AN AGENT-BASED APPROACH TO SUPPORT PERFORMANCE MANAGEMENT FOR DYNAMIC AND COLLABORATIVE WORK

Nora Houari[1] and Behrouz H. Far[2]
Department of Electrical and Computer Engineering, University of Calgary, Calgary, Alberta, Canada

Keywords: Multiagent support systems, modelling, roles, integration, collaboration, coalition.

Abstract: This paper describes a comprehensive multiagent based modelling approach for collaborative and dynamic organizational roles support. The method is a role centred one, where agents collaborate to assist human users through collaboration within the same role, with other roles of the same team, as well as roles of different teams that share tasks dependencies. Agents in the system are not restricted by predefined schemes they can join and/or leave the coalition. We identify the key elements of the role model as rules, agents, and relationships. Our role model integrates both the operational functionalities and the performance management towards specific goals, the former involve human individuals in the loop, whereas the later are performed by software agents to assist in monitoring, control, and adaptation of the performance in dynamic organizations.

1 INTRODUCTION

Software (agent-based) information and support systems have a significant influence on the product, and services, an organization offers. Organizations increasingly rely on computational systems, these systems need to be designed, constructed, and integrated as such to support the feature that justifies this reliance. At the same time, they need to reveal the changes that occur in the environment, such as operational tasks changes, and individuals' reallocations. Hence, this appeal for an explicit examination of knowledge which is generally implicit, for example, who is doing what with whom, etc.

In this paper, we propose a unified and adaptable organizational model for integrating the operational functionality activities and the performance towards specific goals, as well as their dynamics. The agent assistance mechanisms described in this research provides proactive monitoring and control, in addition to keeping track of current events (e.g. task milestones delayed), the agents can automatically join and/or leave their role coalition. Such explicit exploration of this knowledge supply human individuals with an actual view of their progress and allow them take appropriate actions accordingly.

The contribution of this work is threefold: first, we provide an integrated structure of a role that enables us to combine operational activities with performance indicators, second we develop a formal framework for supporting individuals in organization, and third, we outline the process of our approach.

The technical focus of this paper is on developing organizational modelling mechanisms for collaboration and coalition dynamics. Although the presented computational support system was built for software engineering organization, we believe that the method is general enough to be applied to any collaborative environment involving distributed tasks and teams.

This paper is structured as follows: Section 2 briefly discusses the problem and a case study. In Section 3, we illustrate the specification of the role concept in our modelling approach. Section 4 describes our organizational formal model, and in Section 5 we delineate the model process steps. Section 6 summaries related works, and finally conclusions and perspectives of future works are provided in Section 7.

Houari N. and H. Far B. (2007).
AN AGENT-BASED APPROACH TO SUPPORT PERFORMANCE MANAGEMENT FOR DYNAMIC AND COLLABORATIVE WORK.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - AIDSS*, pages 178-184
DOI: 10.5220/0002411801780184

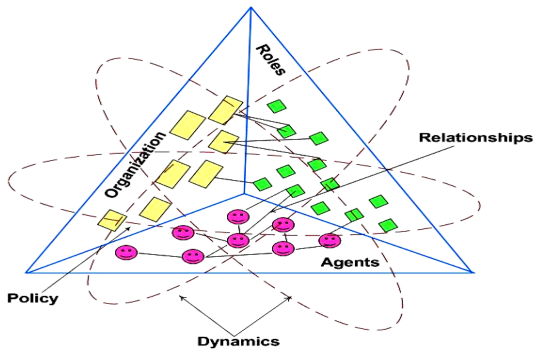

Figure 1: The dimensions of our model.

2 PROBLEM STATEMENT

The application case study considered in this paper is a software engineering organization dealing with the development of software projects development. Software projects are usually multi-team and multi stage development. They can belong to one organisation or be shared among partners. Due to the wide range of distributed data, information and knowledge, it is impossible to effectively locate, keep track, and adapt to changes acquired in these projects. Individuals in these organizations are involved in organizational and projects tasks, they may be reassigned tasks (added and/or substituted), the design and implementation of their support system must reveal these dynamics, and be able to provide awareness about who is doing what when and who is interacting and collaborating with whom.

The aim of this work is to provide a support system that assists human users performing their roles by taking care of the performance management of their roles execution. Since the processing capabilities of human are limited; the distribution of information, resources, and activities among organization members must be set up to guarantee that management of the tasks of each individual remains below his/her capability. By delegating performance management to agents we are (semi)-liberating human individuals from routine tasks and consenting them to focus on processing and execution of the task.

