Formal Transition in Agent Organizations

Eric Matson
Department of Computer Science and Engineering
Wright State University
Dayton, OH 45435-0001
eric.matson@wright.edu

Department of Electrical and Computer
Engineering and Computer Science
University of Cincinnati
Cincinnati, OH 45221-0030

Scott DeLoach
Multiagent and Cooperative Robotics Lab
Department of Computing and Information Science
College of Engineering
Kansas State University
Manhattan, KS 66506
sdeloach@cis.ksu.edu

***Abstract*— Multiagent Systems (MAS) can be employed to solve a great many problems. When combined with an organization model, a MAS becomes an even more useful structure capable of taking on problems that require self-organization, adaptation, and recovery. To capture and understand the concept of organization and reorganization, we must define organization transition. Specifically, we describe a formal process of organization transition.**

I. Introduction

If we view organizations from a human perspective, we see insights that sometimes are not revealed from a purely theoretical approach. For example, consider a profit-oriented company that provides an important service. Assume the company has high turnover of employees, as it is not a pleasant place to be employed. Because of high turnover, the company will not be capable of remaining competitive, or in business, for that matter, it they cannot replace their employees once they leave the company. Further, they must be able to move the specific, remaining employees from one role to another to meet the ever-changing needs of the organization. To adapt and satisfy the dynamic requirements of the company's organization, the company's management must understand the methodology and process of transition, or more generally termed as reorganization. In other words, the ability to modify the organization to overcome environmentally inspired external changes is critical to survival.

Approaching the same problem from a multi-agent perspective, we must consider what it takes to transition an agent organization from one instance to the next. Although agent organizations are not normally as large, as complex or consider as many variables as human organizations, the basic process of organization transition is very similar.

Capturing how humans form and interact with an organization model is a complex task. Applying the same model to software agents requires essentially the same structures and relationships as with human organization models. Often, research will capture specific functional elements of organization applied to a single problem domain or a similar set of problem domains, but will neglect to fulfill a general-purpose sense of organization.

Organizations are typically dynamic in nature over time. The dynamic nature of organizations indicates that change is a constant element affecting their structure and composition. The change, or transition, creates complexity as organization instances propagate from one to the next through their active life. Understanding and formalizing these complexities is required to truly understand the nature of organizations. Self-adapting organizations must possess, at a minimum, the ability to transition from one state to the next, without intervention.

With the need to develop capable organizational models, to satisfy real world problem domains, it is important that we understand how to generally and formally describe and capture organization transition. There are numerous well defined and useful organization models. Dastani, Dignum and Dignum define an agent model for social agents with beliefs and goals that can revise goals via reasoning capabilities [1]. Picard and Gleizes present an alternate an adaptive multi-layered, multiagent approach [5]. Glaser and Morignot examine organization from a more abstract societal model [11], whereas a more complete view of agent organizations comes from Ferber et al. [12]. A specific agent infrastructure is proposed by Omicini and Ricci, well describing key elements of organization [13]. A different approach, by Turner and Turner [4] applies organization theory to a particular domain problem. The specific organizational model we use as a foundation to develop our transition theory was initially proposed by Deloach and Matson [2] with a more refined version to follow [3]. All of the mentioned models focus on structure and agent interaction, but they do not possess elements of transition theory necessary to define how an organization reconstitutes itself when required.

To develop a basis for organization and reorganization process, a formal model for transition in agent organizations is required. Dignum et al. propose a good foundation in which to reason about dynamic reorganization [6] and we will extend and more formally define transition using some shared elements.

In this paper, we will first define the organization model, including the structural, state and transition elements, in section 2. In section 3, a discussion of general transition theory is described. Section 4 defines the formalization of transition theory. Section 5 will describe conclusions, approaches and future research work, in this area. Section 6 proposes future areas of research that derive from transition of agent organizations.

