INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL. 10, NO. 1, MARCH 2005, 11-20

Conflict Resolution With Roles In A Collaborative System

Haibin ZHU

Abstract- **Computer-mediated collaboration is an increasingly prevalent means of connecting individuals and teams. Increasing research is being undertaken to improve virtual environments in order to enhance the ability of collaborators to interact effectively and cooperatively. However, computer-mediated collaboration differs radically from face-to-face human interaction with resultant problems for those relying on technology for collaborative purposes. Even though scientists are working hard to simplify collaboration by providing virtual tools and environments and collaborators work hard to be cooperative, there are still unknown problems for collaborators to find and overcome when using a collaborative system. Conflict resolution is one of the most important problems to overcome.**

This paper briefly introduce an object model for collaborative systems (OMCS) and a multimedia co-authoring system (MCAS); discusses role management and conflict resolution in MCAS systems; emphasizes the usability of roles in conflict resolution. In the last section, it summarizes and concludes that role mechanisms can be very helpful in resolving conflicts in collaborative systems.

***Index Terms*— Conflict resolution, roles, collaborative systems**

1. INTRODUCTION

Collaborative systems combine communication, computer, and decision support technologies to support problem formulation and solution in group meetings [19]. Team members are dependent on mediated interactions for coordination, and are likely to face important deficits in the information they have about the day-to-day activities of their teammates without them [20].

There are numerous differences between collaboration in a computer-supported environment versus a face-to-face environment. Even though scientists are working hard to simplify collaboration by producing systems that provide virtual environments, and collaborators work hard to be cooperative, there are still unknown problems for collaborators to find and overcome when using a collaborative system. In building a collaborative system, conflict resolution is one of the most important problems to overcome.

Early in 1988, scientists recognized this problem and began to study it [19]. Over the past decade, scientists have made numerous contributions. Poole et al. conducted research on how a non-specialized, multipurpose GDSS influences conflict management in groups [19]. Edwards proposed a flexible conflict detection and management method [4]. Jung et al. suggested a conflict resolution method through coordination and argumentation agents [13]. Barber et al. discussed conflict detection in multi-agent systems [1]. Sun et al. proposed a formal specification of a unique combined effect for an arbitrary group of conflict and compatible operations [21]. Other scientists generalized their research by using algorithms in access control to solve this problem [3, 10].

However, even though role mechanisms have been applied very successfully in access control, i.e., Role-Based Access Control (RBAC) [9], there is little research on conflict management with roles [5, 8].

To support collaborative work, a workflow and access control mechanism is very useful [23, 30]. A role is a semantic construct forming the basis of access control policies [17]. Roles are powerful, policy neutral concepts used for facilitating distributed systems management and enforcing access control [14]. Roles provide a natural way for an enterprise administrator or security officer to describe the privileges of various job functions [16].

The other sections are organized as follows. In section 2, an object model for collaborative systems (OMCS) and a multimedia co-authoring system (MCAS) are introduced briefly and the conflict resolution structure with roles used in our system MCAS is described; in section 3, conflict resolution at the management level is discussed and the regulations for role management used in the interface design is demonstrated by Petri nets; in section 4, the conflict resolution at the document level is illustrated and the regulations of editing operations used in the design of a manager object are depicted; in section 5, some implementation issues for the method with roles in the MCAS system are outlined; and in the last section, the paper is concluded that a method with roles can help to resolve conflicts in collaborative systems and suggestions are provided on how to introduce roles into designing collaborative systems.

2. OMCS: AN OBJECT MODEL FOR COLLABORATIVE SYSTEMS

An Object Model for Collaborative Systems (OMCS) is designed and used as a guide to design a co-authoring system [24, 25, 26, and 27].

In terms of object-oriented technology [22 and 29], a collaborative system is addressed as an instance of the class OMCS ::= <CSID, CSDS,CSOP>, where

Manuscript received March 18, 2004; revised October 31, 2004. This work was supported by IBM Eclipse Innovation Grant. This paper is extended from "A role-based conflict resolution method" published at *IEEE International Conference on System, Man and Cybernetics,,* Washington, D.C., USA, Oct., 2003.

H. Zhu is with Dept. of Computer Science and Mathematics, Nipissing University, 100 College Dr., North Bay, Ontario, P1B 8L7, Canada (e-mail: haibinz@nipissingu.ca).

- CSID ::="CoSystem", is a name or an identification. A URL on the Internet can be taken as identification.
- CSDS ::= <A, M, W, D> expresses the structure (or state) of the system, where
    - A denotes the platform architecture of the system, A ::=Totally Centralized | Totally Distributed | Partially Distributed;
    - M denotes the collaborative (or coordination) modes;
    - W denotes a management model for information; and
    - D denotes the support tools for discussion, D ::= Text|Audio|Video|Combined.
- CSOP::=<R, S, E, I> expresses the services provided by the system, where
    - R expresses the services for roles, such as, chairman, lecturer, audience or others;
    - S expresses the services for cooperation, such as, start, enter, leave, late, leave early;
    - E expresses the services for editor, such as text editor, hypertext editor, and multimedia editor; and
    - I expresses the interface design for multi-user requirements, such as WYSIWIS service, and collaborative edit service.

Hence, if people want to design or implement a new collaborative system, they actually create a new instance of this class and set the instance body with new concrete attributes and services.

Based on the above model OMCS, a Multimedia Co-Authoring System (MCAS) [26] have been implemented. The design of a collaborative system is actually a process of instantiation where each attribute or service is implemented. Therefore, MCAS is actually an instance of class OMCS, i.e., MCAS ::= <"MCAS", <a, m, w, d>, <r, s, e, i>>, where

- a ::= partially distributed architecture
- m ::= parallel + sequential + reciprocal
- w ::= WNCH multimedia management model
- d ::= Text + Audio + Video + Combined
- r ::= Chairman + session leader + lecturer + audience
- s ::= Early messages + late comers + early leavers + transitions among roles
- e ::= WNCH editor + special editors for different media such as text, image, audio and video information
- i ::= Multi-user interface

View sharing is the most important mechanism for a synchronized collaborative system [2, 7 and 20]. It means that one user must be aware of the existence of the other users. It is the best way to be aware in a collaborative system. View sharing can help to eliminate unnecessary conflicts among collaborators.

