An Agent-based Framework for Distributed Organizational Memories

Ludger van Elst, Andreas Abecker[1], Ansgar Bernardi,
Andreas Lauer, Heiko Maus, Sven Schwarz

Knowledge Management Department
German Research Center for Artificial Intelligence
DFKI Kaiserslautern, Germany
elst@dfki.de

Abstract: Typical arguments for the use of agent technology in business information systems rely on the heterogeneity of the information landscape, the physical distribution of its components, and the overall complexity of such systems. In the domain of IT support for Knowledge Management, these claims for agent systems typically also hold, but additionally are complemented by characteristics from KM information landscapes that go beyond the purely physical distribution of knowledge sources, like aspects of trust, responsibility, contextuality of knowledge, and others. In this paper, we elaborate on such requirements on IT for Knowledge Management. We present the FRODO framework for distributed organizational memories (DOM) which can be described as a meta-information system with multiple ontology-based structures and a workflow-based context representation. We exemplify the use of socially-enabled agents for balancing individual and organizational concerns in Organizational Memory Information Systems by sketching the agent societies for ontology management, workflow management, and personal information assistance.

1 Introduction

Knowledge Management (KM) is defined as a systematic, holistic approach for sustainably improving the handling of knowledge on all levels of an organization (individual, group, organizational, and inter-organizational) in order to support the organization's business goals, such as innovation, quality, and cost effectiveness (c.f. [Ep02]). KM is primarily a *management discipline* combining methods from human resource management, strategic planning, change management, and organizational behavior. However, the role of *information technology* as an enabling

---

[1] Andreas Abecker is now with the Forschungszentrum Informatik, Karlsruhe, Germany (FZI).

factor is also widely recognized, and a variety of proposals exist showing how to support KM with specialized information systems (e.g., [AE03]).

Often, Information Technology (IT) research for KM focused on the *comprehensive use* of an organization's knowledge, thus aiming at the completeness of distribution of relevant information. Technically, this is typically supported by centralized approaches: knowledge about people, knowledge about processes, and domain knowledge are represented and maintained as information in global repositories which serve as sources to meet a knowledge worker's (potentially complex) information needs. Such repositories may be structured by global ontologies and made accessible, e.g., through knowledge portals [SM00, MS+01]. Or they may be rather "flat" and accessed via shallow (i.e., not knowledge-based) methods like statistics-based information retrieval or collaborative filtering. (This is the typical approach of today's commercial KM tools.)

In the following, we present some KM *characteristics* which, in our opinion, account for serious drawbacks of such centralized IT approaches to KM, and which can immediately be coined into *requirements* for a powerful KM system design:

R1 *KM has to respect the distributed nature of knowledge in organizations*. The division of labor in modern companies leads to a distribution of expertise, problem solving capabilities, and responsibilities. While specialization is certainly a main factor for the productivity of today's companies, its consequence is that both *generation* and *use* of knowledge are not evenly spread within the organization. This leads to high demands on KM:

- Departments, groups, and individual experts develop their particular views on given subjects. These views are motivated and justified by the particularities of the actual work, goals, and situation. Obtaining a single, globally agreed-upon vocabulary (or ontology) within a level of detail which is sufficient for all participants may incur high costs (e.g., for negotiation) and is considered hard work. A KM system should therefore allow balance between (a) *global* knowledge which might have or constitute a shared context, but may also be relatively expensive; and (b) *local* expertise which might represent knowledge that is not easily shareable or is not worth sharing.

- As global views cannot always be reached, a KM system has to be able to handle context switches of knowledge assets, e.g., by providing explicit procedures for capturing the context during knowledge acquisition and for recontextualizing during knowledge support. An example for context capturing is a lessons-learned system which is fed by debriefings after a project is finished [HSK96, HSK98]. Here, a typical question pair is: "What was the most crucial point of the project's success? What are the characteristics of projects where this point may also occur?"

Altogether, we see that distributedness of knowledge in an organizational memory is not a "bug", but rather a "feature", which is by far not only a matter of physical or technical location of some file. It has also manifold logical and content-oriented aspects that in turn lead to derived aspects such as – in an ideal system – the need to deal with matters of

- *trust* (Do I believe in my neighbor's knowledge?),
- *responsibility* (Is my neighbor obliged to maintain his knowledge base because I might use it? And am I obliged to point out errors that I find in his knowledge base?),
- *acknowledgement* (Who gets the reward if I succeed with my neighbor's knowledge?),
- *contextuality* of knowledge (Is my neighbor's knowledge still valid and applicable in my house and my family?),
- ... and many others.