II. Organization Model

To implement teams of autonomous, heterogeneous agents, we created an organizational model, which defines and constrains the required elements of a stable, adaptable and versatile team. While most people have an intuitive idea of what an organization is, there are no standard definitions. However, in most organizational research, organizations have typically been understood as including agents playing roles within a structure in order to satisfy a given set of goals. Our proposed organizational model (O) contains a structural model, a state model and a transition function [2] [9]. Fig. 1 shows the combined structural and state models using standard UML notation.

$$O = (O_{structure}, O_{state}, O_{transition})$$

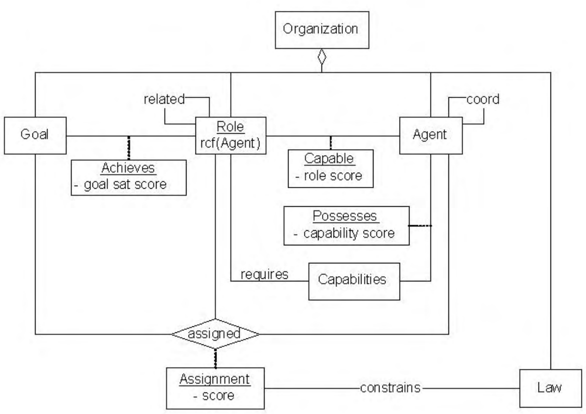

Fig. 1. Organization model

A. Structure

The structural model includes a set of goals (G) that the team is attempting to achieve, a set of roles (R) that must be played to attain those goals, a set of capabilities (C) required to play those roles, and a set of rules or laws (L) that constrain the organization. The model also contains static relations between roles and goals (achieves), roles and capabilities (requires), individual roles (related), goals and subgoals(subgoals) and a unary relation for conjunctivity between subgoals of a goal(conjunctive). Formally, we model the organization structure as a tuple:

$$\begin{aligned} O_{structure} =& < G, R, L, C, achieves, related, requires, \\ & subgoal, conjunctive > \\ where:& \\ & achieves : R, G \rightarrow [0..1] \\ & related : R, R \rightarrow Boolean \\ & requires : R, C \rightarrow Boolean \\ & subgoal : G, G \rightarrow Boolean \\ & conjunctive : G \rightarrow Boolean \end{aligned}$$

Goals are abstract entities that often must be decomposed to have deliverable outputs are used to identify the critical aspects of system requirements. The team goals include the goal definitions, goal-subgoal decomposition, and the relationship between the goals and their subgoals, which are either conjunctive or disjunctive. Therefore, an analyst should specify goals as abstractly as possible without losing the essence of the requirement. This abstraction can be performed by removing detailed information when specifying goals.

A *Role* describes an entity that performs some function within the system, analogous to roles played by actors in a play or by members of a typical company structure. In general, roles may be played by zero, one, or many agents simultaneously while agents may also play many roles at the same time. Each role requires a set of capabilities, which are inherent to particular agents.

Organizational *laws* are used to constrain the assignment of agents to roles and goals within the organization. Generic rules such as "an agent may only play one role at a time" or "agents may only work on a single goal at a time" are common. However, rules are often application specific, such as requiring particular agents to play specific roles.We introduce the notion of laws into the organization, which operationalize norms, sanctions/rewards, and their relationship. Laws should also conform to organizational values. Laws are constraints on actions and thus the law (a, s) prohibits the action a from being taken when state s holds [8].

Capabilities are the abilities that reside within a particular agent. Capabilities are defined by the level of intelligence or some discrete set of agent functions.

Achieves is modeled as a function to capture the relative ability of a particular role to satisfy a given goal. A role that can be used to satisfy a particular goal is said to achieve that goal.

Relationships are dynamically allocated, cohesive links that exist from role-to-role, agent-to-agent, and robot-to-robot during the active organization lifespan. The relationships may be based on communication, delegation, cooperation, or other factors.

Requires defines a boolean relation the specifies a role must have a capability. If this capability is not present then the relation is false.

A *subgoal* defines a boolean relationship between two goals where one goal is a direct subgoal of the other. If the relationship does not hold, the result is false.

The *conjunctive* relation can actually specify whether a goal has a conjunctive or disjunctive relationship with all of its subgoals. A conjunctive relation is defined by true whereas a disjunctive relation is defined by false.

B. State

The second element of the Organization Model is its state. The Organizational State (O_{state}) is an instance of the organizational structure at a point in time. As the Organization Model is a template, the state is an instance of the model. In an instance of an organization state, each of the elements will be bound to a set of values that represent the organization attributes. An organization will possess at least one goal, one role to accomplish the goal, and one agent to play the role where the agent will possess capabilities required by the role. Not every organization state element is required to be populated by an instance variable for creation of a valid organization. The constraints and laws of an organization will govern the requirements of a specific state.

