# Agent-Mediated Knowledge Management

Ludger van Elst, Andreas Abecker

**In this article we review the current state of the art in Agent-Mediated Knowledge Management systems and sketch some directions for future research.**

## 1 Agents and KM

Knowledge Management (KM) is a systematic approach for sustainably improving the handling of knowledge on all levels of an organization (individual, group, organizational, and inter-organizational level) in order to support the organization's business goals, such as innovation, quality, cost effectiveness etc. KM holistically combines activities adressing organization culture, static and dynamic organization structures, as well as ICT infrastructure (cp. [4]).

ICT approaches typically fall into one of two basic system classes: One class—comprising, e.g., Organizational Memory Information Systems (OMs, cp. [1, 20])—aims at acquisition, structuring and high–precision delivery of *explicit* knowledge ("provide the right people with the right information at the right time"). The other class of systems—like expert finder systems or community of practice support—doesn't rely so much on explicitly represented knowledge, but rather brings people together, for instance, to solve a given knowledge–intensive problem (see, for instance [6, 24]).

Based on our practical and research experience with KM solutions, we identify the following requirements as central challenges for next generation KM systems. We can only sketch them briefly, for more detail, please refer to [38, 26].

**R1:** *KM has to respect the distributed nature of knowledge in organizations*: A KM system should therefore allow to balance between (a) *global* knowledge which might have or might 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. Further, as global views cannot always be reached, a KM system must be able to handle context switches of knowledge assets, e.g., by providing explicit procedures for capturing the context during knowledge acquisition and for re-contextualizing during knowledge support. Fully accepting the ideas of distribudness means to face technical as well as organizational problems such as trust, responsibility, and contextuality.

**R2:** *KM systems must reflect the inherent goal dichotomy between business processes and KM processes*[1].: Within an environment of bounded resources, knowledge workers will always concentrate on their first order business processes instead of KM meta processes. This means they optimize their operational goals locally and only invest very little to fulfil strategic, global KM goals. It is clear and pretty well accepted that *having and using* knowledge is important for optimally fulfilling first-order tasks, but the workload and time pressure is nevertheless often so high that the effort invested for preparing this, time for *knowledge conservation, evolution, organization,* etc., is considered a second-order process often neglected in practice. Even cumbersome activities for knowledge search and reuse are often considered to be unacceptable. Therefore, KM processes should be embedded in the worker's first-order processes, and proactive tools should minimze the cognitive load for KM tasks.

**R3:** *Knowledge work as well as KM in general, must be considered as "wicked problem solving" (cf. [14])*: This means that a precise a-priori description of how to execute a task or solve a problem does not exist. 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 in each working step. 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 a fundamentally social process. A KM system should therefore support the necessary complex interactions and supporting processes like planning, coordination and negotiation of knowledge activities.

**R4:** *KM has to deal with changing environments*: KM systems typically reside in environments which are subject to frequent changes, in organizational structures, in business processes, or in the 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 often follows an evolutionary approach where functionalities are not implemented "in one step" for the whole company, but partial solutions are deployed to specific sub-structures. In order to obtain a comprehensive system, these elements then have to be integrated under a common ceiling without disturbing their individual value.

Typically, agents are considered best suited to applications that are modular, decentralized, changeable, ill-structured, and complex (see, e.g., [49]). Although the match between these five salient features and the KM requirements R1 – R4 is already obvious, we will elaborate a bit more on

[1] For a discussion of operational processes vs. knowledge processes, see, for instance, [54, 59].

this match. Let us start with the *weak definition* of agents [61] with the definitional features *autonomy*, *social ability*, *reactive behavior, and proactive behavior*.

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 (R1, 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 (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 accomodating to the reality that knowledge workers typically do not adopt KM goals with a high priority (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 enable the legacy system to interact with other systems much more easily. Furthermore, agent approaches allow for extensibility and openness in situations when it is impossible to know at design time exactly which components and functionalities the system will have in the long run. Both arguments reflect pretty well the technical consequences of abstract requirements R4 and R3 (changing environments demand continuous reconfiguration, the unpredictable nature of wicked problems requires flexible approaches), R2 (competition between operational work and KM meta work call for highly integrated KM solutions), or R1 (already existing local solutions must be confederated).

In the last years, there have been many experimental systems exploring the use of agents for investigating the one or other aspect (such as weakly-structured workflow, ontology mediation, metadata for knowledge retrieval, or contextuality) of KM, as well as comprehensive agent-based KM frameworks (like FRODO, CoMMA, Edamok [3, 10, 29]). In this paper we try to give an overview of today's state of the art in this area and to identify directions for future work.

# 2 A description schema for agent–based KM approaches

In the literature, there exist already many good classification schemas for agent applications. Nevertheless, for the purpose of this paper, we propose a description schema that is, on one hand, more KM-specific and, on the other hand, also captures the whole life cycle of agent–oriented system development. To get an overview of agent approaches for KM, we think that a categorization along three dimensions is especially beneficial:

1. the stage in a system's *development process* where agents are used (analysis, conceptual design, or implementation);
2. the *architecture / topology* of the agent system; and
3. the *KM functionality / application* focused on.

We discuss these dimensions in the following three subsections.

## 2.1 System development level

*Agent–oriented Software Engineering* emphasizes the adequacy of the agent metaphor for design and implementation of complex information systems with multiple distinct and independent components. Agents enable the aggregation of different functionalities (such as planning, learning, coordination, etc.) in a conceptually embodied and situated whole [42]; agents also provide ways to relate directly to these abstractions in the design and development of large systems.

