

An agent-mediated approach to the support of knowledge sharing in organizations

VIRGINIA DIGNUM, FRANK DIGNUM and
JOHN-JULES MEYER

University of Utrecht, The Netherlands;
e-mail: virginia@cs.uu.nl, dignum@cs.uu.nl, jj@cs.uu.nl

Abstract

The realization that knowledge management (KM) is primarily a management science and not a computer science implies a different role for technology in KM. This role is concerned with supporting and extending human interaction and learning, and has therefore the need for intelligence-enhanced, integrated and personalized solutions including agent-based approaches. A people-centered view of KM requires support systems that enable the flexible integration of organizational and individual requirements and objectives. This paper introduces the OperA model for organizations that supports individual initiative and collaboration while prescribing a formal structure for organizational processes. The paper introduces the main aspects of the model and describes a case study where OperA is applied to the development of a knowledge-sharing support system.

1 Introduction

Knowledge has widely been acknowledged as one of the determining factors for corporate competitiveness and advantage. In the past decade, we have witnessed an explosion of approaches to knowledge management (KM). KM is defined as a systematic, holistic approach to the sustainable improvement of the handling of knowledge at all levels of an organization (Eppler, 2002). Practitioners and business managers alike agree that issues of technology, process, people, and content must be addressed in KM to achieve success (Smith & Farquhar, 2000). Moreover, it is becoming increasingly important for organizations to shorten the learning curve (that is, the time to achieve full competence); to rapidly assimilate sophisticated new technologies; and to efficiently fill the gaps in a company's knowledge base—particularly as developments become more complex and operating environments pose increasing demands on people and organizations. Moving forward to be a best-in-class company means transforming everyone in the company into an experienced practitioner in one or more technical or support disciplines.

A close look at how companies really work will show gaps between official work processes—the *a priori* designed flows of tasks and procedures reflecting the ideal activity of the company—and the real-world practices that actually get things done. These gaps are not problems that need fixing; they are opportunities that deserve leveraging (Brown & Gray, 1995). Processes do not do work, people do, and people tend to develop their own ways of doing things. That is, the real assets of organizations are the informal, often inspired ways that real people solve real problems in ways that formal processes cannot anticipate. The realization that such gaps exist is of the utmost importance for the success of knowledge management initiatives. From the starting days of KM, technology has been recognized as an enabling, and often even a leading, factor for connecting (e.g. people to other people or knowledge) and converting (e.g. data into knowledge) (O'Leary, 1998). Comprehensive

KM endeavors, however, have always realized that KM is primarily a *management* science, and not a *computer* science. This implies a different role for technology in KM, that of supporting and extending human interaction and learning, and therefore a need for intelligence-enhanced, integrated and personalized solutions. A KM system that links to the real needs and goals of people in their real-world practices, and facilitates their contacts and interactions, has a much higher chance of success than one that will follow the ‘official’ workflow processes. The above considerations identify a novel direction in KM, that of collaboration management. Collaboration management systems must meet the following requirements (Dignum & Dignum, 2003):

1. Assist people to generate and apply ‘just in time’ and ‘just enough’ knowledge, prevent information overload and stimulate sharing of relevant knowledge in a dynamic, collaborative environment.
2. Preserve individual autonomy and contribute to the creation of an atmosphere of trust between participants.
3. Provide links between individual action and company structure, such that, on the one hand, innovative ways of doing things can be effectively integrated into company processes and, on the other hand, it can be verified whether actions conform to company values and norms.

Agents offer a way to deal with complex systems that have multiple and distinct components, and are often used as a metaphor for autonomous, intelligent entities (Luck *et al.*, 2003). Therefore, agents are, in our opinion, perfectly qualified to model collaboration management systems. In this sense, we define an organization as a set of agents and their interactions, which are regulated by mechanisms of social order and created to achieve common goals. The OperA framework for agent organizations takes the perspective of the organization as a whole and therefore is able to define the global aims of an organization, such as stability over time, some level of predictability, and clear commitment to aims and strategies, as well as the objectives and responsibilities of participants (Dignum *et al.*, 2002; Dignum, 2004).

Agent organizations emerge from the idea that interactions occur not just by accident but aim at achieving some desired global goals. That is, there are goals external to each individual participant (or agent) that must be reached through the interaction of those participants. Desired behavior of a society is therefore often external to the participants. Social structure is determined by organizational design and not dependent on the participants. However, the behavior of individuals is motivated from their own goals and capabilities, that is, people will follow their own goals and motivations and will bring in their own ways of doing things to the society. In other words, the actual behavior of the society emerges from the goal-pursuing behavior of the individual agents within the constraints set by the organization. This creates a need to check conformance of the actual behavior to the desired behavior, a need that has several consequences.

Models for organizations are needed that integrate the realization of organizational requirements and objectives, and at the same time allow participants to have the freedom to act according to their own personalities. Such models for open society support systems must meet the following requirements:

- *Internal autonomy requirement*: interaction and structure of the society must be represented independently from the internal design of participating agent.
- *Collaboration autonomy requirement*: activity and interaction in the society must be specified without completely fixing the interaction structures in advance.

Taking a collaboration perspective on KM implies a different role for technology in KM, that of supporting and extending human interaction and learning, and therefore a need for intelligence-enhanced, integrated and personalized solutions. This is currently leading to an increasing interest in the use of multi-agent concepts for KM, mainly motivated by the fact that, like multi-agent systems, KM domains involve an inherent distribution of sources, problem-solving capabilities and responsibilities (Gandon *et al.*, 2000; Bonifacio *et al.*, 2002; van Elst *et al.*, 2004). In such domains,

it is important on the one hand, to assure that activity conforms to (existing) organizational norms and aims at the realization of global goals, but, on the other hand, the autonomy of participants must be preserved, so that the organization can profit from individual characteristics and skills. This calls for an autonomous and distributed representation of KM systems. Moreover, interactions in KM environments are sophisticated, including negotiation, information sharing and coordination, and require complex social skills with which agents can be endowed. Furthermore, solutions for KM problems cannot be entirely prescribed from start to finish and therefore reactive and proactive problem solvers are required that can respond to changes in the environment, react to the unpredictability of business processes and act on opportunities when they arise.

The OperA model for agent organizations, presented in this paper, meets the above requirements as it enables the separation between the development of the organizational model, representing norms, goals and the social structure of the organization as determined by the organization's owner, and the development of the individual agents that will participate in the organization. Instantiation of social roles and concrete interactions between actors is specified separately, which enables the negotiation of the match of social and individual characteristics and requirements. In this way, OperA attempts to incorporate formal organizational processes and goals with the different individual perspectives of the actors (people, groups and possibly systems) involved (Dignum *et al.*, 2002).

The remainder of this paper is organized as follows. In Section 2, we discuss the use of agents in KM. Section 3 presents the OperA model for organizations. Section 4 introduces the collaboration scenario where a support system for knowledge sharing was developed using OperA. The collaboration support component is described in Section 5. Finally, Section 6 provides concluding remarks and discusses areas for further research.

2 Related work

2.1 Agents in knowledge management

Agent-mediated knowledge management is a new research direction that aims at the cross-fertilization between KM and the intelligent agent research fields (van Elst *et al.*, 2004). Applications of agent technology to KM start from the realization that KM and multi-agent systems have several similarities. Agent-based KM services include (Klusch, 1999):

- search for, acquire, analyze, integrate and archive information from multiple heterogeneous sources;
- inform users when new information of special interest becomes available;
- negotiate for, purchase and receive information, goods or services;
- explain the relevance, quality and reliability of that information; and
- learn, adapt and evolve to changing conditions.

The use of agents in KM can be seen in two perspectives. Firstly, agents are often used to model the organizational environment where the KM system will operate and, alternatively, software agents can be used to implement the services, or functionalities, of KM systems. On the organizational level, van Elst *et al.* (2004) proposed to classify agent-based KM systems according to the degree of sociability. That is, are applications based on single-agent or multi-agent architectures? And, what is the *architecture/topology* of the agent system with respect to the flow of knowledge and information, or with respect to the coordination of decisions? Secondly, multi-agent models are used as the virtual counterpart of real-life societies and organizations, which facilitates the design process since it reduces the conceptual distance between the system and the real-world application it has to model. At the development level, the use of agents in KM can be seen from two perspectives: (1) the stage in a system's *development process* where agents are used (analysis, conceptual design, or implementation); and (2) the *KM functionality/application* focused on.

Applications of agent technology to KM start from the realization that KM and multi-agent systems have several similarities. Agent-based applications are available to support various aspects of KM, from personal information agents to agent-based workflows for business process-oriented KM. Agents are mainly used in dynamic environments where activity and reasoning are determined by the interpretation of perceptions about the actual condition of the environment. Like multi-agent systems, KM environments can be seen as distributed systems where different actors, each pursuing its own goals, need to interact in order to achieve their goals and realize organizational objectives. In such environments, the ability to communicate and negotiate is paramount. Furthermore, the number and behavior of participants cannot be fixed *a priori*, and the system can be expected to expand and change during operation, both in number of participants and in amount and kind of knowledge shared. The use of multi-agent systems in KM is therefore motivated by the following observations.

- KM domains involve an inherent distribution of sources, problem-solving capabilities, and responsibilities (applies the autonomy and social ability of agents).
- The integrity of the existing organizational structure and the autonomy of participants needs to be maintained (uses autonomous nature of the agents).
- Interactions in KM environments are sophisticated, including negotiation, information sharing, and coordination (requires complex social skills with which agents are endowed).
- KM domains call for a functional separation between knowledge use and knowledge sources as a way to incorporate dynamic behavior into system design (agents can act as mediators between source and application of knowledge).
- Solutions for KM problems cannot be entirely prescribed from start to finish and therefore problem solvers are required that can respond to changes in the environment, to react to the unpredictability of the business process and to proactively take opportunities when they arise (uses reactive and proactive abilities of agents).