The organizational state model defines an instance of a team's organization and includes a set of agents (A) and the actual relationships between the agents and the various structural model components.

$$
\begin{aligned}
&O_{state} =< A, possesses, capable, assigned, coord > \\
&where: \\
&\qquad possesses : A, C \rightarrow [0..1] \\
&\qquad capable : A, R \rightarrow [0..1] \\
&\qquad assigned : A, R, G \rightarrow [0..1] \\
&\qquad coord : A, A \rightarrow Boolean
\end{aligned}
$$

Agents coordinate through the organization via conversations and act pro-actively and cooperatively to accomplish global and individual goals.

An agent that *possesses* the required capabilities for a particular role is said to be capable of playing that role. Since not all agents are created equally, possesses is modeled as a real valued function, where 0 would represent absolutely no capability to play a role while a 1 indicates an excellent capability. In addition, since agent capabilities may degrade over time, this value may actually change during team operation.

The *capable* function defines the ability of an agent to play a particular role and is computed based on the capabilities required to play that role.

During the organization process, a specific agent is *assigned* to play a particular role in order to satisfy a specific goal. This relationship is captured by the assigned function, which includes a real valued score that captures how well an agent, playing a specific role, can satisfy a given goal.

When an agent is actually working directly with another agent, it is *coordinating* with that agent. Thus, the state model defines the current state of the team organization within the structure provided by the structural model.

C. Transition

The Organization Transition Function ($O_{transition}$) defines the ability of the organization to reorganize from an instance state to the next instance state over the organization life span. From the initial organization, through its termination, the organization may transition its state model numerous times.

The organization transition function defines how the organization may transition from one organizational state to another over the lifetime of the organization. Since the team members (agents) as well as their individual capabilities may change over time, this function cannot be predefined, but must be computed based on the current state, the goals that are still being pursued, and the organizational rules. In our present research with purely autonomous teams, we have only considered reorganization that involves the state of the organization. However, we have defined two distinct types or reorganization: state reorganization, which only allows the modification of the organization state, and structure reorganization, which allows modification of the organization structure (and may require state reorganization to keep the organization consistent). In this paper, we will mainly focus on state reorganization, although transition encompasses both types of reorganization.

There are two distinct types of transition; organization (initial) and reorganization. The initial step in organizing a multiagent team is to use the existing information production goals to establish the organizational roles required. At the same time, the team of agents making up the team must assess their individual and collective capabilities to determine whether they can fulfill the required roles [2]. If the required roles can be filled, then the capabilities exist to satisfy the information production goals and the team assigns the necessary roles to agents (effectively defining the state of the team's organization). Once the assignments are made, the team may initiate action to satisfy the team goals. Initial organization can be represented by $O_{state(0)} \rightarrow O_{state(1)}$.

The reorganization process follows the same basic steps as the organization process; however, it differs in the point of initiation. Reorganization is initiated by a trigger event, such as capability loss, during the execution of an already existing organization. When such an event occurs, the team must determine if it still has the capabilities to satisfy team information production goals or whether it must reorganize to do so. Reorganization is represented by $O_{state(n)} \rightarrow O_{state(n+1)}$ where $n \in \aleph$.

III. Transition in General

To reason about organizations, they must be thought of in terms of their reason for existence. An organization that has no reason to exist, no mandate, nor set of goals is trivial. Any real organization is bound to accomplishment of some goal, set of goals or task.

An organization's structure and design will differ based on its overall mission and goal set. An organization can be characterized as *finite* or *infinite* based on its defined set of goals and how the goals are characterized. All organizations will be take on one of these two definitions.

Another definition is that of a *successful organization* which represents an organization capable of reaching a final state or acceptable state.

A. Finite Organizations

Finite organizations are required to accomplish a finite, discrete set of goals. When the goals are accomplished, the need for the organization is terminated. At termination, the organization no longer exists as shown in Fig. 2. An example of a finite organization is a software project team. Their mission or absolute goal is to complete a software system. When the software development project is completed, their mission is completed and thus the need for the organization has been completed. This indicates they are part of a finite organization.

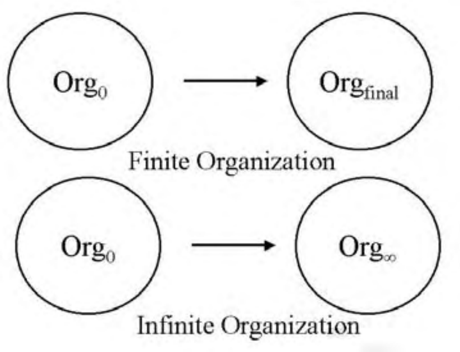

Fig. 2. Finite and Infinite Organizations

B. Infinite Organizations

An infinite organization, or non-terminating organization, is one based on a non-discrete set of goals or an evolving set of goals with no foreseeable termination or end point. An infinite organization may be limited due to external factors, but not by a strict set of discrete, accomplishable goals as shown in Fig. 2. An example of an infinite organization is a software company. A software company has no set of goals that when those goals are accomplished, the organization dissolves. The organization has no distinct set of discrete goals to determine their termination point, but continue on as long as the intention remains to stay in business. In this case, it can be said, that companies don't plan for their own demise.

