

Modeling Corporate Knowledge within the Agent Oriented Abstraction

Pierre Maret and Jacques Calmet*

INSA de Lyon, France and University of Karlsruhe (TH), Germany
E-mail: pierre.maret@insa-lyon.fr, calmet@ira.uka.de

Abstract

An enterprise can be considered in a distributed computational paradigm. Multiagent systems have been invented to tackle such problems within artificial intelligence but are today seen also as a management methodology. The Agent Oriented Abstraction has been proposed to describe in a fully generic way a society of agents. In this paper, we are investigating the application of this model to the abstract modeling of corporate knowledge. We show that such an abstraction mechanism leads to very practical applications whether on the web or on any other support and that it covers the broad scope of corporate knowledge. This approach can be further used to better simulate and support knowledge management processes.

Key Words: Corporate knowledge, abstract modelling, agent society, agent oriented abstraction, knowledge management

1. Introduction

The concept of corporate knowledge has at least two facets. A first one lies fully in business and refers to the amount and quality of know-how and information that is available throughout a company. The second one is set in the framework of information technology and is tightly linked to the various and diverse paradigms that are used to represent knowledge. Nowadays, with the common use of e-business these two concepts are becoming somewhat interrelated. Adopting the IT point of view leads to think of an enterprise as a distributed computational paradigm. Multiagent systems have been invented to tackle such problems within artificial intelligence but are today seen also as a management methodology. It is thus not a new idea to model a company or enterprise as a multi-agent system. It is however a new idea to investigate what abstraction mechanisms for agent methodologies can add to such a modeling.

*Work partially supported by the Calculemus Research Training Network HPRN_CT-2000-00102; The work of Pierre Maret is partially sponsored by the Région Rhône-Alpes, France.

In this paper we intend to investigate the application of the Agent Oriented Abstraction model [2] to the abstract modeling of corporate knowledge. Agent Oriented Abstraction has been proposed to describe in a fully generic way the concept of agents. We show that such an abstraction mechanism leads to very practical applications for corporate knowledge, whether on the web or on any other support. The paper is organized as follows: section 2 gives an overview on existing approaches to corporate knowledge. Section 3 is a short presentation of the Agent Oriented Abstraction. Section 4 consists of the description of the abstract modeling of corporate knowledge within the agent oriented abstraction model. Section 5 is a case study description that serves to illustrate our approach. Our conclusions are presented in section 6.

2. Approaches to corporate knowledge

Corporate knowledge generally covers several domains such as document/knowledge management, discussion forums, skill management and knowledge engineering. The main paradigms to represent knowledge are logic, frames or semantic nets. Inference capabilities are closely associated to the paradigms of knowledge representation. The document/knowledge management domain is linked to document repositories (including Internet), indexing, search engines, information filtering, knowledge extraction through document mining just to name a few related fields. Thesaurus and ontology are also widely used. Documents can be distributed, web-based, structured, semi-structured or unstructured. Discussion forums take advantage of communication facilities within a community of individuals. The main concepts in this domain are topics, participants and messages. Recommender systems also known as collaborative filters [11] do belong to this domain. Skill management consists of identifying, recording and updating individuals competences in order to use them appropriately and to identify lacks of competences [7]. Knowledge engineering concentrates on capturing, structuring, inferring and distributing knowledge. It is linked to specific knowledge engineering techniques and to document repositories. Knowledge

can be recorded manually or extracted automatically, for instance from a company's documents or from web pages. Users' profiles are currently used to ensure the diffusion of the right information to the right users. To decide on how to spread information throughout a company is a key decision in any management. It is nowadays heuristically designed and decided. There is however a lack of formal model in information technology to tackle such problems. Knowledge management can be considered in the perspective of rationalizing and storing internal knowledge, and/or in the perspective of feeding a knowledge base with *imported* knowledge from external sources [12, 10].

The paradigm underlying most of the approaches listed above remains the database paradigm: Information is mostly centralized within a uniform information structure. In agreement with Bonifacio [1] we argue that this approach is not compliant with the very nature of knowledge that is subjective, distributed and contextual. Multiagent systems (MAS) have been introduced as a methodology to address distributed computing problems in artificial intelligence. They have evolved as a management methodology and a software engineering design leading to object-oriented-like systems. MAS are by definition heterogeneous systems. Main software agents developed in the KM area implement functionalities of KM [15]: extraction of knowledge from document bases [6, 5, 8], users' profile identification [6], knowledge targeted diffusion [4, 7]. When adequately considering most of these tools, we observe that they solely cover single issues of corporate knowledge and that they do not propose a broad and generic view on corporate knowledge. Indeed, the level of abstraction remains mostly insufficient and the broad scope of available knowledge is not considered appropriately.

