Agent Societies: towards frameworks-based design

Virginia Dignum
Achmea & University Nyenrode
The Netherlands
Phone: +31-(0)30-2534432
Email: virginia.dignum@achmea.nl

Hans Weigand
Infolab, Tilburg University
The Netherlands
Phone: +31-(0)13-4662806
Email: weigand@kub.nl

Lai Xu
Infolab, Tilburg University
The Netherlands
Phone: +31-(0)13-4668716
Email: l.xu@kub.nl

ABSTRACT

We present a global methodology for the design of agent societies that takes the organisational perspective as starting point and specifies the development steps for the design and development of an agent-based system for a particular domain. So far, most agent-oriented design methodologies haven't considered the influence of the social organisation model on the functionality and objectives of the agent society. Our approach is to provide a generic frame that directly relates to the organisational perception of the problem. Existing methodologies can be used for the development and modelling of each step.

1. INTRODUCTION

The application of the agent paradigm to the development of different application calls for a development methodology that focuses not only on the internal organisation of each of the intervening agents but also on the social aspects of the domain [19]. Generic SE methodologies are not tailored to deal with design aspects specific to agent systems, such as the capture of flexible and autonomous behaviour of agents and the complexity of agent interactions and social organisation of the system [27].

Currently, research on agent specific SE is a topic of great interest [26, 12]. A good overview and analysis of existing methodologies, applicable to Agent-Oriented Information Systems can be found in [1]. However, as yet, there is no well-established and all-encompassing agent-oriented methodology. Most existing methodologies concentrate in just one part of the total picture or are too formal to be applicable in practice. Furthermore, most approaches start from the moment that the decision to use the agent paradigm has been made, and do not guide this choice. A methodology for designing multi-agent systems must be both specific enough to allow engineers to design the system and generic enough to allow the acceptance and implementation of multi-agent systems within an organisation, allowing for the involvement of users, managers and project teams.

In order to make agent technology widely accepted and used in industry it is necessary to clearly specify the type of problems suitable for an agent approach and the benefits of agents above other technologies. An aspect that in our view will contribute to the acceptance and understanding of agents societies in organisations, is that the agent paradigm provides a natural way to view and characterise intelligent systems [24]. Intelligence and interaction are deeply and inevitably coupled, and multi-agent systems reflect this insight. Multi-agent systems can provide insights and understanding about poorly understood interactions between natural, intelligent beings as they organise themselves into groups, societies and economies in order to achieve improvement. Other important contributions to the success and acceptance of agent technology are the development of robust agent tools and of standard methodologies such as FIPA or the AUML effort.

Our aim is to develop a practical methodology that describes all the steps of development of a multi-agent system. The methodology takes the organisational perspective as starting point and specifies the development steps for the design and development of an agent-based system for a particular domain. Once these steps have been identified, existing, specific agent-based methodologies can be used for the development and modelling of each step. We believe that such a generic framework, based on the organisational view, will contribute to the acceptance of multi-agent technology by organisations. Furthermore the methodology proposed gives an answer to the development challenges posed by Sycara [22]:

1. How to engineer practical multi-agent systems.
2. How to decompose problems and allocate tasks to individual agents
3. How to coordinate agent control and communication
4. How to make multiple agents to act in a coherent manner
5. How to make each agent reason about the other agents and the state of coordination
6. How to reconcile conflicting goals between coordinating agents

Based on the organisational coordination model, we define a social framework for agent communities that 'implements' the generic interaction, cooperation and communication mechanisms (2 and 3) that occur in the problem domain. The proposed methodology (1) allows to tailor this generic coordination model to the specific application and to determine the specific agent roles and interactions (5). In the following steps the level of design detail will be successively increased to include the internal organisation and reasoning capabilities of the agents (4). The reconciliation of conflicting goals (6) is one aspect of interaction. Depending on the coordination model, different reconciliation processes need to be specified.

2. COORDINATION MODELS

Software agents are 'advertised' as autonomous entities with reasoning and communicative capabilities, and therefore utmost suitable to implement, simulate or represent real-life entities presenting the same autonomy. Agent societies[1] represent the interactions between agents and are as such the virtual counterpart of real-life societies and organisations. Because of the proactive and autonomous behaviour of agents it is natural to design agent societies mimicking the behaviour and structure of human organisations [28]. Agents model a specific role in the system and interact with others as a means to accomplish their role. This perspective makes the design of the system less complex since it reduces the conceptual distance between the system and the real-world application it has to model.

