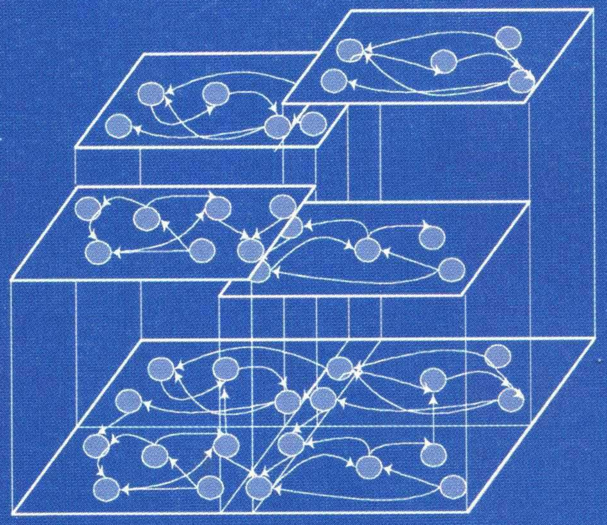
Hot Topics

LNAI 2926

Ludger van Elst
Virginia Dignum
Andreas Abecker (Eds.)

Agent-Mediated Knowledge Management

International Symposium AMKM 2003
Stanford, CA, USA, March 2003
Revised and Invited Papers



Springer

Ludger van Elst   Virginia Dignum
Andreas Abecker (Eds.)

Agent-Mediated Knowledge Management

International Symposium AMKM 2003
Stanford, CA, USA, March 24-26, 2003
Revised and Invited Papers

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Ludger van Elst
Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI) GmbH
Knowledge Management Department
Erwin-Schroedinger-Str., 67608 Kaiserlautern, Germany
E-mail: elst@dfki.uni-kl.de

Virginia Dignum
University of Utrecht
Institute of Information and Computing Sciences
3508 TB Utrecht, The Netherlands
E-mail: virginia@cs.uu.nl

Andreas Abecker
Forschungszentrum Informatik, WIM
Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany
E-mail: abecker@fzi.de

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): I.2.11, I.2.4, I.2, H.5.3, H.4, H.3, C.2.4, K.4.4, J.1

ISSN 0302-9743
ISBN 3-540-20868-2 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Boller Mediendesign
Printed on acid-free paper SPIN: 10976241 06/3142 5 4 3 2 1 0

Preface

In this book, we present a collection of papers around the topic of Agent-Mediated Knowledge Management. Most of the papers are extended and improved versions of work presented at the symposium on Agent-Mediated Knowledge Management held during the AAAI Spring Symposia Series in March 2003 at Stanford University.

The aim of the Agent-Mediated Knowledge Management symposium was to bring together researchers and practitioners of the fields of KM and agent technologies to discuss the benefits, possibilities and added-value of cross-fertilization.

Knowledge Management (KM) has been a predominant trend in business in recent years. Not only is Knowledge Management an important field of application for AI and related techniques, such as CBR technology for intelligent lessons-learned systems, it also provides new challenges to the AI community, like, for example, context-aware knowledge delivery. Scaling up research prototypes to real-world solutions usually requires an application-driven integration of several basic technologies, e.g., ontologies for knowledge sharing and reuse, collaboration support like CSCW systems, and personalized information services. Typical characteristics to be dealt with in such an integration are:

- manifold, logically and physically dispersed actors and knowledge sources,
- different degrees of formalization of knowledge,
- different kinds of (Web-based) services and (legacy) systems,
- conflicts between local (individual) and global (group or organizational) goals.

Agent approaches have already been successfully employed in KM for many partial solutions within the overall picture: agent-based workflow, cooperative information gathering, intelligent information integration, and personal information agents are established techniques in this area. In order to cope with the inherent complexity of a more comprehensive solution, Agent-Mediated Knowledge Management (AMKM) deals with collective aspects of the domain in an attempt to cope with the conflict between the desired order and the actual behavior in dynamic environments. AMKM introduces a social layer that structures the society of agents by defining specific roles and possible interactions between them.

This workshop set the scene for the assessment of the challenges that Agent-Mediated Knowledge Management faces as well as the opportunities it creates. By focusing on agent-mediated interactions, specialists from different disciplines were brought together in a lively and inquisitive environment that provided nice interactions and debates. The main topics for the workshop were:

- collaboration and P2P support,
- agent-based community support,
- agent models for knowledge and organizations,

- context and personalization,
- ontologies and the Semantic Web,
- agents and knowledge engineering.

Besides extended versions of workshop presentations, this volume includes an introductory chapter, and papers originating from the invited talk and from discussion sessions at the symposium. The result is that this volume contains high-quality papers that really can be called representative of the field at this moment.

This volume starts with an introduction to the Agent-Mediated Knowledge Management topic. The paper provides an extended motivation and an overview of research and current developments in the field. The remainder of the volume has been arranged according to the topics listed above.

The first section contains four papers on collaboration and peer-to-peer support. The first paper in this section by Bonifacio et al. proposes a P2P architecture for distributed KM. Graesser et al. discuss the results of a study on the benefits for KM from intelligent interfaces, namely animated conversional agents. The third paper by Guizzardi et al. presents Help&Learn, an agent-based peer-to-peer helpdesk system to support extra-class interactions among students and teachers. The section ends with a paper by Ehrig et al. suggesting a concise framework for evaluation of P2P-based Knowledge Management systems.

The second section contains three papers on agent-based community support. The first paper by Schulz et al. presents a conceptual framework for trust-based agent-mediated knowledge exchange in mobile communities. Kayama and Okamoto examine knowledge management and representation issues for the support of collaborative learning. The last paper in this section, by Moreale and Watt, describes a mailing list tool, based on the concept of a mailing list assistant.

