The Role Transferability in Emergency Management Systems

Haibin Zhu
Department of Computer Science and Mathematics, Nipissing University,
100 College Drive
North Bay, Ontario, P1B 8L7, Canada
email: haibinz@nipissingu.ca

MengChu Zhou
Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights
Newark, NJ 07102, USA
email: zhou@njit.edu

ABSTRACT

Role Transferability is a basic requirement for emergency management systems. Role specification, relationship expression, and transfer regulations are critical elements of this requirement. This paper discusses the role transferability requirement for emergency management systems; emphasizes that role specification is an underlying mechanism for role transfer; proposes a revised E-CARGO (Environment-Class, Agent, Role, Group, Object) model for role transfer in a group; and presents an algorithm to validate role transfer while maintaining group viability.

Keywords

Role, role transfer, emergency management systems.

INTRODUCTION

Emergency management systems (EMS) are computer-based tools that support people dealing with crisis and emergency. They should support people's routine activities and improve their abilities in responding to a crisis. Introducing roles into EMS is very important. They can assist the users of such systems by preventing them from being overwhelmed by too much irrelevant information; reminding people of their rights and responsibilities and helping to maintain a workable system in times of crisis. Roles are necessary to describe an organization. The structures of roles present the ways through which the participants interact and through which the organization achieves its goals (Cyert and MacCrimmon, 1965).

Role transferability is a basic requirement for organizations (Ashforth, 2001) and EMS (Turoff, et al., 2004). It is used to evaluate and check the flexibility of a group when membership and/or roles change. One of the major problems in current information systems is the lack of a mechanism to clearly define and rigorously specify roles while maintaining flexibility when roles change. Role transfer is especially challenging for a group of people during an emergency. "It is impossible to predict who will undertake what specific role in a crisis situation. The actions and privileges of the role need to be well defined in the software of the system and people must be trained for the possibility of assuming multiple or changing roles (Turoff, et al., 2004)."

Every one should be aware of his/her role when a crisis occurs. This is a basic requirement in an emergency situation. When working in China, the first author led a section that was responsible for course offerings in the computer engineering program. A course cancellation due to the shortage of instructors was considered a severe fault for the section. To avoid such a situation, regulations were established requiring that every course have more than one qualified instructor and instructors be able to teach more than one course. This solution helped to deal efficiently with emergency teaching assignments and avoid course cancellations during the five years when he administrated the section. The handled emergency situations were accidents to an instructor, unexpected illness, or urgent family matters.

In fact, in an emergency system, not every one can change and transfer his/her roles. Role transfer is a complex event that may involve many relevant roles. When one person transfers to a new role, his/her original role needs to be adopted by another if the role is still required in the system. This change might initiate a series of role transfers. For example, in a battle field, if a high rank officer is shot, injured or dead, a similar or lower rank officer is needed to play his/her role. The role played by the similar or lower rank officer may need to be played by another, …

In highly-available computer systems, we need duplicate components to guarantee this requirement. It is the same in an emergency management system. One specific role should be played by more than one person. One person should be able to play more than one role (Turoff, et al., 2004, Zhu, 2006).

However, how many roles should a person play or potentially play to deal with an emergency situation? What role-person mapping is effective in dealing with an emergency situation? What group structures can avoid crisis? There are no clear, exact answers for these questions hitherto. This paper intends to demonstrate that in an emergency situation, specially arranged role-person mapping could help a group survive and maintain a workable state. Our role-based collaboration model E-CARGO (Environment-Class, Agent, Role, Group, Object) (Zhu and Zhou, 2006) can be successfully adopted in processing such situations.

In this paper, we discuss roles, their specification and define the fundamental concepts in a role-based emergency system. Next, we demonstrate the revised E-CARGO model that aims at supporting EMS. After that, we describe the requirement for role transfer algorithms and their implementation. Finally, we conclude the paper by discussing future work.

ROLES AND ROLE SPECIFICATION

In our society, people generally adopt two major roles, i.e., server and client. Every one serves others and everyone is served by others. When people play a role, they provide certain services and have rights to ask for services. With this in mind, a role can be considered as a view of persons on the world. When they play a specific role, they have a special view of the surroundings. Their role in a working environment is actually a wrapper with a service interface (Gottlob, et al., 1996, Genilloud and Wegmann, 2000, Steimann, 2001) and request interface (Patterson, 1991) as shown in Figure 1 (Zhu 2006). We can separate a person's role into two parts: the service interface including incoming messages, and the request interface including outgoing messages. In fact, the human icon in Figure 1 can be replaced by an agent, object, group or system. Hence, the roles applied in an information system should be concerned with two aspects of roles: responsibilities and rights. A player (a person, agent, or object) should send out messages to invoke other players' services. In this paper, we concentrate on an agent or person playing a role.

