Towards Agent-Mediated Knowledge Management

Ludger van Elst[1], Virginia Dignum[2], and Andreas Abecker[3]

[1] German Research Center for Artificial Intelligence
– Knowledge Management Department –
elst@dfki.uni-kl.de
[2] Achmea & University Utrecht
virginia@cs.uu.nl
[3] Forschungszentrum Informatik
– Forschungsbereich Wissensmanagement –
abecker@fzi.de

Abstract. In this paper, we outline the relation between Knowledge Management (KM) as an application area on the one hand, and software agents as a basic technology for supporting KM on the other. We start by presenting characteristics of KM which account for some drawbacks of today's – typically centralized – technological approaches for KM. We argue that the basic features of agents (social ability, autonomy, re- and proactiveness) can alleviate several of these drawbacks. A classification schema for the description of agent-based KM systems is established, and a couple of example systems are depicted in terms of this schema. The paper concludes with questions which we think research in Agent-mediated Knowledge Management (AMKM) should deal with.

1 Agents and KM

Knowledge Management (KM) is defined as a systematic, holistic approach for sustainably improving the handling of knowledge on all levels of an organization (individual, group, organizational, and inter-organizational level) in order to support the organization's business goals, such as innovation, quality, cost effectiveness etc. (cp. [33]).

KM is primarily a *management discipline* combining methods from human resource management, strategic planning, change management, and organizational behavior. However, the role of *information technology* as an enabling factor is also widely recognized, and – after a first phase where merely general purpose technology like Internet/Intranets or e-mail[1] were found to be useful for facilitating KM – a variety of proposals exist showing how to support KM with specialized information systems (see, e.g., [4]).

One class of such systems assumes that a huge amount of organizational knowledge is explicitly formalized (or, "buried") in documents, and therefore tries to "connect" knowledge workers with useful information items. Typical systems in this category are Organizational Memory Information Systems (OMs, cp. [1, 24]) which acquire and

[1] Especially large companies often report that these technologies were the first ways to communicate and distribute knowledge across boundaries of hierarchies.

L. van Elst, V. Dignum, and A. Abecker (Eds.): AMKM 2003, LNAI 2926, pp. 1–30, 2003.

structure explicit knowledge and aim at high–precision information delivery services ("provide the right people with the right information at the right time").

On the other hand, expert finder systems or community of practice support don't rely so much on explicitly represented knowledge, but rather bring people together, for instance, to solve a given knowledge–intensive problem (see, for instance [7, 28]). Although such systems also use some explicit knowledge, with respect to the actual knowledge–intensive task, this is more meta than problem–solving knowledge.

Often, Information Technology (IT) research for KM focused on the *comprehensive use* of an organization's knowledge, thus aiming at the completeness of distribution of relevant information. Technically, this is typically supported by centralized approaches: Knowledge about people, knowledge about processes, and domain knowledge is represented and maintained as information in global repositories which serve as sources to meet a knowledge worker's (potentially complex) information needs. Such repositories may be structured by global ontologies and made accessible, e.g., through knowledge portals [75, 52]). Or they may be rather "flat" and accessed via shallow (i.e., not knowledge–based) methods like statistics–based information retrieval or collaborative filtering (this is the typical approach of today's commercial KM tools).

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

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

- Departments, groups, and individual experts develop their particular views on given subjects. These views are motivated and justified by the particularities of the actual work, goals, and situation. Obtaining a single, globally agreed–upon vocabulary (or ontologies) within a level of detail which is sufficient for all participants, may incur high costs (e.g., for negotiation). 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.
- As global views cannot always be reached, a KM system has to be able to handle context switches of knowledge assets, e.g., by providing explicit procedures for capturing the context during knowledge acquisition and for re-contextualizing during knowledge support. An example for context capturing is a lessons-learned system which is fed by debriefings after a project is finished [43, 42]. Here, a typical question pair is: "What was the most crucial point of the project's success? What are the characteristics of projects where this point may also occur?"

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

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

R2 *There is an inherent goal dichotomy between business processes and KM processes*: For companies as a whole as well as for the individual knowledge worker KM processes do not directly serve the operational business goals, but are *second order processes*[2]. Within an environment of bounded resources, knowledge workers will always concentrate on their first order business processes. This means they optimize their operational goals locally and only invest very little to fulfil strategic, global KM goals.[3] 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 usually 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, the 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, is "wicked problem solving" (cf. [15, 21, 22])*: This means that a precise a-priori description of how to execute a task or solve the problem doesn't exist, and consequently, it cannot be said in advance when knowledge should be captured, distributed, or used optimally. An optimal solution for KM problems and the respective knowledge and information flows cannot be prescribed entirely from start to finish, because goals may change or be adapted with each step of working on a task. Therefore knowledge workers and KM systems must be flexible enough to adapt to additional insights and to proactively take opportunities when they arise during work. Solving "wicked problems" is typically a fundamental social process. A KM system should therefore support the necessary complex interactions and underlying, relatively sophisticated processes like planning, coordination and negotiation of knowledge activities.

A phenomenon closely related to this is that KM is very much about *personal relationships*. People want to be recognized as experts, and they are much more willing

[2] There are a couple of exceptions to this, like R&D departments which have knowledge generation as first order goal. For a discussion of operational processes vs. knowledge processes, see, for instance, [68, 78].

[3] In other words, employees will mostly find a way to get their business done, even if processes and tool support are bad, whereas KM tasks will simply be omitted. This has been our experience in KM systems building from our very first requirements gathering on [47].

to share knowledge face-to-face in collaborative problem-solving and expert chats than putting it anonymously into a central knowledge store. Hence flexible point-to-point connections for powerful online communication and collaboration, as well as individual solutions for knowledge storage, identification, and communication must be allowed.

R4 *KM has to deal with changing environments*: In addition to the intrinsic problems described above, KM systems typically reside in environments which are subject to frequent changes, be it in the organizational structure, in business processes, or in IT infrastructure. Centralized solutions are often ill–suited to deal with continuous modifications in the enterprise, e.g., because the maintenance costs for detailed models and ontologies simply get too high.
Furthermore, the implementation of KM systems often follows a more evolutionary approach where functionalities are not implemented "in one step" for a whole company, but partial solutions are deployed to clearly separated 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.[4]

Keeping these requirements in mind, let's have a look at scenarios which are considered to be rewarding tasks for agent-based software solutions. We quote a number of characteristics from [60] (but similar arguments can be found in many books about multi-agent systems) typically indicating that a scenario could be a good application area for agent technology: agents are best suited to applications that are modular, decentralized, changeable, ill-structured, and complex.

