Knowledge Sharing Between Agents in a Transitioning Organization

Eric Matson[1,2] and Raj Bhatnagar[1]

[1] Wright State University
Department of Computer Science and Engineering
Dayton, OH, USA
[2] University of Cincinnati
Department of Computer Science
Cincinnati OH, USA
eric.matson@wright.edu

Abstract. People that interact within a cooperative organization must constantly exchange information on the details of the organization as well as the goals the organization exists to meet. Agent organizations must share knowledge if they are to cooperatively act in the solution of some set of defined goals. The manner in which they share and when they share information varies. In this paper, we present the process to share organization information during the process of transition from one organization state to the next. Some organization models choose to vary the information known between two agents, in relation to the organization. A key element of organization success is that all members operate with the same information so as not to cause divergence in action or purpose.

1 Introduction

Organizations exist in every facet of human existence. People join organizations for reasons such as fulfillment, position or learning. When a person joins an organization, they must learn, or at least be aware, of the others involved in the organization. They must understand the overall structure to fully comprehend their place within the organization. As an example, human organizations commonly use charts to describe where each person fits into the structure. These organization charts exhibit the relationships between positions and people. When a new person joins an organization they are shown where they fit as part of the orientation to the organization. As the organization transitions through changes, the knowledge required for continued understanding of place and position must be updated.

To learn about the organization, the person must exchange organization specific information with others. When they first join, others in the organization transfer information to them to facilitate their organizational learning. The organizational learning is not necessarily classical learning, but instead a process

to share or transfer knowledge. Each agent is previously aware of the knowledge structure and process required to interact with other agents in the organization.

Modeling agent organizations using the inspiration of human organizations, as is commonly done, the designer must create the formalities and implementation to allow the transfer of information between agents. We look at interaction as a basic exchange of information between two agents, but can be extended to any number of agents belonging to an organization. The goal of the exchange is to maintain a state of perfect information between all agents. Perfect organization dictates that all agents must have identical organizational knowledge. The trick is that during transition, initial organization or reorganization, the information will change for at least one agent. That agent then has to insure that all other agents must receive the same knowledge changes. Differences in knowledge between agents will cause potential divergence in goals or roles played by the organization. The effect of bad information, in an agent organization, is much the same as if it were a human organization.

Our logic-based approach to this problem stems from some fundamental work, such as work by Su et al. [13]. Deitterich expressed the need to establish a useful level to approach knowledge, both for storing and learning or exchange [4]. Gordon and Subramanian augmented the approach to knowledge, by establishing the need for finer grain tuning of logic [8]. Baader provides a more general approach to the need for knowledge representation [2]. The basis of these works establishes the fit of logical representations for organization knowledge storage. In this work, the logical representation of organization will mirror the structural representation of organization.

In terms of knowledge sharing, Dignum and Dignum [5] indicate the shift from sharing to collaboration. That is key for this effort, although the basis of our model restricts the knowledge exchanged to a very specific set, lending to the strategy shown by Soller and Busetta [14] to develop a shared understanding between agents. While not strictly a default set of rules, as described by Rybinski and Ryzko [12], our logical structures are standardized, to simplify the body of knowledge exchanged.

An assumption, for this research, is agents are cooperatively participating in an organization where common goals are paramount. Individual agent goals and motivations are not above the needs of the organization. The difference is our approach reflect separating the constitution of the organization from a strictly structural concept. Organizations are normally perceived as structural, with components and relationships. Our approach considers an organization as a mental image of a structural entity. This approach allows better scalability and computation of new states.

In this paper, we describe and demonstrate the organization knowledge exchange between agents belonging to the same organization. In section 2, we describe model elements and processes for sharing of knowledge between organization agents. In section 3, the implementation of this system is described. Results of the implementation are described in section 4. Section 5 explains opportunities for further work.

2 Organization Knowledge Sharing

In this section, we describe the basic structure required to model organizational information to facilitate exchange of information. The foundation of exchange is an organization model [9, 3]. The agent structure is shown first followed by the structural, state and transitional elements of our organization model. Once an organization model is described we extend the model to include processes of exchange. Finally, the model and processes are integrated to show the overall formalities of knowledge transfer between agents.