Coordination is one of the cornerstones of agent societies and is considered an important problem inherent to the design and implementation of MAS [3]. However, the implications of the coordination model for the agent society architecture and design method have usually not been considered. So far, research about coordination in MAS has been limited to the study of 'technical' aspects of coordination, such as control and planning. Little attention has been paid to the organisational aspects of coordination in agent societies.

The emerging research area of **coordination theory** is an interdisciplinary field which, besides multi-agent systems, uses ideas from disciplines such as computer science, management and economics, organisation theory, operations research, cognitive engineering and many others [15]. In our research we are interested in the influence of organisation science research in coordination theory and how this can be applied to the design of agent societies. In our framework, the architecture of the agent society considers and reflects the implications of the coordination model of the real-life organisation being modelled.

2.1 Organisational behaviour

Organisational science and economics have since long researched coordination and organisational structures. Drawing on disciplines such as sociology and psychology, research in organisation theory focuses on how people co-ordinate their activities in formal organisations.

An organisation can be defined as a specific solution created by more or less autonomous actors to achieve common goals. In order to achieve their objectives, organisations depend on a layer of facilitation, which provides the social organisation of the environment, that is, the maintenance of the organisation itself**.** Social interaction emerges from a set of negotiated social norms and is regulated by mechanisms of social control. The way that different societies organise and balance their objectives and their facilitation activities is reflected in their coordination model.

Relationships between and within organisations are developed for the exchange of goods, resources, information and so on. Williamson [25] argues that the transaction costs are determinant for the organisational model. Transaction costs will rise when the unpredictability and uncertainty of events increases, and/or when transactions require very specific investments, and/or when the risk of opportunistic behaviour of partners is high. When transaction costs are high, societies tend to choose a hierarchical model in order to control the transaction process. If transaction costs are low, that is, are straightforward, non- repetitive and require no transaction-specific investments, then the market is the optimal choice. Powell [21] introduces networks as another possible coordination model. Networks stress the interdependence between different organisational actors and pay a lot of attention to the development and maintenance of (communicative) relationships, and the definition of rules and norms of conduct within the network. At the same time, each actor is independent, have its own interests, and can be allied to different networks.

Coordination in markets is achieved mainly through a price mechanism in which independent actors are searching for the best bargain. Hierarchies are mainly coordinated by supervision, that is, actors that are involved in power-dependent relationships act according to routines. Finally in networks it is characterised by mutual interest and interdependency. Figure 1 gives an overview of this classification.

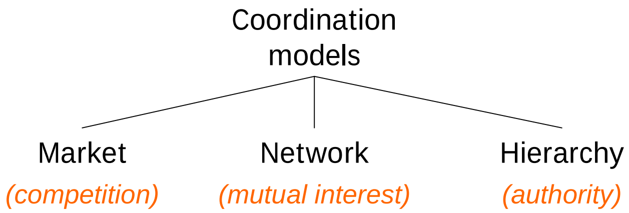

Figure 1 - Classification of coordination models

2.2 Agent societies

The characteristics of the different coordination models used in organisational behaviour theory can be applied to the design of agent societies. In agent societies coordination models describe the way interactions between agents are organised and the way the interface between the society and the 'outside world' is defined. That is, coordination models provide a framework to express interaction between agent activities and social behaviour of the system [7]. We use the organisational coordination models described above to classify agent societies, as illustrated in Figure 2.

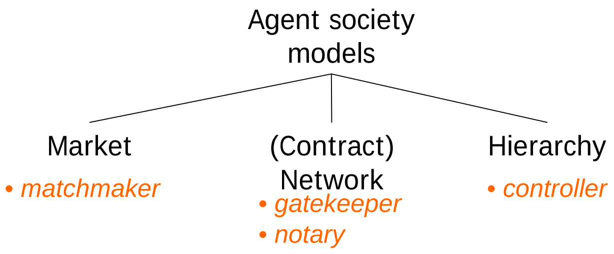

Figure 2 - Classification of agent societies

Our aim is to provide generic facilitation and interaction frameworks for agent societies that implement the functionality derived from the coordination model applicable to the problem domain. The type of coordination will determine the interaction

[1] In order to stress the social aspects of agent interaction we use the term **agent society** instead of **multi-agent system**.

patterns and functionality of the facilitation level of the agent society.