IV. Formalizing Transition

Organization and reorganization are transition processes. To understand these processes, we must first formalize transition. Transition can occur for a number of reasons such as loss of agent, evolving capabilities possessed and required, sub-optimal performance or goal changes. An assumption, to be considered, is that an organization will always try to propagate to a more optimal state than it currently maintains.

In this section, organization properties will be defined. Their definition precedes all other definition because the use of organization properties is the foundation of our organization transition definition. Based on organization properties, the mechanics of the transition function are then defined. The use of the transition function is then extended to finite and infinite organizations.

A. Organization Properties

An organization property Φ is somewhat of an abstract theoretical term. It is abstract to capture to generic nature of what it can define. For general terms, an organization will need a set of properties Φ, for example, *capabilities* or *agents* which by their existence can be the reason for a transition. A major element of defining transition will be the definition of these properties such that individual properties ϕ can be identified as transition triggers. Any individual property ϕ in Φ is eligible to act as a reorganization trigger. Some examples of ϕ include a change in the real value of a capability, the loss of overall capability or agent function, loss of a agent, the reentry of a agent, or the addition of a new agent. As shown in Fig. 3, ϕ will require the organization to engage in a transition from one organization state to another.

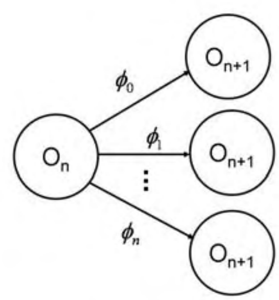

Fig. 3. State Transition

B. Transition Defined

We can reason about transition mechanics being similar to a *finite state automata* [7] or *Kripke Structure* or [10]. Both have a set of finite start states, a transition function and a set of reachable states. An organization does not use an input symbol from an alphabet to transition. It will use a triggering organization property, which in itself can be a transition, such as loss of agent or change in goals. A *finite state automata* has a form like $A = (Q, \Sigma, \delta, q_0, F)$ where A is the name of the automaton $Q$is the set of *states*, Σ is the finite set if *input symbols*, δ is the *transition* function, q_0 is the *initial state* and F is the set of *final* or *accepting* states, our transition function has a similar form. The general form of our theoretic organization model approach is expressed by:

$$O_{transition} = (O, \Phi, \delta, s_n, S_{optimal}, S_{possible} S_{final})$$

Where O is the organization over which the transition will occur, Φ is the set of properties that can trigger a transition of the organization, δ is the transition function, s_n is the set of relative states of the organization, $S_{optimal}$ is the set of optimal states that result from transition and $S_{possible}$ are states that are possible to reach, from the current state. S_{final} is a set of organization states where all goals are satisfied, or the lst goal is satisfied, or it is determined that not all goals can be satisfied. Even though the outcomes are different, each final state draws a conclusion to the organization's set of transitions. Because a organization can only exist as a single entity or instance, the current state s_n is always a unique value. Our transition model has some notable differences from a *non-deterministic finite state machine* (NFA). Our machine does not support epsilon transitions. An organization must transition for a specific and valid reason, otherwise serious side effects can occur, such as *instantaneous transition looping*, where an organization toggles between instances without an exit. Since our transition model does not have a fixed alphabet of input symbols to drive transition, it is not deterministic. It relies on properties of the organization to be dynamically changed for triggering transition.

The basic transition is defined as a product of the O, Φ and S resulting in a set of reachable organization states:

$$\delta : O \times \Phi \times S \rightarrow S$$

So the transition function will be of the form:

$$\delta(O, \phi, s_n) \rightarrow S'$$

Where transition function δ takes the organization O, a *specific* transition property ϕ, and a state of the organization s_n and can transition to a set of new states S' where $S_{optimal} \subseteq S_{possible}$ and $S_{optimal} \subseteq S'$. $S_{final} \subseteq S^{possible}$ for both finite and infinite transition organizations, with the added constraint that $\mid S_{final} \mid \geq 1$ for finite transitions and $\mid S_{final} \mid = 0$ for infinite transitions.