2.1 Agent Core Composition

Before looking at the specific elements of organization, we must first show the overall structure of an agent. An agent is comprised of several knowledge cores, as shown in Fig. 1. An agent has three knowledge cores which are the *organization core*, *communications core* and *task core*. Each core represents the knowledge held by an agent in an area. For example, the communications core represents all knowledge required to communicate with all other agents to which the agent has access. The task core represents it knowledge of each of the capabilities possessed by the agent. An agent may have numerous task cores. While all three cores compose an agents knowledge, the organization core is the one of most interest in this research, and will be the focus of the discussion. This core represents all of the knowledge of the organization in which an agent participates. In simple terms, it is its own internal organization chart defining all components, structural relationships and state relationships of the organization. As the structures contained within the core are discrete, all agents work with an even base in which to share organization knowledge.

2.2 Organization Model Elements

Our organizational model (O) has a structural model, a state model and a transition function [9], described as:

$$O = (O_{structure}, O_{state}, O_{transition}) \tag{1}$$

Before approaching the details of information exchange, we must examine the *structural* and *state* elements of our model. The component and relationship elements are represented as the stored knowledge to be exchanged.

Structure The *structure* is defined by:

$$O_{structure} =< G, R, L, C, ach, rel, req, sub, con > \tag{2}$$

where *ach* is *achieves*, *rel* is *related*, *req* is *requires*, *sub* is *subgoal* and *con* is *conjunctive*, respectively. G describes the set of *goals*, R is the set of *roles*, L is the set of *laws* or *rules* required, and C is the set of *capabilities*. The

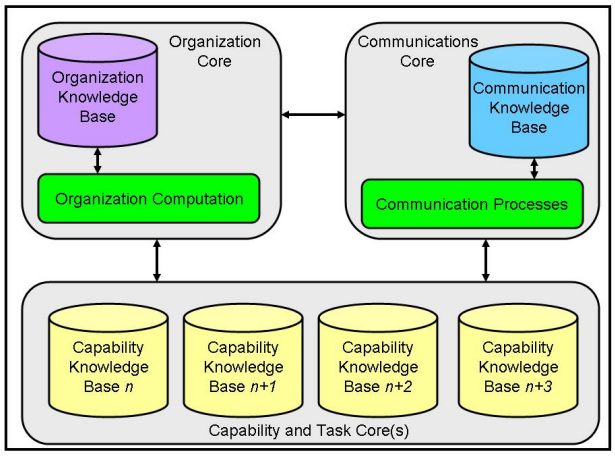

Fig. 1. Knowledge Cores of an Agent

organization structure also contains a set of relations. The achieves relation, $achieves : R, G \rightarrow [0..1]$, states the relative ability of a *role* to satisfy a given *goal*. The *related* function $related : R, R \rightarrow Boolean$ exists only if two *roles* are related. *Roles* require *capabilities* to satisfy a set of *goals* and this is captured by the $requires : R, C \rightarrow Boolean$. The organization may contain subgoal relationships $subgoal : G, G \rightarrow Boolean$. The conjunctive relationship between *goals* is $conjunctive : G \rightarrow Boolean$.

State The *state* is defined by:

$$O_{state} =< A, possesses, capable, assigned, coord > \tag{3}$$

where an A defines a set of *agents* available to participate in the organization. There are several relationships in the state element of the organization. An *agent* capable of playing a certain role possesses the necessary *capabilities* described by the possesses relation, $possesses : A, C \rightarrow [0..1]$. An *agent* is capable of playing a *role* in the organization as described by the capable relation, $capable : A, R \rightarrow [0..1]$. The assigned relation, $assigned : A, R, G \rightarrow [0..1]$, is used to match the best *agent*, *role*, *goal* combination that maximizes the capability of the organization. The coordination relation, $coord : A, A \rightarrow Boolean$, allows a relationship between two *agents*.

Transition Transition is the main topic of knowledge exchange as transition requires that knowledge be exchanged by all agents participating in the organization. There are two specific transition processes, *initial organization* and *reorganization*. From organization $state_0$ to $state_1$ is initial organization. All

other transitions are reorganization. Transition is expressed by:

$$O_{transition} = (O, \Phi, \delta, s_n, S_{optimal}, S_{possible}, S_{final}) \tag{4}$$

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 an organization can only exist as a single entity or instance, the current state s_n is always a unique value [10].

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

$$\delta : O \times \Phi \times S \Rightarrow S \tag{5}$$

So, the transition function is of the form:

$$\delta(O, \phi, s_n) \Rightarrow S' \tag{6}$$

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 $S_{optimal} \subseteq S_{possible}$, $S_{optimal} \subseteq S'$ and $S_{final} \subseteq S_{possible}$.