R2 *There is an inherent goal dichotomy between business processes and KM processes*. For companies as a whole, as well as for the individual knowledge worker, KM processes do not directly serve the operational business goals, but are *second order processes*. Within an environment of bounded resources, knowledge workers will always concentrate on their first-order business processes. This means they optimize their operational goals locally and invest only very little to fulfill strategic, global KM goals. It is clear and pretty well accepted that *having and using* knowledge is important for optimally fulfilling first-order tasks. In face of day-to-day workload and time pressure, however, KM activities like *knowledge conservation*, *evolution*, *organization*, etc., are considered as second-order processes and therefore often neglected in practice. Even the most basic activities for knowledge search and reuse are often considered to be unacceptable. Therefore, the KM processes should be embedded in the worker's first-order processes, and proactive tools should minimize the cognitive load for KM tasks.

R3 *Knowledge work, as well as KM in general, is "wicked problem solving"* (cf. [Bu97, CoWP, DJB97]). This means that a precise a-priori description of how to execute a task or solve the problem doesn't exist, and consequently, it cannot be stated in advance when or what knowledge should be captured, distributed, or used optimally. An optimal solution for KM problems and the respective knowledge and information flows cannot be prescribed entirely from start to finish, because goals may change or be adapted with each step of working on a task. Therefore knowledge workers and KM systems must be flexible enough to adapt to additional insights and to proactively take opportunities when they arise during work. Solving "wicked problems" is typically a fundamental social process. A KM system should therefore support the necessary complex interactions and underlying, relatively sophisticated processes like planning, coordination, and negotiation of knowledge activities.

R4 *KM has to deal with changing environments*. In addition to the intrinsic problems described above, KM systems typically reside in environments which are subject to frequent changes, be it in the organizational structure, business processes, or IT infrastructure. Centralized solutions are often ill-suited to deal with continuous modifications in the enterprise, e.g., because the maintenance costs for detailed models and ontologies simply get too high. Furthermore, the implementation of KM systems often follows a more evolutionary approach where functionalities are not implemented "in one step" for a whole company, but partial solutions are deployed to clearly separated substructures. In order to obtain a comprehensive system, these elements then have to be integrated under a common ceiling without depreciating their individual values.

Keeping these requirements in mind, let's have a look at scenarios which are considered to be rewarding tasks for agent-based software solutions. [Pa98] lists a number of characteristics (but similar arguments can be found in many books about multi-agent systems) typically indicating that a scenario could be a good application area for agent technology: agents are best suited to applications that are modular, decentralized, changeable, ill-structured, and complex. Although the match between these five salient features and the KM requirements R1–R4 listed above is already obvious, we want to elaborate a bit more explicitly on this match. Let us start with the *weak definition* of agents in [WJ95] with the definitional features *autonomy*, *social ability*, *reactive behavior*, and *proactive behavior*. Now we will see why agent-based approaches are especially well-suited to support KM with information technology.

In the first place, the notion of agents can be seen as a natural metaphor to model KM environments which can be conceived as consisting of a number of interacting entities (individuals, groups, IT, etc.) that constitute a potentially complex organizational structure (see R1, but also R4). Reflecting this in an agent-based architecture may help to maintain integrity of the existing organizational structure and the autonomy of its subparts. Autonomy and social ability of the single agents are the basic means to achieve this. Reactivity and proactivity of agents help to cope with the flexibility needed to deal with the "wicked" nature of KM tasks (see R3). The resulting complex interactions with the related actors in the KM landscape and the environment can be supported and modeled by the complex social skills with which agents can be endowed. Proactiveness as well as autonomy help with accommodating to the reality that knowledge workers typically do not adopt KM goals with a high priority (see R2). Regarding primarily the software technology aspects of agents, they represent a way of incorporating legacy systems into modern distributed information systems: Wrapping a legacy system with an agent will provide the legacy system with a clear interface for distributed interaction. Furthermore, agent approaches allow for extensibility and openness in situations where it is impossible to know at design time exactly which components and uses the system will have. Both arguments reflect fairly well the technical consequences of abstract requirements such as R4 and R3 (changing environments demand continuous reconfiguration, the unpredictable nature of wicked-problem solving require flexible approaches), R2 (competition between operational work and KM meta work call for stepwise deployment and highly integrated KM solutions), or R1 (pre-existing local solutions must be confederated).

There are numerous approaches on the basis of agent technologies which tackle various aspects of KM (see [EDA04] for a comprehensive overview). In this paper we choose Distributed Organizational Memories (DOM) as an example to assess the potential of agent technology for KM because their characteristics comprehensively reveal the requirements R1–R4. The following section briefly introduces i) the concept of DOM and ii) with the FRODO project[2] a framework for such DOM that was – from system modeling approach to the technical implementation – designed entirely on the basis of agent technology. Section 3 then exemplifies the use of agents for DOM in more detail.

2 Distributed Organizational Memories (DOM)

Organizational Memory Information Systems, **Organizational Memories** or OMs for short, support the effective handling, conservation, and use of knowledge across time and space and – as far as possible – in person-independent ways. An OM comprises a variety of information sources where information elements of all kinds, structures, contents, and media types are available. The OM has to control and access these information sources in accordance with the users' information needs, which are determined by a combination of personal, organizational and contextual circumstances. The useful interaction with the OM is influenced not only by the current task at hand, but also by the individual's role in the organization, his personal skills and interest profiles (and their overlap with the requirements of the current activity), and his prior knowledge and experience.

