process models as a means for information structuring gain increasing interest, but proactive information support by workflow integration is especially weak in domains where most of the benefits are to be inspected: when supporting knowledge-intensive activities.

The most important reason for this discrepancy is probably due to the characteristics of process models and workflows vs. knowledge-intensive activities: The former rely on the generation of a model a priori which is then enacted multiple times. Consequently a standardized, repetitive type of work processes is the optimal application area. In contrast to that, knowledge-intensive activities, like e.g. scientific work, are by their very nature explorative, non-repetitive in detail, and not completely known a priori. Static process models and classical workflows seem ill-suited to deal with this kind of work. Nevertheless we expect that knowledge-intensive activities will profit from process-oriented support: a suitable standardization and automatic information support can result in improved quality and efficiency of work, while the reflection on modeled work processes facilitates improvements on the process level.

Consequently, an approach is needed which realizes the benefits of explicit process models, workflow-type control and integrated information support for knowledge-intensive activities while coping with the particularities of this domain as described above.

In the research project FRODO (Framework for Distributed Organizational Memories) we developed a comprehensive platform for the realization of organizational memories which support knowledge-intensive activities. The framework realizes two main concepts:

- 1) an agent-based, distributed architecture with appropriate support for handling and merging of different ontologies; this allows to realize organizational memory systems in an evolutionary way by integrating individual solutions for different departments and workgroups under a common roof

- 2) the notion of weak workflows as an approach to handle incomplete process models and integrate modeling and enactment of workflows; this allows to retain the advantages of process-oriented knowledge support with a minimum of a-priori modeling workload or freedom restrictions for knowledge workers.

In the next chapter the core concepts of this approach are outlined. Chapter 3 presents a controlled experiment which has been designed in order to check the key assumptions of the weak workflow approach. The results of this experiment demonstrate the viability and usefulness of the approach. Summary and outlook conclude this paper.

2. Weakly-structured workflows in OMs

2.1. Conceptual Considerations

The overall goal of our concept of weakly-structured workflows in organizational memory information systems (OMIS) is to allow for a balance of potentially conflicting goals, originating in the demands of task-oriented information support on the one hand and the characteristics of knowledge work as well as the work practice of knowledge workers on the other hand. While the first's promise is "the more detailed and explicit the process model the better the information support", the latter objects "explicit modeling of knowledge-intensive processes doesn't pay off; it is expensive in terms of time and money; situations are so unique that knowledge transfer is impossible; and often explicit modeling is even principally not feasible". As one can certainly find examples which support the one or the other position in an extreme way, we suppose that there is a considerable amount of situations where it is worth arbitrating between them. From the above characterization of knowledge-intensive work (for a more detailed analysis and references to relevant literature, see [6]), we derived the following crucial design decisions for weakly-structured workflows:

- 1) **Lazy and late modeling are supported:** It should be possible to start with a partial model of the knowledge-intensive process and to possibly refine it later. The main reason for *lazy* modeling is an "economic" one. Often it is unknown in advance which degree of refinement of a model really pays off. Later, this might become much clearer. The same holds often for the detailed steps of a process which might get clear when actually working on it while others become irrelevant for the actual task.
- 2) **Modeling and execution of process-models is interleaved:** This point is tightly coupled to the first one. In order to be able to already use a partial model

it should be possible to work also on the process model while it is being executed instead of having modeling and execution in two distinct phases (like in traditional workflow approaches).

- 3) **Tasks can be dynamically and hierarchically decomposed/refined:** Hierarchical decomposition is a natural technique to practically implement the late/lazy modeling claim. It is typical for knowledge work that it is well-known on an abstract level that a specific subtask is necessary while the single steps of this subtask are unknown in advance. The hierarchical approach facilitates the specification of abstract elements at the time they are noticed while refinements can still seamlessly be integrated.
- 4) **A rich process logic allows for expressive process representations:** In knowledge-intensive tasks the execution sequence is often highly dependant from interim information processing steps. Therefore, a simple process representation (e.g., only "Task B follows Task A") may not be sufficient for anticipatory modeling. Beyond these standard elements of workflows, constraint-like definitions of dependencies ("Task B needs some information I before it can be started") and respective mechanisms to derive the process flow during runtime seem beneficial.