The way organisational coordination models achieve coordination is determinant to the motivation of coordination in agent societies. In **markets**, agents are self-interested (determine and follow their own goals) and value their freedom of association and own judgement above security and trust issues. Openness is thus per definition a feature of markets. Facilitation is, in the most extreme case, limited to identification and matchmaking activities. Interaction in markets occurs through communication and negotiation.

Network organisations are built around general patterns of interaction or contracts. Relationships are dependent on clear communication patterns and social norms. Agents in a network society are still self interested but are willing to trade some of their freedom to obtain secure relations and trust. Coordination is achieved by mutual interest, possibly using trusted third parties, and according to well-defined rules and sanctions.

Finally, in a **hierarchy** interaction lines are well defined and the facilitation level assumes the function of global control of the society and coordination of interaction with the outside world. In a hierarchy, agents are usually not self interested, and all contribute to a common global goal. Coordination is achieved through command and control lines.

2.3 Coordination of activities

In Computer Science coordination is defined as the art of managing interactions and dependencies among activities. Coordination languages are a class of programming notations that offer a solution to the problem of specifying and managing the interactions among computing agents. From this point of view coordination models are divided into two classes: control-driven and data-driven [20]. Control-driven models are systems made up of a well-defined number of entities and functions, in which the flow of control and the dependencies between entities need to be regulated. The data-driven model is more suited for open societies where the number of entities and functions is not known a priori and cooperation is an important issue. Where the classification of co-operation provided by organisational theory stems from social considerations and transaction costs, this classification is concerned with the way interaction between agents happens. This classification can be seen as an extra dimension of interaction, and can be applied to all agent society models.

In the following chapter we will describe in more detail the social frameworks for agent societies that will implement these coordination models.

3. AGENT SOCIETY FRAMEWORKS

Different application contexts exhibit different needs with respect to coordination, and the choice of a coordination model will have great impact in the design of the agent society. Following this observation we argue that the first step in the development of an agent society is to identify its underlying coordination model.

So far, most agent-oriented design methodologies haven't considered the influence of the social organisation model on the functionality and objectives of the agent society. In many cases the social organisation is left implicit in the design of the agent society. However, the organisation model determines important autonomous activities, which must be explicitly organised into autonomous entities in the conceptual model of the agent society.

Different coordination models result in different frameworks for agent societies. The overall goals of a society are domain dependent but all societies depend on a facilitation layer that provides the social backbone of the organisation. Facilitation activities deal with the functioning of the society itself and are related to the underlying coordination model.

Once the social coordination model of a specific problem domain has been established, the corresponding agent society framework can be applied. In the next stage, this framework is extended with domain specific roles and interaction forms that characterise the problem. We can compare this process to the design a generic enterprise model including roles as accountants, secretaries and managers, as well as their job descriptions and relationships, and then extending it with a 'recipe' to build the functions necessary to achieve the objectives of the given enterprise. These are, for example, designers and carpenters if the firm is going to manufacture chairs, and programmers and system analysts if the enterprise is a software house. While the chosen coordination model determines the social part, domain roles are directly derived from the domain requirements.

3.1 Market model

The market metaphor has been used to describe agent interaction, enhancing the adaptation, robustness, and flexibility of multi-agent systems. In a market model, agents representing (or providing) services and/or competencies compete to perform tasks leading to the satisfaction of their own individual objectives as well as to a possible overall system's goal [18].

The main goal of a market is to facilitate exchange between agents. In a market heterogeneous agents will strive to find partners or clients with whom to trade their services. Being open systems, market architectures assume the heterogeneity of its members, both in structure, goals and ways of acting. The degree of the organisation of the facilitation aspects of the society will determine the level of freedom of agents in markets.

