AGENT TECHNOLOGY FOR DISTRIBUTED ORGANIZATIONAL MEMORIES

The Frodo Project

Andreas Abecker, Ansgar Bernardi, Ludger van Elst
German Research Center for Artificial Intelligence (DFKI) – Knowledge Management Department,
Erwin-Schroedinger-Str, Bldg. 57, D-67663 Kaiserslautern, Germany
Email: {aabecker|bernardi|elst}@dfki.uni-kl.de

Keywords: Agent-Mediated Knowledge Management, Organizational Memory, Distributed Organizational Memory, Socially-Enabled Agents

Abstract: Comprehensive approaches to knowledge management in modern enterprises are confronted with scenarios which are heterogeneous, distributed, and dynamic by nature. Pro-active satisfaction of information needs across intra-organizational boundaries requires dynamic negotiation of shared understanding and adaptive handling of changing and ad-hoc task contexts. In this paper, we present the notion of a Distributed Organizational Memory (DOM) as a meta-information system with multiple ontology-based structures and a workflow-based context representation. We argue that agent technology offers the software basis which is necessary to realize DOM systems. We sketch a comprehensive Framework for Distributed Organizational Memories which enables the implementation of scalable DOM solutions and supports the principles of agent-mediated knowledge management.

1 DISTRIBUTED ORGANIZATIONAL MEMORIES

Knowledge Management envisions the comprehensive use of an enterprise's knowledge, whoever acquired it, wherever it is stored and however it is formulated in particular. *Organizational Memory Information Systems* - shortly **Organizational Memories**, OMs - shall 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 by the actual 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), as well as by prior knowledge and experience.

The internal structure of an OM 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 modelling of business processes as a means for context representation facilitates the situation-specific mark-up and retrieval of information elements; the integration with workflow systems which enact the process models enables *pro-active* information services. Consequently, an OM is best described as a meta-information system with tight integration into enterprise business processes, which relies on appropriate formal models and ontologies as a basis for common understanding and automatic processing capabilities (Figure 1; Abecker et al., 1998 & 2000).

This description seems to motivate a central approach, and in fact a number of OM systems were realized as *central repositories* with globally valid ontologies and structures. However, *centralized* OM approaches have drawbacks with respect to two important aspects:

a) Knowledge generation and use in an enterprise is *distributed by nature*. Departments, groups and individual experts develop individual, differing 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 on a level of detail which is sufficient for all participants is very expensive or even outright impossible. Consequently, an OM should benefit from balancing both *local expertise* – which might represent knowledge which is not easily shareable on a global level–and *overall views* on a more global level. A strictly centralized approach neglects this opportunity.

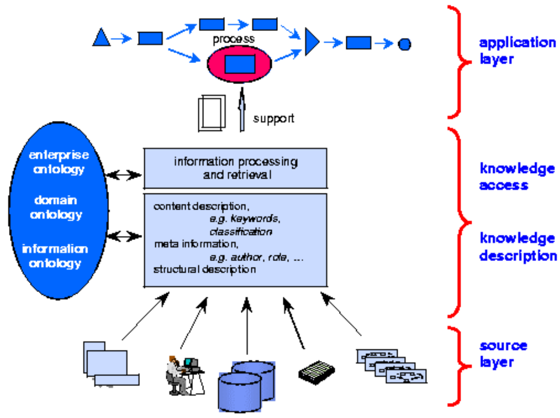

Figure 1: OM as a Meta-Information System

b) Knowledge resides in *changing* environments. A centralized OM is ill-suited to deal with continuous modifications in the enterprise: The maintenance costs for its detailed models and ontologies simply get too high. Furthermore, centralized OMs assume a strict sequence of design, implementation, and use, while in reality a more evolutionary approach seems more promising: OM-like structures evolve in different groups and departments, using appropriate formalizations and conceptualizations. Integrating these elements under a common roof without disturbing their individual value should result in solutions which offer common benefit with reduced efforts while reaching better acceptance on the individual level.