In MACS, the implementation is on a partly distributed system. This means a centralized server managing the shared document and numerous distributed user interfaces which clients use directly as shown in Fig. 1. By this architecture, both view and document sharing are supported.

View sharing means that a system provides the facility which allows each co-author to have the same interface copy to work on as shown in Fig. 1. In Fig. 1, there are four copies of the same interface.

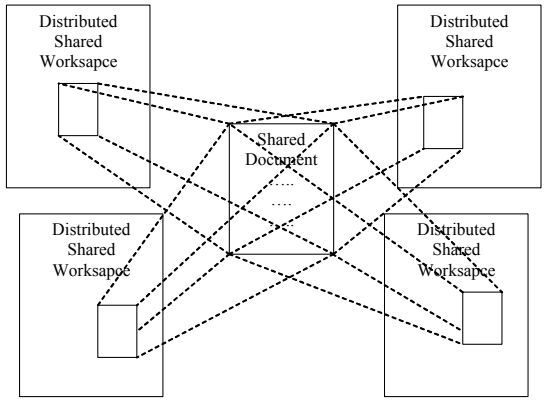

Fig. 1. Sharing by shared workspace.

A collaborative system should provide facilities to support the following activities [24 and 25]:

- announce a collaborative work by earlier messages or calling in time;
- enter or leave at different times;
- designate the roles such as chairman, lecturer, audience, etc.;
- define tasks and priorities, because different roles in different tasks have different priorities at different times; and
- add and control comments, such as comments link, and modification records.

Our MCAS system supports

- both earlier messaging and calling in time to start collaborative work;
- both late comers and early-leavers;
- four roles such as chairman, session leader, lecturer, and audience; and
- role transitions based on co-authors' requests and coordination among the co-authors.

Based on the above view sharing shown in Fig. 1, document sharing is supported by managing shared documents in shared workspaces. The WNCH model [28] is applied in MCAS to manage these documents. In this model, a document (W) object is composed of node (N) objects which may contain content (C) objects and hotspot (H) objects. A content object is used to express different media information such as text, image, audio and video which is stored in a file or a database tuple. A hotspot

object is a special area covering part of a content object and used to express a hyperlink from part of the content object to another node. Through this model, there are several sharing levels, i.e., document sharing, node sharing, content sharing, and hotspot sharing. For each type of sharing, there are different requirements for conflict resolution.

Based on a partially distributed architecture and information management model, there are two levels of conflict control shown in Fig. 2, i.e., the management level for the clients and the document level in the server.

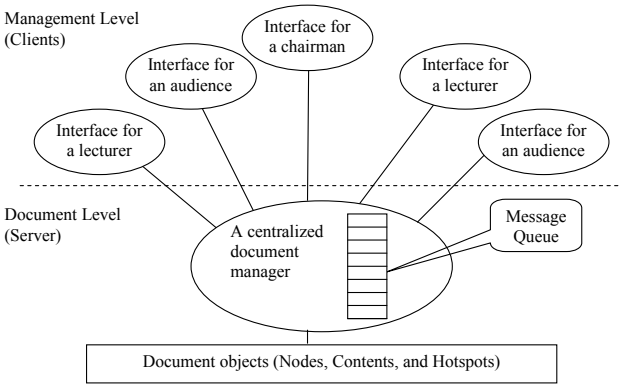

Fig. 2. Two levels of conflict resolution.

- Management Level (Section 3)

Each client is assigned with different roles to, such as chairman, session leader, lecturer and audience. This level will be implemented mainly through interface changing and deals with the operations to shared workspaces, documents and nodes.

- Document Level (Section 4)

There is a centralized information management system in MCAS, where the access messages from users will be received by a manager object before they are sent to relevant objects. By designing a message queue in the manager, conflict resolutions can be enforced by operation regulations. The manager will process the message queue by an interval set or tuned by the system administrator. This level is implemented by a manager incorporated with specific role-based regulations. This level processes the operations to contents and hotspots.

3. CONFLICTS AND RESOLUTION AT THE MANAGEMENT LEVEL

In a collaborative system, there should be task distributions and role assignments [3 and 23]. Without a role-based method, it is difficult to determine what permissions have been authorized for which user [12].

In our co-authoring system, a user is called a co-author. A co-author has a role with specific responsibilities. A role can also be considered as a version of a user object which has a scope of operations of objects in the system.

In our MCAS system, the co-authors are categorized into four kinds of roles, such as chairman, session leader, lecturer, and audience.

- A chairman can open shared workspaces and manage other roles.
- A session leader can work in a shared workspace and manage documents and nodes, but has a temporary role. A chairman is granted a default session leader.
- A lecturer can work in a shared workspace and operate in an opened node.
- An audience can only watch the collaboration and has no right to operate on the shared document. The assignment of this role is to help decrease the possibility of conflict occurrences by decreasing the number of concurrent co-authors.

Through this assumption, role management regulations in the system design are provided as follows:

- Only one chairman exists in MCAS at a time;
- Only one session leader exists in MCAS at a time;
- The chairman is granted the session leadership before assigning it to another lecturer.
- Only one session leader can manage (open, close, delete, or save) a document and a node.
- Only one document can be opened at a time;
- Only one node can be opened at a time;
- A chairman can release his or her chairman role;
- A session leader can release his or her leader role to become a lecturer;
- A session leader can apply to be a chairman;
- A lecturer can apply to upgrade to a session leader;
- A member of the audience can apply to upgrade to a lecturer;
- The chairman can approve audiences' and lecturers' applications for upgrading;
- The chairman can downgrade a session leader to a lecturer, and a lecturer to an audience member;
- The application for chairmanship from a session leader is granted when the chairman releases his or her chairmanship; and
- Non-chairman co-authors can vote to downgrade the current chairman to a lecturer and grant the current session leader the chairmanship. If there is no current session leader, the lecturer who is applying to upgrade, becomes the chairman.