An organizational analysis is often integral part of methodologies for the development of KM systems (see, e.g., the CommonKADS [57], or the DECOR [48] methods). Originating in the realm of *human* collaboration, the notion of agents can be an epistemologically adequate abstraction to capture and model relevant people, roles, tasks, and social interactions. These models can be valuable input for the requirements analysis phase for the development of the KM system.

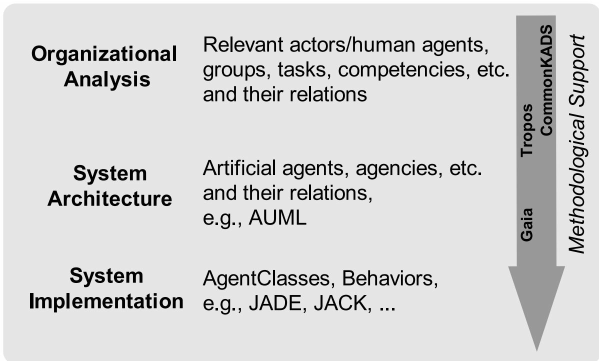


Abbildung 1: Notion of agents at different stages in the development cycle of an agent–based KM system

Figure 1 gives an overview of the use of agents on different levels in the system engineering cycle. Of course, on each level we can have different specific *agent theories* (that is, how agents are conceptualized, what basic properties they have, etc. [61]) and respective *representation languages* (which on the implementation level may be operational programming languages) for defining concrete agents and their relations. *Methodologies* for agent–oriented software engineering like Tropos [32] and Gaia [62] not only define these representation languages for different levels, they are also the glue between them by providing mappings and processes for the transition from one level to another.

In [23], overall design requirements for KM environments were identified, which include the need to separate the specification of the organizational structure for the internal architecture of its component entities, and the need for explicit representation of normative issuesm (see also [21]).

Of course, in real life, pragmatic decisions may often lead to the use of agents at just one or two development levels. On the other hand, having to implement a KM system on the basis of “conventional software” (like relational databases or client/server–based groupware solutions) or on the basis of modern, strongly related technologies like peer-topeer networks or web services should not necessarily hinder an agent–oriented analysis and system design.

## 2.2 Macro–level Structure of the Agent System

*Agent theories*, *abstract agent architectures*, and *agent languages* as defined in [61] mainly take a micro–level view, i.e., they focus on the concept of *one* agent: What properties does an agent have, how can these properties be realized in a computer system, what are the appropriate programming languages for that? For Knowledge Management—which typically employs a strong organizational perspective—the macro–level structure is also of special interest. How many agents do we have? What types of agents? What is the topology with respect to the flow of information, or with respect to the co-ordination of decisions? One possible dimension to characterize the macro–level of an agent–based KM system is the *degree of sociability* as depicted in Figure 2:

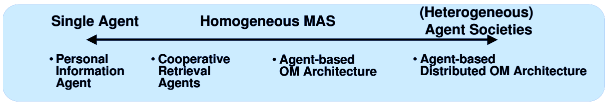


Abbildung 2: Degree of sociability

- *Single–agent architectures*: Typical examples are user interface or personal information agents. These agents can perceive their environment and access some objects like web resources, but they normally have no elaborated interaction (like collaboration or negotiation) with other agents (except for the human user).
- *Homogeneous multi–agent architectures*: Agents can co-operate with other agents in order to solve their tasks. Homogeneity means that the system consists mainly of *one* type / class of agents. These agents have not necessarily the same goals, but their tasks and capabilities are comparable. Agent–based collaborative filtering is a typical example: All agents are seen as peers which can provide information on what entities they use or like, and each agent can collect this information to provide the user with valuable hints about interesting new information.
- *Heterogeneous multi–agent architectures*: contain multiple agent classes which may have completely different purposes, knowledge and capabilities. Various information integration architectures (e.g., Knowledge Rovers [34], MOMIS/MIKS [7]) are described as heterogeneous MAS: Specialists exist for wrapping information sources, agents for integrating different description schemas, and for adequately presenting information to the users.

A characterization of the macro–level structure of an agent–based KM system may, in addition to the description of the number of agents and the system's heterogeneity, also include facets like

- *co-ordination form*: How are decisions and information flow coordinated?
- *open vs. closed system*: (How)Can new agents (agent types) enter the system?
- *implicit vs. explicit social structure*: Do the agents have an explicit representation of their role in the system which allows for a certain assurance of the system's global behavior? Do they even have a machinery for reasoning about their rights and obligations? Are roles globally defined or negotiated?

*Electronic institutions* are a typical example of a complex society architecture. Electronic institutions provide a computational analogue of human organizations in which agents interact through roles that are defined as specified patterns of behavior [60].

Of course, the above examples for different degrees of sociability do not form a discrete, categorical discrimination. On the contrary they are exemplary operating points on a continuous scale. Heterogeneous MAS, e.g., may have sub–societies that are homogeneous themselves.

## 2.3 KM application area

The third dimension for characterizing agent–based KM applications deals with the specific knowledge management functionality of the system: What is the scope of the system? Which KM processes / tasks are supported?

Principally, there are many high–level KM models which could be used to form the vocabulary for this dimension. We will start with the famous KM cycle by Probst et al. [52] which — in addition to the management–oriented tasks of *defining knowledge goals* and *assessing the organization's knowledge* — e.g., identifies six building blocks:

- *Identification processes* analyze which knowledge exists in an organization.
- *Acquisition* is the process of integrating external knowledge.
- *Development processes* generate new knowledge.
- *Distribution processes* connect knowledge containers with potential users.
- *Preservation* aims at the sustainability of knowledge, i.e., that is accessible and understandable over a period time.
- *Utilization* means to operationalize available knowledge in order to solve actual business tasks (better).

