Organization Simulation Based on Normative Knowledge and Role Modeling

Joaquim Filipe*, Kecheng Liu and Bernadette Sharp****

* Escola Superior de Tecnologia do Instituto Politécnico de Setúbal
Rua Vale de Chaves, Estefanilha, 2910 Setúbal, Portugal.
JFilipe@est.ips.pt
** School of Computing of the Staffordshire University
Beaconside, Stafford ST18 0AD, UK
{K.Liu, B.Sharp}@staffs.ac.uk

Introduction

An organizational role is defined as a set of functions together with a set of policies (deontic statements). Several roles can be played by the same human or artificial agent. It is also possible that a set of agents is designated to play a given role.

In modern organizations it is common to find most agents performing several different simultaneous roles. This may cause ethical and deontic conflicts when the same agent must play conflicting roles. In such cases the agent must use some criterion to solve the internal conflict. In the following we will propose a methodology for approaching this kind of problems, maintaining the essential agents' autonomy.

Organizational agents, humans or machines, must be modeled using a social model. This view is envisioned by the computational semiotics community, where six layers are defined (Stamper 1973) including two broad areas: the information technology platform, which consists of three layers (physical world, empirics and syntax) and the human information functions, which consists also of three layers (semantics, pragmatics and social world).

The semantic analysis of business environments can be graphically depicted as an ontology chart, including all agents, roles, resources and their ontological dependencies. An example is shown in figure 1. Additionally to the conceptual knowledge, expressed graphically in the ontology chart, the ontology must also specify behavioral normative knowledge using norms. Norms will be formally defined below, but informally they are seen as social rules that an agent ought to follow. When a clear organizational ontology is not available, where all roles are described and all normative aspects such as authority and power relationships are clear, it might not be possible to understand and predict the agents' behavior.

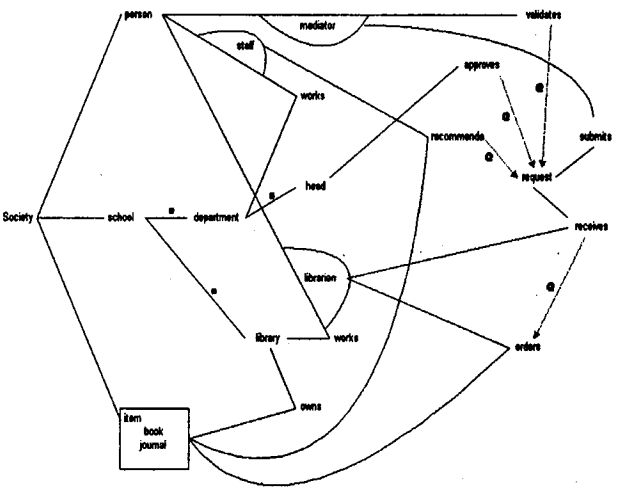

Figure 1: Ontology chart

In the organizational model we envisaged all agents are autonomous. Therefore, it is an essential aspect of our model that agents may choose to violate norms. However, in order to reinforce coordination, there are also norms that are applicable in violation situations (the so-called contrary-to-duties).

Artificial agents are intelligent agents that are designed to perform routine tasks, which require moderate reasoning skills. These agents are built using very abstract software concepts. It is usual to describe them using mental attitudes, such as the BDI architecture model, which is based on the combination of Beliefs, Desires and Intentions (Rao and Georgeff 1992). This model ascribes mental attitudes to agents in order to make it easier to discuss the specification and analysis of intelligent agents, by using an adequate abstract level of discussion..

Typically, coordination is seen as a process that requires communication. This paper presents a framework in which normative semiotic information fields (Stamper 1999), underlie communicative actions of organizational autonomous agents, which co-operate not only through

communication but also by sharing of a common ontology of the organization, and being aware of other agents' roles.

Semiotics And Business Process Modeling

The semiotic approach to computing emphasizes the importance of the integration of computers in social reality. It is very important to make computer-based systems fit into a business organization and integrate information technology with the social aspects that enable the successful fulfillment of business goals. Sometimes there is a risk of applying highly sophisticated technology without a clear understanding of the information circuits and information systems already in place.