Transition Properties Transition properties Φ represent stimuli that can change the organization. They are represented in logical format which capture the generic nature of what they can define. In general terms, an organization will need a set of properties Φ, for example, *capabilities* or *agents*, which can be the stimulus of transition. An individual property $\phi \in \Phi$ 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 an agent, the reentry of an agent, or the addition of a new agent.

Each domain problem, represented by knowledge in a task core, may create a number of task specific transition properties. We will first show general, abstract properties and then discuss specific properties. These general properties can be instantiated to fit specific examples. Some general transition properties are:

1. Loss of an agent participating in the organization
2. An agent loses capability required to play some role
3. A new agent becomes available
4. Capability of an agent increases
5. Capability of an agent decreases
6. A goal is removed
7. A goal is added

8. A goal is relaxed (changed)
9. Change in goals to roles achieves relationship
10. Change in role to capability requires relationship

Changes in organization structure and participants will drive transition activities. Transition properties can be triggered internally or externally. The general transition properties can be split into properties that are external and those that are internal.

Transition Predicates A transition predicate is a formalization of a transition property. The formalization of transition predicates enables the exchange of information. Transition predicates can also be expressed as $\Phi = \{\phi_1 \dots \phi_n\}$. In general, Φ can be expressed as a set of standard, abstract predicates, $\Phi = \{\phi_{lose}, \phi_{add}, \phi_{change}\}$, where ϕ_{lose} is the abstract property dealing with loss, such as losing an agent from the organization or an agent losing capability to play a role. The add property ϕ_{add} describes the action when an agent becomes available for invitation to the organization. The change property ϕ_{change} can either be an increase or decrease and further specializes the change predicate, $\phi_{change} = \{\phi_{decrease}, \phi_{increase}\}$ [11].

The *primitive predicates* exhibit polymorphic behavior as each can be applied to different organization elements to capture different properties.

1. Loss of an agent participating in the organization $\phi_{lose}(a)$
2. An agent loses capability required to play some role $\phi_{lose}(c_i, a)$
3. A new agent becomes available $\phi_{add}(b)$
4. Capability of an agent increases $\phi_{increase}(c_i, a)$
5. Capability of an agent decreases $\phi_{decrease}(c_i, a)$
6. A goal is removed $\phi_{lose}(g)$
7. A goal is added $\phi_{add}(g)$
8. A goal is relaxed (changed) $\phi_{change}(g)$
9. Change in goals to roles achieves $\phi_{change} achieves(r_i, g_j)$
10. Change in role to capabilities $\phi_{change} requires(r_i, c_j)$

Primitive predicates can be used to formalize single properties. If there is a loss of an agent participating in the organization, it can be formalized as the predicate $\phi_{lose}(a)$. An agent a losing some capability can be captured as $\phi_{lose}(c_i, a)$. *Complex predicates* represent the combination of simple predicates logically constructed using common *and* ($\wedge$), *or* ($\vee$) and *not* ($\neg$) relations.

Some predicates will encompass others, but in some cases two properties can be successfully combined to form a single property of transition. In the case that an agent exits an organization, it can be reasoned that all capability of that agent will also exit. So combining the two previous predicates of losing an agent and losing a capability by an agent are redundant, in respect to the capability predicate $\phi_{lose}(a) \wedge \phi_{lose}(c_i, a)$. In another situation, an organization may lose two agents simultaneously. If agents a and b both leave, we can capture that by $\phi_{lose}(a) \wedge \phi_{lose}(b)$, where one primitive predicate does not contain the other.

As there are primitive and complex predicates, another perspective shows *component* and *relationship* predicates. A component predicate is defined as a predicate where the property relates to a component of the organization, such as an agent being added or a goal being deleted. A relationship predicate is defined by a property where a relationship between two components is added, deleted or altered. Relationship predicates can be primitive. Component predicates must be complex as the component must collaborate with a relationship to connect to the organization.

2.3 Exchange Processes

A model is not necessarily sufficient to completely explain the exchange of knowledge. The process must also show how the agents interact to share the information. This definition only describes the basic mechanics of the exchange. It must be further explored to answer questions on what basis is information exchanged. Will the information be shared with anyone who asks? Will the information be shared with all agents? Will it be shared with agents who do not specifically ask for it? These questions not only pose a set of philosophical queries, but also pose some practical problems in exchange. Automatically sending data to an agent that does not need it, as it already possesses the information, is wasteful in terms of resources.