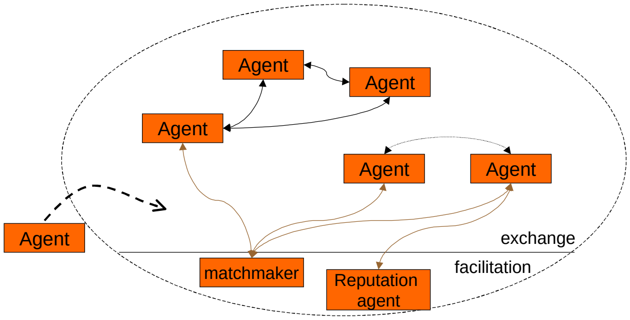

Figure 3 - Free market architecture

In a free market framework, social interaction is open and partners in a transaction are free to make their own agreements and contract rules. Markets are particularly suitable to situations in which resources are overlapping and agents need to compete for them. For example, the market architecture is a good selection to model product or service allocation problems. Being self-interested, agents will first try to solve their own local problem, and then agents can potentially negotiate with other agents to exchange services or goods in shortage or in excess.

The facilitation activities of such agent society are mainly limited to matchmaking, that is, help agents find suitable partners. **Matchmakers** keep track of agents in the system, their needs and possibilities and mediate in the matching of demand and supply of services. Market societies also provide **reputation** facilities to build confidence for customers as well as offering guarantees to its members. Furthermore, it is necessary to define ways to value the goods to be exchanged and determine profit and fairness of exchanges. This is accomplished by some **banking** facilities and currency specification.

3.2 Networks

Networks are coalitions of self-interested agents that agree to collaborate to achieve a mutual goal. Agents in a network society are still self interested but are willing to trade some of their freedom to obtain secure relations and trust. Coordination is achieved by mutual interest, possibly using trusted third parties, and according to well-defined rules and sanctions. These coalitions have been studied in the area of game theory and Distributed Artificial Intelligence (DAI) [23].

Networks provide an explicit shared context, describing rules and social norms for interaction and collaboration. As in any market, the aim of agents when entering the society is to trade their knowledge, goods or services. An agent society based on the model must be able of describing its rules of interaction, regulations, facilities and legal guarantees to applying members.

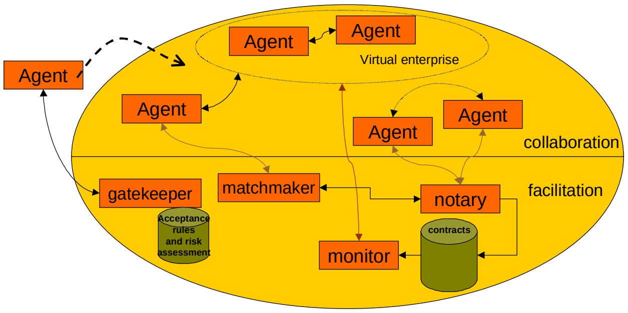

Figure 4 – Contract network architecture

Dellarocas introduces the concept of Contractual Agent Societies (CAS) as a model for developing agent societies [8]. CAS has been inspired by work in the areas of organisational theory, economy and interaction sociology, which model organisations and social systems after contracts. Social contracts govern the interaction of a member with the society. Furthermore, the society is responsible to enforce the contracts formed by its members and punish potential violators (for example, through loss of reputation or eventually banishment). New agents are admitted through a process or socialisation during which the agent negotiates with the society the terms of its membership. As a result the terms of the social contracts of existing members may need to be renegotiated as well.

Facilitation level agents monitor and help form contracts, take care of introducing (teaching) new agents to the rules of the market, keep track of the reputation of agents. Furthermore, they keep and enforce the 'norms' of the agent community and ensure interaction. Besides matchmakers as in market frameworks, other types of facilitation level agents in networks are gatekeepers, notaries and monitoring agents. **Gatekeepers** are responsible for accepting and introducing new agents to the market. Agents entering the marketplace must be informed about the possibilities and capabilities of the market. Gatekeepers negotiate the terms of a social contract between the applicant and the members of the market. **Notaries** keep track of collaboration contracts between agents. **Monitoring agents** are trusted third parties. The marketplace must provide monitoring agents to interested parties. When a contract appoints a (set of) monitoring agents, this is the equivalent to the setting up of a (super) contract between the contracting agents and the environment (here personified by the monitoring agents). This super-contract (which can also be described using the contract language) specifies that the monitoring agents are allowed to check the contracting agents actions (ex. look at agent states) and that the contracting agents must submit to the sanctions imposed.

