An Agent-Mediated Approach to the Support of Knowledge Sharing in Organizations

Virginia Dignum,
Frank Dignum,
John-Jules Meyer

institute of information and computing sciences, utrecht university
technical report UU-CS-2004-019
www.cs.uu.nl

An Agent-Mediated Approach to the Support of Knowledge Sharing in Organizations

Virginia Dignum, Frank Dignum, John-Jules Meyer

University Utrecht, The Netherlands

{virginia, dignum, jj}@cs.uu.nl

ABSTRACT

The realization that 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 on 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 years, 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 on all levels of an organization [17]. Practitioners and business managers alike agree that issues of technology, process, people, and content must be addressed in KM to achieve success [30]. 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 of 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 [5]. 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. Moreover, organizations must keep in mind the limitations of KM and understand that knowledge alone does not guarantee a creative response to decision-making situations. Einstein has stated 'imagination is more important than knowledge'. The realization that such gaps exist is of the utmost importance for the success of knowledge management initiatives. A KM system that links to the real needs and goals of people on their real-world practices, and facilitates their contacts and interactions, has a much higher chance of success than one than 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 [10]:

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 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 are 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 [23]. Therefore, agents are, in our opinion, utmost 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 [15], [12].

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 by 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. That is, the actual behavior of the society emerges from the goal-pursuing behavior of the individual agents within the constraints set by the organizational. This creates a need to check conformance of the actual behavior to the desired behavior 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 [31], [4], [20]. 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, such 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 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 [12].

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. Agents in Knowledge Management

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) [26]. 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. In research as well as in first generation "real world" applications, agent-based applications are available to support various aspects of KM, from personal information agents to agent-based workflows for business process-oriented KM. Basically, the use of agents in KM can be seen in two perspectives. On one hand, agents are used to model the organizational environment where the KM system will operate and, on the other hand, software agents can be used to implement the functionality of KM systems. Most existing KM projects involving agent technology concentrate on the second perspective, that is, use agents as modeling primitives in KM implementation tools.

Agent-Mediated Knowledge Management is a new research direction that aims at the cross-fertilization between the KM and the intelligent agent research fields [31]. Applications of agent technology to KM start from the realization that KM and multi-agent systems have several similarities. 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 as 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 systems 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. Current multi-agent models are not well suited for KM because they either take a centralistic approach to organizational design (cf. for example [33]), 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 [15]. The 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 system to KM, due to the fact that not many applications are available, we believe that agent technology helps to develop KM system that are more flexible, and 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.

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 above traditional MAS models. On the one hand, coordination and interaction in MAS are usually described in the context of the actions and mental states of individual agents [19]. In open societies, however, such 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 a conceptual, fictive behavior. 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).

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 [11], [34]. Castelfranchi 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 [6]. 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 in 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 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.

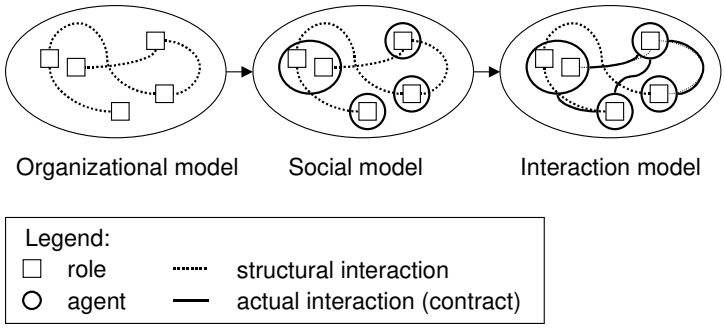

Figure 1: Organizational framework for agent societies

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

[1] A formalism to provide logical semantics to the model is described in [11].

perform society role(s) as a means to realize their own goals according to their own internal aims and architecture. Whereas 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 [7]. Contracts are introduced as a means to 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 [15]. 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, permits to give models a precise semantics, supports the use of structured design techniques and formal analysis, facilitating development, composition and reuse [18], 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 [11]. 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 treats temporal and communicative aspects and furthermore is able to represent deadlines and their influence on the behavior of the model.