Figure 1 depicts our organizational model dimensions that contain: the organization and its policy, roles, agents, their relationships and dynamics.

3 THE ROLE SPECIFICATION

We define a role as “a collection of duties and rights” (Biddle and Thomas, 1966). Duties represent the tasks and interactions that the role is obligated to perform, whereas rights represent the permissions to utilize information entities to perform tasks or interactions. Thus, the concept of role in our system is essentially an abstraction on one hand, for the tasks that are necessary to be performed and/or the interactions that need to occur with other roles to achieve an individual goal, and on the other hand, the information that needs to be accessed or will be generated during the course of performance of those tasks/interactions, and the knowledge that is needed for the successful execution of tasks and interactions towards achievement of the goal.

We identified two types of roles: an organizational *abstract* role and a *conceptual* role.

An abstract role refers to operational functionality tasks and their strategic performance goal; by operational we mean processing and communication towards the execution of the tasks, strategic performance on the other hand deals with measuring the performance of the goal. In our system, an abstract role is assigned to one or more agent (human, software), according to their capabilities, an agent may carry more than one abstract role. Figure 2 delineates the capabilities of an organizational abstract role. Any abstract role requires two types of capabilities: (i) the processing capabilities theses are execution, communication, decision making, etc. (ii) and strategic capability these are the performance management of time, product/service quality, cost, effort etc.

The second type of role we introduced in our model is a conceptual role, which describes a set of capabilities and responsibilities necessary to perform an activity. A conceptual role, often involves relationships (i.e., collaboration, conflict resolution, etc) among abstract roles.

The main idea of our work is to integrate both capabilities of an abstract role in a coherent and unified form (see Figure 3). Our technical approach to this integration is based on building the collaboration and coalition of agents for a dynamic conceptual role (activity). The objective here is that human will be able to assign the performance management of their activities to software agents so that they can fully focus their cognitive capabilities on processing, decision making, etc. Our approach allows an effective shift and delegation of responsibilities to agents playing an abstract role as well as the dynamics in a conceptual role.

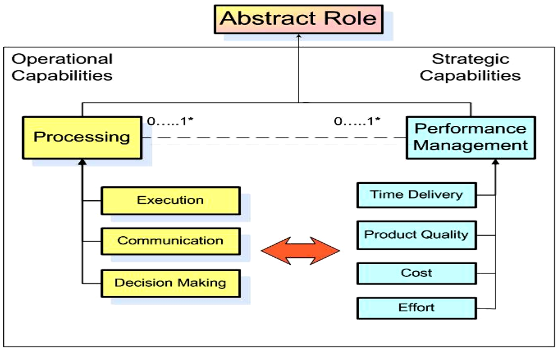

Figure 2: A Dynamic abstract role.

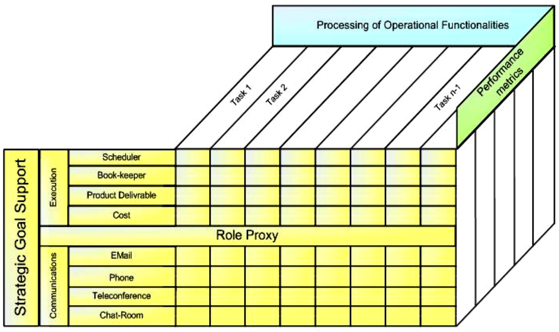

Figure 3: The components of an abstract role.

4 THE FORMAL MODEL

In order to implement our approach that integrates organizational processing functionalities and performance management in a dynamic environment, we first identified key elements such as agents, collaboration, environment and goals in a formal model. Based on this formalism, we highlight the essence of our multiagent support system (in terms of its components and its process). Our proposed role model is a tuple < Rules, Agents, Relationships>. In the next subsections we review briefly these components.

4.1 Rules

Organizational rules are used to constrain one or more of the following: (i) the assignment of agents to roles, (ii) role's capabilities requirement, (iii) role's relationships, and (iv) role's goal achieve. These rules must satisfy the accuracy and functionality requirement, and often are application and domain dependent.

4.2 Agents

Software agents are computational entities that run in an environment, sense it and act upon it to achieve a specific goal. Each agent must possess the required capabilities to play a particular role. We identify two categories of agents the operational and the strategic agents. Operational agents deal with processing and communication to execute the tasks, in our current version of the system they are carried by human individual; however he/she can delegate those routine tasks to software agents as in (Chalupsky et al., 2005). Strategic agents, on the other hand are an added value to the operational agents to keep an eye on the mission goal performance, this category of agent uses metrics for each specific goal.