Another approach of MAS consists of considering agents for modeling the organizational environment. Zambonelli [17] considered this societal approach and proposes an abstraction for computational organization. Shoham [14] proposes what he calls agent-oriented programming (AOP) by analogy to object-oriented programming. However, this is not a real abstraction as in OOP. In [3] we propose an agent-oriented abstraction (AOA) and we show that this approach can be generalized to describe in a fully generic way a society of agents [2] with some required and selected features. It offers a smooth transition between agent level and society level. It is briefly presented hereafter. We will demonstrate in this article that corporate knowledge can take advantage of agent-oriented abstraction.

3. Agent oriented abstraction

The AOA paradigm covers the concepts of agents, annotated knowledge, utility functions and society of agents. AOA associates to agents the usual features (ability to per-

ceive, reason, act, communicate [9]) and it is compliant with a societal approach of agents. Indeed, AOA is based on Weber's classical theory in Sociology [16]. Very briefly, it can be said that Weber states that a society is the result of the actions of individuals.

AOA assumes that agents are entities consisting of both of knowledge component and a decision mechanism system. The knowledge component covers any piece of information available in an enterprise from the technology required to design and produce goods to management decision policy through human relations and internal or external communication. It is partitioned into four components, also called annotations: ontology, communication, cognition and security. The agent's decision mechanism system is related to its tasks and goals. It generates utility functions. Utility functions measure the efficiency of the decisions, for instance choosing the next task to execute or choosing a parameter value within a task. Utility functions are structured into classes. The decision making system is based upon the knowledge component. Agents are then abstractly defined in terms of knowledge and utility. Specializations are made through implementations.

The AOA model extends the abstraction capabilities of the existing Agent-Oriented Programming paradigm (AOP) of Shoham. While AOP does not duplicate what OOP is for programming language, it is purely a societal approach of the design of MAS relying on a BDI (Belief, Desire, Intention) knowledge approach, AOA does it. In this paper, we are investigating the application of this model to the abstract modeling of corporate knowledge. We show that such an abstraction mechanism leads to very practical applications whether on the web or on any other support.

4. The knowledge company abstraction

The aim of this section is to present a general framework addressing knowledge management issue within companies. This abstraction is structured along the same lines that are used to introduce AOA. Our approach is based on distinct basic concepts: agents, annotated knowledge and clusters.

We set as an axiom that the activity within a company is defined through its corporate memory. Corporate memory encompasses then any piece of information available into the enterprise from the technology required to design and produce goods to management decision policy through human relations and internal or external communication. Our approach of corporate knowledge can be summarized in the following definitions:

Definition 1 *A company is composed of agents (individuals and automata). Agents possess knowledge.*

Definition 2 *Agents knowledge is annotated and annotations are classes (or types) structuring the knowledge.*

Definition 3 *Clusters are coherent subsets of agents knowledge. Clusters are defined through filters.*

Definition 4 *Actions and decisions of an agent are performed/taken according to its knowledge and related to its decision making mechanism and evaluated by utility functions.*

Definition 5 *Corporate knowledge is the amount of knowledge provided by individual agents that structure a company. It is made of stored knowledge and of mined knowledge resulting from internal and external communication exchanges.*

The main motivation for this approach is to avoid the separation/dichotomy between agents and knowledge. In other words, we would say that knowledge is not external to agents rather knowledge belongs to individual agents. By taking this point of view, knowledge is kept as near as possible to its use during task processing. Knowledge is therefore not considered and recorded in a database-like paradigm, rather in an individualized frame-like paradigm.