Agent-based models for KM see agents as autonomous social entities (like employees in a company) that exhibit flexible, responsive and proactive behavior and the interactions among these entities give rise to complex dynamics. In this context *agent* is defined as ‘one that has the power or authority to act’ or ‘one that takes action at the instigation of another’. Current multi-agent models are not well suited for KM because they either take a centralistic approach to organizational design (cf., for example, Wooldridge *et al.*, 2000), or have a completely emergent view on agent interactions. Collaboration support systems, as described above, require the integration of individual desires with organizational requirements. The multi-agent organizational model OperA presented in this paper incorporates formal organizational processes and goals and the different individual perspectives of the actors (people, groups and possibly systems) involved. One purpose of this paper is to show the suitability of this model to describe collaboration support systems. Even though, at the moment there is not yet sufficient evidence of the added-value of agent-based systems to KM, due to the fact that not many applications are available, we believe that agent technology helps to develop KM systems that are more flexible, and will lead to a more human-centered, agile and scalable KM support. The Knowledge Market application presented in this paper is an example of this approach.

2.2 Approaches to multi-agent system development

Development methods for multi-agent systems are currently an important research topic, which is demonstrated by the diversity of proposed methodological approaches. Methodologies to design agent societies must be able to describe the characteristics of organizational environments, and therefore should incorporate social concepts such as organization structures, norms and domain language. Furthermore, methodologies must support the development of open societies and the specification of formal institutions. In the remainder of this section, we briefly discuss how some well known models support the social and normative concepts.

Gaia (Wooldridge *et al.*, 2000) is one of the first agent-oriented software engineering methodologies that explicitly takes into account social concepts. *Gaia* aims at providing a coherent conceptual framework for the analysis and design of multi-agent systems. The analysis phase results in (1) the agent model, which specifies system roles and their characteristics in terms of permissions (the right to exploit a resource) and responsibilities (functionalities); and (2) the interaction model, which captures the dependencies and relations between roles by means of protocol definitions. *Gaia* models describe the society level of an agent society and do not capture internal aspects of agent design. However, societies are only considered from the perspective of the individual participants, and therefore *Gaia* does not deal with communication or other collective issues. Furthermore, normative aspects are reduced to static permissions, similar to constraints or rules, and behavior is fixed in protocols. Moreover, *Gaia* is not suited to model open domains, and cannot easily deal with self-interested agents, as it does not distinguish between organizational and individual aspects, and does not provide capabilities for agent interpretation of society objectives, norms or plans.

SODA (Omicini, 2001) is actually an extension to *Gaia* that enables open societies to be designed around suitably designed coordination media, and social rules to be designed and enforced in terms of coordination rules. As *Gaia*, *SODA* distinguishes between an analysis and a design phase. The analysis phase results in three different models: the *role model*, which describes the goals, or tasks of roles and groups; the *resource model*, which describes the environment in terms of available resources; and the *interaction model*, where interaction protocols describe the information required and provided by roles and resources, and the rules governing interaction. During the design phase, roles are mapped to agent classes (the *agent model*), groups are mapped into societies designed around coordination abstractions (the *society model*), and resources are mapped into infrastructure classes (the *environment model*). As an attempt to include a higher abstraction level, *SODA* presents a notion of the context, or environment, of the society, albeit not explicit. However, even though *SODA* distinguishes between agent and collective spaces, it sees roles as the representation of the observable behavior of agents, and therefore cannot represent the difference between the organizational perspective on the activity and aims of individuals from the agent perspective on its own activity and aims. Furthermore, the enactment of roles is fixed in *SODA* as the agent model that maps roles to agent classes without any possibility of accommodating agent preferences or characteristics (agent classes are pure specifications of the role characteristics). There are no normative aspects in *SODA* further than the notion of permission to access infrastructure services. Communication primitives are limited to interaction protocols, and *SODA* provides no explicit representation for the domain ontology. Furthermore, *SODA* does not have a clear and formal semantics.

The *ISLANDER* formalism (Esteva *et al.*, 2001) provides a formal framework for institutions (Noriega, 1997; Rodriguez, 2001) and has proven to be well suited to model practical applications (e.g. electronic auction houses). This formalism views an agent-based institution as a *dialogical system* where all the interactions inside the institution are a composition of multiple dialogic activities (message exchanges). These interactions (or *illocutions*) are structured through agent group meetings called *scenes* that follow well-defined protocols. This division of all the possible interaction among agents in scenes allows a modular design of the system, following the idea of other software modular design methodologies such as modular programming or object-oriented programming. A second key element of the *ISLANDER* formalism is the notion of agent's *role*. Each agent can be associated to one or more roles, and these roles define the scenes the agent can enter and the protocols it should follow. Finally, this formalism defines a graphical notation that not only allows one to obtain visual representations of scenes and protocols but is also very helpful while developing the final system, as they can be seen as blueprints. *ISLANDER* has been mainly used in *e-commerce* scenarios, and was used to model and implement an electronic auction house (the *Fishmarket*). Furthermore, the *e-INSTITUTOR* platform and the *ISLANDER* API enable the animation of models and the participation of external agents. The activity of these agents is, however, constrained by *governors* that regulate agent actions, to the precise enactment of the roles

specified in the institution model. In contrast to the other frameworks discussed here, ISLANDER provides a sound model for the domain ontology and has a formal semantics. However, ISLANDER does not consider the normative aspects of organizations, further than the specification of constraints for scene transition and enactment (the only allowed interactions are those explicitly represented by arcs in scenes).

The *TROPOS* methodology (Castro *et al.*, 2002; Bresciani *et al.*, 2004) spans the overall development process. It distinguishes between an early and a late requirements phase, and between architectural design and detailed design. *TROPOS* starts with an analysis of the organizational setting of the application, in the early requirements phase. The strategic dependency model describes an ‘agreement’ between two actors: the depender and the dependee. The strategic rationale model determines through a means–ends analysis how an actor’s goals (including soft goals) can actually be fulfilled through the contributions of other agents. These two models serve as input for the late requirements phase, in which a list of functional and non-functional requirements is specified. The architectural design defines the structure of a system in terms of subsystems that are interconnected through data, control and other dependencies. The detailed design defines the behavior of each component. The models are implemented using Jack Intelligent Agents (Howden *et al.*, 2001), which is an agent-oriented extension of Java. *TROPOS* is a fairly complete methodology that considers all steps in system development, and it treats both inter-agent and intra-agent perspectives. The main two shortcomings of *TROPOS* are that (a) it is not formal (although there is some ongoing work on providing a formal semantics for *TROPOS*), and (b) it is too organizational-centered in the sense that it does not consider that agents can have their own goals and plans, and not just those coming from the organization. Furthermore, *TROPOS* has no concept representing the normative aspects of an organization. On the other hand, it has effectively been applied in the KM area as a tool to specify support systems to communities of practice (Guizzardi *et al.*, 2004).

3 The OperA agent organization model

In this research, we see the development of agent societies based on two competing goals. On the one hand, the structure and requirements of the society owners must be captured in the society design and, on the other hand, agents must be available that are able and interested in enacting society roles. The OperA model integrates a top-down specification of society objectives and global structure, with a dynamic fulfillment of roles and interactions by participants. The model separates the description of the structure and global behavior of the domain from the specification of the individual entities that populate the domain. This separation provides several advantages to our framework over and above traditional multi-agent system models. On the one hand, coordination and interaction in multi-agent systems are usually described in the context of the actions and mental states of individual agents (Ferber & Gutknecht, 1998). In open societies, however, such an approach is not possible because agents are developed independently from the society and there is therefore no knowledge about the internal architecture of agents, nor possibilities to directly control or guide it. Furthermore, conceptual modeling of agent societies (based on the social interactions) requires that interaction between agents be described at a higher, more abstract level, that is, in terms of roles and institutional rules. On the other hand, society models designed from an organizational perspective reflect the desired behavior of an agent society, as determined by the society ‘owners’. Once ‘real’ agents populate the society, their own goals and behavior will affect the overall society behavior, that is, such social order as envisioned by the society designer is in reality no more than conceptual; abstract behavior seldom is realized exactly in practice. From an organizational perspective, the main function of individual agents is the enactment of roles that contribute to the global aims of the society. That is, society goals determine agent roles and interaction norms. Agents are actors that perform role(s) described by the society design. The agent’s own capabilities and aims determine the specific way an agent enacts its role(s).

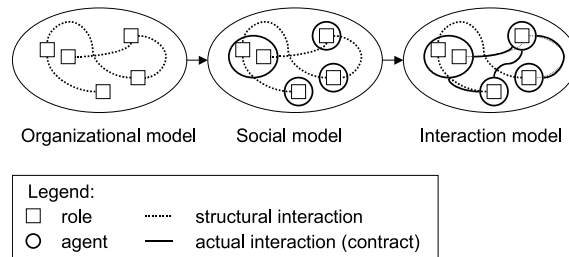


Figure 1 Organizational framework for agent societies

Several authors have advocated such role-oriented approaches to agent society development, especially when it is manifest to take an organizational view on the application scenario (Zambonelli *et al.*, 2001). Castelfranchi (2000) distinguishes between social order, the non-accidental, non-chaotic pattern of interaction in a given system of interacting agents, and social control, agent action aimed at enforcing the conformity of behavior of other agents to some social norm. He argues that due to the autonomous behavior of agents, social control is not enough to deal with the challenge of social order, but agent societies must be able to cope with unintended, emergent behavior of its members.

The OperA framework consists of three interrelated models. The organizational structure of the society, as intended by the organizational stakeholders, is described in the organizational model (OM). The way interaction occurs in a society depends on the aims and characteristics of the application, and determines the way roles are related with each other, and how role goals and norms are ‘passed’ between related roles. For example, in a hierarchical society, goals of a parent role are shared with its children by delegation, while in a market society, different participants bid to the realization of a goal of another role. The agent population of an OM is specified in the social model (SM) in terms of social contracts that make explicit the commitments regulating the enactment of roles by individual agents. Social contracts describe the capabilities and responsibilities of an agent within the society, that is the desired way that an agent will fulfil its role(s). The use of contracts to describe the activity of the system allows, on the one hand, for flexibility in the balance between organizational aims and agent desires and, on the other hand, for verification of the outcome of the system. Finally, given an agent population for a society, the interaction model (IM) describes possible interaction between agents. After all, once 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 capabilities to interact with others according to the society specification. Figure 1 depicts the interrelation between the different models.¹ In the following subsections, we will describe each of the three models in more detail.