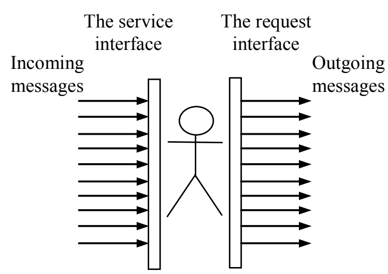


Figure 1. A role as a wrapper of a person in a working environment

To understand Figure 1, we can state the principles or the paradigm of role modeling in information systems:

1) A role is independent of its player.
We can define it separately. It is a common idea that a role is dependent on objects in object systems. In a society, however, people may not care about a specific person when asking for a service. They only want to contact a person who plays a specific role and the role is their major concern. For example, a professor in Nipissing University may ask for services from the UTS (University Technology Service). S/he does not care who comes to provide the service as long as one plays the role of a technician at UTS.

2) A role can be created, changed and deleted.
3) A role includes responsibilities (the service interface) when the player is taken as a server and rights (the request interface) when the player is taken as a client. To specify a role means to specify both responsibilities and rights.
4) A role does not accomplish the tasks specified by the responsibilities. Only players who are playing the role accomplish the tasks.
5) As for the service interface, a role is actually a filter of messages sent to a player.
6) As for the request interface, a role expresses or restricts the accessibility of a player to the system.

7) Roles are taken as a medium for interaction. Roles can be used to support both indirect and direct interactions. Interactions among people, agents or objects are based on their roles, i.e., a request message is sent to the relevant roles. Role-based systems emphasize that the message receivers are roles but not players. A person, an agent or object is not required to be identified to serve. Even though they are actual servers for the clients, they are not directly facing the clients. For direct interactions, each role has exactly one person to play it. For indirect interactions, each role has multiple persons to play. In the former case, identifying the role means identifying the person. In the latter, identifying a role does not mean identifying a person.
8) Playing a role means that the player is attached to a role.
9) A role can be played by one or more players at the same time.
10) A role can be long or short-termed based on the context. Some roles can exist over a long term regardless of the player, a country's president, for example. Other roles may exist only for a special player, such as the guest editor for a special issue of a journal.

THE REVISED E-CARGO MODEL FOR EMS

By establishing a formal model to define and specify a role, we can obtain a much clearer view of such systems. Without a clear specification of components, establishment of successful systems becomes more difficult. A well-defined internal structure helps to guarantee a successful system. It supports the robustness, efficiency and correctness of the entire system. By E-CARGO (Zhu and Zhou, 2006), we mean Environments, Classes, Agents, Roles, Groups and Objects. A role-based system $\sum$ is described as a 9-tuple. $\sum ::= <\mathcal{C}, \mathcal{O}, \mathcal{A}, \mathcal{M}, \mathcal{R}, \mathcal{E}, \mathcal{G}, s_0, \mathcal{H}>$, where,

- $\mathcal{C}$ is a set of classes;
- $\mathcal{O}$ is a set of objects;
- $\mathcal{A}$ is a set of agents who are representatives of human users;
- $\mathcal{M}$ is a set of messages;
- $\mathcal{R}$ is a set of roles;
- $\mathcal{E}$ is a set of environments;
- $\mathcal{G}$ is a set of groups;
- s_0 is the initial state; and
- $\mathcal{H}$ is a set of users.

With the participation of users $\mathcal{H}$, e.g., logging in a system $\sum$, accessing objects of the system, sending messages through roles, and forming a group in an environment, $\sum$ evolves, develops and functions. The results of interaction among objects, agents, and human users are a new state of $\sum$ that is expressed by the values of $\mathcal{C}, \mathcal{O}, \mathcal{A}, \mathcal{M}, \mathcal{E}, \mathcal{G}$, and $\mathcal{H}$. We include $\mathcal{H}$ to express that users may be affected by collaboration. Please note that without the participation of users, the system can only do what the agents can do. This is why users are an essential part of a system. The initial state s_0 is expressed by initial values of all the components $\mathcal{C}, \mathcal{O}, \mathcal{A}, \mathcal{M}, \mathcal{R}$ and $\mathcal{E}$, such as, built-in classes, initial objects, initial agents, primitive roles, primitive messages and primitive environments.