Our approach to knowledge exchange is similar to the *mind-body problem* of Descartes. In the mind-body problem, the mind is differentiated from the matter of the body. The knowledge of an organization, which resides in the individual mind of each agent, within the organization, is different than the physical manifestation of the organization. Each agent carries an image of the organization with all components and relationships. The key is for all agents to have the same image of the organization, in other words, perfect information.

The basic premise is that when each transition occurs, all agents need to be updated with the current organization knowledge. When a human organization requires change, a decision is made and the change is then communicated by the decision maker to those affected. As with human organizations, a single agent will receive the change, ϕ and propagate the change to each of the other agents in the organization.

There exists a risk of a transition property not correctly propagating from the sending agent to the receiving agent. If this occurs, the receiving agent will not compute a new organization and will be different than those agents who successfully received the message. If for some reason, such as an agent being deleted, another agent will sense the agent loss and update the others. It can also be said that each of the others can self update in the event of a loss, but questions whether each is required to recompute. A key goal is to minimize the amount of information transferred for each organization transition.

2.4 Integration

Each agent optimally has the same organization knowledge. This supports the premise that all agents operate on full information. Fig. 2 shows an organization of 4 agents $\{agent_1, agent_2, agent_3, agent_4\}$. $Agent_1$ receives a transition property from either an internal or external force. $Agent_1$ then propagates the predicate to $agent_2$, $agent_3$ and $agent_4$. The organization core represents the part of the *mind* of the agent concerned with where it fits in the organization. The agents themselves represent the physical manifestation, or the *body*.

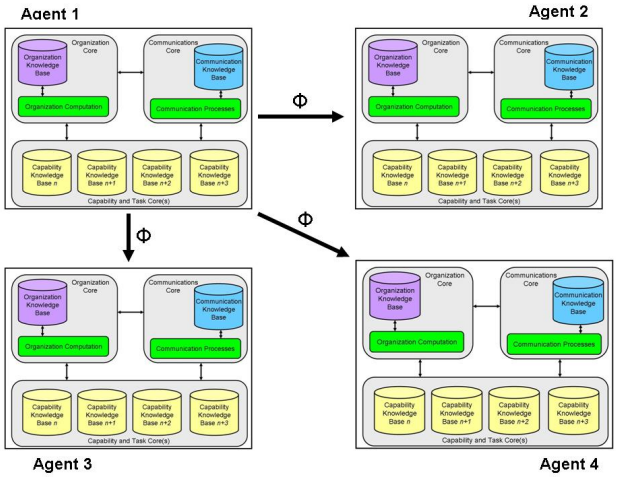

Fig. 2. Knowledge Transfer

3 Implementation

The organization formalisms and knowledge exchange processes have been implemented to complete this work. The implementation is a combination of Java used as the main development platform with *JESS* utilized to implement the knowledge bases. *JESS* has a natural relationship with Java as described by Friedman-Hill [6] and utilizes the rete algorithm of Forgy et al. [7] and Albert [1] shows the computational fit for this algorithm applied to this technical problem.

In this section, the implementation of the structural and state elements as logical constructs in *JESS* are discussed. Each component and relationship are expressed as logical predicates. This logical expression represents the *mind* of the organization. Each predicate is sent to each agent in the *body* and then each agent recomputes a new organization image within their own structure. Thus the *mind* of each agent in the *body* recomputes its own like image of the organization after each change. All *JESS* logical functions are constructed with

rules and facts, based on templates. The organization object is then embedded inside a Java shell for integration with the body of the organization, written in Java.

3.1 Structure and State

Each predicate of the organization model's structure and state can be directly represented by a template in *JESS*. For example, the structural templates are:

```
(deftemplate goal (slot goal))
(deftemplate goal (slot role))
(deftemplate goal (slot capability) (slot score))
(deftemplate achieves (slot role) (slot goal) (slot score))
(deftemplate related (slot role) (slot role))
(deftemplate requires (slot role) (slot capability))
(deftemplate subgoal (slot goal) (slot goal))
(deftemplate conjunctive (slot goal))
```

The state templates are:

```
(deftemplate agent (slot agent))
(deftemplate possesses (slot agent) (slot capability) (slot score))
(deftemplate capable (slot agent) (slot role) (slot goal) (slot score))
(deftemplate coord (slot agent) (slot agent))
```

So a ϕ property of adding a goal, $\phi_{add}(g)$, will exist in JESS as $(goal(goalg))$ added by a rule in *JESS*.