3.3 Hierarchies

Hierarchies coordinate the flow of resources or information through adjacent steps by controlling and directing it at a higher level in the managerial hierarchy. Managerial decisions, and not negotiation and communication as in markets, determine interaction and design. Demand parties do not select a supplier from a group of potential suppliers: they simply work with a predetermined one. *Vertical integration* occurs where the hierarchy is a single firm. In other cases the hierarchy may span several separate firms in a close, perhaps electronically mediated relationship (*vertical coordination*). In vertical coordination hierarchies one can choose to integrate the hierarchies of each of the organisations involved or to model one organisation as an agent participating in the hierarchy modelling the coordination.

In hierarchical systems, an agent reigns over an arbitrarily and usually statically defined sub-hierarchy, in many cases an administrative domain of some kind. For instance, a university could be managed as follows: an agent is in charge in each lab, whereas other agents each oversee a department and a single one rules the university. These domains do not reflect the easy routing parts of the network and do not evolve.

Environments such as automated manufacturing planning and control are also well suited to the hierarchical model. In such systems, reliable control of resources and information flow requires central entities that manage local resources and data but also need quick access to global ones. Hierarchical models of agents have been used for information agents [6] and for management of communication networks [10]. In a hierarchical model of information systems, each information agent is responsible for providing information about a specific domain. Information agents further down the hierarchy provide more specialised information about a domain. In response to a query, an information agent may cooperate with information agents in other domains or sub-domains, in order to generate a response. Communication network solutions are based on a hierarchy of autonomous intelligent agents, which have local decision making capabilities, but cooperate to resolve conflicts. Higher level agents arbitrate unresolvable disputes between peer agents.

In a hierarchic agent model agents at facilitation level are mainly dedicated to the overall control and optimisation of the system activities. Some times, these facilitation activities are concentrated in one agent, typically the 'root' agent of the hierarchy.

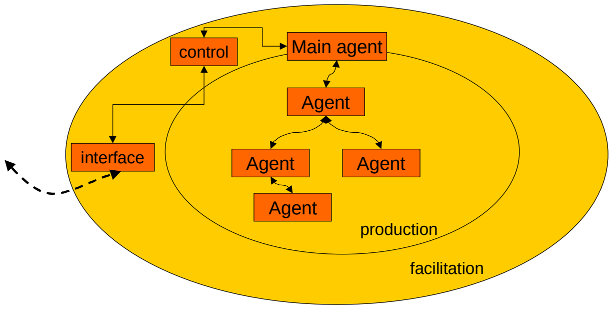

Figure 5 - Hierarchy architecture

Controllers will monitor and orient the overall performance of the system. Autonomous agents have local perspective and their actions are determined by its local state. Therefore, in a hierarchical architecture, is necessary to have an agent which role is to control the overall performance of the system. **Interface agents** in a hierarchical model are responsible for the communication between the system and the 'outside world'. In this architecture communication lines between agents are predefined. Furthermore, agents are usually not free to enter or leave the system. Therefore communication with the outside must be regulated at the facilitation level.

4. DEVELOPMENT METHODOLOGY

In this section we introduce a methodology for the development of agent societies based on the coordination frameworks described above. After accessing the applicability of the agent paradigm to the problem on hand, the methodology consists of the following levels, which provide a growing level of refinement of the resulting system into more structured and precise forms:

- **Coordination**: the coordination structure of the domain is determined and a model is designed based on the collection of coordination models available in the library.
- **Environment**: based on the coordination model design in the previous step, this level describes the interaction between the society and its environment in terms of global requirements and domain ontology.
- **Behaviour**: based on the models above, in this level the intended behaviour of the society is described in terms of agent roles and interaction patterns. These process is supported by a library of roles and interaction patterns
- **Agent**: finally, the internal structure of agents is described in terms of requirements for communication, action, interface and reasoning behaviour.

We intend to develop libraries of models, patterns, behaviours and components to inform this process. The methodological process is described in Figure 6 and will be described in more detail in the following sections.

These steps result in a complete model for the agent society. Furthermore, a verification process is needed to demonstrate that the designed system actually will show the required behaviour. In the very least verification must demonstrate that system objectives and requirements are met by the design.