Alternatively, the model of Nonaka and Takeuchi [47] can be used to describe the KM application area. They distinguish four types of transformation processes between implicit / internal knowledge (e.g., competencies, experiences, skills) and explicit / external knowledge (e.g., facts, coded rules, formal business processes):

- With *socialization*, knowledge that is implicit to a person is transferred to another person by sharing experiences. Apprenticeship learning, for example, makes heavy use of socialization.
- *Externalization* is the process of making implicit knowledge explicit, e.g., by talking about it, writing it down informally or by formalizing it. Knowledge acquisition techniques developed for expert systems mainly aim at externalization.
- *Combination* is the basis for generating new knowledge from external knowledge by relating knowledge pieces with other knowledge pieces. Data mining and machine learning are technical approaches for this.
- *Internalization* is the transformation of explicit knowledge into implicit knowledge and thereby making it applicable.

From these classical models, several further distinctions have been developed in Knowledge Management research that can be utilized to describe the application area. For example, systems can take a more *process–oriented* or a more *product–oriented* view [37, 44]. The latter emphasizes the management of explicit knowledge; the former focuses on human beings and their internal knowledge, i.e., the "process of knowing" and the "process of knowledge exchange" between people.

Furthermore, a KM system can support *individuals* and their tasks at hand, it can support *teams and groups*, or it may take a more global, *organizational perspective*.

# 3 Exemplary Agent-based KM Applications

In this section, we present some examples of agent-based systems developed to support and/or model KM domains. We group these systems by the second dimension (macro–level structure), because this also largely reflects and matches the historical evolution of research in this area.

## 3.1 Predominantly Single Agent Approaches

Most KM support systems that take a single agent approach are *User Interface Agents* or *Information Agents* which embody the metaphor of "a personal assistant who is collaborating with the user in the same work environment" [43]. Virtually all systems in this class are information agents (cp. [40]) that typically

- have access to a variety of information sources;
- handle a model of the user's information needs and preferences; and
- try to provide relevant information to the user in an adequate way, either by filtering incoming information from the sources, or by actively retrieving it.

Prototypical systems in this category use e-mail in-boxes, news forums, dedicated KM databases within the company, intranet documents, or internet search engines as information sources.

A representative architecture for an intelligent information agent that assists the user in accessing a (not agent–based) Organizational Memory, in this case the OntoBroker system, is described in [58]. The agent relies on an explicit model of the actual business process and uses this knowledge of the work context to determine *when* information support may be appropriate and *what* information may be useful in that context.

Rhodes and Maes [56] presented three *just-in-time* information retrieval (JITIR) agents: The *Remembrance Agent* continually presents a list of documents that are related to a document currently being written or read in the Emacs editor, *Margin Notes* uses documents loaded in a Web Browser as context, and *Jimminy* uses the physical environment (location, people in the room, etc.) to determine what information may be relevant. All three agents use the same back–end system *Savant* [55] for the actual information retrieval step.

The primary contribution of such research is the development of adequate *sensors* and *effectors* for personal information agents. Sensors define the way the agents can assess the context of their services, i.e., *when* to perform a service proactively and *what* the user's actual information need is. Here, approaches range from the pre–modelled business processes described above, to observing knowledge workers in their usage of standard office applications like text processors, web browsers or mailing tools (cf. Watson [16] or Letizia [39]).

The effectors of user interface agents, on the other hand, determine the way information can be presented to the user. The JITIR agent *Margin Notes*, for example, automatically rewrites Web pages as they are loaded, and places links to personal information items in a dedicated area of the page. Watson presents suggestions in a dedicated window, and in KnowMore [2], information from the Organizational Memory can be directly handed over to specific fields in a form–based application.

Concerning the level of system development, personal assistant approaches are mostly deployed at the modelling level. The most relevant aspect used from the agent metaphor is that an agent acts *on behalf* of a user who has specific *goals* and *interests*. Regarding the implementation level, personal assistants are today mostly implemented using conventional programming techniques.

With respect to the KM application area, personal assistants are mainly related to the dissemination of knowledge to be used by knowledge workers, in a just–in–time, just–enough fashion. Applications such as OntoBroker take a product–oriented view on knowledge, as they emphasize the management of explicit knowledge sources.

To sum up, we can say that many of the presented ideas are already well-developed in the technological sense, and some of them have even found their way into commercial software products. Although the software functionalities are stable and apparently useful, the logical next step for research and application has seldom been done, namely a rigorous assessment of usability and usefulness. There are a few specific experiments about evaluation of Personal Information Agents and the influence of process-aware, proactive information delivery, respectively (see [15, 17, 27, 56]), but in our opinion there is still a need for broad and long-term experiments about usability issues, user acceptance, and influence on working behavior and working efficiency / effectiveness by KM tools.

## 3.2 Homogeneous Multi–Agent Approaches

An obvious extension to the personal information agents described above is to see each user not only as an information consumer, but also as a provider. In this case, the personal agent could assist the user also in serving as a source of information. A simple example for such agents are the clients for peer-to-peer file sharing like Kazaa, ED2K, or – in the e-Learning domain – Edutella [46]. These agents have specialized interfaces for expressing queries, passing them on to other agents and displaying the results. But they are also able to receive queries and process them by answering with result documents or by passing a query to other agents. In the following, we describe more elaborate approaches.

*MARS* is an *adaptive social network for information access* [63] with a purely homogeneous structure that is based on the idea above. Each agent has basically two competencies: i) to deliver some domain information with respect to a query, and ii) to refer to other agents that may fulfill a specific information need. Additionally, the agents learn assessments of other agents in the network. This means they assess the other agents' ability to produce correct domain answers as well as their ability to produce accurate referrals.