To deal with problems in EMS, we emphasize the relationships among roles and agents in a group. Therefore, we restate the details of an agent, role, environment and group. The definitions of these components are given as follows, where, if χ is a set, $|\chi|$ is its cardinality: $a.b$ means b of a or a's b:

A role is defined as $r ::= <n, I, \mathcal{A}_c, \mathcal{A}_p, \mathcal{N}_o>$ where,

- n is the identification of the role;
- $I ::= <\mathcal{M}_{in}, \mathcal{M}_{out}>$ denotes a set of messages, where, $\mathcal{M}_{in}$ expresses the incoming messages to the relevant agents, $\mathcal{M}_{out}$ expresses a set of outgoing messages or message templates to roles, i.e., $\mathcal{M}_{in}, \mathcal{M}_{out} \subset \mathcal{M}$;
- $\mathcal{A}_c$ is a set of agents who are currently playing this role;
- $\mathcal{A}_p$ is a set of agents who are able (or pre-qualified) to play this role; and
- $\mathcal{N}_o$ is a set of objects including classes, environments, roles, and groups that can be accessed by the agents playing this role.

An agent is defined as $a ::= <n, c_a, s, r_c, \mathcal{R}_p, \mathcal{N}_g>$, where

- n is the identification of the agent;
- c_a is a special class that describes the common properties of users;

- s is the profile of the person whom the agent is representing;
- r_c means a role that the agent is currently playing;
- $\mathcal{R}_p$ means a set of roles that the agent is able to play; and
- $\mathcal{N}_g$ means a set of of groups that the agent belongs to.

Denote by $\mathcal{A}$ the set of all the agents in a system.

An environment and group are defined as follows:

An environment $e ::= <n, \mathcal{B}>$ where

- n is the identification of the environment; and
- $\mathcal{B} = \{< r, q, \Psi>\}$ is a set of tuples of role, number range and an object set. The number range q tells how many users may play this role in this environment and q is a tuple of number $<l, u>$, where l means the lower bound of the number of agents to play this role and u the upper bound. It may be <1, 1>, <2, 2>, <1, 10>, <3, 50>, … . The object set Ψ expresses the complex objects accessed by the agents who play the relevant role. By "complex" we mean that they are composed of other objects. The complex objects in Ψ are mutually exclusive, i.e., one complex object in this set can only be accessed by one agent (user). For each tuple, we have the inequality: $q.l \leq |\Psi| \leq q.u$. In fact, Ψ expresses the resources for agents to access. In an emergency system, we are more interested in l.

Denote by $\mathcal{E}$ the set of all the environments in a system.

In a crisis, the E-CARGO model can be used to indicate the states of a group of people using an emergency system. The agents and related roles in the system clearly tell the people their responsibilities and rights. If one person is called out of the group, another must adopt the role(s) performed by that person. The computer system must quickly reflect new roles to the people in the group due to the role transfer.

With this definition, we know that before creating an environment, we initially need to specify roles. At the same time, before we specify a role, we need to create objects for roles to access.
A group is defined as $g ::= <n, e, \mathcal{J}>$ where

- n is the identification of the group;
- e is an environment for the group to work; and
- $\mathcal{J}$ is a set of tuples of an agent, role, and complex object, i.e., $\mathcal{J} = \{<a, r, o> | \exists q, \Psi \ni (o \in \Psi) \wedge (<r, q, \Psi> \in e.\mathcal{B})\}$.

Denote by $\mathcal{G}$ the set of all the groups in a system.

EMS should directly simulate people's management systems or organizations. From the above discussion, components of the E-CARGO model are comparable to entities encountered in EMS.

ROLE-BASED EMERGENCY MANAGEMENT SYSTEMS

A role-based EMS can be built with the E-CARGO model. The following principles have been set (Zhu and Zhou, 2003, Zhu, 2006):

1) Everything in the world is an object (Kay, 1993).
2) Every system is composed of objects (certainly a system is also an object).
3) The evolution and development of a system is caused by the interactions among the objects inside the system or the interactions between them and outside objects.
4) Classes can be taken as the templates of objects and a class is used to express the commonalities of a group of objects.
5) Every object is an instance of a class and can be created, modified and deleted.
6) A class may have subclasses and superclasses.
7) Object interactions are expressed by exchanging messages.
8) An agent is a special object that represents a person in the system.
9) A role is an entity or data structure that expresses a player's services and requests.
10) An agent can play many roles in a period of time but can play only one role at a definite point of time.
11) A role is independent of agents. It can be defined independently of an agent.