Definition 1 implies that individual and automata are considered under the same agent abstraction. Considering the usual approaches, agents are active objects with the ability to perceive, to reason and to act. In addition, it is generally assumed that agents have explicitly represented knowledge and that they have communication ability. We assess this to be the right level of abstraction because it enables to describe indifferently behavior of individuals, usual software systems or native agent-based software. Societal behavior consists of communication among agents. Each agent must of course be provided with a suitable interface: individuals must be provided with a software user-interface, software systems must be provided with related API (Application Program Interface), for instance. A single example is the transfer of a phone call from a secretariat office to a manager office: it can be done manually by an individual or it can be done automatically by a (more or less sophisticated) computer-based answering machine. With respect to the knowledge component, such a task remains conceptually identical whether it is carried out by the former or the latter actor. The knowledge for this task consists of the phone number of the person to be called. Tasks processed by agents require some existing knowledge and may generate some knowledge production. Following on this previous simple example, the call placed (date, time, caller phone number) can be recorded as a new knowledge item by the agent.

Definition 2 enables the following comments. Classical models for e-commerce do associate an ontology to an

agent. An ontology represents some structured knowledge. It is usually a specification of objects, concepts, and relationships. Such a model is not general enough to assess that the knowledge component of an agent is solely an ontology. Of course this knowledge has most of the time an ontological component. It has been proposed in the AOA approach to extend it to three other components: communication, cognition and integrity mechanisms. The communication component covers the communication abilities of an agent, protocols, message types, etc. The cognition component concerns the reasoning capabilities of the agent. Reasoning can be supported by formalized representations, for instance logic, pragmatics decision making such as intuition or models, for instance heuristics. The security component covers legal aspects, regulations, security issues related to the agent. This distinction into these four dimensions is purely heuristic and intuitive and must not be considered as exhaustive. It can be extended. The concept of annotated knowledge is introduced to structure knowledge of agents. Annotations are classes of knowledge. They enable a hierarchical partitioning of knowledge as well as specialization through inheritance. Inheritance from different annotations and classes is possible, and classes may thus concern several annotations. A knowledge item i can then be annotated by several annotations a_1, \dots, a_n which confers to this item some properties. For instance, messages may be allowed in a communication protocol based upon belief maintenance (communication annotation) and in the same time they may be constrained by the regulations governing the society for this agent (integrity annotation).

A consequence of definition 3 is that clusters and filters are first level abstractions of our approach. A cluster is a sub-part (or sub-class) of an agents knowledge. Clusters can be considered as external views on the agents knowledge (database vocabulary). A cluster k may be, for instance, related to a task or to a given topic. A single concept constitutes the simplest cluster. For a knowledge component consisting in a set C of concepts, the set CL of clusters is defined recursively:

- $\forall c_i \in C : k = (c_i, \{c_i\}, \emptyset) \in CL$ (every concept is a cluster)
- $k = (Head, Subclusters, Relations) \in CL$, where $Subclusters \in 2^{CL} \setminus \emptyset$ is the set of clusters that participate in k , $Head \in Subclusters$ is non-empty. Relations define the relations inside the cluster and in-between subclusters.

Basic operators on clusters may be defined, such as *addition, filtering, search, is_sub-part_of, compare*, etc. Filters are operators for extracting sub-clusters from more extended clusters. For instance, a filter can be applied to remove sensible information from a cluster before broadcast-

ing it. Knowledge annotations and classes are a way to extract relevant clusters.

A purpose for definition 4 is that agents rely on their knowledge to make decisions and to carry out actions. Taking into account their decision mechanism, they define utility functions to evaluate the impact of actions when assessing their state, their environment perception and their assigned goals. Any type of actions is treated similarly whether it is a communication action or a safety directed action or any other. Agents are by definition autonomous and they achieve their assigned goals without any external contribution to the required actions regarding, for instance, knowledge diffusion or query. This implies that they are provided with goals that can be directly or indirectly translated into communication actions. A *greedy* agent may for instance permanently perform queries in order to increase its amount of knowledge. A *secret* agent may refuse to answer queries. This modeling of corporate knowledge supports any type of communication arising from all types of social organization. One may model ranging from a rigid hierarchy as in a trust to a totally anarchical society where each agent has a same weight in the decision making process.

