

From WYSIWIS to WISINWIS: Role-Based Collaboration^{*}

Haibin Zhu, Senior Member, IEEE

Nipissing University

100 College Drive, North Bay, ON P1B 8L7, Canada

haibinz@nipissingu.ca

Abstract - *Traditional CSCW (Computer-Supported Cooperative Work) systems concentrate WYSIWIS (What You See Is What I See) in order to support as many facilities as those in face-to-face collaboration environments, however, many obstacles made these efforts not so successful as was expected. Therefore, we need to consider the other aspects of a CSCW system that can better support collaboration among people, that is, collaboration might be made more productive and more efficient by special computer systems facilities such as role specification. This paper reviews the CSCW systems with concentration on the group awareness support, proposes a new classification of group awareness and illustrates that in collaborative systems WYSINWIS (What You See Is Not What I See) is as important and constructive as WYSIWIS. Then, it proposes role-based collaboration and role specification mechanisms that can be used to a new method to support WYSINWIS.*

Keywords: WYSIWIS, WYSINIS, Role, Role Specification, Role-Based Collaboration.

1 Introduction

People cooperate in order to obtain better or more productive results than individuals do. People also hope to collaborate even when they are at different places and face-to-face collaboration is impossible. That is why we build CSCW systems. There are two kinds of CSCW systems: synchronous and asynchronous systems, where synchronous systems mean that the users use them at the same time and the asynchronous systems mean that users can use them at different times.

Traditional synchronous CSCW systems concentrate more on WYSIWIS or on some kind of collaboration awareness among and between cooperating users [22]. They hope to support people to cooperate on a computer-supported system in the same way as that in the face-to-face environment. Therefore, the technologies required in such systems are real-time group communication, shared views, multi-user interfaces, telepointers, shared applications and shared workspaces [29]. Even further, CVE

(Collaborative Virtual Environment) systems [3] are proposed to simulate face-to-face meeting environment called VR (Virtual Reality) [17]. Microsoft NetMeeting [16], GroupKit [21], MCAS [34] and GAZE [29] are good prototype examples for synchronous CSCW systems. Generally speaking, synchronous CSCW systems support or aim to support WYSIWIS.

However, the applications of CSCW systems are not so successful as what people expected [14], because synchronous CSCW systems cannot be satisfied by the users who want to have a face-to-face-like collaboration on networked computer systems. They encounter many problems such as unsatisfactory communication, frustrated waiting, uneasy environments for discussions and complex operations. Collaboration with traditional CSCW systems is often clumsy, not practical, and frustrating compared to face-to-face collaboration [14]. In reality, synchronous CSCW systems can never provide an environment that can be used to collaborate in the exact same way as that in face-to-face environments. Since relaxed WYSIWIS was proposed [25], many researches and practices were conducted and published to support relaxed WYSIWIS [2, 7, 8, 13, 20]. Relaxed-WYSISWIS view sharing makes groupware more flexible and better matches the way people actually work, especially in large workspaces that contain many artifacts [7]. Colab [25] and DCWA [8] are good examples for systems with relaxed WYSIWIS.

At the same time, asynchronous CSCW systems, CMC (Computer-Mediated Communication) systems [26], or GSSs (Group Support Systems) [27] hope to provide extensions to face-to-face collaboration that is available at any time and any place. They hope to support contributions among users by removing the restrictions of same time conducted by the synchronous collaborative system and providing more acceptable coordination mechanisms [23, 26, 27, 35]. When collaborators cannot collaborate in a face-to-face style, they can collaborate in an asynchronous way. Asynchronous systems provide new environment that can accommodate more creative collaborations than a face-to-face environment. Email systems, BBSs (Bulletin Board System), WebCT [18] and

^{*} 0-7803-8566-7/04/\$20.00 © 2004 IEEE.

WebBoard [15] are good examples of asynchronous systems.

CSCW (Computer-Supported Cooperative Work) systems are computer-based tools that support collaborative activities. They should meet the requirements of normal collaborative efforts among people. They should not only support virtual face-to-face collaboration among people at a distance but also improve face-to-face collaboration by providing more mechanisms to overcome the drawbacks of face-to-face collaboration among people. A role is a set of prescriptions defining how a collaborator should behave in collaboration [4]. Unclear role specification may create dysfunctional ambiguity and conflict in an organization [5]. Therefore, introducing roles into CSCW systems is extremely important. By roles, we can avoid users of CSCW systems being overwhelmed by too much irrelevant information.