3.a. The Organizational Model

Starting point of the Agent Society Model is the organizational model (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 goals that determine agent roles and interaction norms. The organizational model 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, 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.

3.b. 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 organizational model. This means that several populations are possible for each organizational model. Agent populations of the organizational model are described in the social model (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 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 might 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 Social Model 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.c. The Interaction Model

Finally, interaction between agents populating a society is described in the interaction model (IM) by means of interaction contracts. This model accounts for the actual (emergent) behavior of the society at a given moment. Interaction agreements between

agents are described in interaction contracts. Usually interaction contracts will 'follow' the intended interaction possibilities specified in the organizational model. However, because of the autonomous behavior of agents, the interaction model must be able to accommodate other interaction contracts describing new, emergent, interaction paths, to the extent allowed by the organizational and social models.

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 such 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 IM consists of a set of instances of scene scripts (called scenes), described by the interaction contracts between the role enacting agents 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.d. 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 [13]. The methodology provides generic facilitation and interaction frameworks for agent societies that implement the functionality derived from the co-ordination model applicable to the problem domain. Standard society types as market, hierarchy and network, can be used as starting point for development and can be extended where needed and determine the basic norms and facilitation roles necessary for the society. These coordination models describe the different types of roles that can be identified in the society, and such issues as communication forms, desired social order and co-operation 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, as any other software systems, cannot be seen isolated from the (organizational) context where 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 is spent on state-of-the-art ICT systems that are hardly used subsequently. In our opinion, such mistakes are because system development mostly concentrates on the technical aspects of the system and organizational, cultural and users aspects are largely ignored or assumed accomplished. We think that system development must start with the analyses and facilitation of the social environment where it will be inserted. A preliminary step for the OperA methodology is the assessment of the applicability of the agent paradigm to the problem on hand. The methodology we describe in the remainder of this section is structured in three steps. Firstly, it guides the design of the Organizational Model to implement the desired organizational structure of an agent society. Agents are integrated into the society design in the Social Model, by means of social contracts, and agent interactions are fixed in the Interaction Model 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.d.1. Organizational Model design

The first step of the OperA methodology results in the specification of the Organizational Model (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, of which behavior and animation can be verified and compliance to the domain requirements can be checked. These steps result in a complete Organizational Model 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: consists of the analysis of the internal behavior of the system. This process is based on the basic Organizational Model, 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 Organizational Model 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.d.2. Social Model design

The design of the Social Model of an OperA society is in reality much more of an operational issue than a methodological one. That is, where 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 Social Model depends on the activities of specific agents, and is for the most part determined at 'run time'. The design of the Social Model 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 Organizational Model will result in many different Social Models. Based on the Organizational Model specification of an agent society and on a set of specific agents, the social model 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.d.3. Interaction Model design

The creation of an Interaction Model depends on the activities of specific role enacting agents, guided by the description of scenes in the scene scripts specified in the OM. That is, the generation of an Interaction Model for an OperA society depends on:

- The specific role-enacting agents 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 Interaction Models are possible. As with the SM, it allows the incorporation of the specific requirements and characteristics of agents and enables for a more realistic treatment of autonomy. Based on the OM specification of an agent society and on a population of role-enacting agents described in the SM, the interaction model design will describe the interaction contracts for scene scripts.

4. A Knowledge Sharing Scenario

Recent developments show a shift in the focus of Knowledge Management (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 knowledge management 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 community of practice 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 communities of practice and most people belong to at least one of them [5]. Nurturing communities is hard enough when the members are in a single location with good connectivity and increases considerably when the members are spread around different locations, possibly in different areas and with different languages and cultures.

According to [3], 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. Etienne Wenger indicates that such environments are especially suitable for organizations' newcomers, who can learn the common procedures and working practices by informally collaborating with others, and, with time, become a fully integrated and active member of the community [32].

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 [21], [27]. 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 their 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 [9].