Although the match between these five salient features and the KM requirements R1 – R4 listed above is already obvious, we want to elaborate a bit more explicitly on this match. Let us start with the *weak definition* of agents [83] (with the definitional features *autonomy*, *social ability*, *reactive behavior, and proactive behavior*). Now we will see why agent-based approaches are especially well–suited to support KM with information technology:

In the first place, the notion of agents can be seen as a natural metaphor to model KM environments which can be conceived as consisting of a number of interacting entities (individuals, groups, IT, etc.) that constitute a potentially complex organizational structure (see R1, but also R4). Reflecting this in an agent-based architecture may help to maintain integrity of the existing organizational structure and the autonomy of its subparts. Autonomy and social ability of the single agents are the basic means to achieve this.

Reactivity and proactivity of agents help to cope with the flexibility needed to deal with the "wicked" nature of KM tasks (see R3). The resulting complex interactions with the related actors in the KM landscape and the environment can be supported and modeled by the complex social skills with which agents can be endowed.

Proactiveness as well as autonomy help accomodating to the reality that knowledge workers typically do not adopt KM goals with a high priority (see R2).

[4] This requirement of connecting several smaller existing KM islands to create a bigger picture, also fits very well with the frequently suggested KM introduction strategy of looking for "quick wins" (cp. [81]).

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 uses the system will have. Both arguments reflect pretty well the technical consequences of abstract requirements such as R4 and R3 (changing environments demand continuous reconfiguration, the unpredictable nature of wicked-problem solving require flexible approaches), R2 (competition between operational work and KM meta work call for stepwise deployment and highly integrated KM solutions), or R1 (already existing local solutions must be confederated).

There have been a number of more or less theoretical analyses of requirements and ambitious approaches to agent-based solutions for KM (see, e.g., [56, 72]), as well as 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 comprehensive agent-based KM frameworks (like FRODO, CoMMA, Edamok [3, 31, 11, 36], some of them are included in this book). We are well aware that nowadays we are far from reaching a state where we can oversee all methodological, technological, and practical benefits and prospects, problems and pitfalls, and challenges and achievements of Agent-Mediated Knowledge Management. But we hope and we are pretty sure that this paper as well as this volume gives a good idea of the AMKM landscape, opens up some new ways for interesting future work and shows how far we have already come.

2 A Description Schema for Agent-Based KM Approaches

In research as well as in first generation "real-world applications" several agent–based systems exist to support various aspects of Knowledge Management, from *personal information agents* for knowledge retrieval to *agent–based workflows* for business process–oriented KM. In order to be able to compare different agent approaches to KM, we need to describe agent and multi–agent architectures in a way that abstracts from the particularities of individual implementations, but still captures their relevant characteristics. A couple of helpful classification schemas for single agents and multi–agents systems have already been proposed (e.g., Franklin and Graesser's taxonomy of agents [35]), discriminating agents for example by their tasks (information filtering, interface agents etc.), their abstract architecture (e.g., purely reactive vs. agents with state) or concrete architecture (e.g., belief-desire-intention vs. layered) architectures (cf. [82]), or other specific features (mobility, adaptivity, cooperativeness, etc.).

For instance, [61] presented an interesting top-level characterization of agent applications, basically distinguishing three kinds of domains:

1. *Digital* domains where the whole environment of the agents is constituted from digital entities, as is the case, e.g., in telecommunications or static optimization problems.
2. *Social* environments where software agents interact with human beings.

3. Electromechanical environments where agents manipulate and experience the non-human physical world via sensors and actuators, as is the case, e.g., in robotics, factories, etc.

A further classification dimension can be added directly because besides the domain to be handled by the agents we also have to consider the kinds of interfaces to be provided by an agent-based application. Here we have the same options as above: we need social interfaces to integrate people, digital interfaces to interact with other agents, and electromechanical interfaces to link to the physical world. In the case of KM applications we normally have to consider a (highly) social environment with both social and (usually a number of different) digital interfaces.

For the purpose of this paper, we propose a description schema that is on the one hand more specific than these classifications 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 also enable the aggregation of different functionalities (such as planning, learning, coordination, etc.) in a conceptually embodied and situated whole [51]; agents also provide ways to relate directly to these abstractions in the design and development of large systems.

In Knowledge Management, not only are the IT systems highly complex and distributed, but also the organizational environment in which these systems are situated. Especially in more comprehensive KM approaches, the complexity of the organization has to be reflected in the IT architecture. Often, "real world entities" of the organization have a relatively direct counterpart in the computer system, leading to a rather tight coupling between the real and the virtual worlds. Therefore, an organizational analysis is commonly an integral part of methodologies for the development of Knowledge Management IT (see, e.g., the CommonKADS [74], or the DECOR [59] 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.

So, due to the fundamentally social nature of KM applications, the agent paradigm can be — and actually has been — applied at different development levels, such as analysis, modeling and design, and not just to represent technological components of

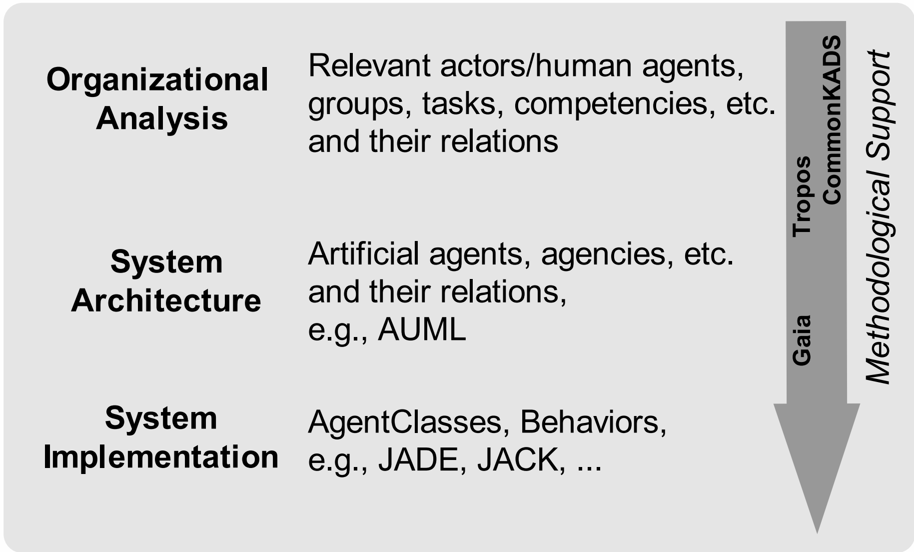

Fig. 1. Notion of Agents at Different Stages in the Development Cycle of an Agent–Based KM System

implemented systems. Figure 1gives 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. [83]) 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 [40] and Gaia [85] 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. The hope is, of course, that on the basis of a high correspondence of the primitives on each level these transitions will be smooth and less error–prone. Even though such methodologies provide a powerful tool to design multi-agent systems, and are currently widely used, they are not always suitable to deal with the complexity of fully fledged KM environments, including openness and heterogeneity. In [27] 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 issues. A recent proposal for a methodology for agent societies that meets these requirements, is presented in [25].