The reality of enterprises' environments thus asks for a *distributed* approach to OM realization: Distributed, heterogeneous OM cells let local expertise prevail while striving for maximal integrated benefit. Evolutionary growth and scalability on all levels is reached by allowing individual OM cells to grow and mature independently while interaction and communication brings enterprise-wide exchange and understanding.

The natural approach for building complex software representations of distributed scenarios is agent technology. In the following, we will outline the characteristics of software agents which are helpful for **Distributed Organizational Memories** (DOM). We will argue that a comprehensive framework for DOMs requests the notion of agent societies. Further, an overview of typical instantiations of agents within the framework is given.

2 AGENT INFRASTRUCTURE FOR THE DOM

2.1 Agent-based Software Systems

The DOM scenario is obviously characterized by a high degree of *heterogeneity and distributedness*, it can easily lead to a highly *complex software system*, and it is an *open environment* in the sense that we have to expect that frequently new components (even formerly unknown ones) may be plugged into the overall system, be replaced by other modules, or plugged out.

Over the last years, the paradigm of *agent-based computing* turned out to be an appropriate means for dealing with such application scenarios. In this paper, we suppose the reader to have basic knowledge about agent-based software systems and engineering. We employ the "weak definition" of agents introduced in (Wooldridge & Jennings, 1995) with the following definitional features: *(i) autonomy; (ii) social ability; (iii) reactive behaviour; and (iv) pro-active behaviour.* Other possible characteristics of software agents - like some level of intelligence, mobility, or techniques for learning and adaptation - may also be relevant for parts of the overall solution we aim at. However, in this paper we will focus on multiagent systems' capabilities for self-organization and social organization as a means for dealing with complex and dynamically changing situations which are mainly constituted by the characteristics mentioned above.

In principle, human as well as software agents can be described with respect to the following dimensions corresponding to Newell's (1982) knowledge level:

- Goals: Agents operate in a regularly changing environment. In doing so, they not only *react* to such changes, but also have their own goals and objectives which they try to achieve.

- Knowledge: Agents have knowledge with respect to the relevant realms of their environment, e.g. objects and other agents, as well as with respect to their own goals.
- Competencies: An agent’s abilities to perceive and manipulate its environment and its own internal state. In a multi-agent environment, the abilities to communicate with other actors are particularly important.

Through communication, knowledge about facts, goals, competencies, etc. can be exchanged. This allows for negotiation and agreements which may lead to a distribution of tasks between agents or to changes of an agent's knowledge and goals.

2.2 Socially-Enabled Software Agents

In Section 3 we will show that for a fully agent-based realization of the DOM scenario a huge amount of agents with possibly diverging goals and maybe highly complex communication and negotiation threads is required. As discussed in much detail by (Schillo et al., 2002), optimal work distribution and collaborative performance in such a group of agents benefits not only from *task delegation* and knowledge exchange, but also from *social delegation* as 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, with better task distribution because of specialization, etc. The phenomenon of society creation and self-organization can be observed in sociology (Bourdieu & Wacquant, 1992) and is a major topic of organizational theory. (Castelfranchi, 2000) considers it a crucial point for the introduction of agents into Enterprise Information Systems to complement the mechanisms for bottom-up control (system behaviour 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 order to achieve this goal, we propose to build a DOM as a set of collaborating societies of socially-enabled agents. These notions are being further refined in (Vicinus, 2002) and are exemplarily illustrated in (Elst & Abecker, 2002). In this paper we sketch the conceptual foundations and sketch their application in the DOM.

Hence we define an **Agent Society** as a set of agents (an agent can be member of several societies at the same time) 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 role concept is not new in agent-oriented analysis and design methods like GAIA (Wooldridge et al., 2000), because analysis and modelling of an application domain is the easier the more similar the modelling paradigm is to the phenomena occuring in the real world. And, obviously, business situations and complex organizations are typically characterized by roles.