As the session leader role is temporary, the three roles are described by Petri nets [12, 18 and 30] where the session leadership becomes a resource and the session leader role a state of a lecturer.

A chairman's state transition can be described by Petri nets as shown in Fig. 3. R_1 expresses the chairmanship in a collaborative system. R_2 expresses the session leadership. P_1 is a co-author waiting to take a chairman's role. P_2 is a working chairman which implicitly means a sufficient number of shared workspaces. P_3 is a chairman working

with the default session leadership. P_4 is a chairman who has released the session leadership. t_1 means an event to ask for the chairman role, t_2 means an event to grasp the session leadership, t_3 means an event to release the session leadership, and t_4 means an event to release the chairman's role. n is a positive integer that expresses the number of co-authors in the system.

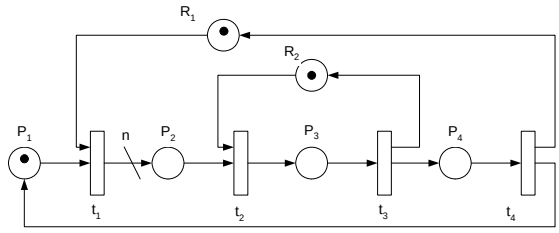

Fig. 3. A chairman's state transition in MCAS

With the same method, a lecturer is described as shown in Fig. 4, where, R_3 is replaced by P_2 to express the opened shared workspace that is created by a working chairman P_2 in Fig. 2. R_2 is the same as R_2 in Fig. 3. P_1 is a logged lecturer waiting for an available shared workspace. P_2 is the lecturer working on the shared workspace. P_3 is a working session leader. P_4 is the lecturer who has been downgraded from a session leader. t_1 means an event asking for the shared workspace, t_2 means an event asking for the session leadership, t_3 means an event to leave the shared workspace, t_4 means an event to release the session leadership, and t_5 means an event to leave the shared workspace.

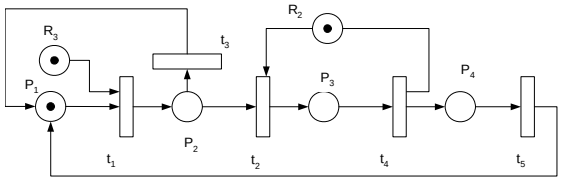

Fig. 4. A lecturer's state transition in MCAS.

In Fig. 5, an audience is depicted. R_3 is the same as that in Fig. 3 and P_1 is an audience acquiring the available shared workspace. P_2 is the audience watching the collaboration of others on the shared workspace. t_1 means an event to acquire the shared workspace and t_2 means an event to leave the shared workspace.

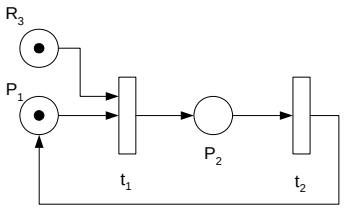

Fig. 5. An audience's state transition in MCAS

Generally, in a collaborative system, the role of a co-author is dynamically assigned. Based on collaboration situations, any co-author could have any role. In Fig. 6, the possible role transitions for a co-author are illustrated. The meanings of all the places and events are as follows.

- R_1, R_2, and R_3 are the same as those in Fig. 3
- and Fig. 4.
- P_{01} is a logged co-author.
- P_{02} is a working chairman who has opened n shared workspaces for all co-authors; P_{03} is a chairman working on the shared workspace and holding the session leadership; P_{04} is a chairman who has released the session leadership; t_{01} means an event to ask for the chairman role, t_{02} means an event to grasp the session leadership, t_{03} means an event to release the leadership, and t_{04} means an event to release the chairman role.

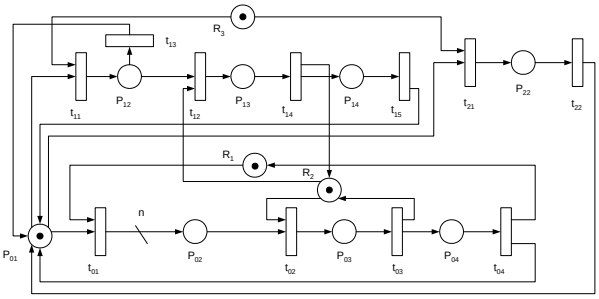

Fig. 6. Role transition of a co-author.

- P_{12} is a lecturer working in a shared workspace; P_{13} is a session leader; P_{14} is a lecturer who has released the session leadership; t_{11} means an event acquiring the shared workspace; t_{12} means an event acquiring the session leadership; t_{13} means an event to leave the shared workspace, t_{14} means an event to release the session leadership, and t_{15} means an event to leave the shared workspace.
- P_{22} is the audience acquiring the collaboration of others on the shared document; t_{21} means an event waiting for the shared workspace, and t_{22} means an event to leave the shared workspace.
- n is a positive integer which expresses the number of co-authors in the system.

Generally in a collaborative system, the roles of collaborators are dynamically assigned. Any collaborator could be any role based collaboration situations. In Fig. 7, the possible role change for a collaborator is portrayed. The meanings of all the places and events are as follows.

- R_1 expresses the only chairman position in collaboration; R_2 expresses the shared document and R_3 expresses an available shared workspace that is created by another collaborator with a chairman role.
- P_{01} is a logged collaborator.
- P_{02} is a working chairman who has opened n shared workspaces for all collaborators; P_{03} is a chairman working on the shared workspace and holding the saving right on the shared document; P_{04} is a chairman who has released the shared

document saving right; t_{01} means an event to ask for the chairman role, t_{02} means an event to grasp the shared document saving right, t_{03} means an event to release the right, and t_{04} means an event to release the chairman role.