In OperA, the organizational model of a society reflects the requirements of the organization’s owners. Agents are seen as autonomous communicative entities that will perform society role(s) as a means to realize their own goals according to their own internal aims and architecture. In contrast, constrained by the organizational design, activity is dependent on the capabilities of actual agents present in the society at a given moment. This means that several agent populations are possible for each organizational model, and the objectives of the society will be achieved in different ways. The characteristics and requirements of the society specified in the society model are then incorporated in the software agents themselves. Agents will thus contain enough information and capabilities to interact with others according to the society specification.

From the point of view of society design, the reasons why an agent wants to enact a role are not relevant. However, from the agent’s perspective, mechanisms must be developed that allow the incorporation of role characteristics into the agent’s architecture. Even though this is not the focus of this paper, we have done some investigative work in this area and have presented some preliminary ideas and results in Dastani *et al.* (2003). Contracts are introduced as a means to

¹ A formalism to provide a logical semantics for the model is described in Dignum *et al.* (2003).

integrate top-down specification of organizational structures with the autonomy of the participating agents. In the social model, contracts describe the agreements between an agent and the society concerning the enactment of a role. In the interaction model, contracts represent the concrete specification of an interaction script according to the participating role enacting agents.

We have also developed a formal theory for the OperA framework (Dignum, 2004). The role of formal methods is to provide a clear and precise description of what a system is supposed to do, rather than a formulation of how it operates. The fact that OperA has a formal semantics makes it possible to give models a precise semantics, supports the use of structured design techniques and formal analysis, facilitating development, composition and reuse (Esteva *et al.*, 1991), and can therefore be used to guide and support the designer while building and refining a conceptual model. The formal model of OperA is based on the language for contract representation, LCR, which is an extension of deontic temporal logic. LCR is a very expressive logic for describing interaction in multi-agent systems (Dignum *et al.*, 2003). LCR makes it possible to describe and verify contracts that specify interaction between agents, and provides a rather realistic representation of a domain, in the sense that it deals with temporal and communicative aspects, and furthermore is able to represent deadlines and their influence on the behavior of the model.

3.1 *The organizational model*

The starting point of the Agent Society Model is the OM that describes the structure and global characteristics of a domain from an organizational perspective. That is, from the premise that it is the society's goals that determine agent roles and interaction norms. The OM is based on the analysis of the domain in terms of the coordination and normative elements. The OM specifies the global objectives of the society and the means to achieve those objectives.

The OM specifies an agent society in terms of four structures: social, interaction, normative and communicative. The social structure specifies objectives of the society, its roles and the model that governs coordination. The global objectives of an organization are represented in terms of objectives of the roles that compose the organization. Roles are tightly coupled to norms, and roles interact with other roles according to interaction scripts that describe a 'unit' of activity in terms of landmarks. The interaction structure gives a partial ordering of the scene scripts that specify the intended interactions between roles. Society norms and regulations are specified in the normative structure, expressed in terms of role and interaction norms. Finally, the communicative structure specifies the ontologies for description of domain concepts and communication illocutions. The way interaction occurs in a society depends on the aims and characteristics of the application, determines the relations between roles, and how role goals and norms are 'passed' between related roles. For example, in a hierarchical society, the goals of a parent role are shared with its children by delegation, while in a market society, different participants bid for the realization of a goal of another role.

3.2 *The social model*

We assume that individual agents are designed independently from the society to model the goals and capabilities of a given entity. In order to realize their own goals, individual agents will join the society as enactors of role(s) described in the OM. This means that several populations are possible for each OM. Agent populations of the OM are described in the SM in terms of commitments regulating the enactment of roles by individual agents. In the framework, agents are seen as autonomous communicative entities that will perform the society's role(s) according to their own internal aims and architecture. Because the society designer does not control agent design and behavior, the actual behavior of the society instance may differ from the intended behavior. The only means the society designer has for enforcing the intended behavior is by norms, rules and sanctions. That is, when an agent applies, and is accepted, for a role, it will commit itself to the

realization of the role goals and it will function within the society according to the constraints applicable to its role(s). These commitments are specified as social contracts that can be compared to labor contracts between employees and companies. The society can sanction undesirable (wrong) behavior as a means to control how an agent will do its 'job'.

The SM is defined by the role-enacting agents (REAs) that compose the society. For each agent, the REA reflects the agent's own requirements and conditions concerning its participation in the society. Depending on the complexity of the implemented agents, the negotiation of such agreements can be more or less free. However, making these agreements explicit and formal allows the verification of whether the animated society behaves according to the design specified in the OM. The SM specifies a population of agents in a society, which can be seen as an instantiation of the OM. When all roles specified in the OM are instantiated to agents in the SM, we say that the SM provides a full instantiation of the society; otherwise, it is a partial instantiation.

3.3 *The interaction model*

Finally, interaction between agents populating a society is described in the IM by means of interaction contracts. This model accounts for the actual (emergent) behavior of the society at a given moment. Inter-action agreements between agents are described in interaction contracts. Usually interaction contracts will 'follow' the intended interaction possibilities specified in the OM. However, because of the autonomous behavior of agents, the IM must be able to accommodate other interaction contracts describing new, emergent, interaction paths, to the extent allowed by the OM and SM.

OperA provides two levels of specification for interactions. The OM provides a script for interaction scenes according to the organizational aims and requirements, and the IM, realized in the form of contracts, provides the interaction scenes as agreed upon by the agents. It is the responsibility of the agents to ensure that their actual behavior is in accordance with the contracts (e.g. using a monitoring agent or notary services provided by the society for that). However, it is the responsibility of the society, possibly represented by some of its institutional roles, to check that the agents fulfill these responsibilities.

The architecture of the IM consists of a set of instances of scene scripts (called scenes), described by the interaction contracts between the REAs for the roles in the scene script. An interaction scene results from the instantiation of a scene script, described in the OM, to the REAs actually enacting it and might include specializations or restrictions of the script to the requirements of the REAs.

3.4 *Development methodology*

Organization theory shows that organizations with different objectives exhibit different requirements for coordination. Coordination models (market, hierarchy and network) are determined by transaction costs and reflect the balance between organizational objectives and activities. For example, the market model fits well in an exchange situation whereas the hierarchical model is better suited for production environment.

A generic methodology to analyze a given domain and determine the type and structure of the agent society that best models that domain is described in Dignum & Weigand (2002). The methodology provides generic facilitation and interaction frameworks for agent societies that implement the functionality derived from the coordination model applicable to the problem domain. Standard society types such as market, hierarchy and network can be used as the starting point for development and can be extended when needed to 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 cooperation possibilities between partners. We distinguish between social—or, facilitation roles—that is, roles needed in order to keep the society going, and operational roles, which will

provide the actual objectives of the society. Facilitation roles are usually played by mutually trusted agents, whereas trust between agents playing operational roles is determined by the type of society organization.

An important lesson learned from the past is that the development of multi-agent systems, like any other software systems, cannot be seen isolated from the (organizational) context in which it is inserted. System goals and structure must, on the one hand, match organizational strategy and processes, and on the other hand, meet user expectations and requirements. An often-heard complaint of managers in organizations is that lots of money and effort are applied to state-of-the-art ICT systems that are subsequently hardly used. In our opinion, such mistakes are because system development mostly concentrates on the technical aspects of the system and organizational, cultural and user aspects are largely ignored or assumed accomplished. We think that system development must start with the analyses and facilitation of the social environment in which it will be inserted. A preliminary step for the OperA methodology is the assessment of the applicability of the agent paradigm to the problem at hand. The methodology we describe in the remainder of this section is structured in three steps. Firstly, it guides the design of the OM to implement the desired organizational structure of an agent society. Agents are integrated into the society design in the SM, by means of social contracts, and agent interactions are fixed in the IM using interaction contracts. This design methodology follows the architecture of the OperA framework, and enables the specification of the OM, SM and IM of an OperA model. In Section 5, the methodology will be applied to the development of an agent society for the support of knowledge sharing in distributed organizations.

3.4.1 *Organizational model design*

The first step of the OperA methodology results in the specification of the OM for an agent society. The OM design methodology consists of three levels, which provide a growing level of refinement of the resulting system into richer and more precise forms. Coordination requirements specify the coordination structure (market, hierarchy or network) of the society. Functional requirements determine the behavior of the society and its relationship with the environment. These requirements are the basis for a basic society model, for which behavior and animation can be verified and compliance to the domain requirements can be checked. These steps result in a complete OM for the agent society.

Coordination level The coordination level starts with the analysis of the social characteristics of the domain, which results in the determination of the purpose, goals, relation forms and communication requirements for the domain. These are at this level used to determine the facilitation architecture of society, which consists of:

- the choice of a facilitation type: market, hierarchy or network; and
- the identification of the basic facilitation roles and interaction structure, associated with the coordination type.

Environment level The main characteristics of a society are identified through the analysis of the (expected) external behavior of the system. This process is based on the output of the coordination level, the identification of stakeholders, the identification of use cases describing overall requirements, and the analysis of the ethical or normative behavior expected in the society. As output of the environment level, a generic organizational architecture of society is obtained, which includes:

- the identification of society stakeholders and overall requirements;
- the specification of the communication primitives needed in the domain (ontologies and illocution primitives);
- the identification of organizational roles associated with stakeholders and their characteristics (described in role tables);
- identification of the ethical or normative behavior expected in the society.

Behavior level This consists of the analysis of the internal behavior of the system. This process is based on the basic OM, obtained from the previous methodological levels (including coordination type, generic role descriptions and communication and normative primitives), and on the functional requirements for roles and interactions as described in use cases for the system. The behavior level results in the complete specification of the OM of an OperA society, which includes:

- the specification of role descriptions for all society roles, as well as their objectives, norms and dependencies;
- the determination of the interaction scenes that will realize the interaction between roles necessary for the realization of their objectives;
- the refinement and specification of social norms, and their classification into role, scene or transition norms.

3.4.2 Social model design

The design of the SM of an OperA society is in reality much more of an operational issue than a methodological one. That is, whereas for the organizational level specific design steps, of increasing levels of detail, were identified which result in a complete society structure, the creation of the SM depends on the activities of specific agents, and is for the most part determined at 'run time'. The design of the SM for an OperA society is based on:

- the role descriptions specified in the OM;
- the way role negotiation scenes are specified in the OM; and
- the characteristics of the agents that apply for society roles.