The third section is devoted to agent models for knowledge and organizations. Filipe discusses the coordination and representation of social structures based on using the EDA agent model for normative agents, combined with the notion of an information field. Lawless looks at the fundamental relations between the generation of information and knowledge, with agent organizations, decision-making, trust, cooperation, and competition. The third paper, by Furtado and Machado, describes an AMKM system for knowledge discovery in databases. Hui et al. report on experience using RDF to provide a rich content language for use with FIPA agent toolkits. The paper by Magalhaes and Lucena, describing a multiagent architecture for tool generation for document classification, closes this section.

The fourth section, on context and personalization, starts with a paper by Louçã who presents a multiagent model to support decision-making in organizations. Novak et al. introduce an agent-based approach to semantic exploration and knowledge discovery in large information spaces. The paper by Evans et al. looks at the use of agents to identify and filter relevant context information in information domains. The section ends with a paper by Blanzieri et al., presenting the concept of implicit culture for personal agents in KM.

The fifth section contains four papers that focus on ontologies and the Semantic Web. Cao and Gandon discuss the benefits of societies of agents in a corporate semantic web. Krueger et al. look at ways to fully realize the potential of the Semantic Web, by automatically upgrading information sources with semantic markup. Hassan investigates interfaces to harness knowledge from heterogeneous knowledge assets. Cassin et al. present an architecture for extracting structured information from raw Web pages and describe techniques for extracting ontological meaning from structured information. The paper by Toivonen and Helin presents a DAML ontology for describing interaction protocols. The last paper in this section, by Petrie et al., discusses the benefits of agent technology to the development of Web services.

The last section of the book contains six papers related to agent and knowledge engineering. The first paper, by Furtado et al., studies the relationship between agent technology, knowledge discovery in databases, and knowledge management. The paper by Molani et al. describes an approach to capture strategic dependencies in organizational settings in order to support the elicitation of requirements for KM systems. Bailin and Truszkowski discuss the role of perspective in conflicts in agent communities. The paper by Tacla and Barthès concerns a multiagent system for knowledge management in R&D projects. Pease and Li introduce a system for collaborative open ontology production. Finally, the paper by Dodero et al. describes an agent-based architecture to support knowledge production and sharing.

We want to conclude this preface by extending our thanks to the members of the program committee of the AMKM workshop and to the additional reviewers who carefully read all submissions and provided extensive feedback on all submissions. We also want to thank all authors who were not only willing to submit their papers to our workshop and rework them for this book, but in addition contributed by their lively participation in a spontaneously organized peer review process.

September 2003

Ludger van Elst
Virginia Dignum
Andreas Abecker

Workshop Organization

Workshop Chairs

Ludger van Elst	DFKI, Kaiserslautern, Germany
Virginia Dignum	Utrecht University, and Achmea, The Netherlands
Andreas Abecker	FZI, Karlsruhe, Germany

Program Committee

Paolo Bouquet	University of Trento, Italy
Rose Dieng	INRIA, Sophia-Antipolis, France
Michael N. Huhns	University of South Carolina, Columbia, USA
Daniel O'Leary	University of Southern California, Los Angeles, USA
Pietro Panzarasa	University of Southampton, UK
Amit P. Sheth	University of Georgia, Athens, USA
Walt Truszkowski	NASA Goddard Space Flight Center, USA
Gerd Wagner	Eindhoven University of Technology, The Netherlands

Additional Reviewers

Nicola Henze
Markus Junker
Alexander Mädche
Heiko Maus
Michael Sintek
Franz Schmalhofer

Table of Contents

Section VI: Agents and Knowledge Engineering

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

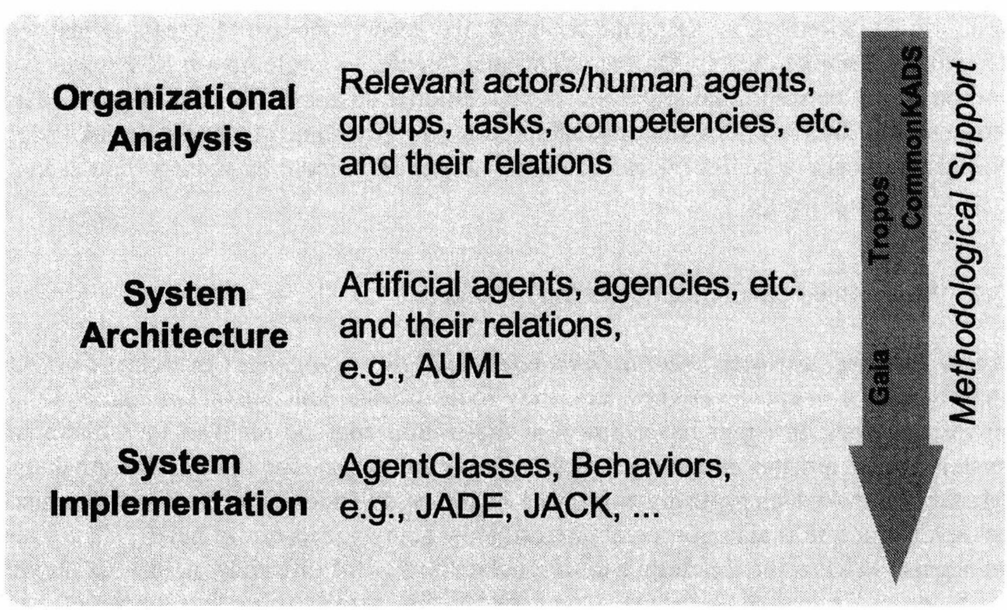


Fig. 1. Notion of Agents at Different Stages in the Development Cycle of an Agent–Based KM System

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,