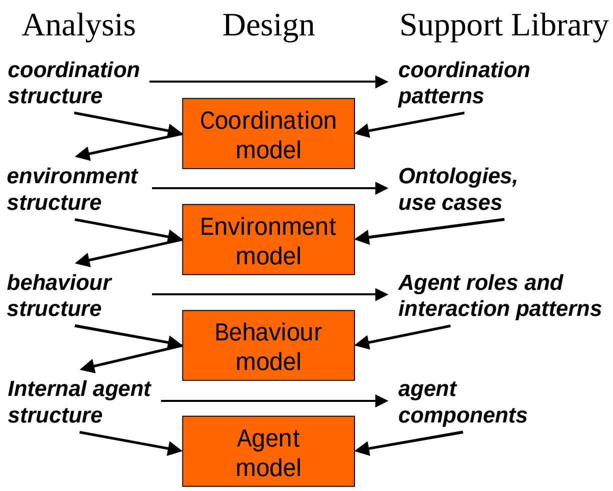

Figure 6 – Development methodology for agent societies

4.1 Determination of agent appropriateness

The aim of this pre-step to the methodology is to motivate the choice of an agent-based approach to the organisation involved in the deployment of the system. It is commonly accepted that the agent paradigm is appropriate for reactive systems. These systems maintain an ongoing interaction with some environment and are inherently quite difficult to design and correctly implement. Moreover, agents scan be seen as 'active objects', that is, objects with goals and communicative and deliberative capabilities. Therefore, agents are applicable in areas were the object paradigm has limitations, especially where communication is involved [11]. However, more research is needed in this area, in order to be able to define methods, libraries and heuristics supporting and informing the application of the agent paradigm to arbitrary problems.

4.2 Coordination level

In this stage a decision is achieved on which coordination model applies to the problem. The table below gives an overview of the specific characteristics of each coordination model, which can be used to determine the applicable model for the domain.

	Market	**Network**	**Hierarchy**
Society goal	Exchange	Collaboration	Production
Agent autonomy	High	Medium	Low
Agent goals	Individual	Both are possible	Global
Relation forms	Negotiation (e.g. Contract Net Protocol)	Negotiable within society norms and rules	Fixed (e.g. Action / Workflow loop)
Communication capabilities of agents	Interaction based on standards; communication concerns exchange only	Both the interaction procedures and exchange can be negotiated	Specified on design
Interface to outside world	Usually open for agents (after identification)	Admittance procedure for agents	Closed for agents; open for data (input and output)

The identification of the appropriate coordination model will point out the type of social laws and norms of conduct and inform the organisation on the interaction patterns and facilitation needs of the agent society, as described in section 3.

So far, agent-based methodologies that consider both the social and the agent levels of analysis and design have been specified with a specific type of society in mind. For example, the Gaia methodology is intended to support the development of societies of agents, which constituents are known at design time and in which all agents are supposed to cooperate towards a common goal, and is thus not suitable to the development of market-like societies [26]. The Coordination Level of our methodology guides the choice of a coordination model and informs the further design process.

4.3 Environment level

At the environment level we describe the external behaviour of the society, that is, the interaction between society and its environment or external world. This includes the identification of the functionality of the society and the specification of the domain ontology.

Organisations and their environments are interdependent, and each can influence the other. The design of agent societies depends in many ways on the environment. In characterising the environment, one has to decide what level of analysis is appropriate. The behaviour of an agent society with respect to its environment can be defined in terms of the expected functionality of the society, that is, what is the society expected to do or produce. This can be expressed in terms of overall requirements or in terms of scenarios or use cases. Requirement Engineering is an important topic of research and several methods have been developed. However these methods are not always applicable to the specific needs of agent systems development. The Temporal Requirement Language, TRL, a language to specify behavioural requirements and scenarios has been developed specifically for agents [13].

Another issue in this level is to identify the concepts and relationships relevant in the domain. Because each agent can have its own ontology, or "view of the world", a common ontology is used to bridge the different ontologies of agents. Depending on the domain and coordination model, the shared ontology can be the merge of the ontologies of each agent or it can concentrate on the concepts relevant for the interaction between agents. The merging approach is not scalable and is costly. Furthermore, the process must be repeated when new agents enter the society, and may not even be possible due to unresolvable inconsistencies between ontologies. In the distributed approach, only articulations between ontologies that serve specific application objectives need to be generated [16].

4.4 Behaviour level

The previous development steps have resulted in a coordination framework (Coordination Level) and the description of the intended behaviour of the agent society in terms of requirements or use cases (Environment Level). The purpose of the Behaviour Level is to populate the framework with the agent roles and interaction patterns needed to achieve the requirements.