The reason behind definition 5 is that corporate knowledge consists of agents with their proper knowledge and their associated communication mechanisms. The concept of knowledge is meant here to be as large as possible. It covers any information an agent either has or can access. Therefore, the societal dimension of knowledge is emphasized. This is to be compared to the more traditional approach to corporate knowledge which is very often monolithic and centralized and which considers knowledge as external to agents. In this model, corporate knowledge management does not require to set objectivity as a global feature. In addition, no knowledge source for a single agent is supposed to be exhaustive. Indeed, it can not be assumed that an agent's knowledge is complete and this knowledge may be contradictory with the knowledge of other agents. Furthermore, tasks carried out by agents are necessarily integrated into a company business process and into any interaction with other agents. Our approach is compliant with this point of view since the knowledge of agents integrates its behavior within the environment.

An additional comment concerns the proposed approach and its link to the AOA approach. The latter proposes to consider agents in a framework composed of two components: the knowledge component and a decision making system. Our approach is a continuation and a validation of the AOA approach applied to corporate knowledge. Thus, emphasis is put on the knowledge component and little on the agent decision making system (with features such as utility functions). Doing so, agents remain undifferentiated whether they consist of a software application (stand alone

or complex/distributed) or of an *intelligent* user-interfaces dedicated to an individual. The decision making system remains conceptually identical in both cases. An important fact is that it is strongly related to the tasks assigned to the agent.

5. Case study

5.1. Single scenario

We introduce a very simple example that will be progressively developed to illustrate our approach. It consists of a one-person consulting enterprise. This person is in charge of all required tasks: advertisement, phone calls, meetings, meetings preparation, letter writing, typing, billing and accounting among many others. Knowledge sharing within the company does not appear since only one person is concerned. Knowledge exchange appears with customers. Let us assume that the business goes so well that the consultant decides to buy an answering machine since he/she leaves the office more frequently. Information is going to be formalized and transmitted within the company, from the consultant to the answering machine and from the answering machine to the consultant. Now, let us assume that the owner decides to hire a bookkeeper because it takes too much time to do the accounting. A flow of information must then be transferred to the new employee, initially first and then regularly each day. As in any company, the business flow is usually not steady. The consultant is sometimes overloaded and sometimes idle. Then, s/he decides to pool with others independent consultants in order to be able to negotiate work transfers with the objectives of sharing/distributing the workload, avoiding a loss of opportunities from clients and hopefully increasing gains.

5.2. Modeling within the AOA

The modeling of corporate knowledge within our approach can be illustrated in the frame of this one manager only example is described below. Agents in this example will take the following names: AC1 (for Agent Consultant 1), EAC (for External Agent Client), AAM (for Agent Answering Machine), ABK (for Agent Book Keeper) and AC2 (for Agent Consultant 2).

AC1 pursue an activity consisting of several tasks: searching for customers (prospecting), signing contracts, gathering information, producing results, etc. These tasks are divided into sub-tasks and are associated to utility functions that serve to select the next task to perform. Following the AOA approach, the knowledge of agents can be divided into 4 classes: ontology, communication, cognition and security. The ontology knowledge of AC1 covers: prospecting (meetings, arguments, etc.), producing

AC1 knowledge regarding EAC	
Ontology knowledge:	company structure and actors, domain of activity, tasks description, consultant's methodology, etc.
Communication:	synchronous/asynchronous, negotiation, cooperation
Safety:	encryption, authentication, authorization

Figure 1. AC1 knowledge regarding EAC

AC1 knowledge regarding AAM	
Ontology knowledge:	AAM identification, secret access code, pre-registered announce, delay before answering, messages led by callers (transferred from AAM), client EAC, EAC-phnb, AC1-urgentCall
Communication:	synchronous, master/slave
Cognition:	rule: Call from client EAC is an "urgent call"
Safety:	encryption method for messages

Figure 2. AC1 knowledge regarding AAM

(clients, documents, meetings, etc.), application domains, etc. We assume that the client is simulated through a single agent called EAC. Interaction protocols between AC1 and EAC may be sometimes synchronous and sometimes asynchronous, they may include negotiation phases and cooperative phases. AC1 may ask questions, organize discussions, observe meetings, etc. Doing so, it collects knowledge transferred by or from EAC. Knowledge is structured or unstructured, complete or incomplete, certain or uncertain. It concerns of course the study carried out by AC1: the firm structure and actors, the field of activity, the tasks description, the methodology, etc. Subclasses of annotations are used to structure knowledge. Specializations of these subclasses are created during the study, either through computations (queries, data mining) or manually. Tips or conclusions are transferred (regularly or at the end of the study) from AC1 to EAC. Filters can be defined within AC1 to create knowledge clusters, for instance for considering different departments at the customers site, tasks, issues, etc. An example of safety rule could be the use of an encryption method for communications. The method could be different for each client. Before exchanging knowledge with a client, AC1 must then select some knowledge (cluster) and choose a communication method. An encryption method is automatically chosen (depending on the recipient) and it is applied to messages during the exchanges. See figure 1 for a brief survey.