3.2 Transition and Exchange

Each agent is a complete independent entity communicating via TCP/IP sockets. All knowledge is exchanged using Java via networking between distributed agents. This technology is employed specifically for loss reduction and error handling abilities in relation to knowledge exchange.

There are three specific change categories which can effect the exchange process. The first is change to a structural element of the organization. Examples of structural change are to add or lose a goal. The second is the change in a state element. An example of state change is an agent gaining or losing capability, thereby requiring a computation of the organization. The third option is the loss or gain of an agent. Each of these changes will be described using the transition predicates and exchange of *JESS* constructs.

Structural Change If a goal is added or lost, the agent first notified must send a message to all others to retain the state of perfect information. If a transition predicate $\phi_{add}(g_n)$ is received by agent x, a_x, then a message must be propagated to all other agents to add the new goal, as a fact. For each organization knowledge core a new fact is added.

State Change If there is a state change such as the capability of agent x increases $\phi_{increase}(c_i, a_x)$, then that agent will propagate the new fact to all other agents.

Agent Change When there is no change to the collection of agents, within the organization, it is straightforward to propagate the new information to all agents. When an agent is gained or lost, the matter of communication takes on a new level of complexity. When an agent is lost, one or more of the agents remaining must recognize the loss. One of the agents must define a predicate $\phi_{lose}(a_x)$, create the update and send to all agents. If an agent is gained to the organization, $\phi_{gain}(a_x)$, the new fact that an agent has been added is sent to all agents by one of the agents, already in the organization.

4 Results

We must first distinguish between results split by the two transition processes, initial organization and reorganization. The result indicates the initial organization is computationally more intense and is based on the number of components and relationships. Since it will only be computed once in each organization's life, its effect is discounted. Reorganization is much smaller, due to the incremental nature of only having to recompute around new components and relationships of the ϕ predicate. If ϕ is quite large, it may alter the computational intensity. For example, if the number of components and relationships in ϕ is equal or greater than the existing organization, reorganization may be computationally large.

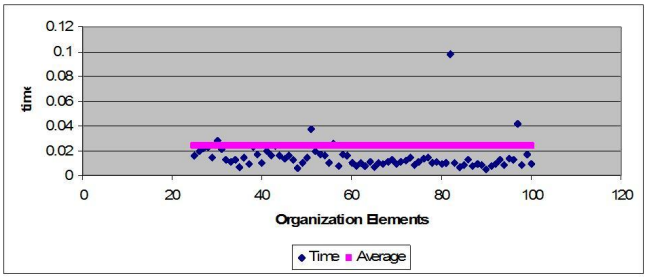

Fig. 3. Results

The computation of a transitioning organization differs from an initial organization to a reorganization. In a strictly structural context, initial organization and reorganization do not differ a great deal. In our mind body approach the difference is significant. For a small organization size of 10 goals, 10 roles, 10 agents and complete relationship set, the time for initial organization is 0.03219 seconds. The time for a reorganization based on one new component and all relationships is 0.01754 seconds. Fig. 3 shows the time to compute a transition against the size of the organization, in elements. The initial organization used in

this analysis has 10 organization components, such as roles, agents or goals, and 15 relationships between those components. The total number of components, on the lower end, is 25. The data shows the time to compute the transition going from 25 components to 100, which is beyond a trivial organization. The transition process is based on computing an optimal organization configuration. The key is that the time to recompute is not significantly different for the larger organization. This is due to the incremental nature of the computation process. This indicates use of this method, is at least initially, scalable.

If we compare this timing to another result by Zhong it shows the difference. In Zhong's research[15], based on a similar model, using only the constructive version of the structural model algorithm to transition, the results of a structural computation yields two interesting points. First, the structural model transition algorithms grows at a fast rate as the number of organization components grow. Secondly, the ability to scale to large organizations will be significantly hindered by a strictly structural approach. This indicates as the size of the organization grows, the difference between our approach and a more structural-based approach will grow, in terms of time to compute.

Instantiating an organization and its transition processes in terms of a mind-body approach has advantages over a strictly structural computational approach. While there are also a few disadvantages, these are overcome by the positives.