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.a, after which we describe the system and its development in detail in section 5.

4.a. 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 leading 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, often they will 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 [8].

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 [14]. The first steps towards the realization of these objectives concerned the development of a Community of Practice, 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, are depicted in Figure 2. The development of the knowledge repository was inspired on work by [16], [25] and [20]. 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 feeding community feeling.

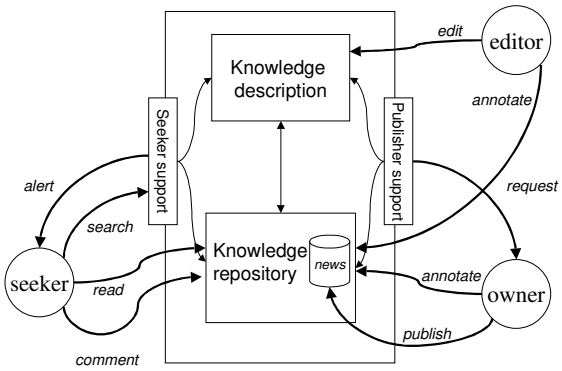

Figure 2: The architecture of KennisNet

4.b. 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. As 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 the previous ones. 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 question from others, and so on. Description of knowledge items includes:

- **Identification**, that is, information about name, datum of publication and status,
- **Content**: describes the actual meaning of an item, using aspects as keywords, abstract, link classification ontology and comments,
- **Context**: contains relevant information surrounding the creation and use of an item. Includes name of submitter, related projects, intended use and reasons for publication,

- **Structure**: describes accessibility and use issues. It includes 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.c. Evaluation of the knowledge repository

After the knowledge repository was running for around one year, we conducted a user satisfaction survey, reported in [28]. 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 [2]. 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 [1]. 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

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

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 conditions[3]. Motivated by the realization that another approach was needed to the problem at hand, the Knowledge Center 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 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 was desired (the collaboration autonomy). 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. It is a generic model for the design of multi-agent systems, which has the added-value of a formal semantics [11], [15] and a customized development methodology [13], and as such suitable to the development of multi-agent system 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

[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.

knowledge owner and reaches the knowledge seeker in a just in time, just-enough basis. In the remainder of this section the system, developed using the Opera Model, is described.

5.a. Design of the OM

In this section, we describe the specification of the Organizational Model 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.a.1. Coordination Level

As described in section 3.d.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 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 [1]. 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 require 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 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.

Table 1: Facilitation roles in the Knowledge Market

ROLE	OBJECTIVES	ABSTRACT NORMS
Gatekeeper	Accept participants	– Obliged 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	– Obliged to register exchanges – Allowed to request exchange information from seekers and owners
Monitor	– Give alerts on deadlines and collaboration terms	– Obliged to alert notary on sanctions
Matchmaker	– Register participants, skills and needs – Accept and distribute exchange requests	– Obliged to distribute requests – Obliged to give distribution requests back to requester

5.a.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.c, 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
- 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 in 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.

Table 2: Stakeholder table for the Knowledge Market society

STAKEHOLDER	OBJECTIVES	DEPENDENCIES
Non-life department	Validate knowledge	Gatekeeper, Editor
	Distribute knowledge within group	Knowledge seeker, Knowledge owner
	Distribute knowledge outside group	Visitor
Knowledge seeker	Get help	Non-life department, Knowledge owner
Knowledge owner	*Get recognition through*: Provide help Publish own knowledge	Knowledge seeker Non-life department Editor

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.