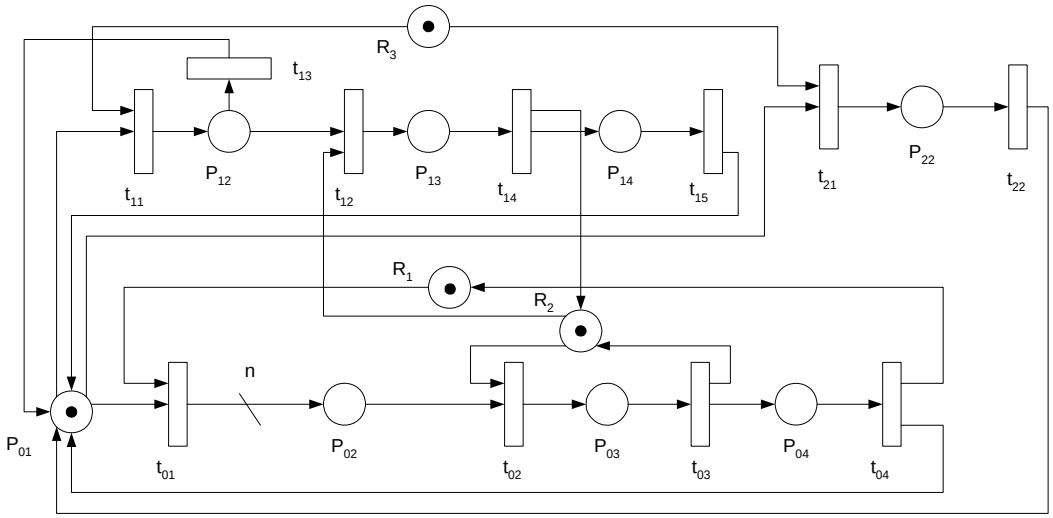

Fig. 7. Role transition of a Collaborator.

- P_{12} is a lecturer working on the shared document; P_{13} is a lecturer holding the shared document saving right; P_{14} is a lecturer who has released the right; t_{11} means an event waiting for the shared workspace; t_{12} means an event acquiring the saving right of the shared document; t_{13} means an event to leave the shared workspace, t_{14} means an event to release the saving right, and t_{15} means an event to leave the shared workspace.
- P_{22} is the audience watching others' collaboration on the shared document; t_{21} means an event waiting for the shared workspace, and t_{22} means an event to leave the shared workspace.
- n is a positive integer that expresses the number of collaborators.

What have been discussed above are only the cases of separate roles. What is the situation in such a system when many different roles are collaborating?

Suppose there are five collaborators who have definite roles, one chairman, two lecturers, and two audiences. The Petri nets are shown in Fig. 8. The meanings of places and transitions of the Petri nets in Fig. 8 are as follows:

- C_0 is a chairman; C_1 and C_2 are lecturers; and C_3 and C_4 are audiences.
- R_1 expresses the only chairman position in collaboration; and R_2 expresses the shared document.
- P_{01} is a logged collaborator waiting to become a chairman; P_{02} is a working chairman who has opened a shared workspace and this also expresses the availability of the shared workspace; P_{03} is a chairman working on the shared workspace and holding the saving right on the shared document; P_{04} is a chairman who has released the shared document saving right; t_{01} means an event to ask for the chairman role, t_{02} means an event to grasp the shared document saving right, t_{03} means an event to release the right, and t_{04} means an event to release the chairman role.

P_{11} and P'_{11} are lecturers waiting the available shared workspace; P_{12} and P'_{12} lecturers working on the shared document; P_{13} and P'_{13} are lecturers holding the shared document saving right; P_{14} and P'_{14} are lecturers who have released the right; t_{11} and t'_{11} mean events waiting for the shared workspace; t_{12} and t'_{12} mean events acquiring the saving right of the shared document; t_{13} and t'_{13} mean events to leave the shared workspace, t_{14} and t'_{14} mean events to release the saving right and t_{15} and t'_{15} mean events to leave the shared workspace.

P_{21} and P'_{21} are logged audiences waiting the available shared workspace; P_{22} and P'_{22} are the audiences watching others' collaboration on the shared document; t_{21} and t'_{21} mean events waiting for the shared workspace, t_{22} and t'_{22} mean events to leave the shared workspace.

When several collaborators are working at the same time, the transitions of roles lead to a very large Petri net. To save spaces, the Petri nets shown in Fig. 9 only deal with two collaborators, the meanings of places and transitions are the same as those in Fig. 8 except that the meanings of P'_{ij} are the same as those of P'_{ij} and the meanings of t'_{ij} are the same as those of t'_{ij} (i= 0, 1, 2 and j = 1, 2, 3, 4, 5).

In Fig. 8 and Fig. 9, there is only one collaborator who is the chairman, and lecturers must wait for the resources created by the chairman. This means that the chairman will open a shared space seen by all the collaborators in MACS that supports real view sharing.

At the management level, many critical operations such as open, save, and delete are allowed only one role at one time.

Therefore, no conflict occurs because there is only one operation at one time.

Through role assignment and operation enabling and disabling regulations, the serialization of operations issued by different co-authors is implemented and conflicts are avoided among operations at the management level.