The internal structure of a single OM (see Fig. 2.1) reflects this principle: by representing explicit interconnections between information elements and formalized models (particularly the domain, the enterprise, and the work context) the content of the information elements is partially made available to *automatic processing and reasoning.* As the various models form a basis for common reference across an enterprise, ranging from lists of shared vocabulary to more detailed ontological representations, *common and shared understanding* is supported by this approach. An explicit modeling of business processes as a means for context representation facilitates the situation-specific markup and retrieval of information elements; the integration with workflow systems which enact the process models enables *proactive* and *context-sensitive* information services. Consequently, an OM is best described as a meta-information system with tight integration into enterprise business processes, and which relies on appropriate formal models and ontologies as a

[2] http://www.dfki.de/frodo/

basis for common understanding and automatic processing capabilities [AB+98, AB+00].

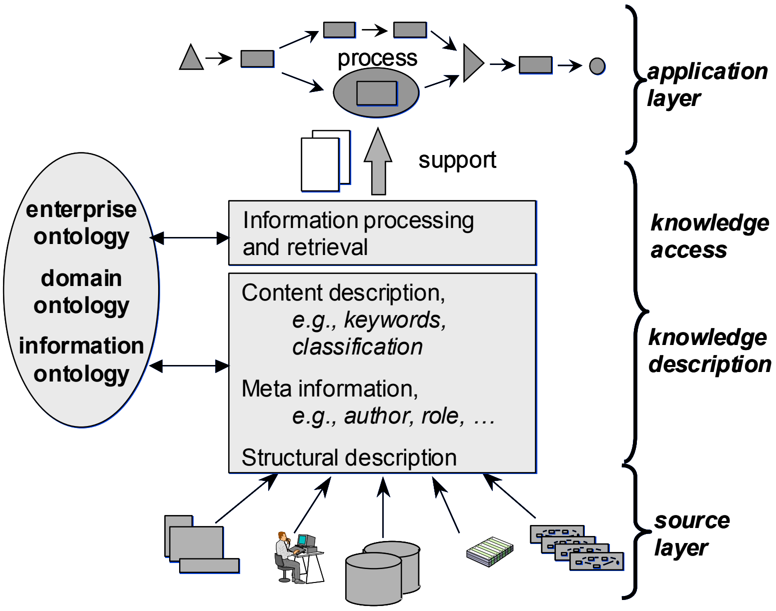

Fig. 2.1: Single OM as a meta-information system

Having considered the analysis in Section 1, we abandoned the concept of one centralized Organizational Memory with global ontologies as shown in Fig. 2.1 and developed the notion of *Distributed Organizational Memories (DOM)*, aiming at a better balance between the needs of smaller units in an organization and the more global KM concerns. The main idea is not to have a global OM (with global processes, ontologies, etc.) for the whole enterprise, but to limit the scope of one OM to a more homogeneous unit (e.g., a group or a department[3]) and to facilitate interoperations between these single OMs. These interoperations can be related to all four levels of the single OM architecture (see Fig. 2.2):

- *Source level*: Here, elements of information sources are shared among different Organizational Memories. Quite complex synchronization procedures may be needed when changes in the source of one OM may affect the application level of another OM.

- *Knowledge description level*: Cooperation among Organizational Memories on the knowledge description level is typically provided by ontology

[3] It is important to note that merging of companies or internal restructuring will usually result in the need of new cooperation and knowledge exchange across formerly independent units; thus requesting such modular approaches.

integration or ontology mapping services. Often this is the catalyst for cooperation on higher levels – i.e., to allow for communication among intelligent information agents.

- *Knowledge access*: If two Organizational Memories cooperate on the knowledge access level, they provide each other complex services, e.g., in the form of information agents. Mechanisms for cooperative information gathering or intelligent information integration are examples for OM integration on the knowledge brokering level. Normally, such services presuppose ontology integration or mediation services.

- *Application level*: There are at least two situations where integration on the application level is needed: i) There is one process that crosses the boundaries of an OM. Typical examples are globally operating companies or virtual enterprises where each site deploys one Organizational Memory. Here cross-organizational workflows are an approach for facilitating integration on the application level. ii) Second-order processes are considered. For example, dedicated knowledge management activities can be realized as second-order processes. Such processes would complement the particular local optimizations of the different Organizational Memories by a superordinate view that allows for *global* optimization.