Table 3: Role Table for Knowledge Market

ROLE	RELATION TO SOCIETY	ROLE OBJECTIVES	ROLE DEPENDENCIES
Applicant	*Potential members*	*Join society*	*Gatekeeper*
Knowledge seeker	Represents stakeholder: Knowledge seeker	Request knowledge	Matchmaker
		Exchange knowledge	Knowledge owner
Knowledge owner	Represents stakeholder: Knowledge owner	Announce offers	Matchmaker
		Exchange knowledge	Knowledge seeker
		Publish knowledge	Editor
Editor	Realization of validation objective of non-life department	Publish validated knowledge	Knowledge owner
		Distribute knowledge	Visitor, seeker
Visitor	Realization of distribution objective of non-life department	Browse repository	Matchmaker Editor

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 result 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** *knowledge-request* **then** *matchmaker* **is** *obliged* **to do** *distribute-request-to-partners*
2. Answer knowledge requests	**Responsibilities**	Initiation: matchmaker Action: knowledge-owner
	Triggers	Pre: matchmaker issues knowledge request Post: owners answer request
	Specification	**whenever** *request-knowledge* **then** *knowledge-owner* **is** *obliged* **to do** *answer-request* **before** *deadline*
3. Apply sanction	**Responsibilities**	Initiator: monitor Action: monitor
	Triggers	Pre: Deadline expired Post: Sanction applied to breaching party
	Specification	**whenever** *contract-breached* **then** *monitor* **is** *obliged* **to do** *apply-sanction(breaching-party)*
	Sanction	

Note that in the examples above, 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.a.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.

5.a.3.1 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.

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	Πo_1 = {get-potential-partners(question, partner-list), choose-best-partner(partner-list, partner), get-answer(question, partner, answer) } Πo_2 = {negotiate-exchange(question, partner, contract), register-contract(notary, contract), exchange-knowledge(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

Facilitation roles are derived from the type of coordination, a network model in this case. As discussed in section 5.a.1, besides the standard network facilitation roles gatekeeper, notary and monitor, the facilitation layer of 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.

Role: Gatekeeper	
Role id	gatekeeper
Objectives	o_1 = *handle*(membership-application(applicant, decision))
Sub-objectives	Πo_1 = { ask-intentions(applicant, 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 = *handle*(request-partner(participant, question)) o_1 = *handle*(register(participant, type))
Sub-objectives	$\Pi o1$ = {find-potential-partners(question, members, potentials) $\forall$p: potentials distribute-request(p, question, YN), answer-request(participant, 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

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 roles seeker, owner and visitor,
- Partner, referring to the roles seeker and owner, 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.

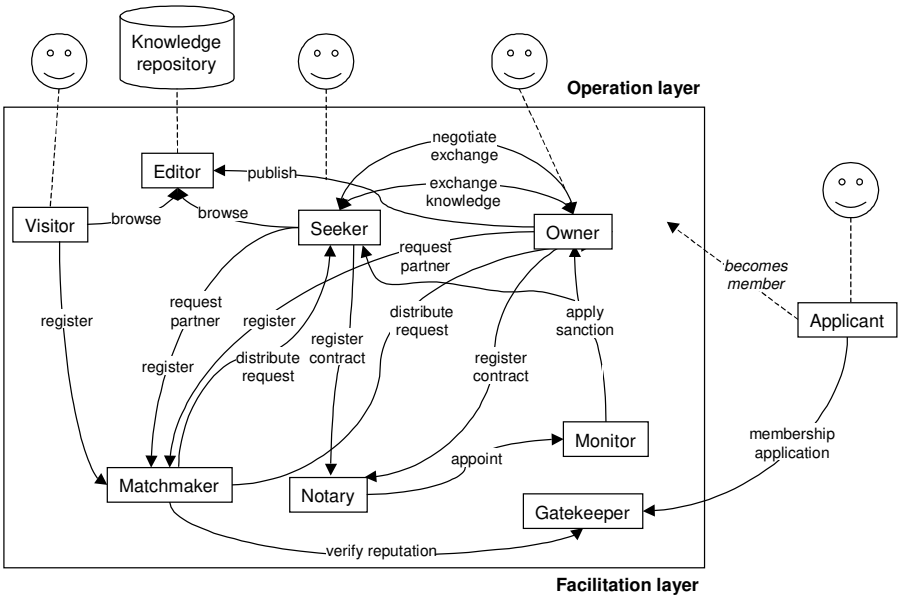

Figure 5: Role dependencies in the Knowledge Market

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 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.