*DIAMS* [19] is a system of distributed, collaborative information agents that help users access, collect, organize and exchange information on the World Wide Web. DIAMS aims at encouraging collaboration among users. Personal agents provide their owners with dynamic views on well–organized information collections, as well as with user–friendly information management utilities. These agents work closely together with each other and with other types of information agents such as matchmakers and knowledge experts to facilitate collaboration and communication. aConnections between users with similar interests can be established with the help of matchmaker agents.

The focus of the research described in [50] is to add *context–awareness* to personal information agents that are (homogeneous) peers in a larger society of agents. The so-called *CAPIA*s (Context–Aware Personal Information Agents) have a model of their social and potential process context (e.g., the user's schedule) as well as of their physical context (time and location). In the COMRIS Conference Center system the CAPIAs are employed for context–sensitive presentation of relevant information, e.g., whether "interesting" conference attendees or events (sessions, exhibition booths) are to be found nearby.

Homogeneous multi–agent approaches in Knowledge Management seem to be a good way for leveraging single–agent approaches by taking advantage of the knowledge of other users in the organization. In the GroupLens project these leveraging effects are systematically investigated [33]. *Let's Browse* [41], the successor of *Letizia* [39], does not model its collaborative web browsing as a cooperation between independent agents, but as one central agent that comprises the profiles of several users. An interesting but open question is, to what extent multi–agent modelling has an "added value" (e.g., wrt. user trust, privacy concerns, willingness to disclose information, ...) compared to a "functionally" equivalent monolithic system.

Also homogeneous MAS applications to KM are mainly seen at the modelling level of development. With respect to the KM dimension, multi–agent approaches are mostly directed to the modelling of collaboration and interaction between users and systems, that is, with socialization issues. While most systems still lean considerably towards a product–oriented view of knowledge, these systems take a more process–oriented view on the management of knowledge, and can support teams and groups, as well as individual users. Homogeneous MAS approaches mostly provide a multiplication of a single–agent, and as such may not be able to support enough depth needed at the analysis and design level for comprehensive KM. Complex KM domains often require the combination of global and individual perspectives, and activities to follow desired structures, while enabling autonomous decisions on how to accomplish results. In order to cope with these requirements, heterogeneous approaches may be more appropriate.

## 3.3 Heterogeneous Multi–Agent and Society–oriented Approaches

Heterogeneous multi–agent systems consist of a potentially high number of agents which may belong to different classes. with diverse competencies and types of goals. The design of many agent–based KM systems emerges from the "standard" three–tier enterprise information architectures that are often the basis for business applications (e.g., [45]):

- The *data layer* manages repositories with knowledge objects such as documents, e-mail, etc.
- The *application layer* realizes the business logic of the system.
- The *presentation layer* organizes the interaction of the system with its users.

*KAoS* [13, 18], a generic agent architecture for aerospace applications, is an early agent–based system for the management of technical information contained in documents. KAoS employs agents on all three layers. In addition, a layer with generic service agents provides the middleware functionality of an agent platform (whitepage and matchmaking services for agents, proxies for connections to other agent domains, agent context management). The data services wrap the information sources by encapsulating indexing, search and retrieval functions, but also monitor them to allow for proactive information push. The prototype system *Gaudi* uses the KAoS platform for situation–specific, adaptive information delivery in the context of training and customer support in the airplane industry [12]. Recent versions of KAoS also incorporate social aspects in agent communities [28]. However, the relevance of this approach for KM applications has not yet been discussed.

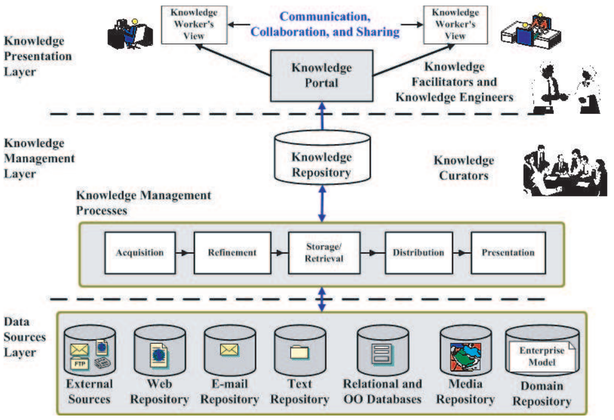


Abbildung 3: Three-layer KM Architecture [35] (reprinted with kind permission)

The focus of KM systems based on a layer architecture like the one presented above is mostly the *reuse* of information contained in the information sources. Consequently, the knowledge flow is mainly from the data layer to the presentation layer. The conceptual model for Knowledge Management that Kerschberg presents with his *Knowledge Rover* architecture [34] does not have this principal restriction. He broadens the presentation layer to a *Knowledge Presentation and Creation Layer*, which also comprises discussion groups and other types of potential knowledge creating services [35] (cf. Figure 3). Hence, knowledge flow from the presentation to the data layer is also taken into account. Consequently, the application layer comprehensively embraces *all* basic KM processes — acquisition, refinement, storage/retrieval, distribution, and presentation of knowledge (cf. Section 2.3).

For a knowledge reuse–oriented view, the *integration of information from various sources* is essential. The fusion of knowledge from multiple, distributed and heterogeneous sources is the focus of the *KRAFT* project [51]. KRAFT has an agent–based architecture with knowledge processing components realized as software agents. The architecture uses constraints as a common knowledge interchange format, expressed in terms of a common ontology. Knowledge held in local sources can be translated into the common constraint language, fused with knowledge from other sources, and is then used to solve a specific problem, or to deliver some information to a user. The generic framework of the architecture can be reused across a wide range of knowledge domains and has been used in a network data services application, as well as in prototype systems for advising students on university transfers, and for advising health care practitioners on drug therapies. The implementation of KRAFT is based on the FIPA standard with RDF as a content language.