However, even when it seems likely that the entire development life cycle for KM applications can benefit from the concept of agents, we are well aware that in concrete, real–life situations often pragmatic reasons[5] may 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-to-peer networks or web services should not necessarily hinder an agent–oriented analysis and system design[6].

2.2 Macro-level Structure of the Agent System

Agent theories, *abstract agent architectures*, and *agent languages* as defined in [83] 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:

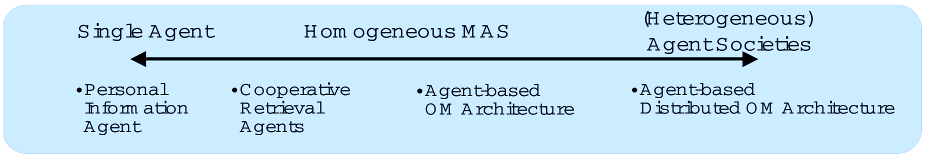

Fig. 2. Degree of Sociability

– *Single–agent architectures* are at one end of the spectrum. Typical examples come from the area of user interface or personal information agents which build a model of a user's interest and behavior, and exploit this knowledge to support him or her by providing relevant information, e.g., from the Web. 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).

[5] In [84], Wooldridge and Jennings nicely describe classes of pitfalls for the development of agent–based systems, including the "overselling", "being dogmatic", and "agents as silver bullet" pitfalls. Parunak [60, 61] also discusses the pragmatics of agent–based software development in real–world settings.

[6] Actually we observe that technologies like P2P and web services incorporate many aspects of the notion of agents when encountering application domains that have characteristics like those described in Section 1 for KM applications.

– *Homogeneous multi–agent architectures* already have a higher degree of sociability. Agents can co-operate with other agents in order to solve their tasks. Homogeneity means that the system consists mainly of *one* type or class of agents. These agents do not necessarily have to have exactly the same goals, but their tasks and capabilities are comparable. Agent–based collaborative filtering is a typical example for this class of MAS: 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. Nevertheless, all agents may have individual information collection and integration strategies.
– *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 [44], MOMIS/MIKS [8]) 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. All these different agent types have to cooperate and bring in their complementary expertise in order to accomplish the overall goal of the system.

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? On the basis of a market model? As a fully connected network? Or in a hierarchical manner?
– *open vs. closed system*: Can new agents enter the system? If yes, does their agent class (competencies, purpose, etc.) have to be known in advance? Or can new types of agents be integrated easily (even at runtime)?
– *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? Or is the agent's social behavior only locally controlled and the system's behavior completely emergent?

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 [79]. Similarly, virtual organizations can potentially take advantage of the new electronic environments through coalition formation among disparate partners to form aggregate entities capable of offering new, different or better services than might otherwise be available. To design such systems requires a theory of organization design, and knowledge of how organizations may change and evolve over time. Sociological organization theory and social psychology are clearly important inputs to the design. Moreover, for the design of open societies, political theory may be necessary. Open systems permit the involvement of agents from diverse design teams, with diverse objectives, which may all be unknown at the time of design of the system itself. How the system as a whole makes decisions or agrees on joint goals will require the adoption of specific political philosophies, for example whether issues are subject to simple majority voting or transferable preference voting, etc. (cp. [51]).

Of course, the above examples for different degrees of sociability — single–agent, homgeneous/heterogeneous MAS — 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. Or, a system may be mainly homogeneous, but has one specialist agent for a certain task. And even between several aspects of communication the structure of the system may differ. So, the topology for making decisions may be a hierarchy, while information may be spread based on a market or fully connected network model [29]. It is also clear that there are dependencies between the three facets of system description. Not all possible combinations fit equally well together, not all of them are equally useful. For instance, if we have a highly structured agent society (like the electronic institutions outlined above) we can normally profit from known social structures when designing effective co-ordination, communication, and decision-making mechanisms, and do not have to use such a general, but "expensive" mechanism as a fully connected network. On the other hand, the more social structure is explicitly implemented into an agent society, the more "closed" this society might be in the sense that entering it will probably be based on a well-specified procedure, depending both on the current status of the society and on the capabilities and goals of a new agent that wants to enter it. On the other hand, if we have a relatively "democratic" way of co-ordination, like a market model, and a completely implicit social structure, it might be pretty easy for such an agent society to act as a pretty "open" system.

2.3 KM Application Area

The two classification dimensions for multi agent systems described in the previous subsections are not directly related to applications in the KM domain. Up to now, we looked at the *level in system development* where the notion of agents is used, and at the *macro–level structure* of the agent system[7]. The third dimension for characterizing agent–based KM applications, described in this subsection, deals with the specific knowledge management functionality of the system: What is the scope of the systems? Which Knowledge Management processes or tasks are supported?
In this paper, we do not want to prescribe a detailed framework for this dimension, but only want to gather and offer some possibilities and general directions.

Principally, all high–level Knowledge Management models can be seen as a starting point to form the vocabulary for this dimension, and there are many such KM models. We will start with the famous KM cycle by Probst et al. [64] 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 what knowledge exists in an organization, what the knowledge containers are, who the stakeholders are, etc.

[7] Though it should be noted that the emphasis on the *degree of sociability* as an important dimension of characterization is strongly biased by our theoretical analysis of KM in Section 1.

- *Acquisition* is the process of integrating external knowledge into an organization.
- *Development processes* generate new knowledge in the organization.
- *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).