Further we define **Socially-Enabled Agents** as software agents equipped with the required mechanisms to process appropriately rights and obligations, which together constitute a role in a society:

- **Rights**: Rights describe a subset of an agent's competencies. They describe under which conditions an agent *is allowed* to do something, like send a message to another agent, change his own goals, or grant rights to other agents.
- **Obligations**: Obligations also describe also a subset of an agent's competencies. They describe under which conditions (i.e. if a certain event occurs or another agent – maybe in a specified role – send a specific message) an agent is obliged to perform some action.

Figure 2 above gives a rough idea of the software agent implementation we did for socially-enabled agents on top of the JADE (Bellifemine et al., 2001) platform. The major design decisions illustrated here are the fact that an incoming message must be first be sorted into the appropriate society module, because an agent may belong simultaneously to several societies. The respective society behaviour implements a Reactive Rule system which encompasses the obligation processing. This leads to a list of candidate actions which is then filtered by the right processing unit before being executed by the agent.

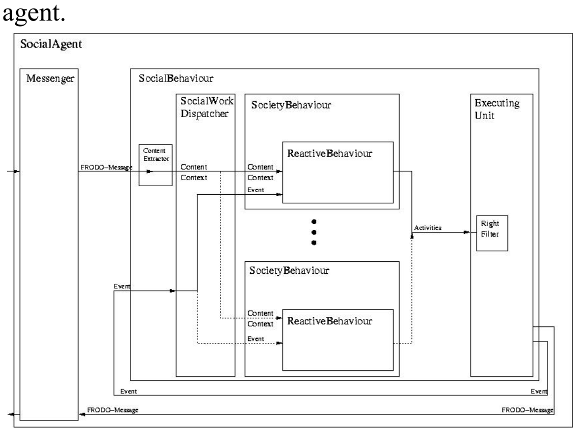

Figure 2: Socially-Enabled Agent

2.3 Competencies as Speech Acts

In order to make the idea of rights and obligations a bit more concrete and to show how their semantics could be defined, we sketch how their introduction leads to speech acts in the agent society. We describe these speech acts similarly to FIPA: The sender, receiver and content of a speech act are specified; feasibility preconditions contain the qualifications; the rational effect shows the reasons for which a speech act might be selected.

Table 1 shows two examples of FRODO speech acts for forming agent societies. With *ApplyForRole* an agent expresses the intention to take a specific role in a society. In the table two alternative specifications are given: a) In the simple specification the sender just wants the receiver to know that it wants to take the role and therefore the semantics of *inform* is used. Here, the receiver itself must infer that an appropriate reaction might be a *GrantRole* or a *Deny*. b) The second alternative is much more specific. Here, a *request* for a *GrantRole* action is used. This action should be applicable as soon as the receiver believes the desired role is possible for the sender. The precondition for *ApplyForRole* is that the sender really wants that role in the respective society and that it not already believes to have the role.

Accordingly, the precondition for a *GrantRole* is that the sender i) has the right to do so (hasRole(sender, society, Manager)), ii) has a belief that the receiver wants the role, and iii) the specific role is appropriate for the receiver. So the manager of an agent society is responsible for forming the society by granting roles to other agents. The operationalization of a role's rights and obligations for a concrete agent is done by a social layer in FRODOs agent platform sketched in Figure 2.

Table 1 - Two Examples of FRODO Speech Acts for Agents Societies.

FRODO speech act	ApplyForRole
Description	An agents wants to take a specific role in a society and therefore sends an application to the manager.
Sender	S
Receiver	R
Content	role, society
Feasibility Precondition	NOT(Believes(S, hasRole(S, society, role))) AND Wants(S, role, society)
Rational Effect	Believes(R, Wants(S, role, society)))
FIPA_action	`(inform :sender S :receiver R :content Wants(S, role, society))`
Comment (alternative specification)	`(request-when :sender S :receiver R :content (action (R, GrantRole(S,role,society)) (Believes(R,possibleRole (S, society, role)))`
FRODO speech act	GrantRole
Description	The manager of a society gives to an applicant a specific role.
Sender	M
Receiver	AP
Content	role, society
Feasibility Precondition	Believes(M, Wants(AP, role, society) AND hasRole(M, society, Manager) AND Believes(M, possibleRole(AP, society, role))
Rational Effect	Believes(M, hasRole(AP, society, role)), Believes(AP, hasRole(AP, society, role))
FIPA_action	`(inform : sender M : receiver AP : content Believes(M, hasRole(AP,society,role))), (inform : sender M : receiver M : content Believes(M, hasRole(AP,society,role)))`
Comment	The second inform just ensures the Rational Effect „Believes(E, hasRole(AP, society, role))".