In an emergency situation, only some agents, roles or groups are critical. They must be easily identified in order to deal with the crisis effectively. To understand the problem of a crisis situation, it is important to provide some common definitions. In considering the following definitions, the terms "agent" and "person" are interchangeable. That is to say, an agent in a system is a representative of a person using the system.

Definition 1: **active role.** A role r is active to an agent if the agent is able or qualified to play this role.

Definition 2: **current role.** A role r is current to an agent if the agent is currently playing this role, i.e., r_c.

Definition 3: critical role. A role r is critical if it has only just enough number of agents currently playing it, i.e., $\exists g, \exists \Psi \ni (< r, q, \Psi > \in g.e.\mathcal{B}) \wedge (|r.\mathcal{A}_c| = q.l)$.

Definition 4: workable group. A group g is workable if for each role r there are enough agents to play it, i.e., $\forall r \in \mathcal{R}, (\exists q, \exists \Psi \ni (< r, q, \Psi > \in g.e.\mathcal{B})) \to (|r.\mathcal{A}_c| \geq q.l)$.

Definition 5: role transfer. If a role loses its playing agent, the group g must re-distribute the agents to roles and guarantee that every role has at least l agents to play it. We call such re-distribution role transfer.

Definition 6: successful transfer. A role transfer in group g is successful if g is workable after the transfer.

Definition 7: critical agent. An agent is critical if there is no successful role transfer to make the group workable once it leaves a group.

Definition 8: lost agent. An agent leaves a group g. We say that g loses this agent and this agent is g's lost agent.

Definition 9: critical group. A group g is critical if for each role in it there are just enough agents currently playing it; Or if each role in it is critical, i.e., $\forall r \in \mathcal{R}, (\exists q \exists \Psi \ni (< r, q, \Psi > \in g.e.\mathcal{B})) \to (|r.\mathcal{A}_c| = q.l)$.

Definition 10: emergency. A level-n emergency occurs if a group has n (≥ 1) lost agents.

Definition 11: strong group. A group is level-n strong if it is workable via successful role transfer after a level-n emergency occurs.

With Definitions 1 and 2, the agent cannot directly respond to messages relevant to this role. $a.\mathcal{R}_p$ is a set of all the active roles of an agent a. The agent can respond directly to messages relevant to its current role, but must transfer to respond to a message not covered by its current role. Note that $r_c \in a.\mathcal{R}_p$.

For an agent, an active role can become current and the current role can become active. An agent can have only one current role. That is to say, an agent can hold many active roles at the same time but one current role at a time. By holding only one current role, we can avoid role-role conflicts (Bostrom, 1980). Active and current roles can also be used to express the significance of role transfer, i.e., changing the current role of an agent to an active one.

In a crisis, people must concentrate on their roles at all times. They cannot leave their roles without players at any time. A critical group in Definition 9 requires that no agent can be lost and still keep the concerned group workable. The system should then request to add new agents or people to the group.

Definitions 10 and 11 suggest that not all agents are critical in a group. Definition 11 means that even though some agents may be lost, the group can remain workable in a finite time through successful role transfer. For a computer-based system, this time is determined by the complexity of algorithms and speed of computers. For a social system, this time might be variable based on the degree of emergency, effectiveness of the organizations and capability of decision makers. It may be minutes, hours, days or weeks.

AN ALGORITHM FOR ROLE TRANSFER

Problem

From the above discussion, one can observe that an algorithm is required to deal with role transfer. From the following fictitious example, we can obtain a more practical requirement.

Suppose that a critical castle in a battlefield contains eight critical ports for soldiers to guard. If one port is taken by the enemy, the castle is lost. The enemy is trying to attack every port simultaneously and continuously. That is to say, at any time, each port must have at least one soldier on guard.