Originating in the management sciences, Probst et al.'s view has been widely adopted and adapted in technology–oriented KM literature (e.g., [1, 76]). Likewise, the classical model of Nonaka and Takeuchi [58]—which focuses on knowledge generation—can be used to describe the KM application area of a system. These authors claim that new knowledge is created by 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 in expert system research 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 of this type of knowledge creation process.
- *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 [47, 54]. The latter emphasizes the management of explicit knowledge contained in ”kowledge products” such as databases, documents, formal knowledge bases etc.; the former focuses on human beings and their internal knowledge, i.e., the ”process of knowing” and the ”process of knowledge exchange” between people. Typical systems with a product–oriented view are document retrieval agents. Expert finder systems, on the other hand, take a more process–oriented view. 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*. The theoretical analysis of Knowledge Management characteristics in Section 1 may be the source of further possible application areas for information technology, e.g., facilitating trust, motivating users to share knowledge, or establishing group awareness.

Concrete agent–based KM applications may deal with one or a few of these aspects, or they may be more comprehensive frameworks that try to cover large parts of the KM cycle. In the following section we will analyze existing agent-based KM applications, illustrative for the different approaches.

3 Exemplary Agent-Based KM Applications

In the previous section, we proposed three dimensions to describe agent–based Knowledge Management systems: i) the system development level (analysis, design, implementation), ii) the macro–level structure of the system (single agent, heterogeneous, or homogeneous MAS), and iii) the KM application area (knowledge distribution, generation, use, etc.). In this section, we will present some examples of agent-based systems developed to support and/or model Knowledge Management 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. Since compiling a complete overview of the systems in all three dimensions is well beyond the scope of this paper, we briefly sketch some systems which we consider typical for the specific approach. Our aim is to present current developments in Agent-Mediated Knowledge Management, indicate their differences to conventional approaches, expose their benefits, and suggest areas for further work.

3.1 Predominantly Single Agent Approaches

Most KM support systems that take a single agent approach are *User Interface Agents* or *Information Agents*. A User Interface Agent embodies the metaphor of "a personal assistant who is collaborating with the user in the same work environment" [53]. Though this rather general definition would comprise agent support for all kinds of KM activities that a knowledge worker can perform (e.g., distribute knowledge, generate new knowledge), virtually all systems in this class are information agents[8]. These agents 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 [77, 73]. The agent relies on an explicit model of the business process the user is engaged with 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.

Two variants of the system are available, a *reactive* and a *proactive* one. In the reactive case, the user triggers the agent by selecting a specific (pre–modelled) query in a specific application context. The agent then tries to retrieve relevant knowledge from the Organizational Memory and passes it on directly to the respective application

[8] For an overview of personal information agents, also for other tasks like expert finding and information visualization see [49].

that triggered the information need and thereby to the user. The reactive agent must have *complete* knowledge about the process context and about the information needs. The proactive agent, on the other hand, relaxes these two requirements, i.e., the application context and the relevant queries may be only partially defined when the agent becomes active. Instead, the agent has a proactive inferencing mechanism which employs heuristics to retrieve relevant information based on uncomplete context and query specifications. In order to cope with the potentially huge number of possible results and related problems (e.g. storage, processing time) from inferencing with underspecified context, the proactive agent is equipped with a mechanism for bounded resource consumption. For their actual knowledge retrieval step, OntoBroker agents, both reactive and proactive, exploit the ontology-based structure of the Organizational Memory .

However, many personal information agents are designed for an environment where such an ontological structure of the information sources cannot be assumed, e.g., the World Wide Web. In this case, agents often rely on standard information retrieval techniques for searching. Rhodes and Maes [70] present three *just-in-time* information retrieval (JITIR) agents: The *Remembrance Agent* continually presents a list of documents that are related to a document that is 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* [69] for the actual information retrieval step.

Nevertheless, the primary contributions of research in personal information agents are not so much the various core retrieval techniques (from statistics–based similarities of text documents up to ontology–based access to formalized knowledge items), but 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, a wide range of approaches are covered in literature, 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 [17] or Letizia [48]).
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* [70], 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.

We now discuss the characteristics of personal assistants along the other two characterization dimensions for AMKM applications described in section 2. 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 currently mostly implemented using conventional programming techniques, i.e., without using a more general "agent development kit for personal information agents". A well–known exception is Letizia, developed at MIT

[48]. With respect to the KM application area, personal assistants, as user–directed approaches, 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 of advanced vendors. In those applications, the agent term is often not used in the narrower technical sense, but merely as a communication or as a design metaphor, but not built upon dedicated agent software platforms. There is a clear, but indirect, link between the functionalities achieved by such systems and our KM software requirements R1 – R4 defined above. Usually one can see that the software functionalities provided here are useful, because they address issues caused by our items R1 – R4 (e.g., in frequently changing environments, push services achieved by personal information agents are much more important than in stable environments, since an agent can continuously monitor whether some relevant change has happened). Altogether, though the software functionalities are stable to some extent and apparently useful, the logical next step for research and application has seldom been done, namely a rigorous assessment of usability and usefulness issues. There are a few specific experiments about evaluation of Personal Information Agents and the influence of process-aware, proactive information delivery, respectively (see [16, 18, 32, 70]), 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

As described in Section 2.2, homogeneous multi–agent systems are formed by several agents belonging mainly to a common "agent class", i.e., on an abstract level they have comparable competencies and goals (albeit they might act on behalf of different users)[9]. *Pure* homogeneous multi–agent systems are rarely found in literature. Typically, facilitation functions (e.g., matchmaking and management of collaboration) are encapsulated as (centralized) service agents, different from the other agents, which might be homogeneous. Examples of such "weakly homogeneous" systems, mostly specialized on *one* KM task, are presented later in this section.

An obvious extension to the personal information agents described in the previous section is to see each user not only as an information consumer, but also as a provider. In this case, besides retrieval and presentation support, the personal agent should assist the user in serving as a source of information. A very simple example for such agents are the clients for peer-to-peer file sharing support like Kazaa, ED2K, or – in the domain of learning resources – Edutella [57]. These agents have specialized interfaces for expressing queries, passing them on to other agents and displaying the results. But they

[9] This definition identifies homogeneous multi–agent systems as close conceptual relatives of peer-to-peer (P2P) systems, even though their implementational basis can be quite different (cf. Section 2.1).

are also able to receive queries and process them by answering with result documents or by passing a query to other agents. Such interaction between different personal assistants can be considered as a multi–agent system. In the following, more elaborate approaches are also described.

MARS, an *adaptive social network for information access* described and evaluated in [86] has a purely homogeneous structure that is based on the idea described in the previous paragraph. Each agent basically has 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 the other agents in the network with respect to the two aspects. This means they assess the other agents' expertise (ability to produce correct domain answers) as well as their ability to produce accurate referrals.