This means that the same OM will result in many different SMs. Based on the OM specification of an agent society and on a set of specific agents, the SM design will describe the social contracts for society roles for those agents. The design of the agents themselves is outside the scope of an OperA model.

3.4.3 Interaction model design

The creation of an IM depends on the activities of specific REAs, guided by the description of scenes in the scene scripts specified in the OM. That is, the generation of an IM for an OperA society depends on:

- the specific REAs and their role enactment agreements, as described in social contracts in the SM; and
- the scripts for interaction scenes specified in the OM.

For the same agent population of an OM, many different IMs are possible. As with the SM, it allows the incorporation of the specific requirements and characteristics of agents and allows for a more realistic treatment of autonomy. Based on the OM specification of an agent society and on a population of REAs described in the SM, the IM design will describe the interaction contracts for scene scripts.

4 A knowledge-sharing scenario

Recent developments show a shift in the focus of KM from knowledge organization to inter-personal collaboration. That is, the aim of KM is no longer just the management of activities related to the creation, preservation and distribution of knowledge assets, but mainly the management and nurturing of collaboration between people. In this view, the basic organizational unit of KM is the community of practice (CoP). A CoP is a group of people sharing a common area of expertise and/or who search for solutions to common problems. A CoP is thus not necessarily an authorized or identified group. People in a CoP can perform the same job, collaborate on a

shared task or work together on a product. What holds them together is a common sense of purpose and a real need to know what each other knows. Most organizations will hold several CoPs and most people belong to at least one of them (Brown & Gray, 1995). Nurturing communities is hard enough when the members are in a single location with good connectivity, and the difficulties increase considerably when the members are spread around different locations, possibly in different areas and with different languages and cultures.

According to (Allee, 2002), CoPs are peers in the execution of real work. What holds them together is a 'common sense of purpose and a real need to know what each other knows'. The use of CoPs in KM (both in businesses and in educational settings) has been motivated by the assumption that knowledge cannot be separated from the communities that create it, use it, and transform it. Wenger *et al.* (2002) indicate that such environments are especially suitable for newcomers to an organization, who can learn the common procedures and working practices by informally collaborating with others, and, with time, become fully integrated and active members of the community.

It is important to note that communities cannot be forcibly created, but they may be fostered, by acquiring from the organization the means to grow and mature within working settings (Preece, 2000; Gongla & Rizzuto, 2001). In addition to that, fostering communities includes creating the conditions for a community to emerge, that is, giving it both social and technological support. The focus of our research is on the technological support, that is, on the design and development of appropriate infrastructures to facilitate knowledge sharing. However, support systems for CoP require a deep understanding of the organization, its goals, values and structures, and must take into account the individual goals of its workers. Furthermore, systems must fit with the specific characteristics of the communities that are going to use them and will only succeed if community members are convinced of the benefits for themselves and for the organization. In order to support this social process, we have developed a method, the SES model (seduce, engage, support), to facilitate the creation and management of communities and emphasize the importance of setting up real targets to CoPs, guaranteeing that their value for the organization be concretely perceived and measured (Dignum & van Eeden, 2003).

In this section, we will describe the development of a prototype agent society for knowledge exchange, using the OperA model and methodology. We start by describing the background and motivation for the project in Section 4.1, after which we describe the system and its development in detail in Section 5.

4.1 Background and motivation

The Knowledge Center Non-Life Insurance (KC) at a large insurance company in the Netherlands is responsible for the development and maintenance of non-life insurance knowledge that will give business units across the company a competitive edge in this area. KC has a need for efficient and goal-directed sharing of information and knowledge. Members of the group (mainly insurance product developers and actuaries) are active across business units, geographically dispersed, and are not part of any existing organizational structure. Their knowledge and expertise are greatly valuable and useful to each other. Nevertheless, because people are not aware of each other's capabilities, they will often discuss their business problems with a direct colleague just because he/she happens to be conveniently close and not because he/she is the best person to consult with (Davenport & Prusak, 1998).

In 2001, a project was started with the objectives of structuring, initiating and organizing the sharing of knowledge between non-life insurance experts across the company, by setting up a framework that assures the continuous availability of consistent and up-to-date knowledge (Dignum, 2002). The first steps towards the realization of these objectives concerned the development of a CoP, the KennisNet, incorporating the facilitation of direct contacts between members and an intranet-based knowledge-sharing server using existing technical infrastructure, a Lotus Notes network, as depicted in Figure 2. The development of the knowledge repository was

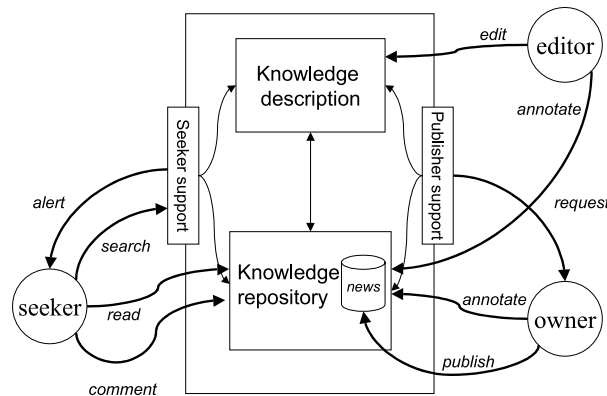


Figure 2 The architecture of KennisNet

inspired by work by Domingue & Motta (1999), Gandon *et al.* (2000) and Mentzas *et al.* (2001). Its functionality enables direct access to contents, publishing and browsing of knowledge items, and allows the implementation of facilities for discussion and broadcast of questions and requests.

Direct contacts between participants were formalized as quarterly workshops with the participation of all members that aimed to

- assure the creation, maintenance and uniformity of domain knowledge (for example, by inviting external authorities in a relevant field and by facilitating structured discussions around a theme); and
- enable participants to learn to know and appreciate each other, and feed community feeling.

4.2 Development of the knowledge repository

The development methodology used for the knowledge repository of KennisNet adapted the usual phases (analysis, design, implementation and evaluation) of system development to the specific case of knowledge management systems. Like organizations themselves, the process of developing knowledge management solutions is a dynamic one, and should be continuously monitored and adapted to the changing goals and structure of the organization. That is, development must be seen as a continuous process, where each step may require changes in previous steps. Furthermore, users and stakeholders must be involved in each level to assure the realization of a system that meets the needs and wishes of the organization and furthermore to assure that development keeps in pace with organizational and environmental changes.

The first step of the development was to identify the strategic goals of the organization or group and the problems that hinder their achievement. Next, problems were analyzed from a knowledge perspective, and solutions were identified and tailored to the specific situation. The objectives and the format of the system were analyzed, discussed and decided upon in a participatory way, during several meetings in which all members of the group participated. Finally, the system was implemented in Lotus Notes.

The focus of the development was on the classification and presentation aspects of the repository. One of the requirements for the repository was that there should be a uniform representation of all types of knowledge items (i.e. documents, Web sites, people, discussions, questions, news, databases and other applications, etc.). That is, one single search request should be able to retrieve documents, experts, related questions from others, and so on. Description of knowledge items includes:

- *identification*, that is, information about name, datum of publication and status;
- *content*, describing the actual meaning of an item, using aspects such as keywords, abstract, link classification ontology and comments;

- *context*, containing relevant information surrounding the creation and use of an item, including the name of the submitter, related projects, intended use and reasons for publication;
- *structure*, describing accessibility and use issues, including the type of item (e.g. document, person, Web site, application), location, contact person (who can tell you more about this item), and access conditions.

The KennisNet system provides automatic support filling in the descriptions of items. Search and retrieval in KennisNet is done on the meta-descriptions of knowledge items.² Furthermore, a classification ontology was developed following a participatory process to which all members of the community contributed. This ontology formed the basis for the classification structure used in the repository.

4.3 Evaluation of the knowledge repository

After the knowledge repository had been running for around a year, we conducted a user satisfaction survey, reported in Pumareja *et al.* (2003). The two main conclusions from this survey were:

1. the face-to-face structure was well appreciated and its value clear; but
2. the benefit and potential of the knowledge server was not always clear to the users, and the server is hardly used.

The survey pointed out that the main reason for this lack of use is that users need a more personal means of interaction to make them comfortable exchanging knowledge. The survey also indicated that knowledge owners prefer to share their expertise within a controllable, trusted group under conditions negotiated for the specific situation and partners. The community of users supported by the KennisNet operates across business unit boundaries, independently of the holding organizational structure. Sharing knowledge therefore implies that knowledge seeker and knowledge owner must be able to find each other and agree on the terms of the exchange.

Other recent studies elsewhere also show that success of knowledge sharing is dependent on the level of trust and dependency between community members and on the kind of culture prevailing in the society (Ali *et al.*, 2002). Knowledge is considered part of one's property and identity, and therefore people wish to keep the decision about sharing knowledge in their own hands, and want to be able to decide on a case-by-case basis whether an exchange is interesting to them or not. Furthermore, reciprocity in exchange is also an important aspect to be considered (Ahuja & Carley, 1998). The above considerations can be summarized in the following requirements for an effective collaboration support system:

- enable exchange within a controllable, trusted group under conditions negotiated by the partners for the specific situation;
- knowledge seekers and owners must be able to find each other and agree on the terms of the exchange;
- as the value of a knowledge item cannot be fixed *a priori*, and knowledge requests are usually not fulfilled by a mere exchange of 'products', but require an often not trivial creation process, mechanisms are needed to dynamically determine exchange conditions.