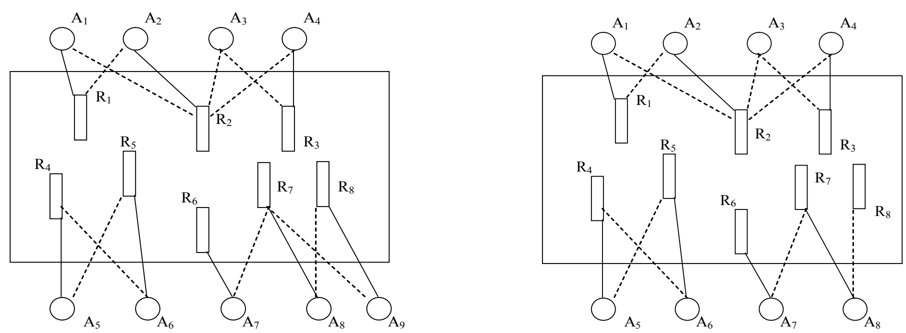
The officers are responsible for commanding a team to guard the ports. Each port is equipped with different special devices to capture required information, contact other soldiers or officers and block enemies. This means that port security requires training that is specialized, expensive and time consuming. Understanding these requirements, the officers hope to find the most economic and quickest techniques to train the soldiers (role assignments) and guarantee the security of all ports even if some soldiers lose the required abilities. That is to say, they cannot train all soldiers to know all devices stationed at all ports due to cost and time constraints.

It is evident that they must send at least eight soldiers to guard the ports. They hope by special training (role assignments) to guarantee that nine solders can maintain port security even if one solder loses the required ability. Suppose that the time required to replace a soldier for a port is not enough for the enemy to acquire a port.

Initially, they consider training each soldier to guard two different ports as shown in Figure 2. They find that this way cannot guarantee the castle's safety when nine soldiers are placed on guard and one soldier (say A_9) loses the ability. In the following figures, we use a box to express a role; a circle is an agent; a dashed arc means a role attachment, meaning the agent is able to play the role, and a solid arc indicates that the agent is currently playing the role. For simplicity, we assume that a role must have at least one agent to play currently it.

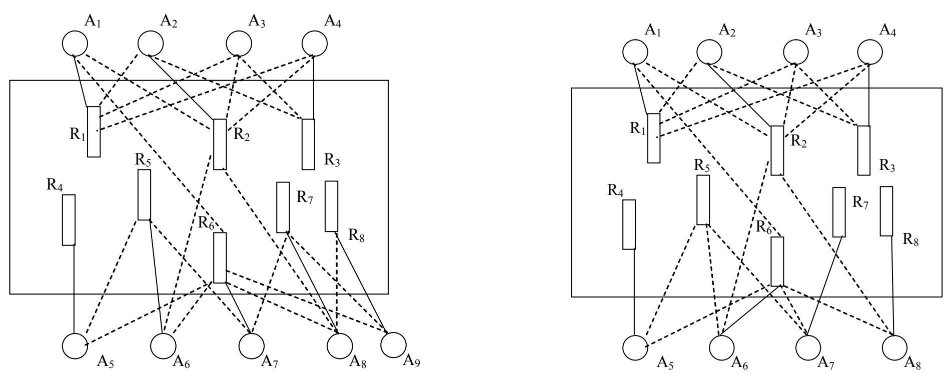
Next, they consider training each soldier to guard three ports as shown in Figure 3. They find that this alternative is not workable if one special soldier (A_9) is lost. If training is re-arranged from Figure 3 to Figure 4, the plan is workable if one special soldier (A_9) is lost.

The following five cases are considered.



(a) (b)

Figure 2. Every role has two agents to play or to be qualified to play



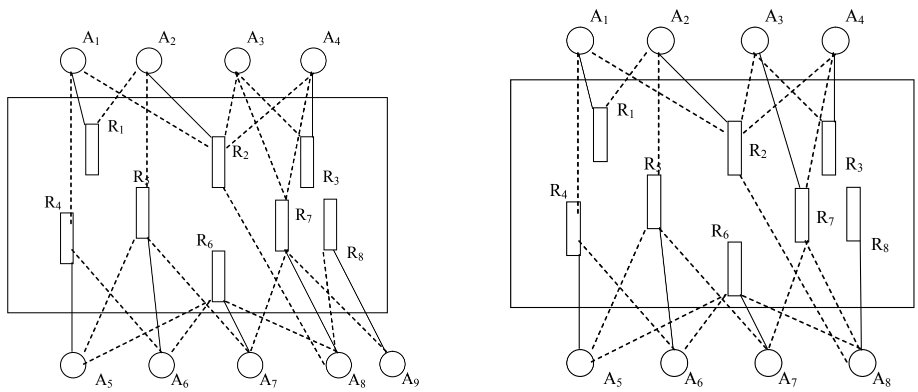
(a) (b)

Figure 3. A new assignment of Figure 2

Case 1*:* Figure 2(a) shows a workable group. Suppose that as the current player of R_8, A_9 leaves the group. The group is shown in Figure 2(b). Is the group still workable by role transfer? In Figure 2(b), we can find that R_8 can be played by A_8. Hence, we can let A_8 play R_8. Consequently, R_7 has no current player. We can let A_7 play R_7, then there is no agent to play R_6, because there are no other agents that can play R_6. This means that the group shown in Figure 2(a) is not *a critical group* but A_9 is *a critical agent*.