At this level, whether or not an agent is intelligent, or how agents are internally structured is not important. We are concerned with only the high-level definition of the types of agents and their goals. The social rules and norms of conduct associated with each coordination framework will determine the interaction patterns between agents in the society. To support the identification and specification of agent roles, role catalogues have been developed (for instance at BT) [14]. These catalogues provide commonly occurring role models, which include many of the facilitation roles we have described in section 3. Interaction patterns can be modelled using for example Petri Nets or AUML interaction protocols [2].

A methodology that accounts for the analysis and design of social behaviour of agent societies is SODA [19]. SODA views coordination as an interaction issue as described in section 2.3 and does not consider the organisational aspects of coordination. Our approach provides a structural way to specify the analysis and design of agent roles and interaction, specifying the coordination framework, and consequently the facilitation structure. This will inform the choice of a specific design methodology for the specification of the behaviour of the agent system. In addition, we plan to develop a library of generic models for different agent roles and interaction patterns.

4.5 Internal Agent level

At Agent Level the functionality of each agent role is specified in terms of specific requirements for interaction, communication, reasoning, goals behaviour and interface.

Depending on the application, agent societies can be open systems where foreign agents (which design cannot be determined a priori) can participate. Therefore, it is not possible to enforce a particular design for the internal structure of agents. However, agents living in a society must, besides the capability of performing its own task, be able to interact. This imposes some conditions to the internal model of the agent which are described in this level.

Furthermore, instead of designing each new agent from scratch a generic agent model can be used. This model can be adapted and filled up with components necessary to achieve the agent’s functionality. GAM (Generic Agent Model) gives a formal design of a compositional agent model that can be reused as a template of pattern for a variety of agent types and application domains [5]. MaSE [9] and DESIRE [4] are other examples of methodologies for agent design.

5. CONCLUSIONS

We have presented a global methodology for the design of agent societies. The methodology takes the organisational perspective as starting point and specifies the development steps for the design and development of an agent-based system for a particular domain. Although there are several agent-based software engineering methodologies (for example, [19, 27, 4, 9]) these are often either too specific or too formal and not easily used and accepted. Our approach is to provide a generic frame that directly relates to the organisational perception of a problem. If needed, existing methodologies can be used for the development, modelling and formalisation of each step. We believe that this approach will contribute to the acceptance of multi-agent technology by organisations. One contribution of our research is that is describes the implications of the coordination model of the organisation for the architecture and design method of the agent society being developed.

We are currently applying this methodology to the development of a Knowledge Management Environment. This will serve as well as a test case to the methodology. Our research will further continue in two directions. On one hand, we intend to further specify each step of the methodology and to develop libraries of conceptual interaction patterns, organisational and facilitation roles and agent components. These libraries will improve and facilitate the design agent societies. We also plan to look at the compatibility and integration of our ideas with current standardisation efforts for agent development (for example the Agent UML effort [17]). On the other hand, we plan to develop tailored frameworks for specific applications domains such as Workflow Management and e-business.

6. REFERENCES

[1] O. Azary and C. Woo, "Analysis and design of Agent-Oriented Information Systems (AOIS)", Working paper 99-MIS-004, Fac. Commerce & Business Administration, Univ. of British Columbia, Vancouver, Canada, 2000.

[2] B. Bauer, J. Mueller and J. Odell, "Agent UML: A Formalism for Specifying Multiagent Software Systems", *Agent-Oriented Software Engineering*, P. Ciancarini and M. Wooldridge (Eds.), LNCS 1957, Springer, 2001, pp, 91-104

[3] A. Bond and L. Gasser, Readings in Distributed Artificial Intelligence, Morgan Kaufmann, 1988.

[4] F. Brazier, B. Dunin-Keplicz, N. Jennings and J. Treur, "DESIRE: Modelling Multi-Agent Systems in a Compositional Formal Framework", *International Journal of Cooperative Information Systems*, M. Huhns and M. Singh (Eds.), 6(1), 1997

[5] F. Brazier, C. Jonker, and J. Treur, "Compositional Design and Reuse of a Generic Agent Model", *Applied Artificial Intelligence Journal*, vol. 14, 2000, pp. 491-538.