In order to support the above collaboration requirements, it was decided to extend the knowledge repository with mechanisms for knowledge exchange and collaboration that keep ownership links between knowledge and people, for the support of negotiation and valuation of exchange

² The knowledge items themselves, such as people, are often not in electronic format.

conditions.³ Motivated by the realization that another approach was needed to tackle the problem at hand, the KC started looking for more adequate models and tools to support collaboration in the group. Multi-agent systems can effectively meet the above requirements. Furthermore, the domain required, on the one hand, solutions to be independent of the design of individual components, representing the needs of each user (the internal autonomy requirement), and, on the other hand, flexibility and dynamic formation of exchanges (the collaboration autonomy requirement). These criteria motivated the choice of a development approach using the OperA model. In the knowledge market, agents can ensure the preservation of individual needs and perspectives, and they can be employed to monitor and assist knowledge exchange, for example by taking care that deadlines are kept, reports are effectively exchanged, and eventual changes are communicated. Furthermore, agents are used to search the network for suitable partners, to publish and search results in the repository on behalf of their owners, and to monitor news and discussion groups.

5 Knowledge market

In collaboration environments, the integrity of the existing organizational structure and the autonomy of participants must be maintained, which calls for an autonomous and distributed representation of KM systems. Interactions in collaboration environments are fairly sophisticated, including negotiation, information sharing and coordination, and require complex social skills with which agents can be endowed. Furthermore, solutions cannot be entirely prescribed from start to finish and therefore reactive and proactive problem solvers are required that can respond to changes in the environment, react to the unpredictability of business processes and act on opportunities when they arise.

These characteristics indicate the applicability of the OperA model to the development of KM environments that focus on the collaboration between people. Hence, the prime application of OperA is the development of a system for knowledge exchange. Nevertheless, OperA appeared to be more widely applicable than its original purpose suggested. It is a generic model for the design of multi-agent systems, which has the added value of a formal semantics (Dignum *et al.*, 2003; Dignum, 2004) and a customized development methodology (Dignum & Weigand, 2002), and as such is suitable for the development of multi-agent systems for a variety of domains.

In this section, we will describe the development of the knowledge market, according to the OperA methodology. The knowledge market aims to support people exchanging knowledge with each other, in a way that preserves the knowledge, rewards the knowledge owner and reaches the knowledge seeker on a just-in-time, just-enough basis. In the remainder of this section the system, developed using the Opera Model, is described.

5.1 Design of the organizational model

In this section, we describe the specification of the OM for the knowledge market, according to the methodological levels presented above, resulting respectively in the coordination, environment and behavior models for the knowledge market.

5.1.1 Coordination level

As described in Section 3.4.1, the design of the OM for an agent society starts with the evaluation of its coordination requirements. At this level, the coordination type of the society is determined. The evaluation of KennisNet shows that collaboration and direct exchange between people are the crucial aspects to realize. People usually agree to share their knowledge with others if they feel that they will gain something from the exchange, and that they can trust their exchange partner. For

³ How much is a specific piece of knowledge worth, at a specific moment, under the specific circumstances holding and to the specific partners involved in the exchange?

Table 1 Facilitation roles in the knowledge market

Role	Objectives	Abstract norms
Gatekeeper	Accept participants	Obligated to check whether applicant is member of KennisNet Allow only KennisNet members to request exchanges Allow external visitors to browse repository
Notary	Register agreements Assign monitors Impose sanctions	Obligated to register exchanges Allowed to request exchange information from seekers and owners
Monitor	Give alerts on deadlines and collaboration terms	Obligated to alert notary on sanctions
Matchmaker	Register participants, skills and needs Accept and distribute exchange requests	Obligated to distribute requests Obligated to give distribution requests back to requester

example, a typical agreement within the KennisNet group says, ‘I will share the result of a market survey I’ve just done with you, if you will let me have a copy of the report you are making for which you want to have those results’. Therefore, a knowledge-sharing system must be able to nurture and support the negotiation and realization of this kind of agreements.

In co-located groups, an exchange of favors relies on the assumption of stability of the community or group cohesiveness. There may be an inherent expectation that—since the relationships within the community are typically long lasting—sooner or later, the favor is likely to be returned. However, in distributed groups, although the common goal binding the members remains long term, contacts and relationships may be relatively fluid, with members entering and exiting as their task needs evolve. In this scenario, exchange of favors is likely to be based on reciprocity in a relatively short time-span (Ahuja & Carley, 1998). That is, collaboration will need to be based on concrete, explicit commitments making clear what each partner is supposed to contribute and expects from the others. Technology can facilitate knowledge sharing, but it is trust that enables it. Sharing knowledge therefore implies that seekers and owners must be able to find each other and agree on the terms of the exchange. Moreover, the value of a knowledge item cannot be fixed *a priori* but depends on many factors, and knowledge and information requests cannot be fulfilled by a mere exchange of finished ‘products’ but requiring an often not trivial process during which the knowledge owner will develop the answer sought by the requester.

The above considerations indicate that the most suitable coordination type for the knowledge market is the network model. However, because the domain requires support for users to find suitable partners, the role of matchmaker is also added to the facilitation layer of the knowledge market. Table 1 describes the facilitation roles of the knowledge market by adapting the generic features of network facilitation roles to the characteristics of the domain.

5.1.2 Environment level

At this level, the global functionality and objectives of the society are determined. The starting point for this level is the elicitation of use cases and requirements. Following the discussion in Section 4.3 on the evaluation and extension of the KennisNet system, the following functionally is desired for the knowledge market:

- possibility to share knowledge that is not available in the knowledge repository;

Table 2 Stakeholder table for the knowledge market society

Stakeholder	Objectives	Dependencies
Non-life department	Validate knowledge Distribute knowledge within group Distribute knowledge outside group	Gatekeeper, editor Knowledge seeker, knowledge owner Visitor
Knowledge seeker	Get help	Non-life department, knowledge owner
Knowledge owner	Get recognition through: providing help publishing own knowledge	Knowledge seeker Non-life department Editor

Table 3 Role table for knowledge market

Role	Relation to society	Role objectives	Role dependencies
<i>Applicant</i>	<i>Potential members</i>	<i>Join society</i>	<i>Gatekeeper</i>
Knowledge seeker	Represents stakeholder: knowledge seeker	Request knowledge	Matchmaker
Knowledge owner	Represents stakeholder: knowledge owner	Exchange knowledge Announce offers	Knowledge owner Matchmaker
Editor	Realization of validation objective of non-life department	Exchange knowledge Publish knowledge Publish validated knowledge	Knowledge seeker Editor Knowledge owner
Visitor	Realization of distribution objective of non-life department	Distribute knowledge Browse repository	Visitor, seeker Matchmaker, editor

- support for coalition formation (in order to develop new solutions when knowledge is not available);
- support for direct exchange between parties where the negotiation of exchange conditions happens on a case-to-case basis.

These requirements indicate the need for both direct exchange, directed at finding relevant partners, and indirect exchange, through the repository, in which case the task of the system is to support publishing the results of direct knowledge exchanges. Furthermore, the knowledge market will use the same domain ontology as the knowledge repository. The stakeholder table is depicted in Table 2. The analysis of the objectives of the different stakeholders identifies operational roles in the society, listed in the dependencies column. The characteristics of operational roles were then further specified in a role table, depicted in Table 3.

Another result of the environment level is the specification of the normative characteristics of the society. Norms related to facilitation aspects have been identified at the coordination level. Other society norms are the result of the requirements and characteristics of the domain. Table 4 gives the results of norm analysis for different situations in the domain. This is not the complete listing of norms in the society, but describes the norms which have been implemented in the prototype.

Table 4 Norm analysis example

Description		Norm Analysis
1. Handling of seeker requests	Responsibilities	Initiator: knowledge seeker Action: matchmaker
	Triggers	Pre: seeker issues request Post: owners are informed of request
	Specification	whenever <i>knowledge-request</i> then <i>matchmaker is obliged to do distribute-request-to-partners</i>
2. Answer knowledge requests	Responsibilities	Initiation: matchmaker Action: knowledge-owner
	Triggers	Pre: matchmaker issues knowledge request Post: owners answer request
	Specification	whenever <i>request-knowledge</i> then <i>knowledge-owner is obliged to do answer-request before deadline</i>
3. Apply sanction	Responsibilities	Initiator: monitor Action: monitor
	Triggers	Pre: Deadline expired Post: Sanction applied to breaching party
	Specification	whenever <i>contract-breached</i> then <i>monitor is obliged to do apply-sanction (breaching-party)</i>
	Sanction	

Note that in the examples in Table 4, we have, for the sake of simplicity, abstracted from the specification of attributes of the concepts used. Later, during the behavior level, it will be determined whether a norm refers to a role, a scene, a transition, or a group. For example, norm 4, refers to a group, that is, both the knowledge-seeker as well as the knowledge-owner roles are affected by this norm.

5.1.3 Behavior level

Finally, the results of the previous methodological steps are combined and refined in the behavior level, to obtain a complete conceptual model for the knowledge market society. In the remainder of this section, we provide a detailed description of the social and interaction structures of the knowledge market.

Social structure In this section, we describe the social structure of the knowledge market. The role table obtained in the environment level is used as basis for the semi-formal role specifications for the external roles. These specifications, shown in Figure 3, can then be transformed in a formal definition using the LCR logic and used in the formal specification and verification of the knowledge market model. Note that, in order to keep the figures simple and readable, we have omitted the parameters of predicates in some cases. This must of course be part of the real specifications.

Facilitation roles are derived from the type of coordination, a network model in this case. As discussed in Section 5.1.1, besides the standard network facilitation roles gatekeeper, notary and monitor, the facilitation layer of the knowledge market also includes the role of matchmaker. In Figure 4, the facilitation roles for the knowledge market are depicted. Note the objectives of facilitation roles are mainly directed to handle requests from operational roles.

Furthermore, groups of roles were identified and group norms formalized in the group description. In particular, the following groups are relevant for the knowledge market:

- participant, referring to the seeker, owner and visitor roles;

Role: Knowledge Owner	
Role id	owner
Objectives	o ₁ = register-skills(matchmaker, skills) o ₂ = answer-request(matchmaker, question) o ₃ = publish-knowledge(editor, knowledge-item)
Sub-objectives	...
Rights	access-repository
Norms	IF request-knowledge(matchmaker, question, deadline) THEN OBLIGED(owner, answer-request(matchmaker, YN, question) BEFORE deadline
Type	external

Role: Knowledge Seeker	
Role id	seeker
Objectives	o ₁ = request-partner o ₂ = exchange-knowledge o ₃ = browse-repository
Sub-objectives	$\Pi_{o_1} = \{\text{get-potential-partners}(\text{question}, \text{partner-list}),$ $\text{choose-best-partner}(\text{partner-list}, \text{partner}),$ $\text{get-answer}(\text{question}, \text{partner}, \text{answer})\}$ $\Pi_{o_2} = \{\text{negotiate-exchange}(\text{question}, \text{partner}, \text{contract}),$ $\text{register-contract}(\text{notary}, \text{contract}),$ $\text{exchange-knowledge}(\text{partner})\}$
Rights	access-repository
Norms	IF agreed-share(partner) THEN OBLIGED (seeker, publish-repository(answer))
Type	external

Figure 3 Definitions of external roles and groups in the knowledge market

- partner, referring to the seeker and owner roles; and
- browser, referring to the seeker and the visitor.