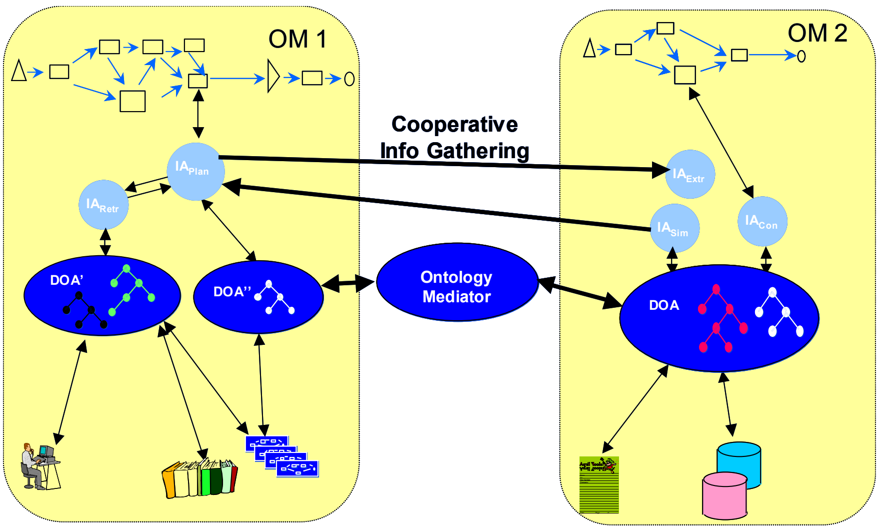

Fig. 2.2: Co-operation between two OMs on the knowledge description and knowledge access layers

In the FRODO project, we have implemented a prototypical agent-based framework for Distributed Organizational Memories. Fig. 2.3 shows the layer architecture we have built on top of the JADE [BPR01] platform:

- The *platform abstraction layer* hides some technical details of the concretely chosen agent platform, refines the FIPA message handling, and provides a comfortable startup component for bootstrapping complete DOM environments.

- The *reactive behavior layer* enables a declarative definition of agent competencies and realizes the mapping of these competencies onto concrete agent behavior. The concrete agent behavior was then implemented as JAVA code or on the basis of more logic-based inferencing engines like SiLRi [DB+98], JESS [Fr03], or TRIPLE [SD02]. This layer also supports the flexible definition of interaction protocols and realizes an agent-internal event listening concept in order to allow for some asynchronicity in the interaction protocols.

- The *social layer* is a quite extensive augmentation to the standard JADE platform. It enables agents to enter into social contracts on the basis of rights and obligations. Rights define conditions under which agents are explicitly entitled to exert a competency or to demand a service from another agent. Obligations define conditions under which an agent has to perform some action. The social layer manages the handling of rights and obligations within an agent and the embedding thereof in its action cycle.

- The *specialist layer* provides classes of agents that implement Knowledge Management functionality (information agents, domain ontology agents, workflow agents, etc.). Examples for these agent classes will be given in Section 3.

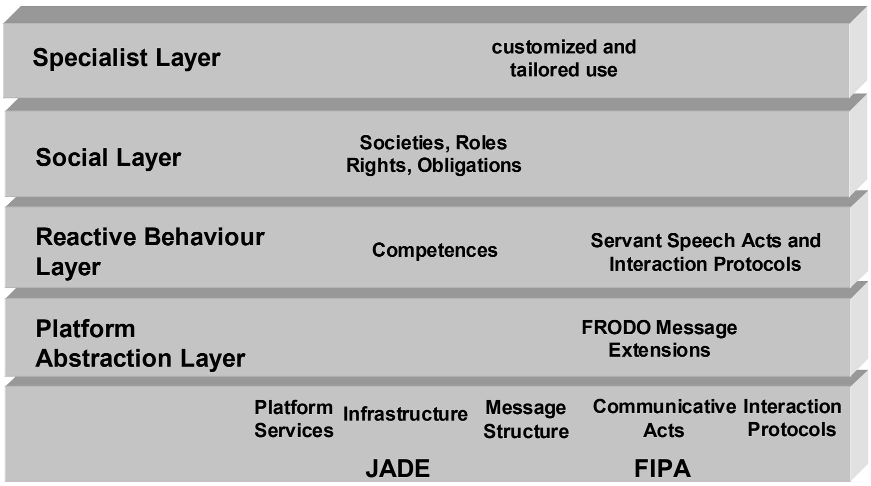

Fig. 2.3: Software Layers of the FRODO Framework for Distributed OM

The most noticeable element of this architecture is certainly the social layer. In Section 3 we will show that for a fully agent-based realization of the DOM scenario a huge number of agents with possibly divergent goals (because of their autonomy) and maybe highly complex communication and negotiation threads are required. As discussed in detail by [SF+02], optimal work distribution and collaborative performance in such a group of agent benefits not only from *task delegation* and *knowledge exchange*, but also from *social delegation.* The latter forms the basis for dynamic self-organization of agent societies, in order to achieve optimal group performance, yet staying flexible enough to cope with changing requirements. Via social delegation, groups of agents constitute *Agent Societies* with less communication effort because of clear responsibilities, better task distribution because of specialization, etc. The phenomenons of society creation and self--organization is considered a crucial point for the introduction of agents into Enterprise Information Systems [Ca00]. They complement the mechanisms for bottom-up control (system behavior emerges from goals and negotiation at the micro level), which are inherent to the agent paradigm, by new mechanisms which appropriately reflect the global directives to be propagated top-down in a stable organization.