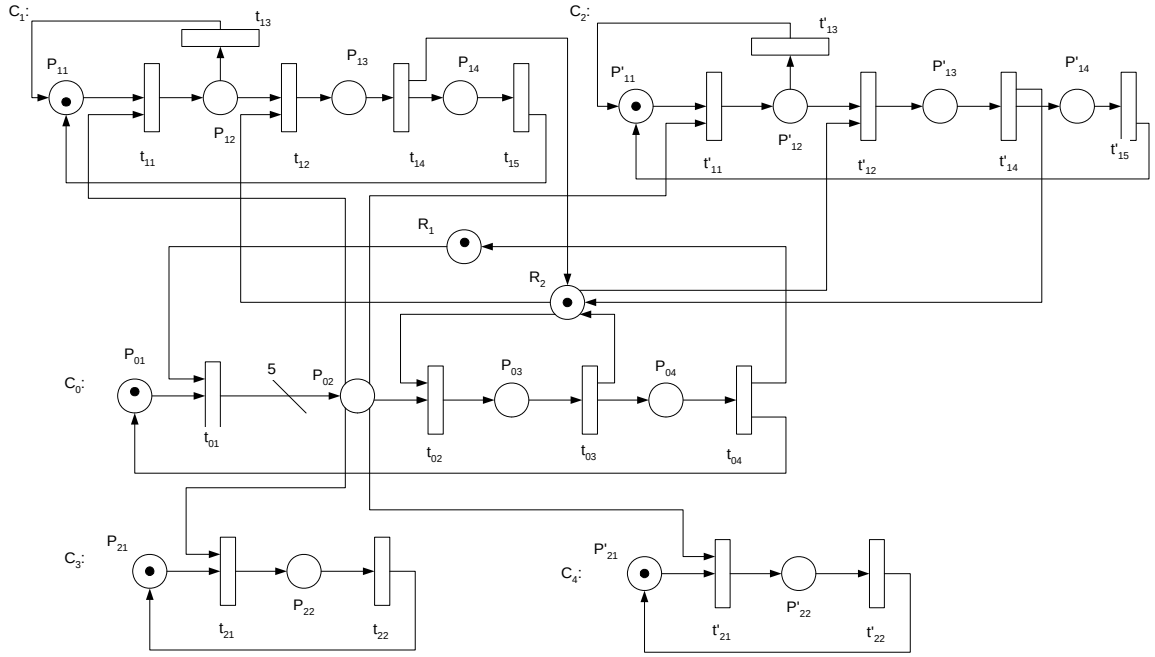

Fig. 8. A Collaboration Situation with one Chairman, two Lecturers, and two Audiences.

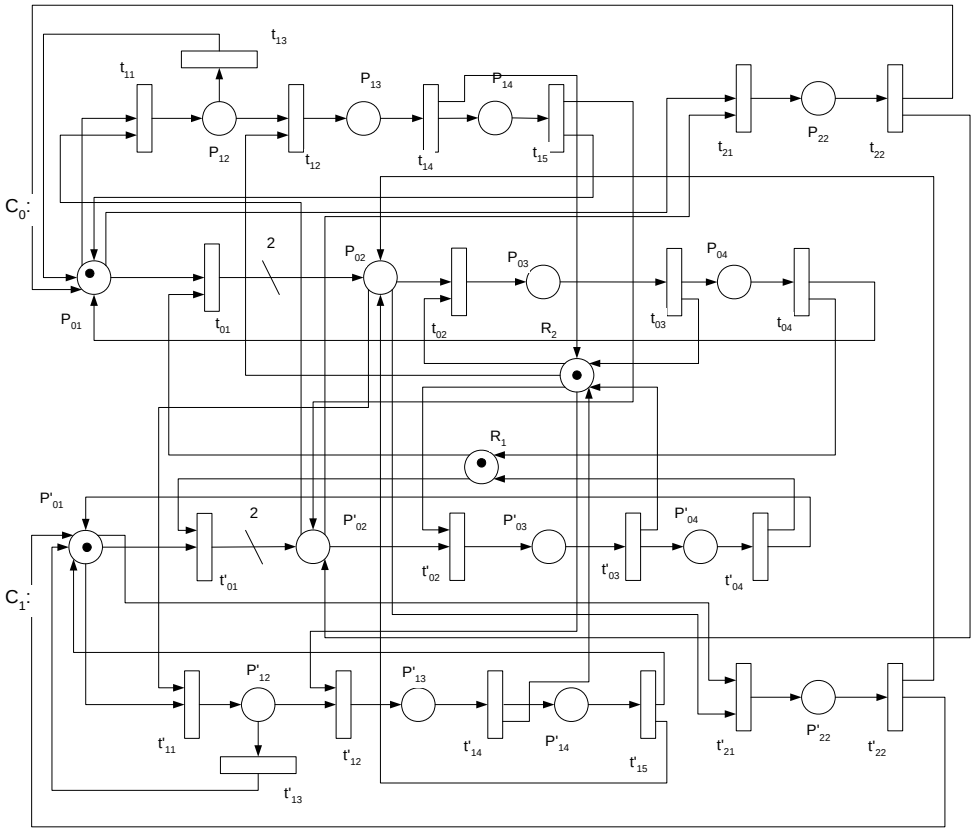

Fig. 9. A Collaboration Situation with Two Collaborators.

4. CONFLICTS AND RESOLUTION AT THE DOCUMENT LEVEL

There are several contributions about document level conflict resolution [4, 13, 16 and 21]. They are mainly concerned with operations on shared objects and their major solutions are dealing with operational transformations.

A conflict is defined simply as a special state that exists between two actions that have been applied, perhaps tentatively, to the artifact [4]. According to the ideas of object-orientation, a conflict can be defined as a state of an object which accepts two messages that are not consistent.

Suppose there are two operations O_1 and O_2, a conflict between these two occurs if and only if [21]:

(1) O_1 is independent of O_2;

(2) The object receiving O_1 is the same as that receiving O_2; and

(3) The result of O_1 is different with that of O_2.

By an object-oriented view, operation O can be defined as a triple, that is,

O ::= <R, M, A>, Where

- R expresses the receiver of the message related to this operation;
- M is the message name; and
- A is the argument attached with the message.

Hence, a conflict between two operations O_1 and O_2 occurs when $O_1.R = O_2.R$, $O_1.M \neq O_2.M$ or $O_1.M = O_2.M$ but $O_1.A \neq O_2.A$.

With this definition, there is a typical conflict of O_1 and O_2, where O_1 = <G, moveTo, p<x_1, y_1>> and O_2 = <G, moveTo, p<x_2, y_2>> when $x_1 \neq x_2$ or $y_1 \neq y_2$, and p<x, y> means a position with coordinates x and y. Here, $O_1.M = O_2.M$ but $O_1.A \neq O_2.A$.