For the group partner the norm that a partner is obliged to register its agreements holds.

Role dependencies and relations to external parties are depicted in Figure 5. The dependencies shown in this figure are related to the role definitions. Dependency relation diagrams, as the one depicted below, display dependencies as labeled arrows between two roles. The source role is the role where the objective is defined, the target role is the role that handles the objective, and the label indicates the objective.

Dependencies in a network society are per default network dependencies. If a given dependency should be of another type, then this must be specified in the dependency definition. Network dependencies identify an authorization equivalence relation between the roles, that is, both are authorized to request the objective. For example, consider the objective register of the role visitor below. A network relation between visitor and matchmaker means that either the visitor can request the matchmaker to handle his registration, or the matchmaker can request the visitor to register.

Interaction structure In this section, we describe the interaction structure of the knowledge market. A scene script is described by its players (roles), its desired results and the norms regulating the interaction. In the OM, scene scripts are specified according to the requirements of the society. The results of an interaction scene are achieved by the joint activity of the participating roles, through the realization of (sub-)objectives of those roles. A scene script establishes the desired interaction

Role: Gatekeeper	
Role id	gatekeeper
Objectives	$o_1 = \text{handle}(\text{membership-application}(\text{applicant}, \text{decision}))$
Sub-objectives	$\Pi_{o_1} = \{\text{ask-intentions}(\text{applicant}, \text{role}),$ describe-society, IF decide-acceptance(applicant, role, yes) THEN negotiate-social-contract(applicant, role, SC) $\}$
Rights	decide-acceptance
Norms	OBLIGED(gatekeeper, inform(applicant, decide-acceptance(applicant, role, decision))).
Type	institutional

Role: Matchmaker	
Role id	matchmaker
Objectives	$o_1 = \text{handle}(\text{request-partner}(\text{participant}, \text{question}))$ $o_1 = \text{handle}(\text{register}(\text{participant}, \text{type}))$
Sub-objectives	$\Pi_{o_1} = \{\text{find-potential-partners}(\text{question}, \text{members}, \text{potentials})$ $\forall p: \text{potentials distribute-request}(p, \text{question}, \text{YN}),$ $\text{answer-request}(\text{participant}, \text{partners})\}$
Rights	
Norms	IF requested(request-partner(participant, question) THEN OBLIGED(matchmaker, distribute-request) IF requested(register(p, type) THEN OBLIGED(matchmaker, verify-reputation(p))
Type	institutional

Figure 4 Definitions of facilitation roles in the knowledge market

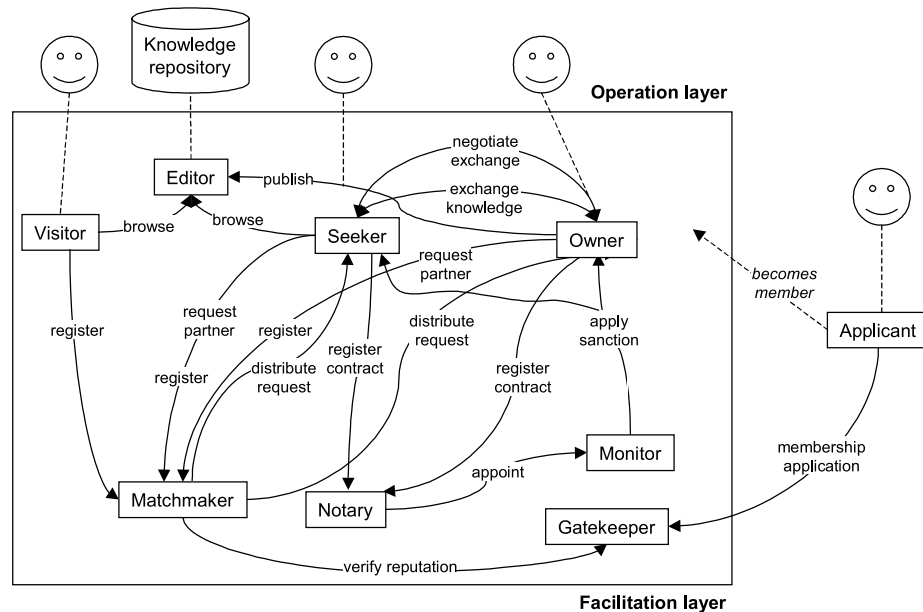


Figure 5 Role dependencies in the knowledge market

patterns between roles, that is, a desired combination of the (sub-)objectives of the roles. An example of a scene script is given in Figure 6.

The methodology prescribes that scenes should be specified for each role dependency identified in the social structure. For the knowledge market this means that a scene script is to be described

Table 5 Scenes in the knowledge market

Scene identifier	Roles	Connected to
Start	Gatekeeper, applicant	Register
Register	Matchmaker, participant	Verify reputation Request partner Publish Browse
Verify reputation	Matchmaker, gatekeeper	Register
Request partner	Partner, matchmaker	Distribute request Negotiate exchange Register End
Apply sanction	Monitor, partner	Exchange knowledge
End	Gatekeeper, participant	—

Interaction Scene: Partner Request	
Description	<i>Seeker requests possible partners that can answer knowledge need</i>
Roles	S: Knowledge-seeker(1), M: Matchmaker (1)
Results	DONE receive-partners(S, M, question, ListPartners)
Patterns	{ request-partner(S, M, question, deadline), distribute-request(M, knowledge-owners, answer-deadline) BEFORE request-deadline, request-deadline BEFORE answer-deadline, answer-deadline BEFORE deadline, receive-partners(S, M, question, List) BEFORE deadline, AND List = {P: DONE (answer-request(P, M, Yes, question) BEFORE answer-deadline))} }
Norms	OBLIGED request-knowledge(M, knowledge-owners, answer-deadline) BEFORE deadline IF request-knowledge(matchmaker, P, question, deadline) THEN OBLIGED answer-request(P, M, YN, question) BEFORE deadline

Figure 6 Scene script for ‘knowledge request’

for each of the labeled arrows depicted in Figure 5. Scenes are first described in informal terms in scene tables that are then translated into formal scene scripts. In Table 5, we list all scenes in the knowledge market, including their participating roles and the target scenes it is connected to. This table is not a complete scene table, but intended to give a summary of the knowledge market scenes and their relationships.

After having decided which scenes to specify for the knowledge market, the relationships between interaction scenes must be identified and formalized. For example, the exchange negotiation scene must occur after a successful partner request scene, and is unique for each group of partners. The scenes for registration of partnerships and exchange are also unique for each partner group, and a new instance should be created each time. Since participants can choose whether to browse the repository, request a partner, or publish a partner, the corresponding scenes are independent of each other and can occur in parallel. Visitors and seekers are not allowed in publish scenes and visitors are furthermore not allowed in partner request scenes, which indicates

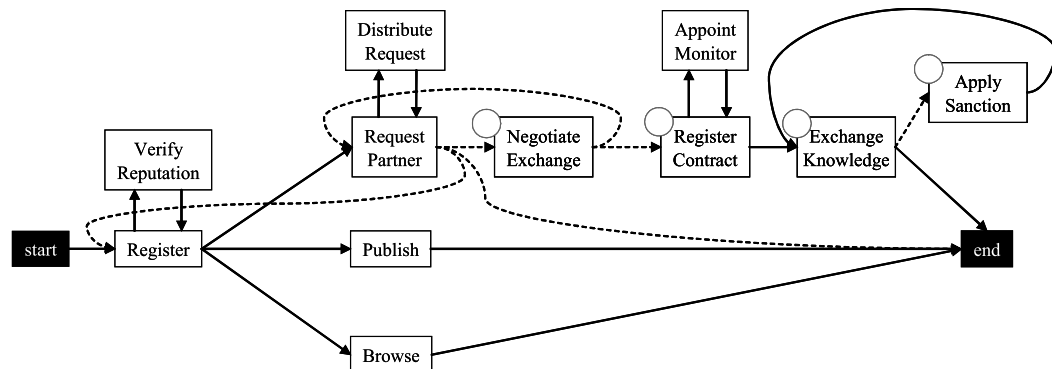


Figure 7 Interaction structure of the knowledge market

a transition norm on the admittance to those scenes. In Figure 6, we provide as an example the scene script for the scene ‘knowledge request’.

OperA uses a landmark-based approach to represent interaction. As described in Section 3, interaction scripts provide the minimum requirements and constraints for interaction that are necessary to achieve the interaction results sought by the design of the society, according to the view of the society. Such an approach allows agents to choose the best applicable action, from their own perspective, to achieve those landmarks.⁴ The concept of a landmark was first introduced in Smith *et al.* (1998), as defining a set of specifiable semantic properties that must hold for the agents involved (e.g. an offer has been made; an offer has been accepted). A landmark is identified by the set of propositions that are true in the state represented by the landmark. The formal definition of landmarks used in OperA is based on Kripke models (Dignum, 2004). Scene scripts can be seen as conversation patterns, and describe a relative sequence or pattern of landmarks in an interaction. In the IM, the pattern of landmarks associated with an interaction script is realized by specifying the protocol consisting of agent actions (actual conversation) for each landmark transition, such that performing those actions, from the source landmark, provably results in the target landmark. In the example depicted in Figure 6, the expected result of the interaction—that is, the seeker receives a list of possible partners, as specified in ‘Results’—is further refined by the landmarks specified in ‘Patterns’, which describe extra constraints on the (partial order of the) results of the scene.