In FRODO, we build a DOM as a set of collaborating societies of socially-enabled agents. We define an *Agent Society* as a set of agents[4] with at least one manager agent (which administers membership, role assignments, etc.) which enact for a certain time one or more *Agent Roles* with respect to this society. The roles themselves are defined by sets of rights and obligations. This means that entering an agent society in a specific role is accomplished by negotiating the respective rights and obligations with the society manager. For details, see [Vi02; EA02]. In the next section we briefly sketch agent (sub-)societies required for building a DOM.

3 Agent Societies for Distributed Organizational Memories

For each of the four layers of the OM framework in Fig. 2.1, there exist several approaches and prototypes that rely on agent technology (e.g., [Ke97]; for an overview see [EDA04]). As discussing the agent societies for all of these levels would be beyond the scope of this paper, we focus on the following key aspects:

- For the *knowledge description level*, we elaborate on distributed ontology management as a central task to realize an encompassing knowledge description across individual components.
- For the *application level*, we present the society-oriented architecture of FRODO's agent-based workflow system and personal user agents. Together, they realize the pro-active coupling of information handling and application processes.

[4] An agent can be member of several societies at the same time.

3.1 Ontology Management

As indicated in Fig. 2.1 and explained in more detail by, e.g., [AB+00; DFH02], the future's corporate-internal and -external information systems will rely too much greater extent than today on ontologies as shared, formalized accounts of domain knowledge structures. Both philosophical and pragmatic reasons suggest that such (typically distributed) ontology-based systems will not keep only one, globally accepted, central ontology, but that different, partially autonomous sites and user groups will maintain their own ontologies, which must interoperate for intelligent information services (cf. [Co98; AB+01]). Since ontologies are *defined* as formal accounts of knowledge *generally agreed upon between a group of actors*, and since their use is typically to exploit *different* information sources and to process their content in an *integrated* manner, it is obvious that creation, maintenance, and use of ontologies should also be understood as a joint effort of several software agents representing the different stakeholders in these processes.

Hence we started our analysis of DOM implementations with the design of the agent society of ontology creators and users described in detail in, e.g., [EA02]. To sum up, we present there the role of the D²OA (*Distributed Domain Ontology Agent*) which mediates between different agent societies holding their own specific domain Ontologies (see Fig. 2.2). These separate societies are represented and managed by their DOAs (*Domain Ontology Agents*) which keep their generally agreed-upon vocabulary, provide an interface to outside the society, are obliged to gather and process update suggestions possibly submitted by ontology users, and are also obliged to broadcast ontology changes or extensions to the actual ontology users as well as to associated D²OAs.

Ontology users' agents can be separated in several groups according to the amount of commitments they enter with respect to ontology use and further developments, as well as to the level of ontology services they want to utilize. For instance, all roles belonging to the group of "*active users*" may have the right to receive update notifications, whereas "*passive users*" may be excluded from regular update services because they are typically palmtop users which only seldom synchronize with the agent network.

3.2 Workflow Management

Since workflow applications are distributed by nature and often, in particular in the case of cross-organizational workflows, are aiming at goals such as reliability, scalability, and efficient load distribution in complex networks, the adequacy of agent technology is fairly obvious (cf. [Pa00]). One of the most prominent agent-based workflow systems has been described by [JF+00]. There, the idea of competencies is built in by the concept of *agencies* which represent specific departments of the company responsible and able to do specific tasks or sub-processes. Internally, such agencies exhibit a master–slave architecture which can be understood as a fixed, hard-wired way of implementing specific rights and obligations.

[YS99] come already closer to our ideas: they show the appropriateness of *role-based* workflow analysis, where roles are defined as a set of rights and obligations. Then they map elementary roles to agent types in their system implementation which negotiate task assignment. Although this approach is much more rigid than ours (where users and resources are represented as agents with *temporarily* assigned roles with respect to a given process instance), the *major* difference is that we propose (like [St01]) to represent tasks as agents, too. In this way, all relevant entities in the real world are represented by software agents, which allows maximum flexibility and scalability. *Task instance agents* gather the resources they need for their execution, and they can, together with a user agent, refine or change their task-specific control flow, thus achieving a maximum level of user control.