The *Campiello* project [36] aims at using innovative information and communication technology to develop new links between local communities and visitors of historical cities of art and culture. The objectives of the project are to connect local inhabitants of historical places better, to make them active participants in the construction of cultural information, and to support new and improved connections with cultural managers and tourists. The system includes a recommender module, a search module, and a shared data space. In order to facilitate the integration, tailoring and extensibility of these components, an agent model was chosen for the services in Campiello. The architecture supports interaction between distributed, heterogeneous agents and is built on top of the Voyager platform[2] which was extended towards an agent platform by adding directory and broker services, administration tools and agent classes.

In an organizational environment, one of the main context aspects is the business process a knowledge worker is involved in. Business process–oriented Knowledge Management (BPOKM, cf. [5]) considers these processes i) as knowledge objects themselves, ii) as knowledge creation context, iii) as trigger, *when* some knowledge objects may be relevant, and iv) as context *what* knowledge may be relevant. The *EULE* system [53] shows an integration of business process modeling and knowledge management. The system takes a micro–level view on business processes by modeling and supporting "office tasks" of a *single* worker by just–in–time information delivery, but does not coordinate complete workflows performed by groups of people. While EULE is not an explicitly agent–based system, in the FRODO framework for Distributed Organizational Memories [3] workflows themselves are first–order citizens in an agent–society for KM in distributed environments. An Organizational Memory in FRODO can be seen 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 [1]. The FRODO four layer architecture for Organizational Memories (OM) contains i) the *application layer*; ii) the *information source layer*; iii) The *knowledge description layer* (metadata layer); and iv) the *knowledge access layer* Agents in a FRODO reside on all four layers:

- *Workflow–related agents* (task agents, workflow model manager, ...) are on the application layer and control the execution of business processes.
- *Personal User Agents* are also on the application layer and provide the interface to the individual knowledge worker.
- On the knowledge access layer, *Info Agents* and *Context Providers* realize retrieval and other information processing services to support the task and user agents.
- The knowledge descriptions are handled by *Domain Ontology Agents*. Dedicated *Distributed Domain Ontology Agents* serve as bridges between several OMs.
- *Wrapper Agents* and *Document Analysis and Understanding Agents* enable access to the sources and informal–formal transitions of information, and are thus located in the knowledge object layer or at the intersection between knowledge objects and knowledge descriptions, respectively.

A FRODO agent is not only described by its knowledge, goals and competencies, but also by its *rights and obligations*, thus forming agent societies. The description of

[2] http://www.recursionsw.com/products/voyager/voyager.asp

ontology societies in [25] exemplifies FRODO's concept of socially–enabled agents for KM. The implementation is based on the FIPA–compliant JADE platform[3].

The *Edamok* project[4] also aims at enabling autonomous and distributed management of knowledge [10]. Edamok completely abandons centralized approaches, resulting in the peer–to–peer architecture *KEx* [9]. Each peer in KEx has the competence to create and organize the knowledge that is local to an individual or a group. Social structures between these peers are established that allow for knowledge exchange between them. In addition to the semantic coordination techniques that are required for this approach, the Edamok project also investigates contextual reasoning, natural language processing techniques and methodological aspects of distributed KM.

In the *CoMMA* project [8]. societies of agents are created for personalized information delivery [31]:

- Agents in the *ontology dedicated sub–society* are concerned with the management of the ontological aspects of the information retrieval activity.
- The *annotation dedicated sub–society* is in charge of storing and searching document annotations in a local repository and also of distributed query solving and annotation allocation.
- The *connection dedicated sub–society* provides white page and yellow page services to the agents.
- The *user dedicated sub–society* manages user profiles as well as the interface to the knowledge worker.

The sub–societies in CoMMA can be organized hierarchically or peer–to–peer. The position of an agent in a society is defined by its role [30]. The system was implemented on top of JADE agent, and special attention was paid to the use of XML and RDF for representing document annotations and queries.

With respect to the question of where in the development cycle the notion of agents is used, most of the systems presented up to now take kind of a *middle–out* approach: All of them have an agent–based description of the system's components. This description is partly motivated by a functional decomposition from an IT point of view, and partly a result of reflecting real–world entities (users, groups, etc.) in the system. Some of these architectures are then implemented using "conventional" software technology, others build upon dedicated agent platforms. Only a few of the described systems complement their architectures with an agent–based KM methodology for guiding the development of such a system in an organizational context.

A recent proposal for a design methodology specifically tailored to agent societies is *OperA* [21]. This methodology is based on a three–tiered framework for agent societies that distinguishes between (a) the specification of the intended organizational structure, and (b) the individual desires and behavior of the participating agents:

1. The organizational structure of the society, as intended by the organizational stakeholders, is described in the *Organizational Model* (OM).
2. The agent population of an OM is specified in the *Social Model* (SM) in terms of social contracts that make explicit the commitments which are regulating the enactment of roles by individual agents.
3. Finally, given an agent population for a society, the *Interaction Model* (IM) describes possible interaction between agents.

After all models have been specified, the characteristics and requirements of the society can be incorporated in the implemented software agents themselves. Agents will thus contain enough information and capability to interact with others according to the society specification. The OperA methodology and framework have been applied to the design of *Knowledge Market*, an agent society to support peer–to–peer knowledge sharing in a Community of Practice; this has been designed in such a way that it preserves and recognizes individual ownership of knowledge and enables the specification and monitoring of reciprocity agreements [22].

The table 1 below characterizes the systems described wrt. the dimensions introduced in Section 2.

# 4 Summary and Outlook

The goal of this paper was twofold, i) to clarify the relationship between typical characteristics of Knowledge Management environments and core features of software agents as a basic technology to support KM, an ii) to provide a framework for the analysis and description of agent–based KM systems.