DIAMS [20] 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. In order to promote easy information sharing and exchange, an object–based structure is used for the information repositories. DIAMS furthermore uses a flexible hierarchical presentation of information integrated with indexed query functionalities to ensure effective information access. Automatic indexing methods are employed to support translation between user queries and communication between agents. Collaboration between users is aided by the easy sharing of information and is facilitated by automated information exchange. Connections between users with similar interests can be established with the help of matchmaker agents.

The focus of the research described in [62] 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 [41]. However, such systems are often not designed as agent systems. Due to their focus on *one* KM task (e.g., recommendation of one specific type of information objects) and a relatively controlled environment, centralized implementations are common. For example *Let's Browse* [50], the successor of the personal information agent *Letizia* [48], 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” (e.g., with respect to the quality of the recommendations) equivalent monolithic system.

As with the single agent approaches presented above, homogeneous multi–agent systems applications to KM are mainly seen at the modelling level of development. On the other hand, in relation 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 than single agent approaches do, and can support teams and groups, as well as individual users. Homogeneous multi–agent 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, such as those described in the next subsection.

3.3 Heterogeneous Multi-agent and Society-Oriented Approaches

Heterogeneous multi–agent systems not only consist of a potentially high number of agents, but these agents also belong to different classes. This means the agents have diverse competencies and types of goals. The heterogeneity can be due to the large number of “real–world” entities of the organization that are reflected in the system, or due to a purely functional decomposition from a software engineering point of view. Also, the more Knowledge Management functions a systems covers, the more heterogeneous the system will be. The systems we present in this section comprise both types of heterogeneity. Some of them only have a limited scope in terms of KM functionality (e.g., storing and retrieving knowledge objects), but encapsulate various service functions in separate specialized agents. Others are meant to be more comprehensive KM backbones and therefore employ agents for more diverse aspects like process support, retrieval support, and personalization. The society–oriented approaches we sketch at the end of this section demonstrate a potential way to cope with this heterogeneity and the complexity of such systems.

The design of many agent–based Knowledge Management systems emerges from the “standard” three–tier enterprise information architectures that are often the basis for business applications (e.g., [55, 34, 45] and others):

– 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 [14, 19], a generic agent architecture for aerospace applications, is quite an early agent–based system for the management of technical information contained in documents, that is based on such a layer model. Aiming mainly at flexible information delivery from heterogeneous information sources in a distributed environment, 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 [13]. Recent versions of KAoS also incorporate social aspects in agent communities [34]. However, the relevance of this approach for Knowledge Management applications has not yet been discussed.

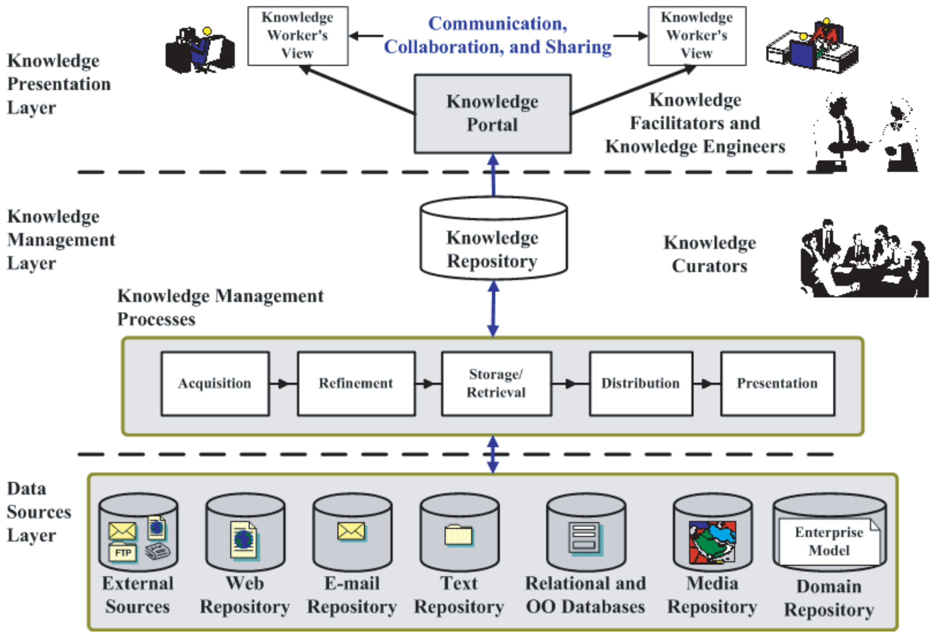

Fig. 3. Three-layer KM Architecture [45] (reprinted with kind permission)

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 [44] 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 [45] (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* (cf. [80]) is essential. One project that deals explicitly with the fusion of knowledge from multiple, distributed and heterogeneous sources is *KRAFT* [63]. KRAFT has an agent–based architecture, in which all knowledge processing components are 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.

Sharing knowledge between people can take place directly, e.g., in face–to–face collaborations or with synchronous media like video conferencing, or indirectly, e.g., via information objects that are exchanged. Even hybrid approaches are possible, for example by analyzing the use of information objects and establishing direct links between people using the same objects. This direction was investigated in the *Campiello* project [46]. Campiello 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[10] 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 [67] 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,

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

which relies on appropriate formal models and ontologies as a basis for common understanding and automatic processing capabilities [1]. Figure 4 shows FRODO's four layer architecture for each Organizational Memory (OM): i) The *application layer* manages the process context in form of weakly–structured workflows [32]. ii) The *source layer* contains information sources with various levels of formalization (process models, text documents, etc.). iii) The *knowledge description layer* provides uniform access to the sources by means of ontologies. iv) By utilizing these descriptions, the *knowledge access layer* connects the application with the source layer. Agents in a FRODO OM 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.

In order to cope with the heterogeneity and complexity, as well as to constrain the overall behavior of the system, agents in FRODO are organized in societies. Therefore, a FRODO agent is not only described by its knowledge, goals and competencies, but also by its *rights and obligations*. The description of ontology societies in [30] exemplifies FRODO's concept of socially–enabled agents for KM. The implementation is based on the FIPA–compliant agent platform JADE[11].

FRODO's approach towards Distributed Organizational Memories is strongly driven by the general considerations of KM presented in Section 1. The overall goal is to find a balance between the organizational KM needs and the individual needs of knowledge workers. This is reflected in the way domain ontologies are handled in the distributed environment. Coming from a comparable analysis of KM characteristics [11], the *Edamok* project[12] also aims at enabling autonomous and distributed management of knowledge. Edamok completely abandons centralized approaches, resulting in the peer–to–peer architecture *KEx* [10]. 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.