[6] A. Castillo, M. Kawaguchi, N. Paciorek and D. Wong, "Concordia™ as Enabling Technology for Cooperative Information Gathering ", Japanese Society for Artificial Intelligence Conference, Tokyo, Japan, 1998

[7] P. Ciancarini, A. Omicini and F. Zambonelli "Coordination Models for Multi-Agent Systems", *AgentLink News*, No. 3, July 1999

[8] C. Dellarocas, "Contractual Agent Societies: Negotiated shared context and social control in open multi-agent systems", Workshop on Norms and Institutions in Multi-Agent Systems, Agents-2000, Barcelona, 2000.

[9] S. DeLoach, "Multiagent Systems Engineering: A Methodology and Language for Designing Agent Systems", *Proc. of Workshop on Agent-Oriented Information Systems (AOIS'99)*, 1999

[10] C. Frei and B. Faltings, "A Dynamic Hierarchy of Intelligent Agents for Network Management", 2nd International Workshop on Intelligent Agents for Telecommunications Applications, IATA'98, Paris, France, 1998, pp. 1-16.

[11] N. Jennings and M. Wooldridge (eds.), *Agent Technology: Foundations, Applications and Markets*, Springer, 1998.

[12] N. Jennings, "On agent-based software engineering", *Artificial Intelligence*, 117, 2000, pp. 277-296.

[13] C. Jonker, M. Klusch, J. Treur, "Design of Collaborative Information Agents", *Cooperative Information Agents IV*, M. Klusch and L. Kerschberg (eds.), LNAI 1860, Springer, 2000.

[14] E. Kendall, "Agent Software Engineering with Role Modeling", *Agent-Oriented Software Engineering*, P. Ciancarini and M. Wooldridge (Eds.), LNCS 1957, Springer, 2001, pp. 163-170.

[15] Malone, T.W. and Crowston, K. "The interdisciplinary study of coordination", *ACM Computing Survey*. 26(1), 1994, pp. 87-120.

[16] P. Mitra, G. Wiederhold and J. Jannink: " Semi-automatic Integration of Knowledge Sources"; *Proceedings of Fusion '99*, Sunnyvale CA, July 1999

[17] J. Odell, H. Van Dyke Parunak and B. Bauer, "Extending UML for Agents", AOIS Workshop at AAAI 2000.

[18] E. Oliveira, "Applications of Agent-based Intelligent Systems", *Proceedings of 4th SBAI- Brazilian Symposium of Intelligent Automation*, S.Paulo, Brazil, 1999.

[19] A. Omicini, "SODA: Societies and Infrastructures in the Analysis and Design of Agent-based Systems", *Agent-Oriented Software Engineering*. P. Ciancarini and M. Wooldridge (Eds.) LNCS 1957, Springer, 2001.

[20] G. A. Papadopoulos and F. Arbab, "Coordination Models and Languages", *Advances in Computers*, Academic Press, 1998, pp. 329-400.

[21] W. Powell, "Neither market nor hierarchy: Network forms of organisation" *Research in Organisational Behaviour*, vol. 12, 1990, pp. 295-336.

[22] K. Sycara, "Multiagent Systems", *AI Magazine*, 19(2), 1998, pp. 79-92

[23] M. Tsvetovat, K. Sycara, Y. Chen and J.Ying, "Customer Coalition in Electronic Markets", Agent-Mediated Electronic Commerce III, F. Dignum and U. Cortés (Eds.), LNAI 2003, 2001, pp.121-138

[24] G. Weiss, *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.

[25] O. Williamson, *Markets and hierarchies: Analysis and antitrust implications*, Free Press, New York, 1975

[26] M. Wooldridge, "Agent-Based Software Engineering", *IEEE Proc. Software Engineering*, 144(1), 1997, pp. 26-37.

[27] M. Wooldridge, N. Jennings and D. Kinny, "The Gaia Methodology for Agent-Orient Analysis and Design", *Autonomous Agents and Multi-Agent Systems*, 3(3), 2000, pp. 285-312.

[28] F. Zambonelli, N. Jennings, A. Omicini and M. Wooldridge, "Agent-Oriented Software Engineering for Internet Applications", *Coordination of Internet Agents: Models, Technologies, and Applications*, A. Omicini et al. (Eds.) Springer, March 2001