Furthermore, because organizations make it possible for more complex activities to take place, scenes must be embedded in a broader context that allows for representation of how the overall objectives of the society can be achieved. OperA therefore enables the description of ordering and synchronization of interaction scenes. Scene scripts are organized into an interaction structure that specifies the coordination of the scene scripts. Transitions describe a partial ordering of the scenes, plus eventual synchronization constraints. Furthermore, the enactment of a role in a scene has consequences for the further enactment of roles in following scenes. That is, the evolution relations between roles must be described. Evolution relations specify the constraints that hold for the REAs as they move from scene to scene in the animated society. Note that several scenes can be happening at the same time and one agent can participate in different scenes simultaneously. Transition scripts must furthermore also describe the conditions for the creation of a new instance of the scene script. For each scene, the interaction structure also specifies an upper bound for the number of instances of that scene that are allowed simultaneously. Finally, each interaction structure definition must include the description of the initial and final scenes. Figure 7 depicts the interaction structure for the knowledge market. Note that dashed arrows indicate an exclusive OR (only one of the paths can be followed). A detailed description of scene transitions can be found in Dignum (2004).

⁴ In the case of non-intelligent agents, such choices can be fixed in the agent architecture by the agent designer, but the approach allows for the maximal use of the agent's capabilities and autonomy within the constraints and requirements imposed by society design.

Social Contract	
Agent	Anne
Role	Knowledge seeker
Clauses	<ol style="list-style-type: none"> 1. PERMITTED(Anne, access-kb([KB1, KB3, KB7]) 2. OBLIGED(Anne, publish-received-knowledge(item, KB3) allows(KO, publish)) 3. $\forall p: \text{contract}(p, \text{Anne}) \rightarrow \text{PERMITTED}(p, \text{publish}(p, \text{Anne's-item}, kb))$

Figure 8 Example of social contract

5.2 Social model

In the SM, the action of independent agents in the society is specified. Such agents seek to enact one of the operational roles in the society. In the knowledge market, agents enacting a facilitation role are controlled by the society. Therefore, external agents cannot apply to a facilitation role. This is not the case in a generic agent society, which allows for independence of facilitation roles. However, in most cases, society design will specify a number of institutional roles in order to keep control over the society in some way or another.

People seeking collaboration through the knowledge market will initiate a personal agent that acts as their avatar in the system. This agent uses the preferences and conditions specified by the user to find appropriate partners and negotiate exchange terms. Depending on the specific task, the personal agent will take either the role of knowledge seeker or knowledge owner. Requirements concerning privacy, secrecy and competitiveness between brands and departments that influence the channels and possibilities of sharing are also described in the specification of the personal assistants. Typically in the KennisNet, members do not have restrictions concerning sharing the knowledge they bring in. However, especially when new products are concerned, it can happen that agents of members involved will require such knowledge to be shared only within a restricted group.

Social contracts describe the agreements between participating agents and the knowledge market society. Negotiation of social contracts is carried out between the applicant agent and the gatekeeper agent, which will watch over the interests of the society itself. For example, Anne is a member of the KennisNet group that is seeking knowledge on price policies from the competition. Anne will initiate an agent enacting the knowledge seeker role in the knowledge market. During the Start scene, the conditions for Anne's agent will be negotiated and fixed in a social contract. Such a contract will, for instance, specify which parts of the repository Anne is allowed to access, what Anne's obligations are concerning the publication of knowledge items received as a result of an interaction, and whether Anne allows items that she provides to be published or not. This negotiation process can be very simple, in which case, Anne is offered a specification of the knowledge seeker role and either she accepts it as it is to be admitted or she refuses and admittance is denied. More sophisticated versions will require that agents are able to reason about goals, norms and objectives.

The example contract depicted in Figure 8 describes the social contract between agent Anne and the knowledge market society, by which Anne is given permission to access some of the knowledge bases in the repository, namely the knowledge bases identified by *kb1*, *kb3* and *kb7*. Anne is obliged to publish all received knowledge in knowledge base 3, given that publishing is allowed by the knowledge owner involved in that exchange, and Anne allows her knowledge (which she may need to release as counter activity in an interaction contract) to be published by her partners, in whichever knowledge bases her partners can access.

5.3 Interaction model

The following example describes a contract between two members. In this example, which is fictitious but typically possible in the domain of non-life insurance, Anne will provide Bob with a

Interaction Contract: 'ID'	
Parties	Anne (A), Bob (B)
Clauses	<ol style="list-style-type: none"> 1. OBLIGED A DONE(A, receive(B, report-concurrent-prices) BEFORE <i>next-week</i> 2. IF received(B, report-concurrent-prices) THEN OBLIGED B (receive(A, comment-report-concurrent-prices) BEFORE <i>3-days</i> AND receive(A, concept-pricing) BEFORE <i>1-month</i>) 3. IF delayed(B, concept-pricing) THEN OBLIGED B inform(A, delayed(concept-pricing))

Figure 9 Example of interaction contract

report about competition prices, on condition that Bob will give her comments on the report (that she will have to present to her unit directors) and eventually share with her his new pricing concept for car insurance. This contract is generated during the 'negotiate partnership' scene and registered in the 'register partnership' scene. In this scene, the notary agent will assign a monitor agent to check the fulfillment of the contract between Anne and Bob. Monitoring can be a very simple activity, where status is checked when a deadline is reached. However, we have chosen to use an agent as monitor because monitors can take a more active role, reminding parties of approaching deadlines or by suggesting possible actions when sanctions occur. Figure 9 gives an informal specification of the clauses of this contract. In the case that one of the agents will not fulfill its commitments, sanctions will be applied. When sanctions are not explicitly specified in the contract, the norms of the society will be used. For instance, the knowledge market follows the norm that agents that do not fulfill their commitments are given less priority in exchanges. In addition, it is possible to consider the publication of a list of best and worse members.

5.4 Implementation

In parallel to the development of the repository, a prototype of the knowledge market was developed. The aim of the prototype was to test the applicability of existing, freely available agent tools to the development of agent societies (Khalil, 2002). As result of this project, knowledge exchange between two agents and mediated by a matchmaker was implemented in both Jade⁵ and Zeus.⁶

In both prototypes, agents exchange knowledge descriptions based on keywords. Instead of a full-blown reciprocity mechanism as specified in the requirements of knowledge market, we chose to use currency-based exchange in the prototype. That is, each agent receives an amount of points that can be use to 'buy' knowledge items and earns points by providing its knowledge to others. Furthermore, we developed a simple heuristic to determine similarity and relevance between knowledge items based on ontological proximity.

However, more work is needed concerning the practical implementation of the knowledge market. Assuming that OperA-based tools for the building of multi-agent systems are available, the implementation of the knowledge market must be extended in at least two directions.

1. *Robust implementation of the complete system.* We envision that this process can be incremental in the sense that a first implementation will be based on homogenous agents (following the current prototype) and be extended with the application of heterogeneous agents (built using different tools and architectures, possibly 3APL (Hindriks *et al.*, 1999) and Jade).
2. *Evaluation of user interaction with the system in a lab environment, as well as in a real environment if possible.* This allows determination of the relative contribution of an agent-based approach compared with traditional means of knowledge exchange, both at the level of the individual users and at the level of the organization.

⁵ <http://sharon.cselt.it/projects/jade>

⁶ <http://www.labs.bt.com/projects/agents/zeus/index.htm>

Matching knowledge supply and demand was one of the main challenges of the project. For example, if the seeker is looking for knowledge items on snow damage in motorcycles and no exact match can be found, would he/she rather get items on snow damage in cars, or generic motorcycle damage? The software system implemented a simple protocol for knowledge matching, based on ontological distances between concepts. More empirical research is needed in this area in order to determine realistic requirements for knowledge matching.

We plan to build a modeling tool for agent societies, according to the OperA framework, by extending the institution specification tool ISLANDER (Esteva *et al.*, 2001) with the organizational concepts formalized in the OperA framework. The resulting tool will allow specifying relations between groups and participants, and different interaction types and constraints over interactions. It can be populated with different types of agents (designed using the agent programming language 3APL (Hindriks *et al.*, 1999) and possibly other agent languages, such as Jade). It will furthermore provide a graphical interface for the animation of agent societies.

6 Conclusions and future work

Current developments in KM show a shift in the focus of KM from knowledge to collaboration. The aim of KM is no longer just the management of activities related to the creation, preservation and distribution of knowledge assets but the management and nurturing of collaboration between people. Such collaboration management systems call for approaches that are reactive and proactive in relation to the needs and expectations of its users. Agent concepts, which originated in artificial intelligence but which have further developed and evolved in many areas of computing, hold great promise for responding to the new realities of knowledge and collaboration management. In this paper, we have presented an agent-based model for organizations that fulfills the specification requirements of collaboration management systems. The model is being applied to the development of a knowledge market for the KC Non-Life Insurance.

Agent concepts can fundamentally alter the nature of knowledge management both in the way KM systems are built as well as the way organizations are analyzed and modeled. On the one hand, the technical embodiment of these concepts can lead to advanced functionality of KM systems, e.g. personalization of knowledge presentation and matching supply and demand of knowledge. On the other hand, the rich representational capabilities of agents as modeling entities allow more faithful and effective treatments of complex organizational processes. In our opinion, one of the main contributions of agent-based modeling of KM environments is that it provides a basis for the incorporation of individual initiative and collaboration into formal organizational processes. Future research in agent-oriented approaches to knowledge management and collaborative systems must therefore include the following:

- methodologies are needed that support the analysis of KM needs of organizations and its specification using software agents and agent societies;
- reusable agent-oriented KM frameworks, including the description of agent roles, interaction forms and knowledge description;
- agent-based tools for organizational modeling and simulation that help determine the knowledge processes of the organization;
- the role of learning in agent-based KM systems, namely, how to use agent learning to support and extend knowledge sharing.

The OperA framework presented in this paper is 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. The organizational structure of the society, as intended by the organizational stakeholders, is described in the OM. The agent population of an OM is specified in the SM in terms of social contracts that make explicit the commitments which regulate the enactment of roles by individual agents. Finally, given an agent population for a society, the IM describes possible interaction between agents.