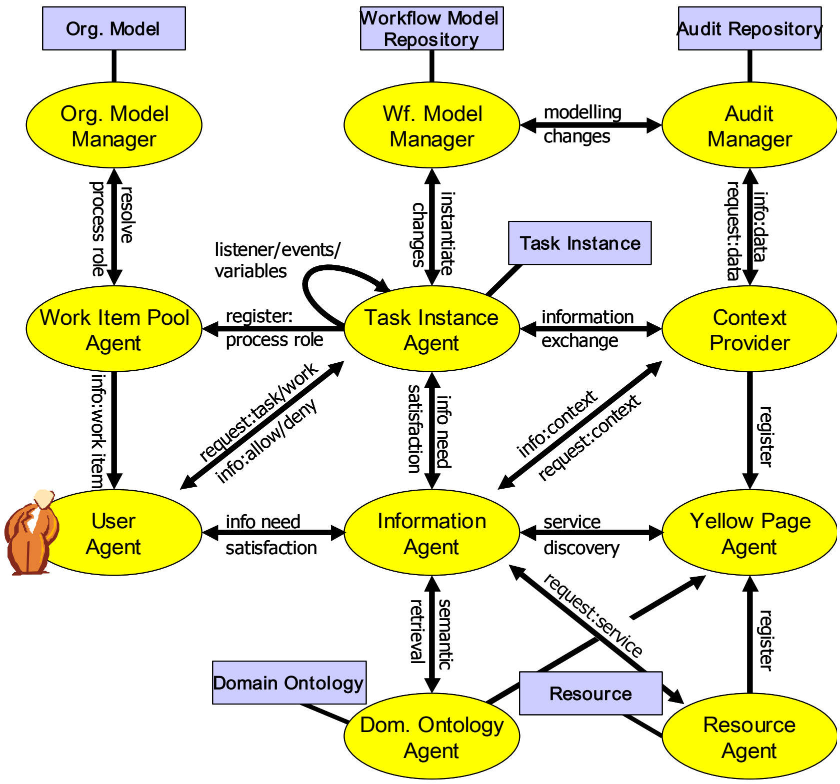

Fig. 3.1: Agent-Role Collaboration in FRODO's workflow system

These task agents are just one class of agents in FRODO's workflow agent society. Fig. 3.1 shows the roles in our agent-based, weakly-structured workflow system with context-sensitive knowledge delivery with some speech acts. We shortly summarize these roles (for a detailed discussion, see [AB+01]): The *model manager* is the access point for starting new workflow instances holding the actual workflow definitions, as well as possible alternatives for specific sub-tasks. The *audit manager* keeps track of all past workflow instantiations, both for documentation purposes and to allow for supporting the learning abilities of the system. Task instance agents belong to an open workflow instance and want to successfully complete a given task by acquiring the necessary user and electronic resources. *Resource agents* offer specific services or represent electronic system resources (like specific software programs) which may be employed for achieving some workflow goal. The *yellow page agent* allows for service and resource publishing and discovery.

Interesting examples for rights and obligations can be found, e.g., at the level of workflow models. In the spirit of a flexible workflow system, task instance agents may, together with the user agent representing the end user's interface actions, change their task model on how to achieve a given goal by an alternative procedure. After the completion of a task they have the obligation to send their execution trace to the audit manager. The model manager has the right to request from the audit manager all workflow traces in the last period of time, and may have the obligation to record and report to possibly open affected workflow instances all interesting changes in the way most users currently enact a given task.

3.3 Personal Information Assistance

Both the ontology management approach discussed above and the workflow integration aim at pro-actively supporting the user with context-specific, potentially useful, and understandable information. Beyond the realization of a shared understanding via a society of ontology agents, and in addition to the context services which are offered by the agent-based workflow, various aspects of a DOM are well-suited for agent-based modeling and realization. In this section we illustrate in particular the concept of Personal User Agents as a means to allow integrated user communication and personalized services, some principal benefits of agent approaches to Information Processing in the DOM setting, and the use of agents to realize appropriate transitions between informal information sources and more formal representations.

Since the advent of software agents, personal information agents and personal assistants for information access and management have been studied. Such *Personal User Agents* (PUA) provide a unique point of access for all system services, offering, e.g., the tasks actually assigned to the user by the workflow system (see Fig. 3.1), as well as an overview of information the system is constantly searching on behalf of the user and according to his permanent information needs.

Besides offering a comprehensive interface, however, our agent approach facilitates the realization and delegation of more active services. As described in, e.g., [AB+98, AB+00] a comprehensive system like ours can and does pro-actively deliver currently relevant information and knowledge to help the user efficiently perform his current task at hand. We understand an actual information need as a function of personal, role, and task-specific information requirements, interests, and preferences [EAM01], which means that specifically useful information and knowledge can be found by taking into account both the short and long-term user work context and his global and local dynamic task context. The use of context for refined information services is mentioned again below and is described in more detail in [Ma01]. A Personal User Agent is an adequate reflection of such individual- and situation-specific information needs and allows their automatic and autonomous satisfaction. A typical PUA might possess the necessary rights and obligations to represent and interact with the user in a complex society. The PUA might have the right to schedule meetings for the user, or to negotiate with task agents about acceptance or rejection of some work item. It might observe the obligation to comprise notifications about important dates or appointments, and about relevant information, or the provision of task-specific support knowledge. Regarding flexible workflow execution, the PUA has the obligation to show to the user all tasks to be executed, and the right to request a change of the task model in reaction to some user GUI activity for changing the way of working on this task.