An approach which is closely related to FRODO and Edamok has been developed in the *CoMMA* project [9]. The CoMMA architecture also employs societies of agents for personalized information delivery [38]:

[11] http://sharon.cselt.it/projects/jade/
[12] http://edamok.itc.it/

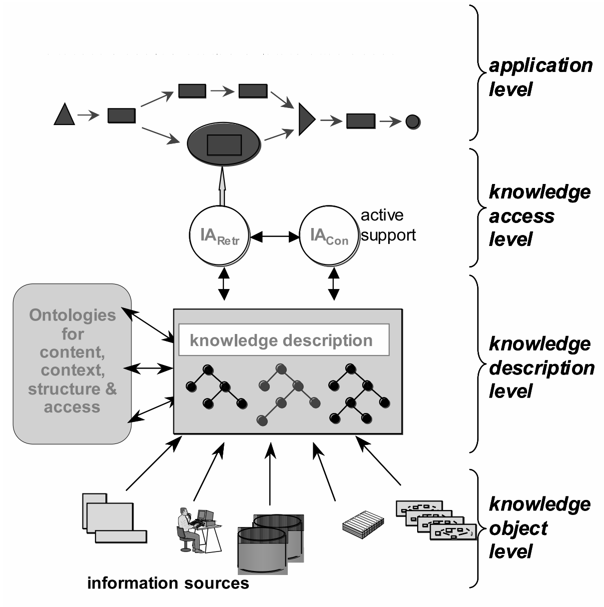

Fig. 4. FRODO Architecture for a Single Organizational Memory

- 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 [39]. The position of an agent in a society is defined by its role [37]. The system was implemented on top of the JADE agent platform, and special attention was paid to the use of XML and RDF for representing document annotations and queries.

As already stated above, business processes play an important role for providing context of knowledge generation and reuse. The utilization of the process context ranges from rather static access structures to knowledge objects (e.g., as a browsing hierarchy in a portal, or as an annotation that can be exploited by a search agent) to workflow–like agent–supported execution and the triggering of proactive information delivery.
An interesting system that uses a concrete, domain–specific process model for its information support is *K-InCA* [71, 6]. In K-InCA, agents are used to guide, monitor and stimulate managers towards the understanding of KM concepts and the adoption of KM practices in organizational contexts, so that the system behaves as a personal

KM coach for its users. The underlying process model of a K-InCA agent describes how changes are adopted by individuals and thereby new knowledge is incorporated into a person's spectrum of working habits. K-InCA agents can be seen as experts on organizational behavior and change management, assisting users in the transition from their current working habits to new habits that integrate some new behavior (e.g. KM practices, entrepreneurial attitude, etc.). The system allows for different modes of interaction (practice and coaching), aiming at bringing the user to adopt a desired behavior. In order to achieve this goal, agents react to the current user activity on the basis of information stored in a domain model and a user model, as well as through interaction with other agents.

With respect to the question of where in the development cycle the notion of agents is used (cf. Section 2.1), most of the systems presented up to now take a kind of *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 (e.g., most user interface agents), others build upon dedicated platforms for agent systems (e.g., based on the FIPA[13] specifications). Only a few of the described systems complement their architectures with an agent–based Knowledge Management methodology for guiding the development of such a system in an organizational context (e.g., Edamok, stemming from the general MAS methodology Tropos and developing it towards KM).

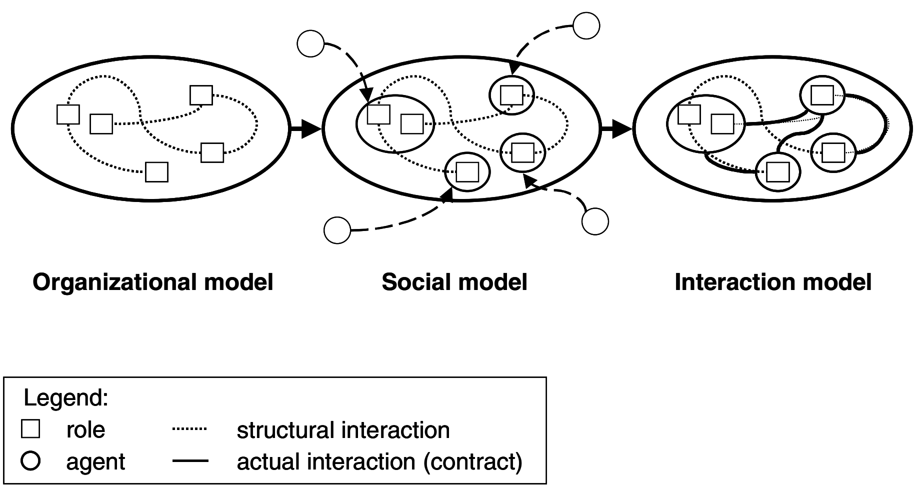

Fig. 5. Relations between the Different Models in OperA

[13] http://www.fipa.org/

A recent proposal for a design methodology specifically tailored to agent societies is *OperA* [25]. This methodology is based on a three–tiered framework for agent societies that distinguishes between the specification of the intended organizational structure and 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. Figure 5 depicts the relation between the different models. The OperA methodology supports the specification of an Organizational Model by analyzing a given domain and determining the type and structure of the agent society that best models that domain is described in [29]. The methodology provides generic facilitation and interaction frameworks for agent societies that implement the functionality derived from the co–ordination model applicable to the problem domain. Standard society types such as market, hierarchy and network, can be used as starting points for development and can be extended where needed and determine the basic norms and facilitation roles necessary for the society. These coordination models describe the different types of roles that can be identified in the society and issues such as communication forms, desired social order and co-operation possibilities between partners. 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 [26].

3.4 Description of Example Systems: Concluding Remarks

In Section 2, we presented a framework for the description of agent–based Knowledge Management systems with the main dimensions *system development level*, *macro–level structure*, and *KM application area*. The analysis of several KM systems in Sections 3.1–3.3 shows that this space is not fully covered by the research approaches and prototypes presented (see also Table 1). Two factors may contribute to this fact:

1. Though at first glance, only the last dimension — the application area — seems to be KM specific, the dimensions are not really independent. If for example, knowledge use and internalization by specialized presentation techniques is the focus of research, an "agentification" of all knowledge sources may well be technological overkill. Or, the other way around, comprehensive KM frameworks may require more powerful agent architectures to cope with the complexity of various KM tasks.