Another typical conflict between O_1 and O_2 is when O_1 = <G, moveTo, p<x_1, y_1>> and O_2 = <G, delete, Null>. Here, $O_1.M \neq O_2.M$.

Through the role mechanisms introduced above, operation O can be redefined as a quadruple by the following:

O ::= <R, M, A, S>, Where

- R, M, and A have the same meanings as above; and
- S means the sender of the message, and there is role information in object S.

By this definition, a conflict between two operations O_1 and O_2 occurs when $O_1.R = O_2.R$, $O_1.S \neq O_2.S$, $O_1.M \neq O_2.M$ or $O_1.M = O_2.M$ but $O_1.A \neq O_2.A$.

The above conflict can be expressed with O_1 and O_2 where O_1 = <G, moveTo, p<x_1, y_1>, S_1> and O_2 = <G, moveTo, p<x_2, y_2>, S_2> when $S_1 \neq S_2$, $x_1 \neq x_2$ or $y_1 \neq y_2$.

Also by this definition, two different operations sent by the same user are not taken as conflicts. Therefore, if there is only one user playing a role, there is no conflict for this role.

From the above definition, if there are definite roles limited in a definite context, conflicts can be avoided by specific regulations. This is demonstrated by the discussion in Section 3.

Conflict management is performed in response to detected conflicts [4], therefore a manager object is used to detect conflicts in system MCAS. In the MCAS system, a time interval is set for the manager to process a message queue. In the message queue, messages created by operations of co-authors are stored in the format of the above definition, i.e., *<R, M, A, S>*. Two operations in different processing intervals of the manager will not create conflicts due to the operations in the former interval having been permanently completed before the operations in the latter interval. The following regulations are used when the manager detects conflicts.

- Only the save operation can change a node permanently.
- More than one co-author can add new content to the current node in the shared workspace because these operations are not in conflicting situations.
- For moving and resizing operations to content and hotspot objects, there are many different moving operations for each content or hotspot object.
  - If there is a session leader, the result will be the leader's operation. For example, suppose there is a conflict with two operations O_1 and O_2 where O_1 = <G, moveTo, p<x_1, y_1>, S_1> and O_2 = <G, moveTo, p<x_2, y_2>, S_2> when $S_1 \neq S_2$, $x_1 \neq x_2$ or $y_1 \neq y_2$. If S_1.role is a session leader (Note that a chairman might be the default session leader), O_1 is accepted and O_2 is discarded. Suppose there is a conflict with two operations O_1 and O_2 where O_1 = <G, moveTo, p<x_1, y_1>, S_1> and O_2 = <G, resize, p<x_2, y_2>, S_2> where p<x_2, y_2> means the new bottom-right of a rectangle, when $S_1 \neq S_2$, $x_1 \neq x_2$ or $y_1 \neq y_2$. If S_2.role is a session leader, O_2 is accepted and O_1 is discarded.
  - If there is no session leader, the result will be the last lecturer's operation upon the receiving timestamps by the manager. Lamport's algorithm [15] is used to deal with the synchronization of clocks, i.e., logical clocks.
- For operations from the same roles, any operation is discarded if there is a deleting operation ahead because of no receiving object.

For example, suppose there is a conflict with two operations O_1 and O_2 where O_1 = <G, moveTo, p<x_1, y_1>, S_1> and O_2 = <G, delete, Null, S_2> when $x_1 \neq x_2$ or $y_1 \neq y_2$. The manager will check the senders' roles.

- If S_1.role is a session leader, O_1 is accepted and O_2 is discarded.
- If S_2.role is a session leader, $O_2$is accepted and O_1 is discarded.

- If S_1.role and S_2.role are both lecturers, the manager checks the operation arriving timestamps. If O_2 is earlier, O_2 is accepted and O_1 is discarded; if O_1 is earlier, O_2 is accepted because it is a deletion.

The accessing shared document can be formalized by a Petri net as shown in Fig. 10. C_0, C_1, C_2, and C_3 express the four collaborators, P_{1i} means a collaborator who is applying the shared document, and P_{2i} means a collaborator working on it (i = 0, 1, 2, 3). t_{1i} means an event acquiring the document, and t_2 means an event releasing it (i = 0, 1, 2, 3). R means the document, and one dot in it means there is only one shared document in the system.

If a role is considered as a resource, a chairman role registration can also be expressed in Fig. 10. The meanings of the places and transitions in Fig. 10 can be reassigned as follows (i = 0, 1, 2, 3):

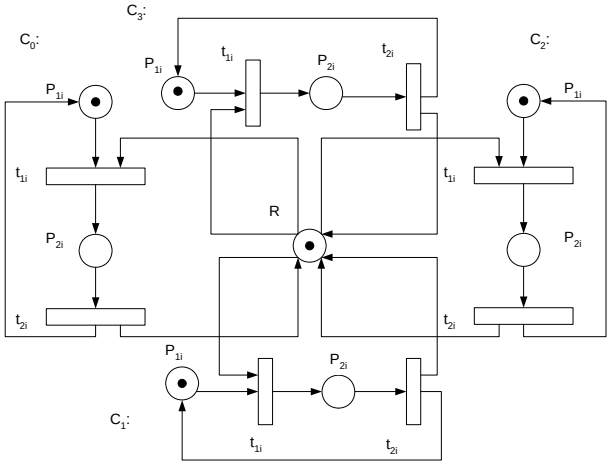

Fig. 10. Shared document and the Chairman Role.

- C_i expresses the collaborators;
- P_{1i} means a registered collaborator waiting for a chairmanship;
- P_{2i} means a collaborator acting as a chairman;
- t_{1i} means asking for the chairmanship;
- t_{2i} means releasing the chairmanship; and
- R means the chairman role; one dot in it means there is only one chairman at one time in the collaboration.