The synopsis of exemplary agent–based KM systems in Section 3 with respect to these dimensions showed how the design space is covered by today's research approaches, prototypes and systems. Though most applications are not entirely agent–based from organizational analysis to system implementation, the potential of agent technology in all phases was demonstrated. On the other hand, it is clear that the majority of KM applications nowadays is *not explicitly* agent–based. Thus, there is still much work to be done in order to fathom the capability of agent technology for KM systems.

We just briefly sketch some possible future research directions:

1. *Socio–technical*: How can the teamwork of human knowledge workers and artificial agents (that might act "on behalf of" people) be balanced? Here, questions from human–computer interaction arise, but also questions of trust, responsibility, etc.
2. *Agent technology and KM functionality*: What agent models and architectures are needed for what kind of KM application? Should concepts of trust, responsibility, rights, obligations be integrated in the models? How can the flexibility of reactivity and proactivity better be exploited for KM tasks?
3. *Methodological and engineering aspects*: Which functionalities can be provided as a kind of "KM middleware" or as modules for building KM applications? How should agent–orientation of design and implementation be reflected in an "agent–based KM methodology"

[3] http://sharon.cselt.it/projects/jade/

[4] http://edamok.itc.it/

in order to facilitate transitions between different phases in the development cycle?

4. *Evaluation of agent–based KM*: How good does the integration of (not agent–based) legacy systems into agent environments work in real–world applications? How easily can new agent–based components really be integrated into an existing system? What evaluation paradigms can be used to make different KM applications more comparable?

At the moment it is hard to argue (and indeed not aimed at in this paper) that agent–based systems can do things that could not also be done with conventional technology, especially at the implementation level. However, we believe that agent technology helps building KM systems faster and more flexibly. We think that the results presented in this paper demonstrate the potential of agent–oriented views to build more human–centered, more agile, and more scalable KM systems.

## Literatur


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

[2] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Context–aware, proactive delivery of task–specific knowledge: The KnowMore project. *Int. J. on Information System Frontiers*, 2(3/4):253–276, 2000.

[3] A. Abecker, A. Bernardi, and L. van Elst. Agent technology for distributed organizational memories. In *Proc. ICEIS-03, Vol. 2*, pages 3–10, 2003.

[4] A. Abecker and L. van Elst. Ontologies for knowledge management. In St. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*, pages 453–474, Berlin, 2003. Springer Verlag.

[5] A. Abecker, K. Hinkelmann, H. Maus, and H. J. Müller, editors. *Geschäftsprozessorientiertes Wissensmanagement*. Springer, 2002.

[6] I. Becerra-Fernandez. The role of artificial intelligence technologies in the implementation of people-finder knowledge management systems. *Knowledge Based Systems*, 13(5):315–320, 2000.

[7] D. Beneventano, S. Bergamaschi, G. Gelati, F. Guerra, and M. Vincini. Miks: An agent framework supporting information access and integration. In M. Klusch, S. Bergamaschi, P. Edwards, and P. Petta, editors, *Intelligent Information Agents - The AgentLink Perspective*, volume 2586 of *LNCS*, pages 22–49. Springer, 2003.

[8] F. Bergenti, A. Poggi, and G. Rimassa. Agent architectures and interaction protocols for corporate management systems. In *Proc. ECAI Workshop on Knowledge Management and Organisational Memories*, pages 39–47, Berlin, Germany, 2000.

[9] M. Bonifacio, P. Bouquet, G. Mameli, and M. Nori. Kex: a peer–to–peer solution for distributed knowledge management. In *Proc. PAKM02*, volume 2569 of *LNAI*, pages 490–500. Springer, 2002.

[10] M. Bonifacio, P. Bouquet, and P. Traverso. Enabling distributed knowledge management. managerial and technological implications. *Novatica and Informatik/Informatique*, 3(1):23–29, 2002.

[11] U.M. Borghoff and R. Pareschi. *Information Technology for Knowledge Management*. Springer, Berlin.

[12] J. M. Bradshaw, S. Dutfield, P. Benoit, and J. D. Woolley. Kaos: Toward an industrial-strength generic agent architecture. In J. M. Bradshaw, editor, *Software Agents*, pages 375–418. AAAI Press/The MIT Press, 1997.

[13] J. M. Bradshaw, S. Dutfield, B. Carpenter, R. Jeffers, and T. Robinson. KAoS: A generic agent architecture for aerospace applications. In T. Finin and J. Mayfield, editors, *Proc. CIKM'95 Workshop on Intelligent Information Agents*, 1995.

[14] S. Buckingham Shum. Representing hard-to-formalise, contextualised, multidisciplinary, organisational knowledge. In *Proc. AAAI Spring Symposium on AI in Knowledge Management*, Stanford University, Palo Alto, CA, 1997. AAAI Press.

[15] J. Budzik and K. Hammond. Watson: Anticipating and contextualizing information needs. In *Proc. 62nd Annual Meeting of the American Society for Information Science*, 1999.

[16] J. Budzik, K. Hammond, and L. Birnbaum. Information access in context. *Knowledge based systems*, 14(1-2):37–53, 2001.

[17] J. Budzik, M. Ruberry, H. Stein, and K. J. Hammond. Proactively searching the web: Can we agree on what's relevant? In *Demo and Poster Proceedings of ACM HyperText*, 2002.

[18] J. M. Bradshaw R. Carpenter, R. Cranfill, R. Jeffers, L. Poblete, T. Robinson, A. Sun, Y. Gawdiak, I. Bichindaritz, and K. Sullivan. Roles for agent technology in knowledge management: Examples from applications in aerospace and medicine. In B. R. Gaines and R. Uthurusamy, editors, *Papers AAAI Spring Symposium on AI in Knowledge Management*, pages 9–16. AAAI Press, Menlo Park, 1997.