In the advanced evolution stages of such a system, PUAs might have the right (or even the obligation) to establish alliances between groups of PUAs in order to make, for instance, information search more efficient by exchanging individual search strategies or query feedback and compiling it in group-relevant knowledge, as it is done in *Collaborative Filtering*.

4 Summary and Outlook

With the advent of the networked economy, virtual enterprises, and ubiquitous computing, it is clear that we need new computing and software design paradigms to cope with the huge complexity of software systems and application problems in tomorrow's Enterprise Information Systems. The general *Organizational Memory* architecture shown in Fig. 2.1 was the basis for numerous research and several successful application projects. The logical next step is to proceed to the *Distributed OM* (DOM) approach roughly sketched in this paper.

As briefly discussed in Section 3, all relevant areas to be addresses in such a system have already been tackled with agent technology with promising results. Our main message in this paper is that all these approaches must be combined in a homogeneous design and implementation approach in order to fully exploit the synergy potential and to allow for new ideas which are not possible if one considers these areas in an isolated manner. However, building such integrated systems introduces a new level of complexity into software design and implementation. In order to deal with this complexity, but also from the analysis of general KM characteristics like physical and logical distributedness of knowledge (Section 1), we introduced the notion of *Socially-Enabled Agents* and the concept of *Agent Societies* defining *Roles with Rights and Obligations*. These notions allow a most adequate modeling of the components and interactions in DOM systems.

In the area of agent-based workflow, role-based modeling proved already to be a useful system analysis paradigm for mapping the processes occurring in the real world. We are currently extending this approach to the whole scope of DOMs and are preparing the software basis for implementing such systems with the same mechanisms used for system analysis. Besides the basic framework, our FRODO project presented here focused on the realization of distributed ontology management and agent-based, weakly-structured workflow solutions. An experimental evaluation of the workflow concept for supporting knowledge work was performed and is described in [EA+03]. The evaluation shows the helpfulness of the weak workflow concept and the task-embedded information support in an OM scenario. In the successor project EPOS[5], we concentrate on leveraging for organizational use the efforts a knowledge worker spends on his “personal” knowledge management. By investigating the role of single knowledge workspaces as the providers and consumers of knowledge in a DOM scenario, we thus intend to offer a feasible solution to the goal dichotomy identified as requirement R2.

At the moment it is hard to argue (and indeed not the point of this paper) that agent-based systems can do things in KM that could not also be done using conventional technology, especially when only the implementation level is considered. However, we believe that agent technology helps building KM systems faster and more flexibly. The work in the FRODO project, but also many other research on Agent-Mediated Knowledge Management [EDA04] strengthened our hope that an agent-oriented view (not only with respect to implementation technology, but also to organizational analysis and system modeling) leads to a more human-centered, more agile, and more scalable KM support.

Acknowledgement

The work described here has been supported by the German Ministry for Research under grant 01 IW 901 FRODO and 01 IW C01 EPOS. We also thank our three anonymous reviewers. Their detailed remarks were extremely valuable for preparing the final version of this paper.

---

[5] http://www.dfki.de/epos/

Literature

[AB+98] Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., and Sintek, M.: Toward a technology for organizational memories. IEEE Intelligent Systems, 13(3):40–48, 1998

[AB+00] Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., and Sintek, M.: Context-aware, proactive delivery of task-specific knowledge: The KnowMore project. Int. J. on Information Systems Frontiers, 2(3/4):139–162, 2000.

[AB+01] 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. Document D-01-01, DFKI, Kaiserslautern, 2001.

[AE03] Abecker, A., and van Elst, L.: Ontologies for knowledge management. In St. Staab and R. Studer, (Eds.): Handbook on Ontologies in Information Systems, pp. 453–474, Springer, 2003

[BPR01] Bellifemine, F., Poggi, A., and Rimassa, G.: Developing multi agent systems with a FIPA-compliant agent framework. Software – Practice And Experience, 31:103–128, 2001

[Bu97] Buckingham Shum, S: Representing hard-to-formalise, contextualised, multidisciplinary, organisational knowledge. In Proc. AAAI Spring Symposium on Artificial Intelligence in Knowledge Management, Stanford University, AAAI Press, 1997

[Ca00] Castelfranchi, C.: Engineering Social Order. In: A. Omicini, R. Tolksdorf, F. Zambonelli (eds.), Engineering Societies in the Agent World, 1st Int. Workshop, LNCS 1972, Springer, 2000.

[Co98] Colomb, R.M.: A digital library needs many indexes. 1998. http://www.itee.uq.edu.au/~colomb/Phronesis/

[CoWP] Conklin, E. J., and Weil, W.: Wicked problems: Naming the pain in organizations. White Paper of Group Decision Support Systems, Inc.