From the Petri nets, it is needed to note that there might be a starvation situation if one collaborator always grasps the shared document or the chairman role. This demonstrates the importance of negotiation by other video and audio tool to avoid the starvation. The collaborators can form regulations how to share the shared document, and there might be one person should be always the chairman.

The Petri net model in Fig. 10 clearly shows that the system allows one person changing his/her role from chairman to another.

5. IMPLEMENTATION

The implementation of MCAS system's interfaces is in a window style with windows, dialogue boxes, buttons, menus and menu bars.

At the management level, different interfaces for the different default roles are designed specially for co-authors. All co-author's interfaces are similar, but different users with different roles will view different menus or menu items in a window.

The interface is dynamically changed when the co-authors change their roles. This is implemented by enabling or disabling some operation menu items in the menu bars of a window.

The objects, operations and their meanings managed at this level are as follows.

Shared Workspace:

- Open: open shared workspaces for co-authoring. This means there is a workspace on each client controlled by the chairman; and
- Close: close all the shared workspaces at all clients.

Document:

- New: create a new document for co-authoring;
- Open: open an existing document;
- Save: save the current document;
- Delete: delete the current document. This operation only deletes the structuring information among nodes related to this document, and the nodes related to this document will be kept and used for other documents; and
- Close: close the current document.

Node:

- Create: create a new node;
- Structure: link the current node to the current document;
- Open: open an existing node;
- Save: save the current nodes;
- Delete: delete the current node. This operation does not delete the content objects related to this node, and they are still used for creating other nodes. Because hotspot objects are related to a special part of a content object belonging to a node, they are deleted; and
- Close: close the current node.

At the document level, a document contains only node objects. The interface will be enabled when a document and a node are opened. The objects, operations and their meanings are as follows.

Content:

- Add: add a new content object to the current node;
- Delete: delete the selected content object;
- Move: move the selected content to a new position which is decided by a new position for the object's top-left point of a rectangle.

- Resize: resize the selected content which is decided by a new position for the object's bottom-right point of a rectangle.

Hotspot:

- Add: add a new hotspot object to the current node;
- Delete: delete the selected hotspot object;
- Move: move the selected hotspot to a new position which is decided by a new position for the object's top-left point of a rectangle.
- Resize: resize the selected hotspot which is decided by a new position for the object's bottom-right point of a rectangle.
- Hyperlink: link another node to the selected hotspot.

With the regulations discussed in Sections 3, a multi-interface is implemented with the help of logged role information. If a co-author changes his or her role, the operation menu will be changed by disabling or enabling some specific operation menu items. With the regulations discussed in Section 4, a manager object is designed to detect and resolve conflicts.

6. CONCLUSIONS

By using role management and role regulations, the task of conflict resolution can be simplified. Many possible conflicts are avoided by the role management and regulations programs. The role mechanisms are a practical way to support conflict resolution in collaborative systems. From what have been discussed, it is valuable to point out that this successful practice comes from the combination of a document management model WNCH, a partially distributed architecture, and a group of role regulations or policies. From a systems engineer's view, a stand-alone technology or methodology would not be successful unless in combination with other related ones. Role mechanisms would have the same results.

It is needed to clarify that any method may have pros and cons. The role management methods could affect the system concurrency to some extent. It is recommended that the designers and builders of collaborative systems should make a trade-off between role management and concurrency requirements based on the special requirement of the system.

Role management is a complex task that must be paid more attention. The more roles there are, the more complicated the Petri nets are for describing transitions among roles, and the more complicated algorithms and regulations are to solve conflict situations. Therefore, there should not be too many roles in one situation at one time in a system, i.e., roles must be managed with separate times and contexts.

ACKNOWLEDGEMENT

The authors would like to thank the comments provided by the anonymous reviewers and editor, which help the authors improve this paper significantly.

REFERENCES

[1] Barber, K. S., Liu, T. H., Ramaswamy, S., "Conflict Detection during Plan Integration for Multi-Agent Systems", *IEEE Trans. On Systems, Man, and Cybernetics- Part B: Cybernetics*, Vol. 31, No. 4, Aug. 2001, pp. 616-628.

[2] Begole, J., Rosson, M. B. and Shaffer, C. A., "Flexible Collaboration Transparency: Supporting Worker Independence in Replicated Application-Sharing System", *ACM Transactions on Computer-Human Interaction*, Vol. 6, No. 2, 1999, pp. 95-132.

[3] Dewan, P., and Shen, H., "Controlling access in multiuser interfaces", *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 5, No. 1, March 1998, pp. 34 – 62.

[4] Edwards, W. K., "Flexible Conflict Detection and Management in Collaborative Applications", *UIST'97*, Alberta, Canada, 1997, pp. 139-148.

[5] Edwards, W. K., "Policies and Roles in Collaborative Applications", *CSCW'96*, Cambridge, USA, 1996, pp. 11-20.

[6] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., and Chandramouli, R., "Proposed NIST Standard: Role-Based Access Control", *ACM Transactions on Information and system Security*, Vol 4, No 2, Aug. 2001, pp. 224-274.

[7] Gutwin, C. and Greenberg, S., "The Effect of Workspace Awareness Support on the Usability of Real-Time Distributed Groupware", *ACM Transactions on Computer-Human Interaction*, Vol. 6, No. 3, 1999, pp. 243-281.

[8] Guzdial, Mark , Rick, Jochen, and Kerimbaev, Bolot, "Recognizing and supporting roles in CSCW", *Proceeding of the ACM 2000 Conference on Computer supported cooperative work*, Philadelphia, Pennsylvania, United States, December 2000, pp. 261-268.

[9] http://csrc.nist.gov/rbac/.

[10] Jaeger, T., Edwards, A., and Zhang, X., "Managing Access Control Policies Using Access Control Spaces", *SACMAT'02*, Monterey, California, USA, June 3-4, 2002, pp. 3-12.