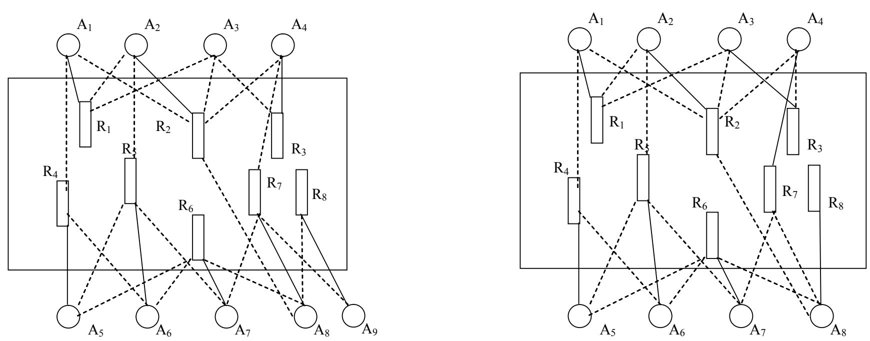
Case 2: Figure 3(a) shows a group similar to Figure 2(a). The only difference is that the agents' qualifications are more than those in Figure 2(a). After A_9 leaves the group, A_8 can play R_8, A_7 can play R_7, and A_6 can play R_6. Although A_5 can play R_5, we cannot find another qualified agent to play R_4. This means A_9 is a critical agent for the group shown in Figure 3(a). After A_9 leaves, the group is not workable as shown in Figure 3(b).

Case 3: Figure 4(a) shows a group similar to Figure 3(a). The only difference is that the agents' qualifications are rearranged. After A_9 leaves the group, we can find A_3 to play R_7. The problem is solved as shown in Figure 4(b). This means that A_9 is not a critical agent for the group shown in Figure 4 (a).



(a) (b)

Figure 4. A new arrangement of Figure 3

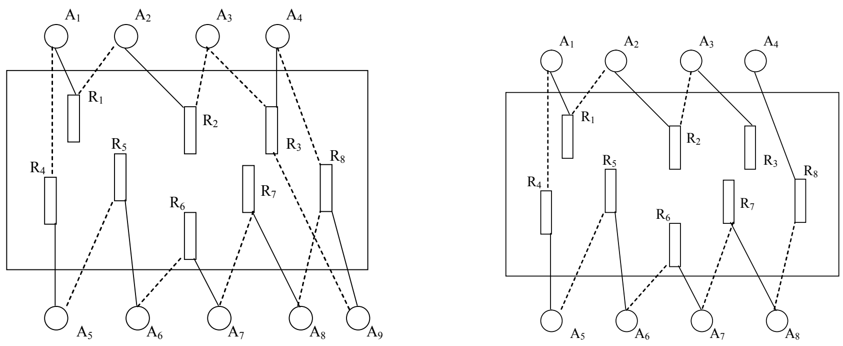


(a) (b)

Figure 5. A new arrangement of Figure 4

Case 4: Figure 5(a) shows another arrangement of Figure 4(a). After A_9 leaves the group, we can have A_8 play R_8. Now, R_7 has no free agent to play it. We can have A_4 play R_7 and A_3 play R_3. Then, the problem is solved as shown in Figure 5(b). This shows that A_9 is not a critical agent in the group shown in Figure 5(a). We can also check and know every agent in Figure 5(a) is not critical, i.e., the group shown in Figure 5(a) is *a level-1 strong group*.

Case 5: Figure 6(a) shows a re-arrangement of Figure 2(a). If A_9 leaves the group, we can make it a workable group shown in Figure 6(b) and make A_9 not *a non-critical agent*. In fact, we can verify that Figure 2(a) is a *level-1 strong group*.



(a) (b)

Figure 6. A new arrangement of Figure 2

They also want to know, if Figure 4 remains workable after an agent A_i (i= 1, 2, …, 9) is lost. If this is solvable by well-educated assistants, how can they prepare the duty exchanges after one soldier is lost? How can they deal with more complex groups? How can computer technology help the officers solve this problem? These questions evidently produce a requirement for an automatic algorithm.