3 AGENT SOCIETIES FOR THE DOM

In this section we briefly sketch the agent (sub-) societies required for building a DOM which arise directly from going through the several layers of the architecture in Figure 1.

3.1 Ontology Management

As indicated in Fig. 1 and explained in more detail, e.g., by (Abecker et al., 2000; Davies et al., 2002), the future's corporate-internal and external information systems will rely to much more 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 (cp. (Colomb, 1998; Abecker et al., 2001)).

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, e.g., in (Elst & Abecker, 2002). To sum up shortly, we present there the role of the D²OA **Distributed Domain Ontology Agent** which mediates between different agent societies holding their own specific domain ontologies. 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 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 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 synchronize only seldom with the agent network.

3.2 Workflow Agents

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 (cp. Pang, 2000). One of the most prominent agent-based workflow systems has been described by (Jennings et al., 2000). 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.

(Yu & Schmid, 1999) 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 about 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 Stormer, 2001) to represent also tasks as agents. In this way, all relevant entities in the real world are represented by software agents, what allows maximum flexibility and scalability. **Task 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.

In (Abecker et al., 2001) we describe the several roles in our agent-based, weakly-structured workflow system with context-sensitive knowledge delivery. We shortly summarize these roles: 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 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** (analoguely to user agents) represent electronic system resources (like specific software programs) which may be employed for achieving some workflow goal. The **Resource Manager** administers the resource agents available in the system and coordinates their communication with task agents.

Interesting examples for rights and obligations can be found, e.g. at the level of task models. In the spirit of a flexible workflow system, task 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 the most users currently enact a given task.

3.3 Personal User Agents

Since the begin 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, as well as an overview of information the system is constantly searching on behalf of the user according to his permanent information needs.

However, this does not yet exploit the full potential of our approach. As described, e.g., in (Abecker et al., 1998 & 2000) a comprehensive system like ours should and can be able to pro-actively deliver currently relevant information and knowledge to help the user efficiently perform his actual task at hand. We understand an actual information need as a function of personal, role and task-specific information requirements, interests, and preferences (Elst et al., 2001) 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 (Maus, 2001).

Typical rights of a PUA could be – within given limits of autonomy – to schedule meetings for the user, or to negotiate with task agents about acceptance or rejection of some work item. Typical obligations 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 activitiy for changing the way of working on this task.

In 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*.

3.4 Information Processing

Since the major reason for the system described here is to provide a user with purposefully selected, aggregated and processed data, information, and knowledge, the information processing agent society is at the core of our considerations.

There is a whole bunch of literature describing agent types and abstract functionalities occuring in multi-agent systems for information gathering, integration, and presentation, from Wiederholds Wrapper-Mediator approach (Wiederhold & Genesereth, 1997) up to Kerschberg's sophisticated agent typology in his Knowledge Rover architectures (Kerschberg, 1997). Such approaches show that complex, distributed information management problems can profit much from structured, agent-based software architectures. Nevertheless, the concepts of agent societies are almost not discussed in this community. Some authors use metaphors from the real-world to describe innovative functionalities, like the "digital reference librarian", which essentially amounts to a role definition in an information processing society, but an explicit role mechanism is usually not employed. The only relevant work in this direction which is known to us is described in (Röscheisen, 1997). The author employs a *relationship-based approach* to achieve trust and network security in the Internet environment. As a conceptual and technical means to realize this relationship-based approach, he introduces the notion of "commpacts – communication pacts" for encapsulating the boundary conditions of a social relationship, e.g., legal contracts or informal conventions. We considers these commpacts as a realization of kind of a "peer-to-peer" version of our rights and obligations.