[11] Jeffrey, P. and McGrath A., "Sharing Serendipity in the Workplace", *Proceedings of the conference on Collaborative Virtual Environments (CVE 2000)*, San Francisco, CA, USA, 2000, pp. 173-179.

[12] Jeffrey, J. M., "Using Petri Nets to Introduce Operating System Concepts", *ACM SIGCSE Bulletin,* Vol. 23, No. 1, 1991, pp. 324-329.

[13] Jung, H., Tambe, M., and Kulkarni, S., "Argumentation as distributed constraint satisfaction: applications and results", *Proc. of the fifth international conference on Autonomous agents* , 2001 , Montreal, Quebec, Canada, pp. 324 – 331.

[14] Kern, A., Kuhlmann, M., Schaad, A., and Moffett, J., " Role Engineering: Observations on the role life-cycle in the context of enterprise security management", *Seventh ACM Symposium on Access Control Models and Technologies 2002* , Monterey, California, USA, 2002, pp. 43-51.

[15] Lamport, L., "Time, Clocks, and the Ordering of Events in a Distributed System", *Communication of the ACM*, Vol. 21, No. 7, July, 1978, pp. 558-564.

[16] Nyanchama, M. and Osborn, S., "The role graph model and conflict of interest", *ACM Transactions on Information and System Security (TISSEC),* Vol. 2, No. 1, February 1999, pp. 3-33.

[17] Park, J. S., Sandhu, R. and Ahn, G. J., "Role-based access control on the web", ACM Transactions on Information and System Security (TISSEC), Vol. 4, No. 1, February 200, pp. 37-71.

[18] Peterson, J. L., "Petri Nets", *Computing Surveys*, Vol. 9., No. 3., Sept. 1977, pp. 223-252.

[19] Poole, M. S., Homes, M., DeSanctis, G., "Conflict management and group decision support systems", *Proceedings of the conference on Computer-supported cooperative work*, Portland, Oregan, USA, January 1988, pp. 227-243.

[20] Steinfield, C., Jang C. Y. and Pfaff, B., "Suppoting Virtual Team Collaboration: The TeamSCOPE System", *GROUP'99*, Phoenix, Arizona, USA, 1999, pp. 81-90.

[21] Sun, C., and Chen, D., "Consistency Maintenance in Real-Time Collaborative Graphics Editing Systems", *ACM Transactions on Computer-Human Interactions*, Vol. 9, No. 1, March 2002, pp. 1-41.
[22] Zhu, H., *Object Oriented Technology: Principles and Designs*, Publishing House of Changsha Institute of Technology, Oct. 1992.
[23] Zhu, H., Cai, K., Fan, A. and Song, H., *Principles and Designs of Distributed Systems*. Publishing House of Changsha Institute of Technology, Sept. 1997.
[24] Zhu, H., Turoff, M. and Li, Z., "An Object Model for Collaborative systems and a Toolkit to Support Collaborative Activities", *Proceedings of the 2000 American Conference on Information Systems (AMCIS'00)*, Long Beach, USA, Aug. 2000, pp. 590-594.
[25] Zhu, H., Wang, P., and Hu, S., "CoWNCH: An Object-Oriented Model for Computer Supported Hypermedia Co-Authoring", *Proc. of 1995 European Simulation Symposium (ESS'95*), Germany, Oct., 1995.
[26] Zhu, H., Wang, P., and Hu, S., "The Design of a Multimedia Co-authoring Platform based on Client/Server Model", *ACTA Electronica Sinic*a, Vol. 25, No. 5, 1997.
[27] Zhu, H., Wang, P., and Hu, S., "A Multimedia Co-authoring System Object Model AMWD/RSEI", *Chinese J. of Computer,* Vol. 20, Suppl., 1997.
[28] Zhu, H., Wang, Pu, and Hu, S., "An Object-Oriented Multimedia Management Model WNCH", *Software Journal,* Vol. 7, Suppl., 1996.
[29] Zhu, H., Yang, G., and Liu, Z., Object Oriented Principle and its Applications, *Publishing House of Changsha Institute of Technology*, Sept. 1998.
[30] Zhu, H. and Zhou, M., "Formalizing the Design of a Collaborative System", *Proceedings on IEEE International Conference on Systems, Man, and Cybernetics (SMC'02)*, Tunisia, Oct. 2002.

Haibin Zhu received B.S. degree in computer engineering in 1983 from Institute of Engineering and Technology, China, and M.S. and Ph.D. degrees in computer science in 1988 and 1997 from the National University of Defense Technology (NUDT), China. He is an Assistant Professor of the Department of Computer Science and Mathematics, Nipissing University, Canada.

He was a visiting professor and a special lecturer in the College of Computing Sciences, New Jersey Institute of Technology, USA from 1999-2002. He was a lecturer, an associate professor and a full professor from 1988 to 2000 at NUDT. He has published more than 50 papers, three books and one book chapter on object-oriented programming, distributed systems, collaborative systems and computer architecture.

He served as a program committee member for the 2005, 2004, and 2003 Int'l Conference on Systems, Man & Cybernetics (ICSMC'05, '04, '03) and the 2004 Canadian Conference on Computer and Software Engineering Education (C3SEE'04).

He is the receipt of the Best Paper Award from the 11th (International Society of Production Enhancement) ISPE International Conference on Concurrent Engineering (ISPE/CE2004), the 2004 and 2005 IBM Eclipse Innovation Grant Awards, the Educator's Fellowship of OOPSLA'03, a 2nd Class Nation-Level Award of Education Achievement for contributions to computer education from Ministry of Education of China (1997), a 2nd Class Nation-Level Award of Excellent Textbook from the Ministry of Education of China (2002), three 1st Class Ministry-level Research Achievement Awards from DOD of China (1997, 1994, and 1991), and a 2nd Class Excellent Textbook Award of the Ministry of Electronics Industry of China (1996).

Dr. Zhu is a senior member of IEEE, a member of ACM and a life member of the Chinese Association for Science and Technology, USA.