Though it seems clear to us that there is a strong demand for standardization of knowledge-intensive processes this type of work can certainly not be seen as "assembly line" working. The four above features should therefore be integrated into an *assistance system* where typically many elements of "what" and "when" are underspecified, leaving the knowledge worker freedom for individual work practice, adaptation and creativity.

In order to enable the desired *information support* the representation of tasks has to be extended by the description of respective information needs (cf. [7]). Such information needs may be more static information items or more dynamic. In the latter case, the actual information need may be a function of previously executed tasks in a process or it should be processed by a more complex information agent when the task is being executed (see, e.g., [8] for a discussion on the determination of an *actual information need*).

The approach presented here very much acknowledges that many parts of knowledge-intensive tasks might not pre-determined (e.g., by a business-process engineer). Thereby, the knowledge worker is not only in control of the process, but also seen as a source for the acquisition of potentially business-critical process knowledge (by features 1 & 3). In order to make this process knowledge available to the enterprise, the *process models* as well as the *instances* of these models (which might describe valuable experiences) become part of the organizational memory (OM), ready for reuse. Seeing the processes

themselves as knowledge containers allows to describe their life- and use cycle as a variant of the standard case-based reasoning (CBR) cycle ([4], [5]). Figure 1 depicts this case-oriented view: When a new business problem arises, an appropriate process model is retrieved and instantiated. The execution of the new instance embodies the reuse of the process knowledge as well as of the knowledge contained in the attached information needs. Late/lazy modeling and hierarchical refinement are typical revision steps. Due to the hierarchical process structure other (partial) models might be inserted by additional retrieval steps. Furthermore, instances of similar processes can be used as additional knowledge sources. After a process has been executed it is added to the OM. Here, probably a more complex maintenance procedure is needed (cf. [5]).

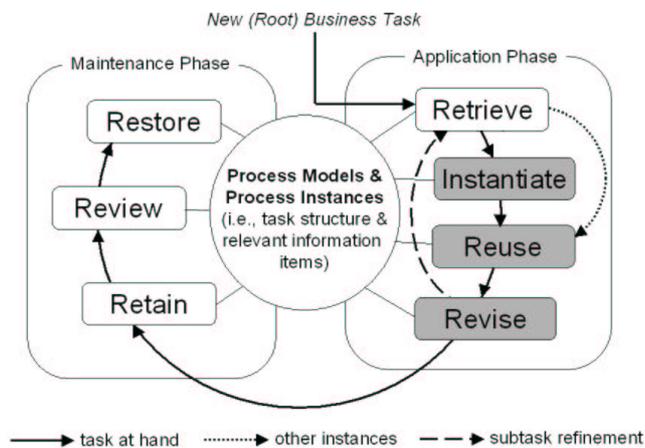


Figure 1. CBR-oriented lifecycle of weakly-structured workflows in FRODO

In general, this leads to a tight integration of knowledge management and CBR (see [1] for an overview). [3] argue that this kind of case-oriented workflow configuration a double-loop learning with an inner cycle in the application phase and an outer maintenance cycle is well suited for organizational learning, one of the major KM goals.

2.2. Implementation in the FRODO platform

In this paper we can only give a rough sketch of the implementation principles of the FRODO platform. The principal coupling of a workflow system with information management has already been realized in the KnowMore project [10], the prototype demonstrated pro-active knowledge delivery based on explicitly modeled information needs which rely on the current process instance as context information; the delivery is triggered by the workflow system. FRODO has extended this approach towards a distributed architecture which supports integrated modeling and enactment of weak workflows as a basis for information support for

knowledge-intensive activities. The FRODO TaskMan system (cf. figure 2) is the central component for user interaction. This system integrates a worklist handler (presenting user-specific views on relevant tasks and their activity states) with a process presentation and modification tool (allowing for dynamic modification of the workflow) and an integrated information browser (facilitating the description of information needs and pro-actively presenting relevant information). The implementation relies on the agent paradigm: the individual worklists, the task instances, the complete workflow, and the information access components are implemented as autonomous agents with well-defined communication protocols. The various OM structuring elements (e.g., organizational roles, domain ontologies, information sources) are modeled in formal ontologies using the well-known Protégé tool; a wrapper encapsulates these models and acts as an ontology agent for the complete system.