AC1 communicates with AAM in a master/slave mode, that is, AC1 controls AAM behavior parameters. The main tasks of AC1 related to communication with AAM are: set of parameters, put on, put off and ask for messages. A task is executed when its associated utility function is predomi-

nant (for instance the utility function for put on is maximal when AC1 is very busy). AC1 transfers to AAM the values of parameters such as: secret access code, pre-registered vocal announce or delay before answering. AC1 can also ask AAM for messages left by callers. A more sophisticated relation between AC1 and AAM is the following: AC1 is provided with the fact that it must not miss a call from client EAC. EAC phone number is known (EAC-phnb), and AC1 will be joinable at the phone number AC1-urgentCall. Information is transferred to AAM. It has the ability to identify phone callers and to enact appropriate actions. Regarding security issues, AC1 can ask AAM to encrypt recorded messages before their transfer. To this end, the encryption method is transmitted to AAM. AC1 will then be the only agent being able to understand these messages. For a short presentation see figure 2

The accountant is an employee of the consultant and obeys her/his directives (in the framework of existing laws and of their respective competence). AC1 communicates with ABK in a collaborative mode. Indeed, even if ABK obeys to AC1 orders, it acts in the framework of legality in its domain of competence. AC1 and ABK decide how they will communicate (orally, by email, once a day, etc.). Tasks of AC1 linked to book keeping are for example: start new expenses, negotiate new credit line with banker, start invoicing process, etc. Utility functions related to these respective tasks can be: expected return on investment, credit negotiation opportunity (companys good results, change of banker) or completion of a study. Ontology knowledge of AC1 regarding the collaboration with ABK consists for instance of ABK identification, planned expenses, actual expenses, incomes, turnover, clients, suppliers, banker or bor-

AC1 knowledge regarding ABK	
Ontology knowledge:	ABK identification, expenses (planned, actual), incomes (planned, actual), turnover, clients, suppliers, banker, credit limit, etc.
Communication:	synchronous/asynchronous, collaboration
Cognition:	rule: "do not plan more expenses than allowed"
Safety:	all expenses and incomes must be transmitted to ABK

Figure 3. AC1 knowledge regarding ABK

AC1 knowledge regarding AC2	
Ontology knowledge:	AC2 identification, competences, workload (planned, actual), clients, daily invoice amount, etc.
Communication:	synchronous/asynchronous, negotiation
Cognition:	strategic positioning
Safety:	respect subcontract engagement

Figure 4. AC1 knowledge regarding AC2

rowing limit. As communication is based on a collaborative protocol, AC1 is provided with the ability to query ABK and to answer to queries from it. Decision functions must be transferred from AC1 to ABK: criteria for setting up the delay of payment for sellers or customers, criteria for automatic authorizing on expenses (small/significant, common/uncommon), etc. ABK possesses also its own knowledge and intrinsic characteristics. ABK obeys obviously to the laws and rules of its domain of competence: accountancy. For example, it elaborates monthly the account listing and asks AC1 to validate it. A presentation of this relationship can be found in figure 3.

AC1 communicates with AC2 in a synchronous or asynchronous negotiation mode. The list of tasks comprises: to maintain a list of colleagues, to send subcontract bids, to answer to subcontract offers, to check planning, etc. The ontological knowledge of AC1 regarding the collaboration with AC2 consists of: AC2 identification, competences, workload (planned, actual), clients, daily invoice amount, etc. Cognition rules may for instance concern strategic positioning: accept reduced daily invoice amount in order to acquire experience in a targeted competence area. The safety component of knowledge is for instance the engagement to not negotiate for contract with a client (see above the relation with EAC) who was met/encountered during a subcontract (do not *steal* the client from AC2). The knowledge base is represented in figure 4

It would be too long to describe the knowledge and the tasks of each agent of our example. We showed however that our modeling of corporate knowledge encompasses the knowledge and the tasks of agents and that it is general

enough to model individual and system behaviors. Agents behavior consists of evaluating utility functions and selecting the next task to be executed. Tasks regarding knowledge exchange consist of capturing and annotating knowledge, defining filters and knowledge clusters, transferring and receiving clusters. Since agents are not necessarily purely automated agents, they can be provided with a friendly user interface. In some cases, the agent is an automated system and in some others, it plays the role of a software assistant to the user.