5.a.3.2 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 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 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.

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	- -

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 a transition norm on the admittance to those scenes. In Figure 6, we provide as example, the scene script for the scene 'Knowledge Request'.

Interaction Scene: Partner Request	
Description	*Seeker requests possible partners that can answer knowledge need*
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'

OperA uses a landmark based approach to represent interaction. As described in chapter 3, interaction scripts provide the minimum requirements and constraints for interaction that are necessary to achieve the interaction results sought by the society design, according to the view of the society. Such approach allows agents to choose the

best applicable action, from their own perspective, to achieve those landmarks.[4] The concept of landmark was first introduced in [29], as defining a set of specifiable semantic properties that must hold of 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 [15]. Scene scripts can be seen as conversation patterns, and describe a relative sequence or pattern of landmarks in an interaction. In the Interaction Model, 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 in organizations activities that are more complex can take place, scenes must be embedded in a broader context, that allows representing how the overall society objectives 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 role-enacting agents 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.

[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

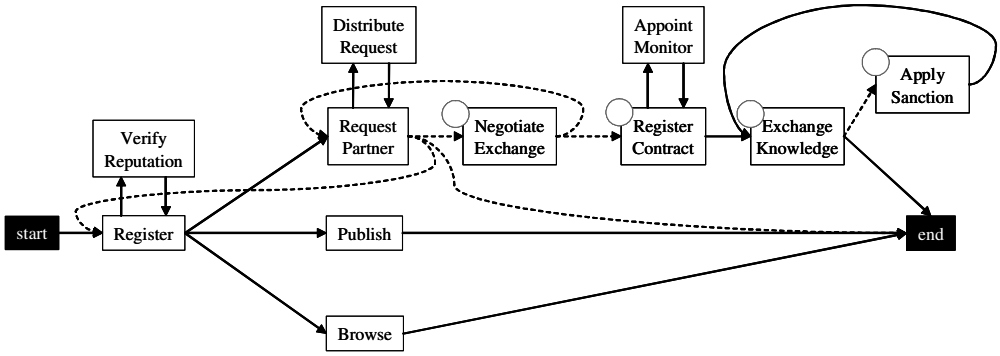

Figure 7: Interaction structure of Knowledge Market

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 [15].

5.b. Social Model

In the Social Model, 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 of 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 done 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 contract will, for instance, specify which parts of the repository Anne is allowed to access, which are the obligations of Anne concerning the publication of knowledge items received as result of an interaction, and whether Anne allows for 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.

Social Contract	
Agent	Anne
Role	Knowledge seeker
Clauses	
	1. PERMITTED(Anne, access-kb([KB1, KB3, KB7])
	2. OBLIGED(Anne, publish-received-knowledge(item, KB3) | allows(KO, publish))
	3. ∀p: contract(p, Anne) → PERMITTED(p, publish(p, Anne's-item, kb))

Figure 8: Example of social contract

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 base that her partner can access.

5.c. Interaction Model

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

Interaction Contract: '*ID*'	
Parties	Anne (A), Bob (B)
Clauses	
1.	OBLIGED A DONE(A, receive(B, report-concurrent-prices) BEFORE *next-week*
2.	IF received(B, report-concurrent-prices) THEN OBLIGED B (receive(A, comment-report-concurrent-prices) BEFORE *3-days* AND receive(A, concept-pricing) BEFORE *1-month*)
3.	IF delayed(B, concept-pricing) THEN OBLIGED B inform(A, delayed(concept-pricing))

Figure 9: Example of interaction contract

5.d. 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, free available agent tools to the development of agent societies [23]. As result of this project, knowledge exchange between two agents and mediated by a matchmaker was implemented in both Jade[5] 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 in the prototype for a currency-based exchange. That is, each agent receives an amount of points that can be use to 'buy' knowledge items and earn 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 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 [22] and Jade).
2. *Evaluation of user interaction with the system in a lab environment, as well as in a real environment if possible*. This in order to determine 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 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 [18] 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 [22] 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, 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:

- Methodologies are needed that support the analysis of knowledge management needs of organizations and its specification using software agents and agent societies

- Reusable agent-oriented knowledge management 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 knowledge management 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 Organizational Model (OM). The agent population of an OM is specified in the Social Model (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 Interaction Model (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 co–ordination model applicable to the problem domain. Standard society types such as market, hierarchy and network, can be used as starting points for development and can be extended where needed and determine the basic norms and facilitation roles necessary for the society. These coordination models describe the different types of roles that can be identified in the society and issues such as communication forms, desired social order and co-operation possibilities between partners.

The 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 [15]. 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 social model, such as OperA, with formal agent architectures. Such a language will enable 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 enables the design, implementation and verification of multi-agent systems. Such tool 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 and support 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.

7. References

[1] Ahuja, M., Carley, K. 1998: Network Structure in Virtual Organizations, *Journal of Computer Mediated Communication*, **3**(4), June.

[2] Ali, I., Pascoe, C., Warne, L. 2002: Interactions of organizational culture and collaboration in working and learning. IEEE Journal Educational Technology & Society. 5(2).

[3] Allee, V. 2002: Knowledge Networks and Communities of Practice. Organizational Development Practitioner Online, vol. 32, no. 4, at: http://www.odnetwork.org/odponline/vol32n4/knowledgenets.html

[4] Bonifacio, M., Bouquet, P., Traverso, P. 2002: Enabling Distributed Knowledge Management: Managerial and Technological Implications, Novatica and Informatik/Informatique, III (1).

[5] Brown, J., Gray, E. 1995: The People are the Company. Fast Company, (1):78-82, Boston, MA. URL: http://www.fastcompany.com/online/01/people.html

[6] Castelfranchi, C. 2000: Engineering Social Order. In: Omicini, A. Tolksdorf, and Zambonelli, F. (Eds.): Engineering Societies in the Agent World, LNAI 1972, Springer-Verlag.

[7] Dastani, M., Dignum, V., Dignum, F. 2003: Role Assignment in Open Agent Societies. In: *Proc. of AAMAS'03, 2nd International Joint Conference in Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia.

[8] Davenport, T. and Prusak L. 1998: Working Knowledge: How organizations manage what they know, Harvard Business School Press.

[9] Dignum, V. van Eeden, P. 2003: Seducing, Engaging and Supporting Communities at Achmea, In: *Proceedings of the 4th European Conference on Knowledge Management*, Oxford, UK, September 18-19.

[10] Dignum, V., Dignum, F. 2003: The Knowledge Market: Agent Mediated Knowledge Sharing. Marik, V., Muller, J., Pechoucek, M. (Eds.): Proceedings of CEEMAS'03, Prague, June 16 – 18. LNAI, Springer-Verlag.

[11] Dignum, V., Meyer, J.-J., Dignum, F., Weigand, H. 2003: Formal Specification of Interaction in Agent Societies. In: Hinchey, M., Rash, J., Truszkowski, W., Rouff, C., Gordon-Spears, D., (Eds.): *Formal Approaches to Agent-Based Systems* (FAABS'02). LNAI 2699, Springer-Verlag, 2003.

[12] Dignum, V., Meyer, J-J., Weigand, H., Dignum, F. 2002: An Organizational-oriented Model for Agent Societies. In: Proc. Int. Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA'02), at AAMAS, Bologna, Italy.

[13] Dignum, V., Weigand, H. 2002: Towards an Organization-Oriented Design Methodology for Agent Societies. In: V. Plekhanova (Ed.): *Intelligent Agent Software Engineering*. Idea Publishers.

[14] Dignum, V. 2002: A Knowledge Sharing Model for Peer Collaboration in the Non-Life Insurance Domain. In: Proc. German Workshop on Experience Management, Lecture Notes in Informatics, German Society for Informatics, Berlin.