Consequently, the system is transparent with respect to any level of distribution: Model modifications, task instantiations, information handling results in appropriate messages being sent across the agent community, and whoever is concerned by such activities receives the necessary messages. As the technological basis for this implementation we used the JADE platform; to cope with the particularities of distributed, role-oriented OM integration we realized our agents with a specialized social layer which allows the definition of appropriate rights and responsibilities to control the inter-OM cooperation. The OM-specific agents can be freely configured based on formal descriptions of their roles and functionalities. Wherever possible we employed web-enabled representation and communication technology, like e.g. RDF as central representation formalism and the http protocol for inter-agent exchange. More implementation details can be seen e.g. in [11]

3. Experimental Evaluation

While the assessment of several building blocks of IT support for KM is quite standardized (e.g., precision/recall of IR or usability studies for GUI), more complex assumptions are rarely tested in a controlled way (cf. [2]). As a complete experimental evaluation of comprehensive frameworks for Organizational Memories is not feasible we decided to concentrate on three aspects which should be investigated in a laboratory experiment: i) weakly-structured workflows are a useful basis for support of knowledge-intensive activities, ii) integration of process execution and information support is accepted and considered of benefit, iii) process-embedded information is a means of knowledge sharing and transfer.

3.1. Method

In order to experimentally test these conjectures we conducted an experiment where subjects had to perform a knowledge-intensive task (“visit planning”) with the FRODO-TaskMan. The chosen domain appeared suitable due to four conceptual and pragmatic characteristics: i) The task of visit planning includes room for flexible interpretation by the subjects. ii) It is directed towards a specific result, namely a schedule. iii) The web can be used as a rich information base. iv) The task can be understood and processed in reasonably short time.

The two main factors in the experimental design were the flexibility mode (strict vs. weakly-structured workflow) and information support (information needs given vs. not given). For practical reasons, we didn’t test the whole process lifecycle, but concentrated on the main application steps (see shaded boxes in figure 1).

3.1.1. Subjects. 25 students from different departments of the University of Kaiserslautern participated in the experiment. There were no special prerequisites with respect to prior knowledge. The students received 25€ for their participation.

3.1.2. Apparatus. Each subject worked on a web-connected PC equipped with an adapted version of the FRODO-TaskMan (see figure 2). The adaptation restricted on the one hand the functionality of the TaskMan to those features related to the experiment. On the other hand it allowed to easily switch between an operation mode *with weakly-structured workflow support* vs. a mode which allowed *only strict workflows*, i.e., without late/lazy-modeling facilities. The graphical user interface consisted of three main areas. On the left, a graphical representation of the workflow (tasks with their relations and status) can be used to navigate and edit the workflow. The right pane handles the information items attached to the selected task. Relevant information (e.g., webpages, keywords, memos, concepts of a domain ontology) can be explored and added to the task. The functionality of this area is tightly integrated with the Microsoft InternetExplorer and with Google as a simple “search agent”. In particular, information from the InternetExplorer could be simply added to a task by drag&drop. Search behavior of the subjects was recorded via the InternetExplorer’s history functionality. The third area in the GUI is used to switch the various working modes (work on a task, edit the workflow, etc.)