4.3 Relationships

Relationships refer to instance interaction relationships, the dynamics of activities, the roles, and agents performing them towards achieving a mission goal. In our system we consider the collaboration for activities among team members, and the dynamic in joining and leaving the activity.

5 THE MODEL PROCESS STEPS

The process for building our support system consists of the following steps:
1. *Identify the performance indicators (PIs)* for a particular goal; for each operational task we identify the performance metrics towards a particular goal.
2. *Assign agents:* define the required agents to support each task, and their grouping for an abstract role.
3. *Identify the type of interactions:* Interactions are constrained by organization law, the application specific process, and individual privacy.
4. *Build a collaboration algorithm* between teams' members leading to accomplishment of the goal.
5. *Construct the* coalition method that automatically manages the performance in the case of an agent leaving or joining the coalition.

Because of the limitation of the paper length, details of some steps are provided in (Houari and Far,. 2007). The individual process steps are briefly summarised in the subsequent sub-sections.

5.1 Performance Indicators Identification

For any given task, we identify the key performance indicators (PIs) to accomplish a particular goal. Details of an example of metrics for achieving the goal of delivering work on time were presented in (Houari and Far, 2007).

5.2 Assignment of Agents

We identify two categories of agents to support operational functionalities. The first category support the execution, and the second the communication. In this work we consider only the formal. The later (communication) was addressed by works like (Chalupsky et al., 2005). Future extension of this work will integrate both categories of support. For an abstract role we assigned five support agents: H_P: *Human Proxy;* R_S: *Role Scheduler*; R_B: *Role Bookkeeper*; R_D: *Role product Deliverable*; R_C: *Role Cost.* There is also an additional agent support for the team leader: *Team Leader Proxy* (H_{TLP}).

This set of agents capture the information of all the tasks assigned to the role. We set the human proxy agent to consolidate the other agents of the role, and allow it as the only agent to talks to the human user.

5.3 Identify the Interactions

In our system, we identified three types of interactions (see Table 2 and Figure 4).

Table 2: Agent interaction types.

	Between	Definition
1	$H_{P1} \propto H_1$	Interaction between human proxy agent and the human individual it supports
2	$R_{i1} \propto R_{j1}$	Interaction among agents that support the same human individual
3	$H_{P1} \propto H_{P2}$	Interaction between one human proxy and another human proxy

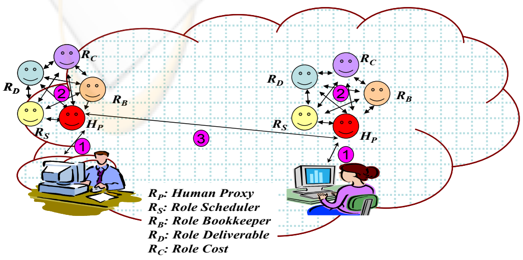

Figure 4: The types of agent interactions.

5.4 Building the Collaboration

The set of an individual's assigned project and organization tasks are specified by P_{Ta}, O_{Ta} respectively. Team's project and team's organization tasks are denoted by P_{TT} and O_{TT} correspondingly. The members of the team are T_H and task's dependencies function is defined by tdf. Each individual is assisted by five agents: the proxy, scheduler, bookkeeper, product deliverable, and cost ($H_P;R_S;R_B;R_D;R_C$) respectively (step 2). These agents are assigned the individual's organization and project tasks, and their dependencies (steps3). For every task (project or organization) we allow the interaction among the supporting agents of the individual role according to the interaction types identified in section 5.3 (steps4-5). If the task has no dependencies we add the collaboration with the team leader proxy (step 6). If the task has dependencies (i.e. carried by more than one individual) we allow interaction between proxy agents of the individuals sharing the task as well as between the agents of the teams' leaders (step7). The next level of collaboration, deals with the team leader (step 11-14), where he is assisted by a proxy, scheduler, product deliverable, and cost support agents (step12). The team's project and organization tasks of the team members are assigned to the supporting agents of the team leader.

Figure 5 outlines the pseudo code of the collaboration of agents within the role and between roles.