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

[16] Domingue, J. B. and Motta, E. 1999: A Knowledge-Based News Server Supporting Ontology-Driven Story Enrichment and Knowledge Retrieval. URL: http://kmi.open.ac.uk/people/motta/papers/planet_onto.pdf. Proc. EKAW'99.

[17] Eppler, M. 2002: Glossary definition: Knowledge management. NetAcademy: www.knowledgemedia.org/modules/glossary/view def.php?id=mediamanagement-20.

[18] Esteva, M., Rodriguez, J., Sierra, C., Garcia, P., Arcos J.: On the formal specifications of electronic institutions, In: Dignum F., Sierra C. (Eds.): *Agent-mediated Electronic commerce (The European AgentLink Perspective)*, LNAI 1991, Springer, 2001, pp. 126-147.

[19] Ferber, J., Gutknecht, O. 1998: A meta-model for the analysis and design of organizations in multi-agent systems. Proc. of ICMAS'98, IEEE Press.

[20] Gandon, F., Dieng, R., Corby, O. & Giboin, A. 2000: A Multi-Agent System to Support Exploiting an XML-based Corporate Memory. Proceedings PAKM'00, Basel.

[21] Gongla, P., Rizzuto, C. 2001: Evolving Communities of Practice: IBM Global Services Experience. *IBM Systems Journal* 40(4).

[22] Hindriks, K., de Boer, F., van der Hoek, W., Meyer, J-J.: Agent Programming in 3APL. *Journal of Autonomous Agents and Multi-Agent Systems* **2**(4), Kluwer Academic Press, 1999, pp. 357-401.

[23] Khalil, A.: *Multi-Agent System for Achmea*. MSc. Thesis, Utrecht University, 2002

[24] Luck, M., McBurney, P., Preist, C. 2003: Agent Technology: Enabling Next Generation Computing: A Roadmap for Agent Based Computing. AgentLink II.

[25] Mentzas, G., Apostolou, D., Young, R., Abecker, A. 2001: Knowledge networking: a holistic solution for leveraging corporate knowledge. Journal of KM, MCB, 5(1).

[26] O'Leary, D. 1998: Knowledge Management Systems: Converting and Connecting. IEEE Intelligent Systems, May-June, pp. 30-33.

[27] Preece, J.: *Online Communities: Designing Usability, Supporting Sociability.* J. Wiley & Sons, 2000.

[28] Pumareja, D., Bondarouk, T., Sikkel, K. 2003: Supporting Knowledge Sharing Isn't Easy - Lessons Learnt from a Case Study. *Information Resource Management Association International Conference (IRMA'03)*, Philadelphia.

[29] Smith, I., Cohen, P., Bradshaw, J., Greaves, M., Holmback, H.: Designing Conversation Policies Using Joint Intention Theory. Proceedings of the Third International Conference on Multi Agent Systems (ICMAS-98), Paris, IEEE Press, 1998, pp. 269-276.

[30] Smith, R, Farquhar, A. 2000: The Road Ahead for Knowledge Management: An AI Perspective, AI Magazine, AAAI, 21(4):17-40.

[31] van Elst, L., Dignum, V., Abecker, A. 2003: Towards Agent-Mediated Knowledge Management. In: L. van Elst, V. Dignum, A. Abecker (Eds.): *Agent-Mediated Knowledge Management: Selected Papers*, LNAI 2926, Springer-Verlag,.

[32] Wenger, E., McDermott, R., Snyder, W. 2002: *Cultivating Communities of Practice*. Harvard Business School Press.

[33] Wooldridge, M., Jennings, N., Kinny, D. 2000: The Gaia Methodology for Agent-Orient Analysis and Design. Autonomous Agents and Multi-Agent Systems, **3**(3).

[34] Zambonelli F., Jennings, N., Wooldridge, M. 2001: Organizational Abstractions for the Analysis and Design of Multi-Agent Systems. In: Ciancarini P., Wooldridge, M. (eds.): Agent-Oriented Software Engineering, LNCS 1957, Springer-Verlag.