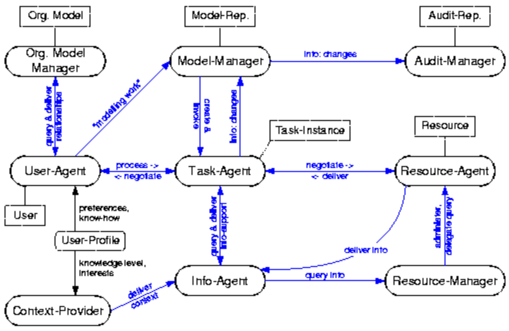

Figure 3: Agent-Role Collaboration

In our approach, at least the following agent roles are required**: Info Agents** know how to answer specific questions in a given context, or, respectively, how to come to an answer by delegating subproblems and integrating the results coming back. To this end, they may employ resource agents – which manage, e.g., databases – or specific search or problem-solving knowledge which might refer to domain, information, or enterprise ontologies managed by the respective ontology agents. Info Agents are supported by the **Context Provider** which is a specialist for the question which context facets may be helpful for improving what information processing task. On request from Info Agents, it gathers relevant context information, e.g., from user profile information delivered by the PUA, or from task and process information delivered by Task Agents and Model Manager. This context information is then sent to the respective Information Agent to support his goal of precise, situation-specific knowledge delivery. Figure 3 sketches the way how several agent roles interact in order to achieve the goal of workflow management with integrated, context-sensitive information delivery.

3.5 Formal-informal Transitions

We assume that most of the higher-level value adding services for knowledge and information processing need more formal representations than usually available in legacy information systems, or in the Internet. Instead of a text in a web site, we need the ZIP code of an address, instead of a tech report about a given topic, we need just this topic for retrieval purposes, instead of a JPEG representation of a technical drawing, we need the name of the product part it refers to. *Semantic Web* approaches suppose to have comprehensive metadata for such purposes. Our experience is that it will always be unrealistic to expect that all the metadata will be attached a-priori to informal knowledge representations which might be required for some later processing step. Instead, we need both approaches to deal with the informal representations and combine their results with more exact information (like the combination of metadata-based retrieval and fulltext retrieval on the basis of document similarity), and we need automatic techniques to extract and create metadata from informal inputs.

There are many approaches to integrate *wrappers* into multi-agent information gathering systems to extract data from Internet sites. (Lesser et al., 2000) present a more comprehensive approach which integrates document classification and information extraction agents of different level of sophistication, covering services from rough page-to-topic classifications to heavyweight document understanding. (Maus, 2001) shows that integrating such wrapper services into a comprehensive OM scenario can improve the quality and efficiency of algorithms. (Klein & Abecker, 1999) show that Document Analysis and Understanding (DAU) can itself be understood as a multi-agent process. However, up to now, agent-based DAU is not a topic of major interest, and its integration into more complete OM scenarios neither. So, this topic is solved today in a pragmatic way, but a thorough role-based analysis is still missing. Nevertheless, some possibilities are obvious. For example, the Context Provider introduced above might be obliged to continuously update the expectation store of DAU agents by analyzing newly started workflow processes.

4 SUMMARY

With the advent of the networked economy, virtual enterprises, and ubiquitious 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. 1 was the basis for numerous research and several successful application projects. The logical next step is to proceed to the **Distributed OM** 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 ist 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 you consider these areas in an isolated manner. A good example for such a synergy are the quality improvements possible for document analysis algorithms when taking into account workflow expectations (Maus, 2001). However, building such integrated systems introduces a new level of complexity into software design and implementation. In order to deal with this complexity, we introduced the notion of **Socially-Enabled Agents** and the concept of **Agent Societies** defining **Roles with Rights and Obligations** in Section 2. In the area of agent-based workflow, role-based modeling proved already to be a useful system analysis paradigm for mapping the processes occuring in 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.