Using the semiotics framework to cover the main stages of systems lifecycle, a collection of methods has been defined, which can be applied to all the systems development activities along the systems development lifecycle. These were developed under the MEASUR research program – Method for Eliciting, Analyzing and Specifying User Requirements (Stamper et al. 1988). This program, actually a suite of methods, was developed including methods for semantic analysis, ontology representation, and norm analysis. Norms represent business rules, social goals, constraints and other structural aspects of the organization and are essential for defining an agent's roles, including the specification of its functions and obligations.

The adopted approach views a business process as a process-oriented network of autonomous agents. Agents can represent individuals or collectives, including external stakeholders such as customers, regulators or suppliers, and internal entities such as staff, departments, or systems. The syntax and semantics we adopt for representing knowledge inside the agents' minds is based on Predicate Logic and Deontic Logic. At the pragmatics level we model the communication between agents using a specialized language, similar to KQML, based on speech-act theory (Searle 1969). At the social level the normative knowledge acquires the most prominent role. Norms are the source of commitments and contracts the agents establish with each other, which autonomous agents may choose to violate. Social obligations and social goals are however different from individual goals, in the sense that although their adoption depends on the agent, there are social costs involved if they are dropped.

Normative Knowledge Modeling

The formal representation of norms can also be done using different knowledge representation paradigms. For example, NORMA (Liu 1993) is a knowledge representation language inspired in frame systems, and using a specific methodology and notation for representing norms, which was developed in the scope of the MEASUR project, aforementioned. In NORMA a norm has the basic structure:

<Norm>::= If <Condition> then <D> <Agent> <Action>

where D is a Deontic operator (obligation, permission or prohibition), Agent is a responsible entity and Action is what the agent does if the norm is adopted.

Furthermore, time-aspects (triggers and deadlines) and indications concerning the target domain of the norm are specified in NORMA using norm frames.

In NORMA a norm controlling the behavior of an agent at the social level, i.e. an action norm, is viewed has a constraint. However, besides their proscriptive nature we believe that there is also a prescriptive nature, i.e. norms may act as a mechanism of goal generation. Norms can be seen as a helpful tool for handling bounded rationality: the cognitive apparatus obtains better and more consistent behaviors than rational calculus probably would

Organizational Roles

An organizational role specifies a set of functions, obligations and rights associated with a job position. In this context, roles are conceptual structures that describe the behavior of organizational agents. This includes the declarative specification of the relevant ontology: the relevant entities and their relationships, the available set of actions the agent can do and their specification in some formal executable language.

Roles are generic, independent of the specific agents that fulfill them at a certain point in time. In that sense roles are abstract structures that define an organization structure.

The role structure we propose here is a first step in the design of an agent organization, where agents interact autonomously in the way described in the previous section.

Roles are structured in two parts:

1. A normative part that enumerates all the obligations and rights applying to the role, which is specified as a list of Deontic logic rules;
2. A functional part, where the actions that can be performed in the scope of the role are defined. This part enumerates all the conversation plans that the agent playing the role can choose to use.

Both parts are described mainly declaratively. The use of a declarative paradigm for describing actions is less efficient than the procedural paradigms but has the advantage of providing flexibility, allowing agents to adapt their actions in "run-time" even when they meet unexpected situations.

Organizational structure defines power relations that influence the authorizations and obligations of each role. If a role is in direct line dependency on another one then the superior role has the right to request or command the subordinate role and the subordinate role is obliged to obey the requests or commands of the superior role, within the scope of its functions.