[19] J. Chen, S. Wolfe, and S. D. Wragg. A distributed multi-agent system for collaborative information management and sharing. In *Proc. CIKM-2000*, pages 382–388. ACM Press, New York, NY, USA, 2000.

[20] R. Dieng-Kuntz and N. Matta. *Knowledge Management and Organizational Memories*. Kluwer, 2002.

[21] V. Dignum. *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. PhD thesis, Utrecht University, to appear in 2004.

[22] V. Dignum and F. Dignum. The knowledge market: Agent–mediated knowledge sharing. In V. Marik, J. Müller, and M. Pechoucek, editors, *Multi–Agent Systems and Applications III*, volume 2691 of *LNAI*. Springer.

[23] V. Dignum and F. Dignum. Modelling agent societies: co-ordination frameworks and institutions. In P. Brazdil and A. Jorge, editors, *Proc. 10th Portuguese Conf. on AI*, LNAI 2258, pages 191–204. Springer, 2001.

[24] V. Dignum and P. van Eeden. Seducing, Engaging, and Supporting Communities at Achmea. In *Proc. 4th European Conf. on Knowledge Management, Oxford, UK*, September 2003.

[25] L. van Elst and A. Abecker. Ontologies for information management: balancing formality, stability, and sharing scope. *Expert Systems with Applications*, 23(4):357–366.

[26] L. van Elst and A. Abecker. Domain ontology agents for distributed organizational memories. In *[20]*, 2002.

[27] L. van Elst, F.-R. Aschoff, A. Bernardi, H. Maus, and S. Schwarz. Weakly–structured workflows for knowledge–intensive tasks: An experimental evaluation. In *IEEE WETICE (KMDAP2003)*. IEEE Computer Press, 2003.

[28] P. Feltovich, J. M. Bradshaw, R. Jeffers, and A. Uszok. Order and KAoS: Using policy to represent agent cultures. In *Proc. AAMAS 03 Workshop on Humans and Multi-Agent Systems*, Melbourne, Australia, 2003.

[29] F. Gandon. *Distributed Artificial Intelligence and Knowledge Management: ontologies and multi-agent systems for a corporate semantic web*. PhD thesis, INRIA and University of Nice - SophiaAntipolis, 2002.

[30] F. Gandon. A multi–agent architecture for distributed corporate memories. In *Third Int. Symp. From Agent Theory to Agent Implementation at the 16th European Meeting on Cybernetics and Systems Research (EMCSR 2002)*, pages 623–628, Vienna, Austria, April 2002.

[31] F. Gandon and R. Dieng–Kuntz. Distributed artificial intelligence for distributed corporate knowledge management. In M. Klusch, S. Ossowski, and O. Shehory, editors, *Cooperative Information Agents VI : 6th International Workshop, CIA 2002*, volume 2446 of *LNCS*. Springer, 2002.

[32] F. Giunchiglia, J. Mylopoulos, and A. Perini. The tropos software development methodology: processes, models and diagrams. In M. Gini, T. Ishida, C. Castelfranchi, and W.L. Johnson, editors, *Proc. AAMAS'02*, pages 35–36. ACM Press, July 2002.

[33] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. M. Sarwar, J. L. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In *Proc. AAAI-99*, pages 439–446, 1999.

[34] L. Kerschberg. Knowledge rovers: Cooperative intelligent agent support for enterprise information architectures. In P. Kandzia and M. Klusch, editors, *Proc. CIA-97*, volume 1202 of *LNAI*, pages 79–100, Berlin, February 26–28 1997. Springer.

[35] L. Kerschberg. Knowledge management in heterogeneous data warehouse environments. In Y. Kambayashi, W. Winiwarter, and M. Arikawa, editors, *Proc. 3rd Int. Conf. on Data Warehousing and Knowledge Discovery*, volume 2114 of *LNAI*, pages 1–10. Springer, 2001.

[36] M. Koch. Knowledge management and knowledge agents in Campiello. In B. Lees, H. Müller, and C. Branki, editors, *Proc. Workshop on Intelligent Agents in CSCW*, pages 44–52, Dortmund, Germany, 1998.

[37] O. Kühn and A. Abecker. Corporate memories for knowledge management in industrial practice: Prospects and challenges. In *[11]*, 1998.

[38] A. Abecker L. van Elst, V. Dignum. Number 2926 in LNAI. Springer.

[39] H. Lieberman. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, *Proc. IJCAI-95*, pages 924–929, Montreal, Quebec, Canada, August 1995. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[40] H. Lieberman. Personal assistants for the web: An MIT perspective. In M. Klusch, editor, *Intelligent Information Agents: Agent–Based Information Discovery and Management on the Internet*, pages 279–292. Springer, 1999.

[41] H. Lieberman, N. Van Dyke, and A. Vivacqua. Let's browse: A collaborative web browsing agent. In Mark Maybury, editor, *Proc. IUI-99*, pages 65–70, N.Y., January 5–8 1999. ACM Press.

[42] M. Luck, P. McBurney, and C. Preist. Agent technology: Enabling next generation computing — a roadmap for agent–based computing, version 1.0, 2003. available at http://www.agentlink.org/roadmap/.

[43] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):30–40, July 1994.

[44] G. Mentzas, D. Apostolou, R. Young, and A. Abecker. *Knowledge Asset Management: Beyond the Process-centred and Product-centred Approaches*. Advanced Information and Knowledge Processing. Springer, 2002.