Where a *finite state automata* uses a string of symbols to transition, as normally used to validate languages our transition function takes as input a string of transition properties ϕ as input. This string of properties is not predetermined, but will be generated dynamically as the organization interacts in its environment as represented by $\phi_0, \phi_1 \ldots \phi_n | \phi \in \Phi, n \in \aleph$.

Anytime any element of the organization changes, a new organization state will be instantiated. Even the smallest change in the organization state changes the structural integrity of the organization state, and therefore, a transition must occur.

When a property change initiates a reorganization, the organization may transition to a new state s_{n+1} or it can also result in being in the same state (s_n) where, even though the property changed, the values and relationships of the organization instance did not change.

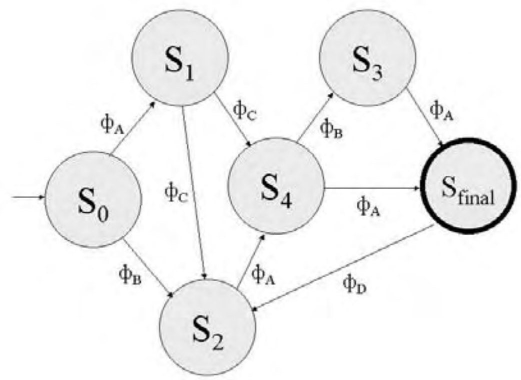

Fig. 4. Organization Transition Machine

The graphical view of an example organization transition machine is shown in Fig. 4. Stating the same transitions, using the transition function, is defined by:

$$O_{transition} = \{O, \{A, B, C, D\}, \delta, \{s_0, s_1, s_2, s_3, s_4, s_{final}\} \\ , S_{optimal}, S_{possible}, S_{final}\}$$

and the following machine transitions:

$$\delta(O_1, \phi_A, s_0) \vdash s_1$$
$$\delta(O_1, \phi_B, s_0) \vdash s_2$$
$$\delta(O_1, \phi_C, s_1) \vdash s_2$$
$$\delta(O_1, \phi_C, s_1) \vdash s_4$$
$$\delta(O_1, \phi_A, s_2) \vdash s_4$$
$$\delta(O_1, \phi_D, s_2) \vdash s_{final}$$
$$\delta(O_1, \phi_A, s_3) \vdash s_{final}$$
$$\delta(O_1, \phi_B, s_4) \vdash s_3$$
$$\delta(O_1, \phi_A, s_4) \vdash s_{final}$$

Due to the model non-deterministic definition, we can use an alternate shorthand notation and express transitions by:

$$\delta(O_1, \phi_C, s_1) \vdash \{s_2, s_4\}$$

Mathematical induction can be used to prove theoretical elements of this multiagent transitional theory. We do not supply proofs due to space considerations and basic understanding of how induction can be applied to similar instances for finite state machines.

C. Finite Transition

To formally express the definition of a finite organization transition, the condition must hold for the set of transitions to eventually result in a final state being reached. The transition:

$$\delta(O_i, \phi, S) \vdash^* S_{final}$$

states that there is a set of transitions that will lead from some initial machine state to a S_{final} or final state. This will also indicate that, if designed correctly, a finite transition organization possibly is a *successful organization*.

D. Infinite Transition

An infinite organization transition can be expressed in a similar manner to the finite. By definition, an infinite organization, never reaches a final state, so the expression will result in S_{∞}, instead of S_{final}.

$$\delta(O_i, \phi, S) \vdash^* S_{\infty}$$

This indicates that an infinite organization will not reach a final state and will continue on transitioning indefinitely. For this reason, $\mid S_{final} \mid = 0$ will be an enforced constraint for any infinite transition organization.

V. Conclusion

We have defined the basic elements of formal transition for a multiagent organization. The foundational elements stem from the well-conceived definitions of finite state machine and general automata theory. While this is not a fully developed and implemented system, it is instead, the basic theory necessary to understand the nature and mechanisms of transition for complex multiagent systems.

The theoretical elements, of this work, are directly linked to an existing and usable model for developing multiagent organizations in a a number of research domains. It defines a basic theory of transition, as well as, finite and infinite organizations.

A new finite state automata is defined to support the transition of multiagent organizations, in both finite and infinite modes. The new machine allows an organization to move from state to state during the life of the organization. This basic foundation will support the concept of self-organization and adaptability to the problem domain environment.