Some traditional CSCW systems, or groupware, indeed applied the concept of roles [1, 6, 10, 19, 24, 26]. Some of asynchronous systems use role concepts to show different interfaces or information to different users with different roles such as administrator, students, instructors or others to facilitate different responsibilities of users. Actually, they are the first steps of WYSINWIS.

However, they did not produce exact and clear role definition and specification. As a result, there is no commonly accepted concept of roles; there are no definite methods to express a role that is dynamically changeable in a collaborative system. Traditional CSCW applications can only support pre-defined or static roles [10] and the users have no supporting facility to tune their roles in order to make collaboration more productive and more efficient.

This paper is arranged as follows. The second section discusses the fundamental requirement of group awareness in collaborative systems and reviews the evolving from WYSIWIS to WYSINWIS and states that WYSINWIS can also support awareness among collaborators efficiently. The third section discusses what we mean by role-based collaboration; the fourth section proposes a new method to describe or specify a role that can be applied to support WYSINWIS in CSCW systems. The last section concludes the paper and proposes the issues that require more comprehensive research.

2 Group awareness, WYSIWIS, relaxed WYSIWIS, and WYSINWIS

“Awareness” plays an important role in CSCW systems [22]. Without awareness, there would be no collaboration. The basic meaning of awareness in collaborative systems is to know who the collaborators are, what they are doing, what they have done and what they want to do. Hence, it is also called group awareness. With group awareness, the collaborators have a basic idea of his

or her collaborating working environment that he/she is using and contributing.

Traditional CSCW systems apply four kinds of interleaved awareness [13]: informal awareness, social awareness, group-structured awareness and workspace awareness, wherein,

- Workspace awareness is about who is present in the workspace, where they are, and what they are doing;
- Informal awareness means that the general sense of who is around and what they are depending on;
- Social awareness is the information that a person maintains about others in a social or conversational context; and
- Group-structured awareness involves knowledge about such things as people’s roles and responsibilities, their positions on an issue, their status, and group process.

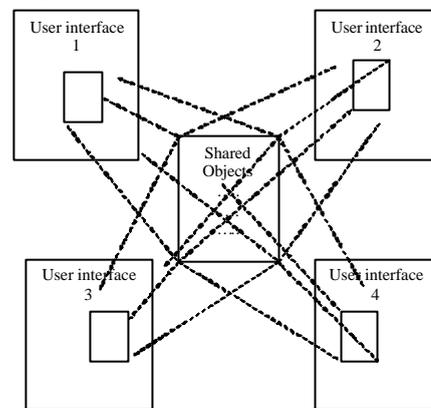


Figure 1. The Same Views to the Same Shared Object
Based on time factors, we can discuss the differences between synchronous awareness and asynchronous awareness. Synchronous awareness is directed at supporting a face-to-face environment and asynchronous awareness is directed at email systems. Evidently, workspace awareness is synchronous awareness and other three are asynchronous awareness. The informal, social and group-structured awareness are mainly used to support asynchronous awareness.

WYSIWIS means What You See Is What I See. This is the aim of most synchronous CSCW systems. Synchronous systems do not only want to support the same views to the shared documents shown in Fig. 1 but also the shared views of each collaborator, because in a face-to-face environment, all the collaborators share the views to the environment.

WYSIWIS is one of the prevalent approaches in designing systems to support synchronous interaction and cooperation, that is, it provides the users with all their individual and familiar tools [7]. WYSIWIS emphasizes that collaboration requires the commonalities among the collaborators.

WYSIWIS enforces strict synchronization of different user views onto a shared workspace. Relaxed WYSIWIS coupled with techniques for promoting multi user awareness and concurrency control mechanisms for interleaving users actions have focused on enabling the possibility of collaboration while retaining a high degree of individual autonomy. WYSIWIS supports that different users at different displays were forced to see the same part of a virtual workspace. Relaxed WYSIWIS was led by a less strictly coupled approach where different user's views could diverge [2].

Although the WYSIWIS idealization recognizes that efficient reference to common objects depends on a common view of the work at hand, WYSIWIS was found to be too limiting and relaxed versions were proposed to accommodate personalized screen layouts [20].

Relaxed WYSIWIS actually means that groupware should not always support full or strict WYSIWIS principles at all time and all places, it should have flexible choices for users to choose.