Algorithm 1: The Collaboration Algorithm
Inputs: $P_{Ta}[t_1..t_n]$; $O_{Ta}[t_1..t_m]$; $P_{TT}[t_1..t_q]$; $O_{TT}[t_1..t_l]$; $T_H[H_1..H_k]$; tdf
Outputs: Human's (H_a) support agents' collaboration binding
1. **for** each individual
2. $H_a := <H_{Pa}; R_S; R_B; R_D; R_C>$
3. $<H_{Pa}; R_S; R_B; R_D; R_C> := [P_{Ta}, O_{Ta}, tdf]$
4. **for** $\forall [t_i, t_j] \in <P_{Ta}, O_{Ta}>$
5. $(H_{pa} \infty H_a) \wedge (H_{pa} \infty R_S \infty R_B \infty R_D \infty R_C)$
6. **if** $(t_i \neg\exists\ tdf)$ **then** $(H_{Pa} \infty T_{LPa})$
7. **else** $(H_a \infty H_k) \wedge (H_{Pa} \infty H_{Pk}) \wedge (R_{ai} \infty R_{ki}) \wedge (H_{Pa} \infty T_{LPa}) \wedge (T_{LPa} \infty T_{LPk})$
8. **end if**
9. **go** to 11
10. **end for**
11. **for** $\forall [t_i, t_j] \in < P_{TT}, O_{TT}>$
12. $H_{TL} := < H_{TLP}; R_S; R_D; R_C >$
13. $<H_{TLP}; R_S; R_D; R_C> := [P_{TT}, O_{TT}, T_H]$
14. **end for**

Figure 5: The pseudo code of the collaboration.

5.5 The Coalition Dynamics

The last step in the process of our computational support system consists of building the conceptual role coalition dynamics mechanism. Our approach in building the dynamic coalition is rule driven; getting agents involved in tasks triggers automatic activity coalition creation, deletion or integration. Automatic coalition management enhances the dynamics of collaboration: agents are not restricted by predefined schemes and they can join or leave the coalition.

5.5.1 Formal Notation

A role in our computational system is a *task* centred one. A task represents anything which has a role involvement, attributes, properties and role means of interactions. Our conceptual role coalition is founded on the notion of *activity* that associates a goal with a set of individuals and with a set of tasks. According to the following semantics; the individuals involved in the activity have a common goal to achieve through the accomplishment of the tasks of the activity.

Performing a task requires a binding to it. A conceptual role coalition dynamically binds a set of tasks and agents from a given activity. This binding is normalized by rules specified bellow and managed by an algorithm illustrated through an example in the next subsection. An agent a assigned to activity V performs a task t from V, if and only if, he has a binding on t.

We call $c_v(A, T)$ a conceptual role coalition of the activity V binding a set of agents A to a set of tasks T from V, such that every agent of A is assigned to a task from T. The conceptual role coalition combines the collaborations of the agents with the tasks. A given coalition shows who is sharing tasks with whom, and which tasks are shared. The conceptual role coalition is based on the following rules:

Rule 1: The set A of agents of the conceptual role is maximum that is A contains all the members of the activity bound on any of the tasks of the activity.

Rule 2: The set T of tasks of the conceptual role is maximum that is T contains all the tasks of the activity that the members of the conceptual role are sharing (a task is shared if and only if, more than one binding exist)

To ensure rules 1 and 2, we define the following operation.

Operation: Two conceptual roles can be merged if and only if they have the same agent members or they have the same tasks.

The result of merging two conceptual role coalition $c_{1V}(A;T)$ and $c_{2V}(A';T')$ is an other conceptual role $c_{3V}(A \cup A'; T \cup T')$.

For example $c_{1V}(\{a_1,a_2\};\{t_1,t_2,t_3\})$ is mergeable with $c_{2V}(\{a_1,a_2\};\{t_4\})$ and the result is $c_{3V}(\{a_1,a_2\};\{t_1,t_2,t_3,t_4\})$. Similarly $c_{1V}(\{a_1,a_2\};\{t_1\})$ and $c_{2V}(\{a_3,a_4\};\{t_1\})$ are mergeable in $c_{3V}(\{a_1,a_2,a_3,a_4\}; \{t_1\}$.

The binding of agents and tasks control the evolution of the coalition set of an activity but must guarantee rules 1 and 2.

5.5.2 Illustrative Example

We demonstrate the coalition dynamic method through an example, in this example an activity V has four agent members: a_1, a_2, a_3 and a_4 that are responsible for t_1, t_2, t_3, t_4 and t_5 in this way:

- a_1, a_2 and a_3 are collaborating on the tasks t_1, t_2, and t_3.
- a_1 and a_2 are collaborating on t_4.
- a_4 is active on t_5.