The OperA methodology supports the specification of an agent society by analyzing a given domain and determining the type and structure of the agent society that best models that domain. This methodology provides generic facilitation and interaction frameworks for agent societies that implement the functionality derived from the coordination 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 to 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 cooperation possibilities between partners.

The formal semantics for OperA formalizes these three models but does not include the specification of the players (i.e. the agents themselves) in a society. As it is now, the OperA formalism does not result in an implemented system but in a conceptual model. This means that typical system properties such as liveness cannot be verified directly, system traces cannot be generated. An extension to the formal language to support the specification of the internal aspects of agents would enable the complete specification of an animated society.

In our opinion, the development of open multi-agent systems will increasingly take place in ways such as the one proposed by OperA. That is, separating the specification of social issues from design of the individual agents, as stated in the first autonomy requirement. Therefore, a novel area of research is the description of formal languages to describe the integration of a formal SM, such as OperA, with formal agent architectures. Such a language will enable are to determine the exact semantic relations and system properties of open agent societies.

A main direction for future research is the development of practical tools to build OperA models that enable the design, implementation and verification of multi-agent systems. Such tools should be able to guide the engineering process, by following the development methodology, and enable the specification and the automatic configuration of agent societies according to the OperA model, as well as supporting the verification of models. We are currently initiating a project to develop a computational system to specify an OperA model and automatically generate a multi-agent system that implements that model. The resulting multi-agent system should include the functionality of the OM, institutional agents to enact the facilitation roles, and the capabilities to enable the incorporation of external agents that will enact operational roles. The tool should furthermore provide software mechanisms for security and robustness to enable building real-world applications, beyond pilot implementations.

References

- Ahuja, M and Carley, K, 1998, "Network structure in virtual organizations" *Journal of Computer Mediated Communication* **3**(4).
- Ali, I, Pascoe, C and Warne, L, 2002, "Interactions of organizational culture and collaboration in working and learning" *IEEE Journal Educational Technology & Society* **5**(2).
- Allee, V, 2002, "Knowledge networks and communities of practice" *Organizational Development Practitioner Online* **32**(4) <http://www.odnetwork.org/odponline/vol32n4/knowledgenets.html>.
- Bonifacio, M, Bouquet, P and Traverso, P, 2002, "Enabling distributed knowledge management: Managerial and technological implications" *Novatica and Informatik/Informatique* **III**(1).
- Bresciani, P, Giorgini, P, Giunchiglia, F, Mylopoulos, J and Perini, A, 2004, "TROPOS: An agent-oriented software development methodology" *Journal of Autonomous Agents and Multi-Agent Systems* **8**(3) 203–236.
- Brown, J and Gray, E, 1995, "The people are the company" *Fast Company* **1**(1) 78–82, <http://www.fastcompany.com/online/01/people.html>.
- Castelfranchi, C, 2000, "Engineering social order. In A Omicini, Tolksdorf and F Zambonelli (eds) *Engineering Societies in the Agent World (Lecture Notes in Artificial Intelligence, 1972)*, Berlin: Springer.
- Castro, J, Kolp, M and Mylopoulos, J, 2002, "Towards requirements-driven information systems engineering: The TROPOS project" *Information Systems* **27** 365–389.
- Dastani, M, Dignum, V and Dignum, F, 2003, "Role assignment in open agent societies" in *Proceedings of AAMAS '03, 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia.

- Davenport, T and Prusak, L, 1998, *Working Knowledge: How Organizations Manage What they Know*, Harvard Business School Press.
- Dignum, V, 2002, "A knowledge sharing model for peer collaboration in the non-life insurance domain" in *Proceedings of the German Workshop on Experience Management (Lecture Notes in Informatics)*, Berlin: German Society for Informatics.
- Dignum, V, 2004, A model for organizational interaction: based on agents, founded in logic, PhD Thesis, Utrecht University.
- Dignum, V and Dignum, F, 2003, "The knowledge market: Agent mediated knowledge sharing" in V Marik, J Muller and M Pechoucek (eds) *Multi-Agent Systems and Applications, CEEMAS 2003, Prague (Lecture Notes in Artificial Intelligence, 2691)*, Berlin: Springer, pp. 168–179.
- Dignum, V, Meyer, J-J, Dignum, F and Weigand, H, 2003, "Formal specification of interaction in agent societies" in M Hinchey, J Rash, W Truszkowski, C Rouff and D Gordon-Spears (eds) *Formal Approaches to Agent-Based Systems (FAABS '02) (Lecture Notes in Artificial Intelligence, 2699)*, Berlin: Springer.
- Dignum, V, Meyer, J-J, Weigand, H and Dignum, F, 2002, "An organizational-oriented model for agent societies" in *Proceedings of the International Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA '02)*, AAMAS, Bologna, Italy.
- Dignum, V and van Eeden, P, 2003, "Seducing, engaging and supporting communities at achmea" in *Proceedings of the 4th European Conference on Knowledge Management*, Oxford, UK, 18–19 September.
- Dignum, V and Weigand, H, 2002, "Towards an organization-oriented design methodology for agent societies" in V Plekhanova (ed.) *Intelligent Agent Software Engineering*, Idea Publishers.
- Domingue, JB and Motta, E, 1999, "A knowledge-based news server supporting ontology-driven story enrichment and knowledge retrieval" *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW '99)*.
- Eppler, M, 2002, "Glossary definition: Knowledge management". NetAcademy. http://www.knowledgemedia.org/modules/glossary/view_def.php?id=mediamanagement-20.
- Esteva, M, Rodriguez, J, Sierra, C, Garcia, P and Arcos J, 2001, "On the formal specifications of electronic institutions" in F Dignum and C Sierra (eds) *Agent-mediated Electronic Commerce (The European AgentLink Perspective) (Lecture Notes in Artificial Intelligence, 1991)*, Berlin: Springer, pp. 126–147.
- Ferber, J and Gutknecht, O, 1998, "A meta-model for the analysis and design of organizations in multi-agent systems" in *Proceedings of the Third International Conference on Multi Agent Systems (ICMAS-98)*, Paris, Los Alamitos, CA: IEEE Press.
- Gandon, F, Dieng, R, Corby, O and Giboin, A, 2000, "A multi-agent system to support exploiting an XML-based corporate memory" *Proceedings of PAKM '00*, Basel.
- Gongla, P and Rizzuto, C, 2001, "Evolving communities of practice: IBM global services experience" *IBM Systems Journal* 40(4).
- Guizzardi, R, Perini, A and Dignum, V, 2004, "Providing knowledge management support to communities of practice through agent-oriented analysis" in *Proceedings of I-KNOW '04, 4th International Conference on Knowledge Management*, Graz, Austria, 30 June–2 July.
- Hindriks, K, de Boer, F, van der Hoek, W and Meyer, J-J, 1999, "Agent programming in 3APL" *Journal of Autonomous Agents and Multi-Agent Systems* 2(4) 357–401.
- Howden, N, Ranquist, R, Hodgson, A and Lucas, A, 2001, "Jack: summary of an agent infrastructure" in *Proceedings of the 5th International Conference on Autonomous Agents*.
- Khalil, A, 2002, Multi-agent system for achmea, MSc Thesis, Utrecht University.
- Klusck, M (ed.) 1999, *Intelligent Information Agents: Agent-based Information Discovery and Management in the Internet*, Berlin: Springer.
- Luck, M, McBurney, P and Preist, C, 2003, *Agent Technology: Enabling Next Generation Computing: A Roadmap for Agent Based Computing*. AgentLink II.
- Mentzas, G, Apostolou, D, Young, R and Abecker, A, 2001, "Knowledge networking: A holistic solution for leveraging corporate knowledge" *Journal of KM* 5(1).
- Noriega, P, 1997, Agent-mediated auctions: The fishmarket metaphor. IIIA Monograph Series, 8. PhD Thesis, Institut d'Investigaci'o en Intel.ligència Artificial (IIIA).
- O'Leary, D, 1998, "Knowledge management systems: Converting and connecting" *IEEE Intelligent Systems* May–June, 30–33.
- Omicini, A, 2001, "SODA: Societies and infrastructures in the analysis and design of agent-based systems" in P Ciancarini and M Wooldridge (eds) *Agent-Oriented Software Engineering (Lecture Notes in Artificial Intelligence, 1957)*, Berlin: Springer, pp. 185–193.
- Preece, J, 2000, *Online Communities: Designing Usability, Supporting Sociability*, New York: John Wiley & Sons.
- Pumareja, D, Bondarouk, T and Sikkil, K, 2003, "Supporting knowledge sharing isn't easy: Lessons learnt from a case study" *Information Resource Management Association International Conference (IRMA '03)*, Philadelphia, PA.

- Rodriguez, JA, 2001, On the design and construction of agent-mediated electronic institutions. PhD Thesis, Institut d'Investigaci'o en Intel·ligència Artificial (IIIA).
- Smith, I, Cohen, P, Bradshaw, J, Greaves, M and Holmback, H, 1998, "Designing conversation policies using joint intention theory" *Proceedings of the Third International Conference on Multi Agent Systems (ICMAS-98)*, Paris, Los Alamitos, CA: IEEE Press, pp. 269–276.
- Smith, R and Farquhar, A, 2000, "The road ahead for knowledge management: an AI perspective" *AI Magazine* 21(4) 17–40.
- van Elst, L, Dignum, V and Abecker, A, 2004, "Towards agent-mediated knowledge management" in L van Elst, V Dignum and A Abecker (eds) *Agent-Mediated Knowledge Management: Selected Papers (Lecture Notes in Artificial Intelligence, 2926)*, Berlin: Springer.
- Wenger, E, McDermott, R and Snyder, W, 2002, *Cultivating Communities of Practice*, Harvard Business School Press.
- Wooldridge, M, Jennings, N and Kinny, D, 2000, "The Gaia methodology for agent-orient analysis and design" *Autonomous Agents and Multi-Agent Systems* 3(3).
- Zambonelli, F, Jennings, N and Wooldridge, M, 2001, "Organizational abstractions for the analysis and design of multi-agent systems" in P Ciancarini and M Wooldridge (eds) *Agent-Oriented Software Engineering (Lecture Notes in Artificial Intelligence, 1957)*, Berlin: Springer.