Macro–level Structure	**Single–agent System**	**Homogeneous MAS**	**Heterogeneous MAS**
Example application	OntoBroker [77], Jimminy [70], Remembrance Agent [70], MarginNotes [70], Watson [17], Letizia [48], Lets Browse [50]	MARS [86], DIAMS [20], GroupLens [41], CAPIA [62]	KAoS [34], Knowledge Rover [45], KRAFT [63], Campiello [46], FRODO [3], CoMMA [38], KEx [10], K-InCA [71], OperA [25]
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, connecting 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

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

2. Sparsely populated areas in the design space spanned by the description framework just may not yet be investigated by current research.

While the first case covers operating points that simply make no sense for agent–based Knowledge Management, the second may lead towards new research aspects. We think that some papers in this book are well suited for stimulating thoughts in new directions.

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, and ii) to provide a framework for the analysis and description of agent–based KM systems.

In Section 1 we emphasized four main characteristics of Knowledge Management which in our opinion fundamentally account for the suitability of agent–based systems for supporting KM:

– The *distributed nature of knowledge* may — from a technical point of view — raise special challenges, but for an organization and its individuals it is the only way to cope with the complexity of knowledge and should therefore be seen as an imperative and not as a nuisance. Agents are a natural form to represent that knowledge is created and used by various actors with diverse objectives. Socially–enabled agents can also help to tackle derived questions like accountability, trust, etc.
– The *inherent goal dichotomy between business processes and KM processes* leads to the fact that knowledge workers typically do not adopt KM goals with a high priority. Proactive agents may be able to stand in for (or at least remind the knowledge worker) when KM tasks fall behind.
– *Knowledge work as well as KM in general is "wicked problem solving"* without a fixed a-priori description of goals and solution paths. Reactive and proactive behavior of agents help to reach the necessary degree of flexibility. Social skills of agents can facilitate the management of the complexity of interactions that are typical for wicked problem solving.
– *The continuously changing environments* are not entirely an intrinsic KM characteristic, but nevertheless any IT support for KM has to deal with this given factor. Agent approaches allow for extensibility and openness in situations where it is impossible to know at design time exactly which components and uses the system will have.

In Section 2 we developed a framework for the description of agent–based KM systems with the main dimensions

– *system development level* (analysis, design, implementation),
– *macro–level structure* (single agent, heterogeneous, or homogeneous MAS), and
– *KM application area* (knowledge distribution, generation, use, etc.).

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 a fact that the vast 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 information systems.

As the development of a comprehensive “Agents–in–KM Roadmap” is well beyond the scope of this paper, we just briefly sketch a couple of directions that may be interesting for future research:

1. *Socio–technical*: How can the teamwork of human knowledge workers and artificial agents (that might act “on behalf of” people) be balanced? Questions from human–computer interaction arise here, 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 be better exploited for KM tasks? Which *new* functionalities can agent–based systems offer to KM?
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” in order to facilitate transitions between different phases in the development cycle?
4. *Evaluation of agent–based KM*: How well does the integration of (non agent–based) legacy systems into agent environments work in real–world applications (case studies)? How easily can new agent–based components really be integrated into an existing system? Which evaluation paradigms can be used to make different KM applications more comparable (agent–based vs. agent–based, but also agent–based vs. “traditional”)?

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 using conventional technology, especially when only the implementation level is considered. However, we believe that agent technology helps building KM systems faster and more flexibly. We think that the results presented in this paper and in the other contributions in this book have the potential to strengthen the hope that an agent–oriented view (regardless of the implementation technology) leads to a more human–centered, more agile, and more scalable KM support.

References

[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. *International Journal 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 *Proceedings of the 5th International Conference On Enterprise Information Systems, 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. To appear.

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

[6] A. Angehrn, T. Nabeth, L. Razmerita, and C. Roda. K-inca: Using artificial agents for helping people to learn new behaviours. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2001)*, pages 225–226, Madison USA, August 2001. IEEE Computer Society.

[7] 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.

[8] 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 *Lecture Notes in Computer Science*, pages 22–49. Springer, 2003.

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

[10] M. Bonifacio, P. Bouquet, G. Mameli, and M. Nori. Kex: a peer–to–peer solution for distributed knowledge management. In *Proceedings of the Fourth International Conference on Practical Aspects of Knowledge Management (PAKM02)*, volume 2569 of *Lecture Notes in AI*, pages 490–500. Springer, 2002.

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

[12] U.M. Borghoff and R. Pareschi. *Information Technology for Knowledge Management*. Springer, Berlin, Heidelberg, New York, 1998. Some articles of this volume also appeared in a special issue of the Journal of Universal Computer Science, vol. 3, no. 8, Springer, 1997.

[13] 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.

[14] 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, *Proceedings of the CIKM '95 Workshop on Intelligent Information Agents*, Baltimore, Maryland, 1995.

[15] S. Buckingham Shum. Representing hard-to-formalise, contextualised, multidisciplinary, organisational knowledge. In *Proceedings of the AAAI Spring Symposium on Artificial Intelligence in Knowledge Management*, Stanford University, Palo Alto, CA, 1997. AAAI Press.

[16] J. Budzik and K. Hammond. Watson: Anticipating and contextualizing information needs. In *Proceedings of the Sixty-second Annual Meeting of the American Society for Information Science*, 1999.

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

[18] 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.

[19] 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 from the AAAI Spring Symposium on Artificial Intelligence in Knowledge Management*, pages 9–16. AAAI Press, Menlo Park, 1997.

[20] J. Chen, S. Wolfe, and S. D. Wragg. A distributed multi-agent system for collaborative information management and sharing. In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 382–388. ACM Press, New York, NY, USA, 2000.

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

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

[23] R. Dieng and J. Vanwelkenhuysen, editors. *Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop KAW'96, Special Track on Corporate memory and Enterprise Modeling*. November 1996.

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

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

[26] 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-Verlag.

[27] V. Dignum and F. Dignum. Modelling agent societies: co-ordination frameworks and institutions. In P. Brazdil and A. Jorge, editors, *Progress in Artificial Intelligence: 10th Portuguese Conference in Artificial Intelligence*, Lecture Notes on Artificial Intelligence 2258, pages 191–204. Springer-Verlag, 2001.

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

[29] V. Dignum, H. Weigand, and L. Xu. Agent societies: Towards framework-based design. In M. Wooldridge, G. Weiss, and P. Ciancarini, editors, *Agent-Oriented Software Engineering II*, Lecture Notes on Artificial Intelligence 2222, pages 33–49. Springer-Verlag, 2001.