Case-study

This case-study concerns an organizational process of

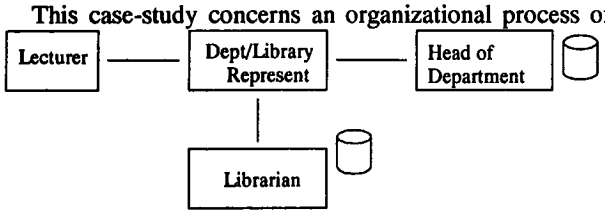

Figure 2: Role Relationships Diagram (brief)

the Polytechnic Institute of Setúbal depicted in figure 2. The book acquisition process is started by the lecturer, who merely expresses the desire to obtain a certain book. This is picked up by the agent that performs the mediator role between the department to which the lecturer belongs and the library. This mediator must enquire if the president of the department considers the book worth being acquired and has enough money to buy it. If the answer is positive then the library will purchase the book and will inform the lecturer as soon as the book becomes available. Each organizational agent may be a human agent or an artificial agent. However, artificial agents are acting in the organizational network on behalf of users who are ultimately responsible for these agents.

Semantic Analysis

In figure 1 is depicted the ontology chart that results from the semantic analysis of the aforementioned organizational process. Role specification for each role indicated in figure 2 was made using norm rules with the format described above and coded using a knowledge-based system.

Technical Aspects

The programming language that supported the implementation of the case-study was JAVA™. However, some of the software abstract layers are embedded in development tools that we used.

One of these tools is JINI™, a network-centered software, created by Sun Microsystems Inc. as an extension of Java, the company's cross-platform programming language. JINI is fully object based and objects communicate via the standard Java RMI interface through basic operations in JavaSpaces™ (a system that manages features such as object processing, sharing and migration). With JavaSpaces JINI includes facilities for transaction coordination (with two-phase commit) and persistency. JINI provides plug-and-execute capabilities where a new device (agent) on the network can gain instant recognition (boot, join and discover protocol) by the network and have access to any services for which it has authority.

JESS (JAVA Expert-System Shell) is a rule-based development tool that includes a simple programming interface for running knowledge-based systems structured using rules and facts. The inference engine supports forward-chaining and backward chaining and it is possible to represent uncertain knowledge using certainty factors.

We have developed the logic model for each agent in JESS, and the communication aspects in JINI.

The Agent Architecture

Figure 3 depicts the internal architecture of an artificial agent.

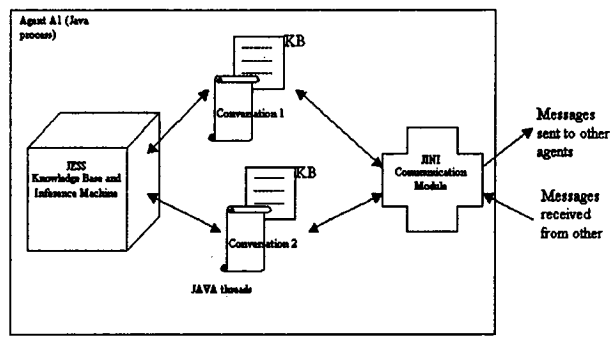

Figure 3: Agent Architecture

Artificial agents are not totally autonomous in the sense they are mere assistants, who have a human "master" who determines the initial goals of the agents.

Agents are basically message exchanging entities. However, in order to be able to manage several conversations with different agents, we need to create, inside each agent, an object type to represent a conversation. Before the agent initiates a conversation it must define in which role the agent is going to have the conversation. The conversation is itself a process thread that holds a knowledge base and an inference engine of its own, where a role has been loaded

Acknowledgments

This paper has been supported by the research project 1/97 of the Polytechnic Institute of Setúbal.

References

Liu, K. 1993. Semiotics Applied to Information Systems Development. PhD Thesis, Twente University, Holland.

Stamper, R. 1973. Information in Business and Administrative Systems. John Wiley & Sons.

Stamper, R. 1999. The Information-Field Paradigm and New Directions for Systems Analysis and Design

Rao, A. and M. Georgeff 1992. An Abstract Architecture for Rational Agents. In C. Rich, W. Swartout and B. Nebel (Eds) *Proc. 3rd Int'l Conference on Principles of Knowledge Representation and Reasoning.* Morgan Kaufman Publishers, San Mateo, CA.

Searle, J. 1969. Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press.