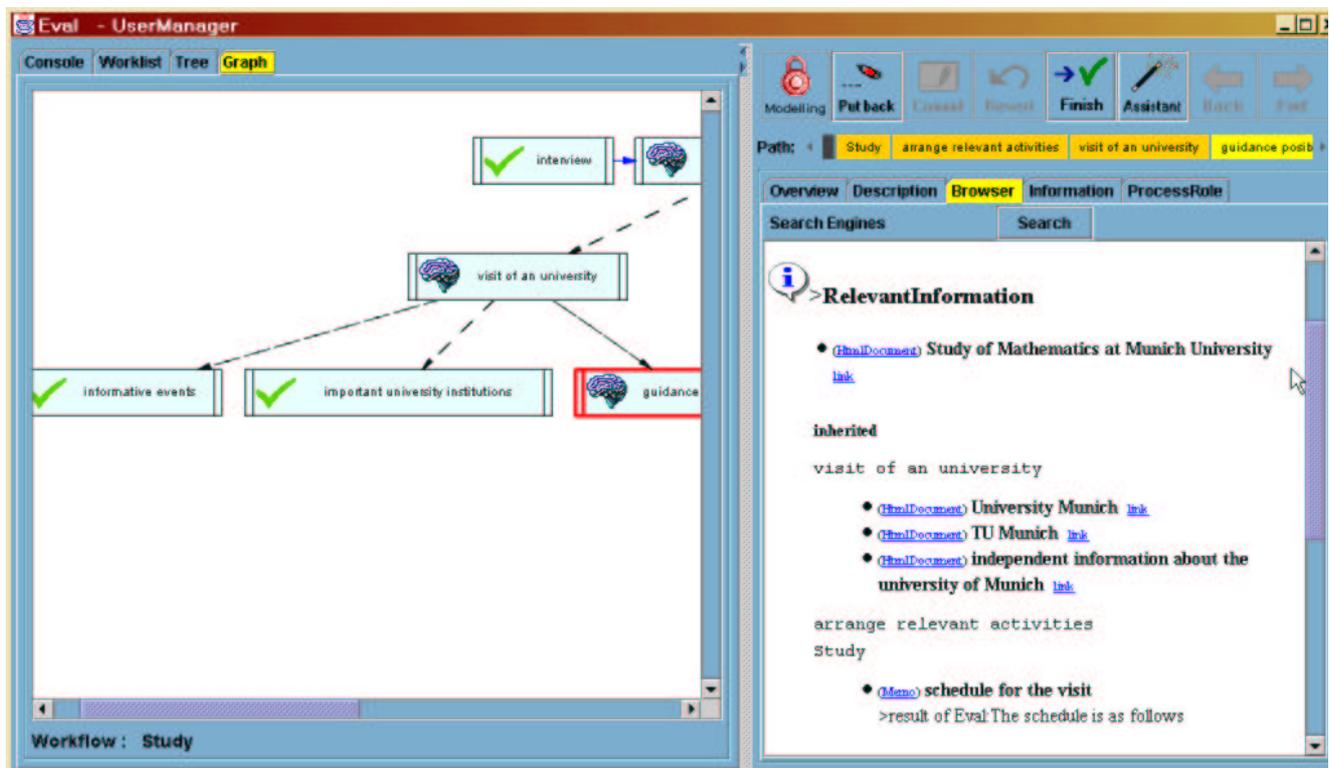


Figure 2. Main FRODO-TaskMan GUI

3.1.3. Materials. Besides instructions concerning the concept of workflows and handling of the TaskMan the materials consisted of *scenario descriptions* and *work instructions* for the subjects. The scenarios were structured as follows: A foreign person is going to visit a German university with a certain objective. The visit is limited to three days. Therefore, the university wants to provide the visitor with an informative schedule. The following is an example text translated from the original German material:

“Please imagine a French university professor who is planning to attend a conference in Saarbrücken. Following the conference he has three days off for sightseeing in and around Saarbrücken. Your task is to compile a 3-day agenda for the professor. Please use the internet to investigate elements for the agenda. The result of your work should be a realistic schedule for the three days which might actually be executed by the professor.”

A structurally equivalent text was designed for a second domain where a foreign prospective student wants to visit Munich to plan her studies. In the course of the experiment the subjects also were provided with initial workflows for the visit planning task. These models were structurally similar and specified basically a three step procedure: i) Analyze requirements of the visitor. ii) search for relevant information. iii) Commit and prepare plan. For the professor domain, we had initial models on two levels of complexity (12 vs. 29 subtasks).