[45] L. Moreau, N. Gibbens, D. De Roure, S. El-Beltagy, W. Hall, G. Hughes, D. Joyce, S. Kim, D. Michaelides, D. Millard, S. Reich, R. Tansley, and M. Weal. SoFAR with DIM Agents: An agent framework for distributed information management. In Jeffrey Bradshaw and Geoff Arnold, editors, *Proc. PAAM-2000)*, pages 369–388, Manchester, UK, April 2000. The Practical Application Company Ltd.

[46] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. Edutella: A p2p networking infrastructure based on rdf. In *Proceedings WWW-2002*. ACM Press, 2002.

[47] I. Nonaka and H. Takeuchi. *The Knowledge–Creating Company*. Oxford University Press, 1995.

[48] G. Papavassiliou, G. Mentzas, and A. Abecker. Integrating knowledge modelling in business process management. In *Proc. ECIS-2002*, Gdansk, Poland, 2002.

[49] H. Van Dyke Parunak. What can agents do in industry, and why? An overview of industrially-oriented R&D at CEC. volume 1435 of *LNCS*, pages 1–18, 1998.

[50] E. Plaza and J.-L. Arcos. Context–aware personal information agents. In M. Klusch and F. Zambonelli, editors, *Proc. CIA-2001*, number 2182 in LNAI, pages 44–55. Springer, September 2001.

[51] A. D. Preece, K. Hui, W. A. Gray, P. Marti, T. J. M. Bench-Capon, D. M. Jones, and Z. Cui. The KRAFT architecture for knowledge fusion and transformation. *Knowledge Based Systems*, 13(2-3):113–120, 2000.

[52] G. Probst, S. Raub, and K. Romhardt. *Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen*. Th. Gabler, Wiesbaden, 1998.

[53] U. Reimer, B. Novotny, and M. Staudt. Micro–modeling of business processes for just–in–time knowledge delivery. In R. Roy, editor, *Industrial Knowledge Management: A Micro–Level Approach*. Springer, 2001.

[54] U. Remus. *Prozessorientiertes Wissensmanagement: Konzepte und Modellierung*. PhD thesis, University Regensburg, 2002.

[55] B. J. Rhodes. *Just-In-Time Information Retrieval*. PhD thesis, MIT Media Lab, May 2000.

[56] B. J. Rhodes and P. Maes. Just-in-time information retrieval agents. *IBM Systems Journal*, 39(3/4):685–704, 2000.

[57] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. van de Velde, and B. Wielinga. *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press, 1999.

[58] S. Staab and H.-P. Schnurr. Smart task support through proactive access to organizational memory. *Knowledge–based Systems*, 13(5):251–260, 2000.

[59] G. Stefanidis, D. Karaginnis, and R. Woitsch. The PROMOTE Approach: Modelling Knowledge Management Processes to Describe Knowledge Management Systems. In *Proc. OKLC 02, Athens, Greece*, April 2002.

[60] J. Vázquez-Salceda and F. Dignum. Modelling Electronic Organizations. In *Multi-Agent Systems and Applications III: 3rd. International/Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'03)*, volume 2691 of *LNAI*, pages 584–593. Springer, 2003.

[61] M. Wooldridge and N. R. Jennings. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[62] M. Wooldridge, N.R. Jennings, and D. Kinny. The gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.

[63] B. Yu, M. Venkatraman, and M.P. Singh. An adaptive social network for information access: Theoretical and experimental results. *Journal of the Applied Artificial Intelligence*, 17(1):21–38, 2003.

## Kontakt

Ludger van Elst
Deutsches Forschungszentrum für Künstliche Intelligenz
Postfach 2080, 67608 Kaiserslautern
Tel.: +49 (0)631 205-3474
Fax: +49 (0)631 204-3210
e-mail: elst@dfki.de
WWW: http://www.dfki.uni-kl.de/

Bild

Andreas Abecker manages the "Knowledge Management" department (WIM) at FZI (Forschungszentrum Informatik Karlsruhe) and co-ordinates the FZI Competence Center for Business Software. His research interests include knowledge representation and reasoning, data and text mining, business-process oriented knowledge management, and agent-mediated knowledge management, as well as technologies for the Semantic Web and Semantic Web Services.

Bild

Ludger van Elst is a research scientist at the DFKI KM department currently preparing his PhD on Ontology Societies in Distributed Organizational Memories. His further research interests include Cognitive Science, Text Comprehension, and Evaluation of KM systems.

| Macro–level Structure | Single–agent System | Homogeneous MAS | Heterogeneous MAS |
|---|---|---|---|
| **Example application** | OntoBroker [58], Jimminy [56], Remembrance Agent [56], MarginNotes [56], Watson [16], Letizia [39], Let´s Browse [41] | MARS [63], DIAMS [19], GroupLens [33], CAPIA [50] | KAoS [28], Knowledge Rover [35], KRAFT [51], Campiello [36], FRODO [3], CoMMA [31], KEx [9], OperA [21] |
| **System–development Level** | | | Organizational analysis (seldom) |
| | Design (*acting on behalf of*–metaphor) | Design (restricted notion of agents) | Design (more comprehensive notion of agents: belief-desire-intention architectures, speech acts) |
| | Implemented mostly with conventional techniques | Implemented on top of middleware for distributed systems (Web, Peer–to–Peer) | Implementation with dedicated agent platforms and/or Semantic Web technology |
| **KM Application Area** | Distribution and utilization of knowledge | Distribution, utilization and preservation of knowledge | Often KM *frameworks* |
| | Adequate presentation to ease internalization | Presentation for internalization, connect people for socialization | Aiming at covering large areas of the knowledge cycle |
| | Mostly "knowledge as product" | Product and (rudimental) process view | Product and/or process view |

Tabelle 1: Typical Operation Points within the Design Space of Agent–based KM Systems.