This is represented by three conceptual role coalitions $c_{1V}(\{a_1,a_2,a_3\};\{t_1,t_2,t_3\})$;$c_{2V}(\{a_1,a_2,\};\{t_4\})$ and $c_{3V}(\{ a_4\};\{t_5\})$(see Figure 4(a)).

5.5.3 The Agent Coalition Binding Scheme

If the agent a_3 will be involved in task t_4 , he has to bind to t_4, then the conceptual role coalition involving t_4 must show that a_3 joins the group of members responsible for t_4, so a_3 will be added to c_{2V} which becomes $c_{2V}(\{a_1,a_2,a_3\};\{t_4\})$. However this will break the rule 2, because conceptual role coalitions c_{1V} and c_{2V} have the same members. The two conceptual role coalitions have to be merged; and the coalition set becomes: $c_{1V}(\{a_1,a_2,a_3\};\{t_1, t_2, t_3, t_4\})$; and $c_{3V}(\{a_4\};\{t_5\})$ (see Figure 4(b)).

If the agent a_4 becomes involved on t_3, then a_4 has to be added to coalition c_{1V}, but this will violate rule 1 because a_4 is not bound to t_1, t_2, t_4. Therefore the coalition c_{1V} will be split into $c'_{1V}(\{a_1, a_2, a_3\};\{t_1, t_2, t_4\}$ *and* $c''_{1V}(\{a_1,a_2, a_3, a_4\};\{t_3\})$. These two new conceptual role coalitions are not mergeable with any of the other coalitions of the activity, so the resulting coalition set is: $c'_{1V}(\{a_1,a_2,a_3\};\{t_1,t_2,t_4\}$ and $c''_{1V}(\{a_1,a_2,a_3,a_4\};\{t_3\}$, $c_{3V}(\{a_4\};\{t_5\})\}$.(see Figure 4(c)).

We construct the conceptual role join coalition algorithm as fellow:

Let (m,t) be the new bindings, in order to ensure rules 1 and 2, we present the algorithm in two steps. The first step consist in finding the conceptual role coalition c_V involving task t, if such coalition exists, and then adding the member m to c_V member set. The second step is the merge of the coalitions.

We arrange the coalitions in three sets:

`ToChange`: includes the coalition that will be modified.

`Mergeable`: contains the possible mergeable coalitions: it is composed of the remaining coalitions having agent member m as a member.

`UnChanged`: contains the coalitions that will be left unmodified, it is composed of the remaining coalitions.

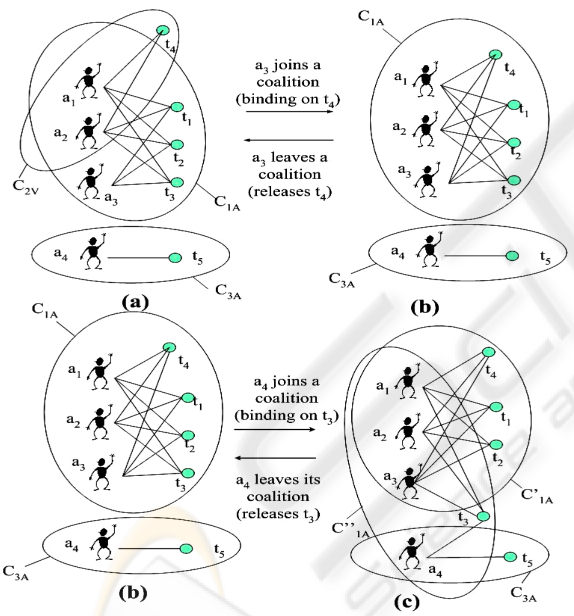

Figure 6: Conceptual role coalition dynamics.

5.5.4 The Agent Coalition Release Scheme

When the agent leave the coalition and release the binding on task t, this must be reflected in the relevant conceptual role coalition. The first step is to find the relevant coalition that is finding the coalition involving the task t. The second step is to remove the agent from the list of the agent of this conceptual role coalition, and to guarantee that the resulting coalition does still correspond to the coalition definition and still respects rules 1 and 2.

This restriction will imply a reorganization of the conceptual role coalition. Using the previous example, we had the following coalition set: $\{c'_{1V}(\{a_1,a_2,a_3\};\{t_1,t_2,t_4\}), c''_{1V}(\{a_1,a_2,a_3,a_4\};\{t_3\}), c_{3A}(\{a_4\};\{t_5\})\}$(See Figure 4(c)).