3.1.4. Procedure. The subjects were randomly assigned to one of two groups. Each group performed three phases: In a 1-hour practice phase, the subjects had to *model* a workflow for the visit planning process. So, they got acquainted with the concept of traditional workflows as well with the software environment. In two 1-hour work phases the subjects had to execute the visit planning process for the two domains (university professor and

prospective student). Here, we provided the subjects with standardized models that were similar to those models the subjects created in the practice phase. Figure 3 shows how the different experimental conditions were allocated to the groups and phases.

While due to resource restrictions we couldn't balance all experimental factors in a full design, the approach has the advantage of reflecting a major part of a workflow lifecycle, from initial modeling to execution and reuse in latter instances. In the second work phase, the subjects have been provided with additional requirements for their task (e.g., “The professor is accompanied by his spouse. She's interested in wellness facilities.”) in order to test the handling of external changes. At the end, the subjects had to answer a questionnaire to gain direct measurements with respect to the hypotheses. The processed workflow models and the explorer history were recorded as indirect measurements.

3.2. Results

3.2.1. Questionnaire. 76 percent of the participants felt more flexibly supported when they were able to solve their tasks with the weak workflow concept. A chi²-analysis calculated for these statements was highly significant ($\chi^2(2, N = 17) = 14.588, p = .001$). Only 18 percent felt more flexible while working with the classical a-priori strict workflow concept, 6 percent did not recognize any difference. The opposite question led to respective results: 65 percent of the participants felt more restricted, when they had to solve their task the “classical a-priori strictly modelled” workflow concepts. A chi² - Analysis calculated for these statements was significant ($\chi^2(2, N = 17) = 7.529, p = .023$). (18% - weak workflow, 18% no difference). Participants were asked in which part of the experiment they had to invest unnecessary effort to solve their tasks. 71 percent did not recognize any difference between the weakly-structured workflow concept and the classical workflow, even though some lazy/late modeling took place. Only 18 percent indicated that they had to invest more unnecessary effort while working with the weak workflow concept (12% for the strict case). A chi² - Analysis calculated for these statements was significant ($\chi^2(2, N = 17) = 10.706, p = .005$). 59% of the participants reported that in general they felt better supported when they solved their tasks with the weak workflow concept. ($\chi^2(2, N = 17) = 5.765, p = .056$). 29% felt better supported during the phase with the classical workflow concept. Some (2) explained this preference with the clearer and easier to understand structure of the classical workflow task and did not refer to the difference between the weak workflow concept and the classical concept. 94% of the participants (16) considered the possibility to add tasks during run-time as

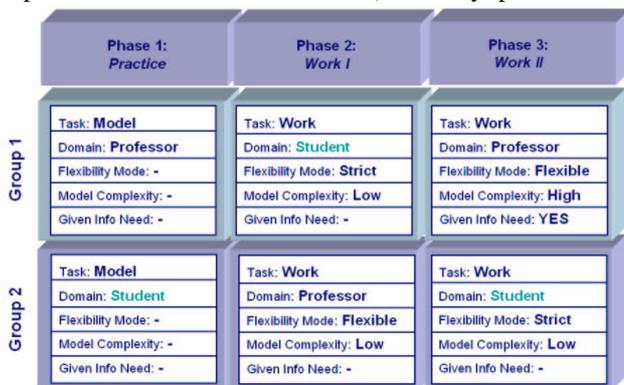


Figure 3. Experimental design, comprising the model-work-refine phases of a workflow lifecycle

helpful or wrote positive remarks in the free remarks field. One person remarked that the possibility to erase tasks was not good. Only one person stated that she did not use this feature. Nobody found it inconvenient. Nobody stated that he did not recognize this feature.

3.2.2. Indirect Measurements. When subjects had the possibility to dynamically change the workflow-structure, late/lazy modeling was used extensively. By adding tasks participants enlarged the given workflow model on average by 19%. Since deleting tasks was also possible the workflow model was on the other hand diminished by averagely 12%, resulting in a net increase of 6%.