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

[31] L. van Elst and A. Abecker. Domain ontology agents for distributed organizational memories. In *[24]*, 2002.

[32] 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 Workshop on Knowledge Management for Distributed Agile Processes: Models, Techniques, and Infrastructure (KMDAP2003)*. IEEE Computer Press, to appear 2003.

[33] M. Eppler. Glossary definition: Knowledge management. NetAcademy: `www.knowledgemedia.org/modules/glossary/view\_def.php?id=mediamanagemen%t-20`, 2002.

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

[35] S. Franklin and A. Graesser. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In *Intelligent Agents III. Agent Theories, Architectures and Languages (ATAL'96)*, volume 1193 of *LNCS*, Berlin, Germany, 1996. Springer-Verlag.

[36] 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.

[37] F. Gandon. A multi–agent architecture for distributed corporate memories. In *Third International Symposium 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.

[38] 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*, Berlin, Germany, 2002. Springer-Verlag.

[39] F. Gandon, R. Dieng, O. Corby, and A. Giboin. A multi–agent system to support exploiting an xml–based corporate memory. In *[66]*, 2000.

[40] 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, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, pages 35–36. ACM Press, July 2002.

[41] 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 *Proceedings of the 1999 Conference of the American Association of Artifical Intelligence (AAAI-99)*, pages 439–446, 1999.

[42] G. van Heijst, R. van der Spek, and E. Kruizinga. Organizing corporate memories. In *[23]*, 1996.

[43] G. van Heijst, R. van der Spek, and E. Kruizinga. The lessons learned cycle. In *[12]*, 1998.

[44] L. Kerschberg. Knowledge rovers: Cooperative intelligent agent support for enterprise information architectures. In P. Kandzia and M. Klusch, editors, *Proceedings ot the First International Workshop on Cooperative Information Agents*, volume 1202 of *LNAI*, pages 79–100, Berlin, February 26–28 1997. Springer.

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

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

[47] O. Kühn and A. Abecker. Corporate memories for knowledge management in industrial practice: Prospects and challenges. In *[12]*, 1998.

[48] H. Lieberman. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, Montreal, Quebec, Canada, August 1995. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[49] 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-Verlag, 1999.

[50] H. Lieberman, N. Van Dyke, and A. Vivacqua. Let's browse: A collaborative web browsing agent. In Mark Maybury, editor, *Proceedings of the 1999 International Conference on Intelligent User Interfaces (IUI-99)*, pages 65–70, N.Y., January 5–8 1999. ACM Press.

[51] 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/.

[52] A. Maedche, S. Staab, N. Stojanovic, R. Studer, and Y. Sure. SEAL – A framework for developing SEmantic Web PortALs. *Lecture Notes in Computer Science*, 2097:1–22, 2001.

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

[54] 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-Verlag, 2002.

[55] 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, *Proceedings of the 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 2000)*, pages 369–388, Manchester, UK, April 2000. The Practical Application Company Ltd.

[56] T. Nabeth, A. Angehrn, and C. Roda. Towards personalised, socially aware and active knowledge management systems. In *Proceedings of E-2002 e-Business and e-Work Annual Conference; Prague, Czech Republic*, 2002.

[57] 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 of WWW-2002*. ACM Press, 2002.

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

[59] G. Papavassiliou, G. Mentzas, and A. Abecker. Integrating knowledge modelling in business process management. In *Proceedings of the Xth European Conference on Information Systems (ECIS-2002)*, Gdansk, Poland, 2002.

[60] H. Van Dyke Parunak. What can agents do in industry, and why? An overview of industrially-oriented R&D at CEC. *Lecture Notes in Computer Science*, 1435:1–18, 1998.

[61] H. Van Dyke Parunak. Industrial and practical applications of DAI. In Gerhard Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 9, pages 377–421. The MIT Press, Cambridge, MA, USA, 1999.

[62] E. Plaza and J.-L. Arcos. Context–aware personal information agents. In M. Klusch and F. Zambonelli, editors, *Cooperative Information Agents V, Proceedings of the 5th International Workshop CIA 2001*, number 2182 in Lecture Notes in Artificial Intelligence, pages 44–55. Springer-Verlag: Heidelberg, Germany, September 2001.

[63] 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.

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

[65] U. Reimer. *PAKM-98: Practical Aspects of Knowledge Management. Proc. of the Second Int. Conference*. October 1998.

[66] U. Reimer, editor. *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management, PAKM2000*, Basel, Switzerland, 2000. http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-34/.

[67] 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–Verlag, 2001.

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

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

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

[71] C. Roda, A. Angehrn, and T. Nabeth. Matching competencies to enhance organisational knowledge sharing: An intelligent agent approach. In *Proceedings of the 7th International Netties Conference*, pages 931–937, Fribourg, Switzerland, September 2001.

[72] C. Roda, A. Angehrn, T. Nabeth, and L. Razmerita. Using conversational agents to support the adoption of knowledge sharing practices. *Interacting with Computers*, 15:57–89, 2003.

[73] H.-P. Schnurr and S. Staab. A proactive inferencing agent for desk support. In S. Staab and D. O'Leary, editors, *Proceedings of the AAAI Symposium on Bringing Knowledge to Business Processes*, pages 44–52. AAAI Press, Menlo Park, 2000.

[74] 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.

[75] S. Staab and A. Maedche. Knowledge portals: Ontologies at work. *The AI Magazine*, 22(2):63–75, 2000.

[76] S. Staab, H. Schnurr, R. Studer, and Y. Sure. Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16(1):26–34, 2001.

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

[78] G. Stefanidis, D. Karagiannis, and R. Woitsch. The PROMOTE Approach: Modelling Knowledge Management Processes to Describe Knowledge Management Systems. In *Proc. of the third European Conference on Organizational Knowledge, Learning, and Capabilities (OKLC 02), Athens, Greece*, April 2002.

[79] 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)*, Lecture Notes on Artificial Intelligence 2691, pages 584–593. Springer-Verlag, 2003.

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

[81] K. Wiig. Perspectives on introducing enterprise knowledge management. In *[65]*, 1998.

[82] M. Wooldridge. Intelligent agents. In Gerhard Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 1, pages 27–78. The MIT Press, Cambridge, MA, USA, 1999.

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

[84] M. Wooldridge and N.R. Jennings. Pitfalls of agent-oriented development. In K.P. Sycara and M. Wooldridge, editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 385–391, New York, May 9–13, 1998. ACM Press.

[85] 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.

[86] 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.