Smith et al implements their Kansas system with WYSIWITYS (What You See Is What I Think You See) [24]. They believe that people in a group play various roles and roles are supported by the system's treatment of outputs and inputs. Greenburg argues that we need to build personalized groupware in order to obtain wider acceptance of groupware. Personalized groupware should offer a system that conforms to the individual needs of participants and of groups [12]. Chang, K. H. et al mentioned a WYSINWIS style in their collaborative writing tool for user's own viewing spaces [8]. We suggest that the terminology WYSINWIS is used to emphasize the differences among peoples' requirement to different inputs and outputs.

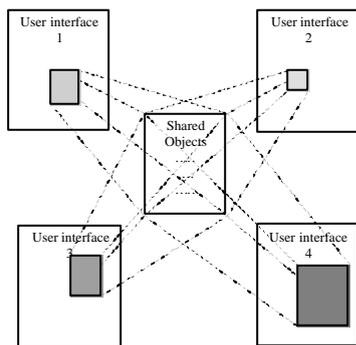


Figure 2. Different Views to the Same Shared Object

WYSINWIS stands for What You See Is Not What I See. We use this terminology to emphasize the real collaboration among people in a group take different responsibilities and have different rights. On a computer system that supports collaborative work among people should support this requirement: WYSINWIS (Figure 2).

In reality, every person has his or her view to the world. They ask, serve, and interact with the world in their different ways. WYSINWIS definitely does not want to support the explicit awareness, but it still can support collaboration in an efficient way. WYSINWIS emphasize the differences among the collaborators. WYSINWIS will use the implicit awareness to help collaborators to contribute in a collaborative work. WYSINWIS emphasizes that one user's view is not totally the same as others' view on shared objects.

Evidently, traditional CSCW systems emphasize more on WYSIWIS than WYSINWIS. WYSIWIS is the aim of explicit awareness. The problem is that we still need to group awareness when WYSIWIS cannot be practically applied. Therefore, relaxed WYSIWIS, WYSIWITYS, and WYSINWIS are proposed to the CSCW world. They are all mechanisms that can be used to support implicit awareness. Actually, WYSINWIS touches another extreme point that emphasizes the personalized view instead of shared view and role-based collaboration is the direct way to support it.

We can describe the spectrum of group awareness mechanisms as shown in Figure 3. From the spectrum, we can compare WYSIWIS and WYSINWIS in the following aspects:

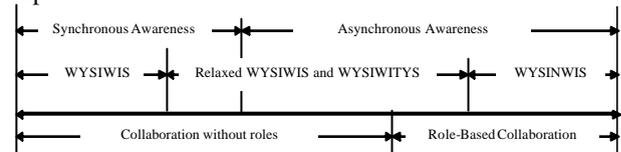


Figure 3. The Awareness Spectrum in Collaborative Systems

- WYSIWIS emphasizes commonalities requirements in collaborative environments and WYSINWIS emphasizes the personalities in the working environments.
- WYSIWIS emphasizes the benefits of face-to-face meeting and brainstorming for creative collaboration and WYSINWIS emphasizes the benefits of efficient working when collaborators have clear rights and responsibilities.
- WYSIWIS still has a long way to be satisfied by the users and WYSINWIS has just begun noticed and has many potential applications.
- WYSIWIS coupling has two main disadvantages. It can lead to wars whenever users are forced to share changes they do not want to share. WYSIWIS requires the overhead of communicating all user changes to all of the other users, which is inefficient, specially when the users do not wish to share some of these changes. WYSINWIS does not have these disadvantages but does not have any of the advantages of WYSIWIS either [9].

Either WYSIWIS or WYSINWIS has its pros and cons. One does not take precedence over the other. Therefore,

they are of the same importance in building collaborative systems.

3 Role-based collaboration

It is a fact that awareness and roles are intertwined [24]. Our approach is to use roles as fundamental components to build a collaborative system. In such systems, one first plays a special role and then get a special user interface. Role concepts are fundamental to support awareness in WYSINWIS.

One of the major problems in traditional CSCW systems is determining how to define and specify roles clearly and rigorously, while maintaining the flexibility for collaboration. Many traditional CSCW systems with roles have lost flexibilities because they can only provide static role mechanisms based on intuitive role concepts. There are no flexible mechanisms for role tuning, changing and transitions because we lack a special mechanism to express a role.