Computation minimization is the best result of this approach. While larger, more complex organizations must be tested, the early results show promise. The computation is performed locally and in parallel, which allows the transition process to be completed more rapidly. The intent is for each agent to work with perfect information and each agent will have the same organization image, without transferring the entire structure each transition cycle. The rate of message growth is small. Even with a large change, all computation is local. This will allow a near linear growth rate during organization augmentation. This will reduce temporal computation problems in transition processes.

There are a few negative side effects of this approach. Perfect information requires that information is transferred from agent to agent without interruption or error. If there is a transfer loss, the synchronization of the organization image maps will suffer. Recovering from loss, during exchange, is key for the design. There must be synchronization allowing each agent to recompute simultaneously with all others. If there are lag times, it can create temporal problems in transition.

5 Further Work

The initial algorithm will be extended to a complete distributed model and a hybrid model, which allows an integration of command mode and complete distributed behavior. Larger organizations will be theoretically analyzed and empirically analyzed to determine performance over large, distributed agent organizations and societies. The scalability question will be further developed to see if there is a breaking point of the design.

References

1. Luc Albert, Average Case Complexity Analysis of Rete Pattern-match algorithm and Average Size of Join in Databases. Rapports de Reserche, No. 1010. Institut National de Reserche en Informatique and Automatique, Rocquencourt, France, April, 1989.
2. Franz Baader, Logic-based Knowledge Representation. Artificial Intelligence Today, Recent Trends and Developments, no. 1600. Springer-Verlag, M. Wooldridge and M. Velosa (eds.), 13-41, 1999.
3. Scott DeLoach, Eric Matson. An Organizational Model for Designing Adaptive Multiagent Systems. Agent Organizations: Theory and Practice at the National Conference on Artificial Intelligence (AAAI-04), July 25-29, 2004, San Jose, CA.
4. Thomas Dietterich, Learning at the Knowledge Level. Machine Learning, 1:287-316, 1986, Kluwer Academic Publishers, Boston, MA, USA.
5. Virginia Dignum, Frank Dignum.The Knowledge Market: Agent-Mediated Knowledge Sharing, Lecture Notes in Computer Science Springer Berlin/Heidelberg, vol. 2691, Multi-Agent Systems and Applications III: 3rd International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2003, Prague, Czech Republic, June 16-18, 2003.
6. Ernest Friedman-Hill. JESS in Action: Rule Based Systems in Java. Manning Publications, Inc., Grennwich Connecticut, USA, 2003.
7. Charles Forgy, Allen Newell, Anoop Gupta. High-Speed Implementation of Rule-Based Systems. ACM Transactions on Computer Systems, Vol. 7, no. 2, May 1989, pages 119-146.
8. Diana Gordon, Devika Subramanian, A MultiStrategy Learning Scheme for Agent Knowledge Acquisition, Informatica, 17:4, 1993.
9. Eric Matson, Scott DeLoach. Organizational Model for Cooperative and Sustaining Robotic Ecologies, Proceedings of Robosphere 2002 Workshop, NASA Ames Research Center, Moffett Field, California, November 14-15, 2002.
10. Eric Matson, Scott DeLoach, Formal Transition in Agent Organizations, IEEE International Conference on Knowledge Intensive Multiagent Systems (KIMAS '05), Waltham, MA, April 18-21, 2005.
11. Eric Matson, Raj Bhatnagar. Properties of Capability Based Agent Organization Transition. 2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT-2006), Hong Kong, December 18-22, 2006.
12. H. Rybinski and D. Ryzko. Knowledge Sharing in Default Reasoning-Based Multiagent Systems, IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2003. October 13-16, 2003, pp. 576 - 579.
13. Kaile Su, Xudong Luo, Huaiqing Wang, Chengqi Zhang, Shichao Zhang, Qingfeng Chen. A Logical Framework for Knowledge Sharing in Multi-agent Systems, Lecture Notes in Computer Science, vol. 2108, Springer Berlin/Heidelberg. Computing and Combinatorics : 7th Annual International Conference, COCOON 2001, Guilin, China, August 20-23, 2001,
14. Amy Soller and Paolo Busetta. An Intelligent Agent Architecture for Facilitating Knowledge Sharing Communication, in Proceedings of Workshop on Humans and Multi-Agent Systems, International Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, 2003, pp. 94-100.
15. Christopher Zhong, Scott A. DeLoach. An Investigation of Reorganization Algorithms. Proceedings of the International Conference on Artificial Intelligence (IC-AI'2006). June 2006, Las Vegas, Nevada, CSREA Press, 2006.