As could have been expected participants added more tasks to the given model with low complexity (22% in comparison to 17% added to the high-complexity model) and deleted considerably more tasks from the model with high complexity (16% in comparison to only 4% deleted from the low-complex model). The model with low complexity had a net increase of 18% whereas the model with high complexity only had a net increase of 1% percent. This might indicate that the high-complexity model is somehow saturated.

One of the central quality aspects for the utility of process-triggered information support is the precision of the assignment of information needs to tasks. Table 1 shows the mean difference of information items per task after the two work phases for both domains.

Table 1. Difference information items per task before/after processing a workflow

Condition	Mean Δ Information Items per Task		
	Static	Dynamic	Total
Strict Workflow			
LowComplexity, Phase2, Student	1.64	0.3	1.94
LowComplexity, Phase3, Student	1.42	0.46	1.88
Weak Workflow			
LowComplexity, Phase2, Professor	0.88	0.41	1.28
HighComplexity, Phase3, Professor	1.04	0.44	1.49

Static information items are URLs and text memos linked to a task, dynamic information items are keywords and concepts from a domain ontology. The latter have one step of indirection to retrieve the actual information. The results show that much more static items are attached than dynamic. In the strict workflow condition we had a total of 1.9 added information items per task, in the weakly-structured workflow condition 1.5. Due to the relatively small numbers of subjects per condition, we couldn't

expect this difference to be statistically significant. Moreover, there was quite a high variance as there were some tasks especially on the more abstract levels where nearly no information items have been added and several more concrete subtasks with many items attached. However, a survey of the models showed that – in line with the data in table 1 - the lower information item per task ratio in the weakly-structured workflow groups resulted in a higher precision of the task-information structure. This holds especially for those information items which are related to the “change of requirements” situation in the second work phase. Here, the weakly-structured WF group typically inserted a new subtask with respective information needs while the strict group just added the information needs with a low precision of classification.

Concerning the use of proactive information support an analysis of web access logs showed that about 30% of all information access resulted from pre-given information needs, reflecting re-use of earlier knowledge.

3.3. Discussion

From the results presented we can clearly derive that the subjects appreciated the flexibility of weakly-structured workflows. The possibility of lazy/late modeling and interleaved modeling and execution of workflows seems very important for this kind of knowledge-intensive work. An interesting observation is that the subjects more acknowledged the benefits of flexibility after they lost them than the other way around, having flexibility as an additional feature after working with strict workflows. It could also be demonstrated that proactive information support is useful while not completely blocking creativity of the knowledge worker. The experiment showed the advantage of the flexible concepts especially in situations of changing requirements. Furthermore, the results indicate that the precision of proactive information support is higher in the flexible case. Of course one has to be careful with respect to the outer validity of the results. However, we think that the scenarios have been designed in a way that has high potential for transferability to real world situations of knowledge-intensive work. The experimental design has certainly drawbacks with respect to statistical coverage for some aspects related to the indirect measurements. In return, we gained valuable insights into many aspects of our approach. Altogether we can say that the conceptual building blocks of IT-supported KM are very well supported by the evaluation.

4. Summary & Outlook

In this paper, we presented the concept of weakly-structured workflows for process-oriented Knowledge Management support. We briefly described the FRODO platform as an implementation of the approach and reported design and results of an experimental evaluation which demonstrates the usefulness of our basic claims.

From the characterization of our approach as a kind of CBR (cf. figure 1) with reuse and continuous adaptation (and hopefully improvement) of knowledge-intensive processes in an organization, we can also derive next research steps. Especially the very important retrieval step and the maintenance steps have to further be elaborated. *Retrieval* is vitally from the user's point of view due to direct implications for practicability and hence acceptance. In FRODO we proposed a task concept ontology [12] as an index for weakly-structured workflows. In the EPOS project we will investigate support of process retrieval by analyzing knowledge workers' interactions with their standard applications. The *review step* is the basis for the overall quality of an Organizational Memory with weakly-structured workflows as facilitator of organizational learning. Here, abstraction from and integration of individual workflow adaptations might be supported by machine learning and data mining techniques.