In reality, we can see many of these flexibilities in normal collaboration for a special person who takes a position. Generally, roles seem to be defined clearly with natural languages in many organizations. Why can a person with a definite role do more or less than another person with the same role? This situation is due to the ambiguous meaning of roles defined by natural languages. People can do more or less than what a role has specified because they have their own understanding about the role. That is why many traditional collaborative systems did not introduce exact role specification; essentially, they were afraid of losing the flexibilities of collaboration.

However, we can think in a different way. If we can build a CSCW system providing flexible role tuning and transition mechanisms based on exact role specification, then the people using such kind of CSCW systems can work efficiently with clear role specification, thus increasing productivity by active role tuning and transitions. We can also allow a user with a special role to create new objects in the system and give him or her more opportunities to contribute creative work. A creative person can do more than what a role specifies because he/she can play different roles in different situations.

Therefore, clear role definition and specification can help a person collaborate in a group [4]. If we can support both mechanisms for clear role specification and easy role tuning, changing and transitions, we can keep both the flexibility and efficiency of a CSCW system.

Role is not an isolated concept. Without the support of object technology, it would be impossible to specify a role. With object-oriented principles, we can conceptually construct the underlying components of a role-based collaborative system including classes, objects, messages, interfaces, agents, roles, groups and human users. In this

way, we can easily express a role's incoming messages with class mechanisms. Therefore, we can use a special object, such as an agent to express a human user that provides services in a collaborative system. It is a normal idea for current object-oriented methodologies to specify a role as a server. That is to say, when an object plays a role, it responds specific incoming messages relevant to this role [11, 28, 30, 31, 32, 33]. However, no object system and no object-oriented methodology can specify a role as a client with definite outgoing messages.

Based on observations of our society from the viewpoint of object-orientation, we found that, in collaboration, people generally have two kinds of existences. One is a server that expresses the responsibilities and the other a client that expresses the rights. When a person plays a role, he or she provides specific services and possesses specific rights to ask for services. In this sense, a role can be considered as a person's view to the world. When a person plays a specific role, he or she has a special view of the surroundings. A role in a collaborative environment is actually a wrapper with a service interface and a request interface [32, 33]. That is to say, we can separate a person's role into two parts: one is the service interface specifying incoming messages and the other the request interface specifying outgoing messages. Based on this view of roles we have formed our basic infrastructure for role-based collaboration [33].

4 Support WYSINWIS with roles

In role-based collaboration, human users who enter the system with initial roles may interact with the system and other users who have a clear understanding about the responsibilities and rights involved. During collaboration, they can ask for role tuning and transitions and their roles can be tuned or changed by other roles. In this way, every human user has a clear role definition, which helps him/her work efficiently. The system allows the human users to impose creative contributions through efficient role tuning and transitions.

The scenario of collaboration based on this kind of system can be described as follows.

- A collaborative system built with role mechanisms is installed on a server.
- Each human user uses an ordinary interface like a web browser to log into the system on his/her client computer with a role.
- Some human users create classes, objects, agents or roles in the system.
- A human user plays one role at one time.
- A human user may change his role based on the requirement in collaboration.

- The human users will directly use the request interfaces relevant to their roles while their agents accept messages relevant to their roles.
- Roles can be modified or tuned by some human users with specific roles based on the requirement in collaboration.
- Through the roles, while their agents provide services, the human users access objects, their agents, contact other agents, join a group, and contribute to the collaboration.
- The result of the collaboration is reflected by the states of objects and agents in the system.

To help specify roles in a collaborative system, we need to provide the following tools:

- Class management;
- Object management;
- Group management;
- Message management; and
- Role management.

In each management tool, we need to provide the following operations:

- Add: add a new class (an object, a group, a message or a role) in the system;
- Delete: delete a class (an object, a group, a message or a role) from the system and all its instances;
- Search: search a class (an object, a group, a message or a role) with key words included in class names and class description;
- Display: display a class (an object, a group, a message or a role) in a specific format; and
- Edit: change different attributes of a class (an object, a group, a message or a role).

With the management of classes, objects, groups and messages, we can define a new role by assigning messages for this role. With the management of roles, we can create interfaces by selecting messages from the interfaces as follows for a specific role. We can use hierarchical and structural methods to make the selections easily and efficiently.

- A list of all the agent classes and their interfaces;
- A list of all the other classes and their interfaces;
- A list of all the objects including agents and their interfaces; and
- A list of all the groups and their interfaces.

The key point is that we can use an internal structure to express a role. With this internal structure, we can form dynamic interfaces. With the help of roles, we can get a software system that will show different interfaces to different users depending on their roles. We can improve all the four kinds of awareness mentioned in section 2 (Figure 4).