6. Conclusion

Corporate knowledge consists both of information that is available throughout a company and of information technology frameworks and paradigms. Considering an enterprise as a distributed computational paradigm, multi-agent systems have been proposed to address knowledge management issues within a company. We propose in this paper a new abstract modeling of corporate knowledge. Our approach is based on the agent oriented abstraction paradigm. This paradigm provides a high level of abstraction. From the perspective of corporate knowledge management, agents cooperate to resolve a unique high-level problem, namely the success of a company. There are numerous strategies for achieving this goal. We consider agents that have the ability to process their own tasks or solve problems, which could gain in productivity, time or quality by importing knowledge from other agents. The broad scope of available knowledge is then appropriately considered. Our work gives now the possibility to better simulate and

support knowledge management processes, and therefore to innovate with new methods in this field. It is also well-suited to filter the amount of knowledge that is transmitted throughout a company. This is one of the decisions shaping up how management is conducted within a company. This will be the topic of another publication. It is also possible to introduce a concept of virtual knowledge communities for distributed knowledge management as shown in [13].

References

- [1] M. Bonifacio, P. Bouquet, and R. Cuel. Knowledge nodes: the building blocks of a distributed approach to knowledge management. *Journal of Universal Computer Science*, 8(6):652–661, 2002.
- [2] J. Calmet, A. Daemi, R. Endsuleit, and T. Mie. A liberal approach to openness in societies of agents. In *Proc. of Engineering Societies in the Agents World*, Lecture Notes in Artificial Intelligence. Springer Verlag, 2003. To appear in post-proceedings.
- [3] J. Calmet, P. Maret, and R. Endsuleit. Agent oriented abstraction. Submitted to Royal Academy of Sciences Journal. Special Issue on Symbolic Computation and Artificial Intelligence, 2004.
- [4] Campiello home page. Available electronically at: <http://klee.cootech.disco.unimib.it/campiello/>, 2004.
- [5] J.R. Chen, S.R. Wolfe, and S.D. Wragg. A distributed multi-agent system for collaborative information management and sharing. In *Proc. of the 9th ACM International Conference on Information and Knowledge Management*, pages 382–388, 2000.
- [6] CoMMA home page. Available electronically at <http://www.si.fr.atosorigin.com/sophia/comma/Htm/HomePage.htm>, 2004.
- [7] V.R. Benjamin et al. Skills management in knowledge-intensive organizations. In *Proc. of the EKAW 2002 International Conference*, volume 2473 of *Lecture Notes on Computer Science*, pages 80–95. Springer Verlag, 2002.
- [8] FRODO home page. Available electronically at <http://www.dfki.uni-kl.de/frodo/>, 2004.
- [9] M. Huhn and L. Stephens. Multiagent systems and societies of agents. In *Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence*, chapter 2, pages 79–120. Cambridge, MA, USA, MIT Press, 1999.
- [10] K-InCA home page. Available electronically at <http://www.calt.insead.edu/project/K-InCA/>, 2004.
- [11] H. Kautz, B. Selman, and M. Sha. Referral web: combining social networks and collaborative filtering. *Communication of the ACM*, 40(3):63–65, 1997.
- [12] KRAFT home page. Available electronically at <http://www.csc.liv.ac.uk/~kraft/>, 2004.
- [13] P. Maret, M. Hammond and J. Calmet. Agent Based Virtual Knowledge Communities for Distributed Knowledge Management. Submitted, 2004.
- [14] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60:51–92, 1993.
- [15] L. van Elst, V. Dignum, and A. Abecker. Towards agent-mediated knowledge management. In *Agent-Mediated Knowledge Management: Selected Papers*, volume 2926 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 2003.
- [16] M. Weber. *Economy and Society*. University of California Press, 1986.
- [17] F. Zambonelli, N.J. Jennings, and M. Wooldridge. Organisational rules as an abstraction for the analysis and design of multi-agent systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(3):303–328, 2001.