Algorithm

From the above discussion, we know that not all agents are critical in a workable group. Whether a workable group is strong or not is up to the organization of the group and relationships between roles and agents. We need the help of computers to determine a role transfer procedure. Therefore, we need both specific data structures and algorithms. Below is an implementation of a role transfer algorithm. The E-CARGO model, discussed in the third section, is qualified to provide such structures. A descriptive algorithm is as follows. For simplicity, we suppose that at least one agent is required to play a role, i.e., for group g, $\forall q (\exists r, \exists \Psi \ni (< r, q, \Psi > \in g.e.\mathcal{B})) \rightarrow q.\ell = 1)$. Suppose that $\mathcal{S}$ is a set and $s \in \mathcal{S}$, we use $\mathcal{S}$.remove(s) to express "removing s from $\mathcal{S}$". Suppose $e \notin \mathcal{S}$, we use $\mathcal{S}$.add(e) to express "adding e to $\mathcal{S}$".

Input: A set of roles $\mathcal{R}$, a set of agents $\mathcal{A}$, a set of <role, agent> mapping for current agents $\mathcal{U}$ and a set of role mapping <role, agent> for qualified agents $\mathcal{V}$, and a role w without a current agent, i.e., $\forall a \in \mathcal{A} <w, a> \notin \mathcal{U})$.

Output:

Success: A set of roles $\mathcal{R}$, a set of agents $\mathcal{A}$, a set of <role, agent> mapping for current agents $\mathcal{U}$, a set of role mapping <role, agent> for qualified agents $\mathcal{V}$, $\forall r \in \mathcal{R}, \exists a (<r, a> \in \mathcal{U})$.

Failure: A set of roles R, a set of agents A, a set of <role, agent> mapping for current agents $\mathcal{U}$ and a set of role mapping <role, agent> for qualified agents $\mathcal{V}$ and a role without current agent w, i.e., $\forall a \in \mathcal{A} <w, a> \notin \mathcal{U})$.

Step 1: Preparing for role transfer:

1) Build a set of all the roles in the group $\mathcal{R}$.
2) Build a set of all the agents in the group $\mathcal{A}$.
3) Set w the role currently without a playing agent.
4) If $|\mathcal{R}| > |\mathcal{A}|$ report failure, i.e., there is no successful role transfer.
5) Else preparation completes, continues to Step 2.

Step 2: Role transfer:

1) If $\mathcal{A} = \Phi$ and $\mathcal{R} \neq \Phi$ then report failure and exit.
2) If $\mathcal{R} = \Phi$ then report success and stop.
3) For all $a \in \mathcal{A}$ do
{
If $w \in a.\mathcal{R}_p$ (i.e., $<w, a> \in \mathcal{V}$) then
{
If $a.r_c$ is empty then set $a.r_c$ with w.

```
                Report success and exit.
                }
        }
        4) For all a ∈ 𝒜 do
        {
                If  w ∈ a.ℛp (i.e., <w, a> ∈ 𝒱) then
                        {
                                Save a.rc to q;       // Note that, q is a temporary space to store the current role.
                                Set a.rc with w;
                                Set 𝒜 with 𝒜.remove(a);
                                Set ℛ with ℛ.remove(w);
                                Set w with q;
                                Go to Step 2;
                                If Role transfer is successful then report success and exit;
                                Else
                                {
                                        Set 𝒜 with 𝒜.add(a);
                                        Set ℛ with ℛ.add(w);
                                        Set w with q;
                                }
                        }
        }
```

Step3: Report failure and exit.

Complexity Analysis

Suppose there are M agents and N roles in a system, we can analyze the complexity of the above algorithm.

Because $M \geq N$, we will simplify them to M.

For Step 1, there is a need to enumerate all agents and roles. The complexity is $O(M+N) = O(M)$.

For Sub-step 3) of Step 2, there is a need to enumerate all agents. The complexity is $O(M)$.

For Sub-step4) of Step 2, for each agent, we need a new recursive search in a set with the agent removed, therefore, the complexity is $O(M*(M-1)* \ldots *1) = O(M!)$

Therefore, the total complexity is $O(M!) + O(M) + O(M) = O(M!)$.

From this complexity analysis, we may understand that role transferability problem is NP-hard. Heuristic or artificial intelligent methods are needed for large size systems.