[DJB97] Davenport, T.H., Jarvenpaa, S.L., and Beers, M.C. Improving knowledge work processes. Sloan Management Review, 37(4):53–65, 1997.

[DFH02] Davies, J., Fensel, D., and van Harmelen, F. (eds.). Towards the semantic web: ontology-driven knowledge management. Jon Wiley & Sons, 2002.

[DB+98] Decker, S., Brickley, D., Saarela, J., and Angele, J.: A Query and Inference Service for RDF. In: QL'98 – The Query Languages Workshop, W3C Workshop.

[EAM01] Elst, L. van, Abecker, A., and Maus, H.. Exploiting user and process context for knowledge management systems. In: Workshop on User Modelling for Context-Aware Applications at UM-2001, Sonthofen, Germany, 2001.

[EA02] Elst, L. van, and Abecker, A.. Ontologies for information management: balancing formality, stability, and sharing scope. Expert Systems with Applications, 23(4), pp. 357–366. Elsevier Science, Amsterdam, 2002.

[EA+03] Elst, L. van, Aschoff, F.-R. , Bernardi, A., Maus, H., and Schwarz, S.: Weakly-structured Workflows for Knowledge-intensive Tasks: An Experimental Evaluation. In: Proc. 12th IEEE Int. Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2003), pp. 340–345, IEEE Press.

[EDA04] Elst, L. van, Dignum, V., and Abecker, A. (Eds.): Proc. of the AAAI Spring Symposium on Agent-Mediated Knowledge Management AMKM-03, Springer LNAI 2926, 2004.

[Ep02] Eppler, M. Glossary definition: Knowledge management, 2002 at NetAcademy: www.knowledgemedia.org/modules/glossary/viewdef.php?id=mediamanagement-20

[Fr03] Friedman Hill, E.: Jess in Action: Java Rule-Based Systems, Manning Publications, 2003.
[HSK96] van Heijst, G., van der Spek, R., and Kruizinga, E.. Organizing corporate memories. In R. Dieng and J. Vanwelkenhuysen, (Eds.) 10th Knowledge Acquisition for Knowledge-Based Systems Workshop KAW'96, November 1996.
[HSK98] van Heijst, G., van der Spek, R., and Kruizinga, E.: The lessons learned cycle. In U. M. Borghoff and R. Pareschi. Information Technology for Knowledge Management. Springer, 1998.
[JF+00] Jennings, N., R., Faratin, P., Norman, T.J., O'Brien, P., and Odgers, B.: Autonomous agents for business process management. Int. Journal of Applied Artificial Intelligence 14(2):145–189, 2000.
[Ke97] Kerschberg, L.: The role of intelligent software agents in advanced information systems. Invited Paper for the 15th British National Conference on Databases, Birkbeck College, University of London, 1997.
[MS+01] Maedche, A., Staab, S., Stojanovic, N., Studer, R., and Sure, Y.: SEAL – A framework for developing SEmantic Web PortALs. Springer LNCS 2097:1–22, 2001.
[Ma01] Maus, H.: Workflow context as a means for intelligent information support. In: V. Akman, P. Bouquet, R. Thomason, R.A. Young (eds.): Modeling and Using Context. 3rd Int. and Interdisciplinary Conference. Springer LNAI 2116, 2001
[Pa00] Pang, G.: Implementation of an agent-based business process. Diploma Thesis. Universität Zürich, 2000
[Pa98] Parunak, P. H. Van Dyke, 1998. What can agents do in industry, and why? An overview of industrially oriented R&D at CEC. Springer LNCS 1435:1–18.
[SF+02] Schillo, M., Fley, B., Florian, M., Hillebrandt, F., and Hinck, D.: Self-organization in multiagent systems. In 3rd Int. Workshop on Modelling Artificial Societies and Hybrid Organizations (MASHO), 2002.
[SD02] Sintek M., and Decker, S.: TRIPLE – A Query, Inference, and Transformation Language for the Semantic Web. Int. Semantic Web Conference (ISWC), 2002.
[SM00] Staab, S., and Maedche, A. Knowledge portals: Ontologies at work. The AI Magazine, 22(2):63–75, 2000.
[St01] Stormer, H.: A flexible agent-workflow system. In: Workshop on Agent-Based Approaches to B2B. at the 5th Int. Conf. on Autonomous Agents (AGENTS), 2001.
[Vi02] Vicinus, R.: Soziale Agenten für das Management von Domänen-Ontologien in verteilten Organizational Memories. Diploma Thesis, Universität Kaiserslautern, 2002.
[WJ95] Wooldridge, M., and Jennings, N. R.: Intelligent Agents: Theory and Practice. The Knowledge Engineering Review, 10(2):115–152, 1995.
[YS99] Yu, L. and Schmid, B.F.: A conceptual framework for agent-oriented and role-based workflow modeling. In G. Wagner and E. Yu (eds.), 1st Int. Workshop. on Agent-Oriented Information Systems, 1999.