The requirement of multiagent organizations to be adaptable, re-configurable and possess the capability to survive are all supported by the definition of transition and reorganization theory. Without the ability to transition, the system cannot be adaptable. This is true for software-based multiagent systems as well as human organizations.

VI. Future Work

The are a number if issues and opportunities with the formal definition of transition in multiagent organizations. A paramount issue is controlling the state explosion problem. A second opportunity is to integrate this theoretical approach, with an implementation. A third refinement of importance, is the theoretical description of organization properties.

The first problem is that of state explosion. With real valued functions playing a part in the definition of an organization's structure a side effect is the ability to reorganize to an infinite number of new states. This creates a problem, not only in theory, but also in the practical matter of managing the organization transition. It also potentially violates the definition of a finite state structure. An area of exploration will be to examine methods of abstraction for minimizing or controlling the state explosion in organization transition.

We will integrate the theoretical definition of the formal agent organization transition system into the already existing organization model to create a fully implemented and extensible system. This system will then be applicable to a number of problem domains, including, but not limited to, robotic teams, agent organization simulation and enterprise systems development.

Organization properties are a key factor in driving transition. External factors will often force the organization to transition to a new state. It is important to fully understand and develop the idea of organization properties.

VII. Acknowledgment

This work is sponsored by the Air Force Office of Scientific Research (AFOSR) under grant number F49620-02-1-0427.

References

[1] M. Dastani, V. Dignum, F. Dignum, Organizations and Normative Agents. In Proceedings of the First Eurasian Conference on Advances in Information and Communication Technology (EurAsia-ICT 2002) Tehran, Iran, October 29-31, 2002.

[2] Eric Matson, Scott DeLoach. Organizational Model for Cooperative and Sustaining Robotic Ecologies, Proceedings of the Robosphere 2002 Workshop, NASA Ames Research Center, Moffett Field, California, November 14-15, 2002.

[3] Scott DeLoach, Eric Matson. An Organizational Model for Designing Adaptive Multiagent Systems. The AAAI-04 Workshop on Agent Organizations: Theory and Practice (AOTP 2004). July 25-29, 2004, San Jose, California.

[4] R. Turner,E. Turner. A Two-Level Protocol Based Approach to Controlling Autonomous Oceanographis Sampling Networks. IEEE Journal of Oceanographic Engineering, vol. 26, no. 4, pp. 654-666, October, 2001.

[5] Gauthier Picard, Marie-Pierre Gleizes. An Agent Architecture to Design Self-Organizing Collectives: Principles and Application, Adaptive Agents and Multi-Agent Systems: Adaptation and Multi-Agent Learning, Lecture Notes in Computer Science (LNAI) 2636, pp. 141-158, Springer, 2003.

[6] Virginia Dignum, Liz Sonenberg, Frank Dignum: Dynamic Reorganization of Agent Societies, Proceedings of CEAS: Workshop on Coordination in Emergent Agent Societies at ECAI 2004, Valencia, Spain, 22-27 September 2004.

[7] John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. Introduction to Automata Theory, Languages and Computation, 2nd Edition. Pearson Education, Patparganj, India, 2001.

[8] Y. Shoham and M. Tennenholtz. On Social Laws for Artificial Agent Societies: Off-Line Design. Artificial Intelligence 73(1995)231-252.

[9] Eric Matson, Scott DeLoach, Enabling Intra-Robotic Capabilities Adaptation Using An Organization-Based Multiagent System. IEEE International Conference on Robotics and Automation (IEEE ICRA 2004), April 26 May 1, 2004. New Orleans, LA USA.

[10] National Institutes of Standards and Technology (NIST), Dictionairy of Algorithms and Data Structures, http://www.nist.gov/dads/HTML/kripkeStruct.html

[11] Norbert Glaser, Philippe Morignot. The Reorganization of Societies of Autonomous Agents, Lecture Notes in Artificial Intelligence 1237, Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW '97), Ronneby, Sweden, May 1997.

[12] J. Ferber, O. Gutknecht, F. Michel. From Agents to Organizations: An Organizational View of Multi-Agent Systems, Proceedings of the 2nd International Joint Conference Autonomous Agents and Multiagent Systems (AAMAS 2003), Melbourne, Australia, ACM Press, 2003.

[13] Andrea Omicini, Alessandro Ricci, MAS Organization within a Coordination Infrastructure: Experiments in TuSCoN, Proceedings of the 4th International Workshop on Engineering Societies in the Agents World IV (ESAW 2003), London, UK, October 2003.