In our FRODO project there is not yet a full-fledged software demonstrator for the whole solution. We focussed first on the areas of distributed ontology management and agent-based, weakly-structured workflow which are already running (Elst & Abecker, 2002; Vicinus, 2002; Abecker et al., 2001). Further implementations use the concepts of socially-enabled agents for workflow-embedded, ontology-based information management in the areas of music information and research publication management (Chaouch, 2002; Hofmann, 2002). Besides building further software demonstrators for specific partial problems and specific synergies within the whole scenario, an important next step should be a sound theoretical analysis and "political" harmonization of all activities in our direction which may be subsumed under the term of **Agent-Mediated Knowledge Management** (cp. Dignum, 2002).

REFERENCES

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

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

Abecker, A., Bernardi, A., van Elst, L., Lauer, A., Maus, H., Schwarz, S., and Sintek, M., 2001. *FRODO: A framework for distributed organizational memories. Milestone M1: Requirements analysis and system architecture.* Document D-01-01, DFKI, Kaiserslautern.

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

Blake, M.B., 2002. An agent-based cross-organizational workflow architecture in support of Web services. In *11th IEEE WETICE 2002*. IEEE Computer Society Press, Pittsburgh.

Bourdieu, P., and Wacquant, L., 1992. An invitation to reflexive sociology. Polity Press, Chicago.

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

Chaouch, Kaies, 2002. *Intelligent ontology-based information access in the music area.* Diploma Thesis. Universität Kaiserslautern.

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

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

Dignum, V., 2002. *An overview of agents in knowledge management.* http://www.cs.uu.nl/people/virginia/-dignum_ekaw.ps

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

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

Hofmann, K., 2002. *Prozeßorientierte Unterstützung wissensintensiver Tätigkeiten am Beispiel der Publikationsplanung.* Diploma Thesis. Universität Kaiserslautern.

Jennings, N., R., Faratin, P., Norman, T.J., O'Brien, P., and Odgers, B., 2000. Autonomous agents for business process management. *Int. Journal of Applied Artificial Intelligence* 14(2):145-189.

Kerschberg, L., 1997. 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.

Klein, B., and Abecker, A., 1999. Towards distributed knowledge-based parsing for document analysis and understanding. *In IEEE Forum on Research and Technology Advances in Digital Libraries, Sixth IEEE ADL '99.* IEEE Computer Society, Los Alamitos.

Lesser, V., Horling, B., Klassner, F., Raja, A., Wagner, T., and Zhang, S., 2000. BIG: An agent for resource-bounded information gathering and decision making. *Artificial Intelligence Journal*, 118(1-2):197-244.

Maus, H. 2001. 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.

Motik, B., Maedche, A., and Volz, R., 2002. A conceptual modeling approach for building semantics-driven enterprise applications. In*: ODBASE-02*, *1st Int. Conf. on Ontologies, Databases and Application of Semantics.*

Newell, A., 1982. The knowledge level. *Artificial Intelligence*, 18: 87-127.

Pang, G., 2000. *Implementation of an agent-based business process.* Diploma Thesis. Universität Zürich.

Schillo, M., Fley, B., Florian, M., Hillebrandt, F., and Hinck, D., 2002. Self-organization in multiagent systems. In *3rd Int. Workshop on Modelling Artificial Societies and Hybrid Organizations (MASHO).*

Stormer, H., 2001. A flexible agent-workflow system. In: *Workshop on Agent-Based Approaches to B2B* at the 5th Int. Conf. on Autonomous Agents (AGENTS 2001).

Vicinus, R., 2002. *Soziale Agenten für das Management von Domänen-Ontologien in verteilten Organizational Memories.* Diploma Thesis, Universität Kaiserslautern.

Wiederhold, G., and Genesereth, M.R., 1997. The conceptual basis for mediation services. *IEEE Expert* 12(5): 38-47 .

Wooldridge, M., and Jennings, N. R., 1995. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2).

Wooldridge, M., Jennings, N. R., and Kinny, D., 2000. The Gaia methodology for agent-oriented analysis and design. In *Autonomous Agents and Multi-Agent Systems, 3,* 285-312, Kluwer Academic Publishers.

Yu, L. and Schmid, B.F., 1999. 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.*