Implementation

To facilitate the algorithm, we put three special variables into a group RoleSet, AgentSet and eRole, where RoleSet is a set of all the roles in the group; AgentSet is a set of the agents in the group and eRole is the role currently having no current role, i.e., $\mathcal{R}$, $\mathcal{A}$, and w in the above algorithm.

We have implemented two functions: *prepareRoleTransfer()* is used to set the above three variables (Step 1) and *transferRoles()* is used to transfer roles in the group (Step 2). The key point of the implementation is that we need a graph retrieval algorithm and a back traceable search algorithm.

All the special cases discussed above are simulated by a Java program. We obtain the same results as shown in Figure 2-6.

CONCLUSION

Role transferability is important in an emergency management system (EMS). A Chinese saying "预则立, 不预则废 (Yu Ze Li, Bu Yu Ze Fei)" tells us "Preparation leads to success and otherwise failure". Specially and carefully assigning current and active roles is a good and required preparation for EMS. Based on the revised E-CARGO model, we define the properties of EMS and the concept of role transferability. By implementing an algorithm, we can exactly recognize whether a group in EMS is strong or not and find a transfer scheme. Our current algorithm deals with only the situation that one role loses its player and at least one agent is required to play every role. It is valuable for us to find a more generalized algorithm to deal with the situation that n (>1) roles lose their players at the same time and more than one agents are required to play a role. Moreover, the complexity of the proposed algorithm O(M!) is unsatisfactory for large systems. A heuristic algorithm is necessary. It is also worth considering a graphical interactive tool to specify a group.

ACKNOWLEDGMENTS

This research is in part supported by IBM Eclipse Innovation Grant. Thanks to Professor Mike Brewes of Nipissing University for his proofreading this article.

REFERENCES

1. Ashforth, B. E.(2001) Role Transitions in Organizational Life: An Identity-based Perspective, Lawrence Erlbaum Associates, Inc.
2. Bostrom, R. P.(1980) Role Conflict and Ambiguity: Critical Variables in the MIS User-Designer Relationship, *Proc. of the 17th Annual Computer Personnel Research Conference*, Miami, Florida, United States, 88-115.
3. Cyert, R. M and MacCrimmon, K. R. (1968) Organizations, *Lindzey, G. and Aronson, E. eds., The Handbook of Social Psychology, Vol. 1,* Addison-Wesley, 568-611.
4. Genilloud, G. and Wegmann, A. (2000) A Foundation for the Concept of Role in Object Modelling", *The 4th International Enterprise Distributed Object Computing Conference (EDOC2000)*, Makuhari, Japan, 76-85.
5. Gottlob, G., Schrefl, M. and Röck, B. (1996) Extending Object-Oriented Systems with Roles, *ACM Trans. on Information Systems (TOIS)*, 14, 3, 268-296.
6. Guarino, N. (1992) Concepts, Attributes and Arbitrary Relations: Some Linguistic and Ontological Criteria for Structuring Knowledge Bases, *Data & Knowledge Engineering*, 8, 249-261.
7. Kay, A. C. (1993) The Early History of Smalltalk, *The Second ACM SIGPLAN History of Programming Languages Conference, ACM SIGPLAN Notice*, 28, 3, 69-75.
8. Patterson, J. F., (1991) Comparing the Programming Demands of Single-User and Multi-User Application, *The Fourth Symposium on User Interface Software and Technology*, ACM Press, 87-94.
9. Steimann, F. (2001) Role = Interface: A merge of concepts, *Journal of Object-Oriented Programming*, 14, 4, 23-32.
10. Turoff, M., Chumer, M., Van de Walle, B. and Yao, X. (2004) The Design of a Dynamic Emergency Response Management Information System (DERMIS), *Journal of Information Technology Theory and Application (JITTA)*, 5, 4, 1-35.
11. Zhu, H. (2005) Encouraging Participants' Contribution in Collaboration*", Proc. of the 2005 IEEE International Conference on Systems, Man, and Cybernetics*, Hawaii, USA, 1574-1579.
12. Zhu, H. (2006) Role Mechanisms in Collaborative Systems, *International Journal of Production Research*, 44, 1, 181-193.
13. Zhu, H. and Zhou, M.C. (2006) Role-Based Collaboration and its Kernel Mechanisms, *IEEE Trans. on Systems, Man and Cybernetics, Part C,* to appear.
14. Zhu, H. and Zhou, M.C. (2003) Methodology First and Language Second: A Way to Teach Object-Oriented Programming, *Companion of 2003 International Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA'03)*, California, USA, 140-147.