If agent a_3 releases the task t_4, it has to be removed from the member associated with t_4 in c'_{1V}. Rule 2 implies splitting the coalition into $c_{1V}(\{a_1,a_2,a_3\};\{t_1,t_2\})$ and $c_{2V}(\{a_1,a_2,\};\{t_4\})$ which leads the conceptual role coalition set to be partitioned as follows:

`ToChange`: Is the set which if not empty, includes the coalition that will certainly be modified.

`Mergeable`: contains the coalition that can be modified to ensure rule1.

`UnChanged`: contains the coalitions that will be left unmodified, it is composed of the remaining coalitions.

6 RELATED WORK

There have been several attempts at formalizing the concepts of teamwork within organization using multiagent systems: works of (Chalupsky et al., 2005), (Sycara et al., 2003) and (Kogut et al., 2004) propose systems and architectures to assist organizational human teams in routine tasks, such as meeting scheduling. In contrast to our approach, these systems deal with only the communication support part of the task performance, and ignore the processing part. We believe that our approach is more general since it integrates both support for the processing and communication required in performing any task. In addition none of these efforts have included techniques dealing with the different kind of the possible interactions and collaborations of team members, and their dynamics.

Some other works such as (DeLoach and Matson, 2004), (Krauth et al., 2005) and (Xu and Zhang, 2005) have looked at modelling organization using the notion of agents roles, goals and indicators; in distinction to our work, they do not capture the binding between the human and his/her tasks, these models fail to take into account the human perspective, thus is human in the loop, as it is a key element in any concrete application.

7 CONCLUSIONS AND FUTURE WORKS

In this paper, we described a multiagent modelling approach for collaborative and dynamic organizational roles support. The method is a role centric, where agents collaborate to assist human users by performing collaboration within the same role, between other roles of the same team, as well as roles of different teams that share tasks dependencies. We consider also the dynamics within the role; that is, agents joining and or leaving the coalitions. Ongoing work deals with simulation of the model using the third generation distributed dynamic decision making DDD-III (Simulator, 2006) for time performance. Future work will consider incorporating several other performance indicators such as performance indicators for quality, effort, cost, etc.

ACKNOWLEDGEMENTS

The Authors would like to acknowledge the clarifications and suggestions made by Dr. Joerg Denzinger on early versions of this work. We would also like to thanks Dr Kevin M. Gildea and Dr. Georgiy Levchuk of Aptima, Inc. for providing us with the DDD-III simulator to validate the model.

REFERENCES

Biddle, B.J. and Thomas, E.J., 1966. Role Theory: Concepts and Research, John Wiley & Son, Inc, 1966.

Krauth, E., Moonen, H. Popova, V. and Schut, M., 2005. "Agent-Based Modeling of Third-Party Logistics Performance Indicators", Intelligent Agent Technology IEEE/WIC/ACM International (IAT 2005) pp. 262-265, 2005.

Chalupsky, H. Gil, Y. Knoblock, C. A K. Lerman, Oh, J. Pynadath, D. V., Russ, T. A and Tambe, M. 2005.: "Electric Elves: Applying Agent Technology to Support Human Organizations", IAAI 2001, pp 51-58, 2005.

Xu, H. and Zhang, X. 2005. "A Methodology for Role-Based Modeling of Open Multi-Agent Software Systems", 7th International Conference on Enterprise Information Systems (ICEIS 3, 2005), pp. 246-253, 2005.

Sycara, K.P. Paolucci, M.,. Velsen, M. V, and Giampapa, J.A. 2003 "The RETSINA MAS Infrastructure", Autonomous Agents and Multi-Agent Systems 7 (1-2), pp 29-48, 2003.

Houari, N. and Far, B.H. 2007. "Improving Project Performance with Role Driven Computational Support System", IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2007), April 2007.

Houari, N. and Far, B.H. 2007. "A multiagent Coalition Support Tool for Collaborative and Dynamic Roles", IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2007), April 2007.

Kogut, P.A. Yen, J., Leung, Y., Sun, S, Wang, R. Mielczarek, T. and Hellar, B., 2004 "Proactive Information Gathering For Homeland Security Teams", Commun. ACM 47(3), pp 48-50, 2004.

DeLoach, S.A., and. Matson, E, 2004 "An Organizational Model for Designing Adaptive Multiagent Systems". The AAAI-04 Workshop on Agent Organizations: Theory and Practice (AOTP), 2004.

Simulator, 2006. "The DDD-III: Distributed Dynamic Decision-making" Simulator Provided by Aptima, Inc, 2006.