Often, research in IT for Knowledge Management is rather positivistic; technology and architectures are being proposed, sometimes their scopes and limits are exemplified by single case studies. Experimental evaluations like the one reported here complement the repertoire of assessment techniques and has the potential to significantly contribute to scientific progress in this important area.

Acknowledgements. The FRODO project was supported by the German Ministry for Education and Research (Grant 01 IW 901). We would like to thank Andreas Lauer who implemented major parts of the FRODO platform.

10. References

[1] Aha, D., Becerra-Fernandez, I., Mauer, F., and Munoz-Avila, H., *Exploring Synergies of Knowledge Management and Case-Based Reasoning – Papers from the AAAI Workshop*, AAAI Press, Menlo Park, 1999.

[2] Aschoff, F.-R., and van Elst, L., "Empirical Evaluations of Organizational Memory Information Systems: A Literature Overview", *unpublished DFKI Discussion paper*, 2001. available at <http://serv-4100.dfki.uni-kl.de:8000>

[/~docbase/dokana/WWW/D00000606.pdf](#)

[3] Wargitsch, C., Wewers, T., Theisinger, F., WorkBrain: "Merging Organizational Memory and Workflow Management Systems", in: *Workshop "Knowledge-Based Systems for Knowledge Management in Enterprises"*, 21. KI-Jahrestagung '97, Freiburg, Germany, 1997.

[4] Aamodt, A., and Plaza, E., "Case-based reasoning: Foundational issues, methodological variations, and system approaches", *AI Communications*, 7(1), March 1994.

[5] Reinartz, Th., Iglezakis, I., & Roth-Berghofer, Th., "Review and Restore for Case-Base Maintenance". *Computational Intelligence - An International Journal*, Volume 17 (2), May, 2001.

[6] Schwarz, S., Abecker, A., Maus, H., Sintek, M., "Anforderungen an die Workflow-Unterstützung für wissensintensive Geschäftsprozesse". Workshop "Geschäftsprozessorientiertes Wissensmanagement", 1. Konferenz Professionelles Wissensmanagement (WM2001), 14. - 16. März, Baden-Baden, 2001. (in german), Extended abstract in: G. Stumme, H.-P. Schnurr, St. Staab, R. Studer, Y. Sure (Hrsg.) *Professionelles Wissensmanagement*, Shaker Verlag.

[7] Holz, H., Process-Based Knowledge Management Support for Software Engineering, *Doctoral Dissertation, University of Kaiserslautern, 2003 (to appear)*.

[8] van Elst, L., and Abecker, A., "Integrating Task, Role, and User Modeling in Organizational Memories". In: I. Russell and J. Kolen (Eds.), *FLAIRS-2001, The 14 Int. FLAIRS Conference*, pp. 295-99, AAAI Press. Key West, Florida, USA, May 21-23, 2001.

[9] Maurer, F., Holz, H., "Process-Centered Knowledge Organization for Software Engineering", In: [1], 1999.

[10] Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., and Sintek, M., "Context-Aware, Proactive Delivery of Task-Specific Knowledge: The KnowMore Project", *Int. Journal on Information Systems Frontiers*, 2(3/4):139-162, 2000.

[11] Abecker, A., Bernardi, A., van Elst, L., Lauer, A., Maus, H., Schwarz, S., and Sintek, M., "FRODO: A Framework for Distributed Organizational Memories. Milestone M1: Requirements Analysis and System Architecture", *DFKI Document D-01-01*, DFKI GmbH, March 2001.

[12] Schwarz, S., "Task-Konzepte: Struktur und Semantik für Workflows" (in German), To appear in: 2. *Konferenz Professionelles Wissensmanagement, Workshop „Wissensmanagement im Kontext der Modellierung und Ausführung wissensintensiver und schwach strukturierter Geschäftsprozesse“*, Luzern, 2003.