By these mechanisms, we can conceive that we implement WYSINWIS by providing a totally customizable user interface for every different user. The video and audio information can be shown on the interfaces by video and audio objects accessed by some roles.

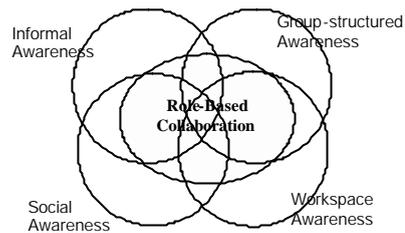


Figure 4. Different kinds of awareness and role-based collaboration

5 Conclusions

Collaborative systems should not only support virtual face-to-face collaboration among people at a distance but also improve face-to-face collaboration by providing more mechanisms to overcome the drawbacks of face-to-face collaboration among people.

Introducing roles into CSCW systems is very important. Using roles, we can avoid overwhelming users of CSCW systems with too much irrelevant information. Roles are fundamental mechanisms that support practical WYSINWIS requirement in collaboration.

Role-based collaboration imposes challenges not found in traditional CSCW systems. The achievement obtained from our research on role-based collaboration will bring exciting improvements to both the development and application of CSCW systems.

WYSINWIS plays an important role in building collaborative systems. It emphasizes that collaborators take different views with the collaborative systems when they play different roles. We need to pay more attention to the mechanisms that support this view requirement. The role specification mechanism is one proposed method to support this idea. To conclude, a role decides what a user can see on the interface. To really accomplish the tasks to support WYSINWIS thoroughly, we need to clarify the relevant issues as follows.

- Creation of an efficient and practical role specification mechanism including tools to help role facilitators to assign objects that might be accessed by a specific role;
- Creation of efficient and practical role presentation mechanism that help user be aware about the objects in collaboration required for him or her; and
- Development of an adaptive and automatic interface producer to reflect smoothly the role transition and role tuning.

Acknowledgement

This research is supported by 2004 IBM Eclipse Innovation Grant.

References

- [1] Becht, M., Gurzki, T., Klarmann, J. and Muscholl M., ROPE: Role Oriented Programming Environment for Multiagent Systems, *4th IECIS International Conference on Cooperative Information Systems*, Sept. 1999, pp325-333
- [2] Benford, S., Bederson, B. B., et al, "Designing Storytelling Technologies to Encourage Collaboration Between Young Children", *The SIGCHI conference on Human factors in computing systems table of contents*, The Hague, The Netherlands, 2001, pp556 - 563
- [3] Benford, S., Greenhalgh, C., Rodden, T. and Reocke, J., "Collaborative Virtual Environment", *Communications of the ACM*, vol. 44, no.7, July, 2001, pp79 - 85
- [4] Biddle, B.J. and Thomas, E.J.(Ed), *Role Theory: Concepts and Research*, John Willey & Sons, Inc., 1966
- [5] Bostrom, R. P., "Role conflict and ambiguity: Critical variables in the MIS user-designer relationship", *Proceedings of the 17th annual computer personnel research conference*, 1980, Miami, Florida, United States, pp 88 - 115
- [6] Botha, R. A. and Eloff, J. H. P., "Designing Role Hierarchies for Access Control in Workflow Systems", *25th Annual International Computer Software and Applications Conference (COMPSAC'01)*, October 2001, pp. 117-122
- [7] Carstensen, P. H. and Schmidt, K., "Computer Supported Cooperative Work: New Challenges to Systems Design", in *Kenji Itoh (ed.) Handbook of Human Factors*, Tokyo 1999
- [8] Chang, K. H., Gong, Y., Dollar, T., Gajiwala, S., Lee, Byong and Wear A. W., "On Computer Supported Collaborative Writing Tools for Distributed Environments", *ACM 23rd Conference on Computer Science*, Feb. 1995, pp222-229
- [9] Dewan, P. and Choudhary, R., "An Editing-Based Characterization of the Design Space of Collaborative Applications", *Journal of Organizational Computing*, vol. 4, no. 3, 1994, pp219-240
- [10] Edwards, W. K., Policies and Roles in Collaborative Applications, *Proceedings of ACM 1996 Conference on Computer-Supported Cooperative Work (CSCW'96)*, Cambridge, USA, 1996, pp11-20
- [11] Gottlob, G., Schrefl, M. and Röck, B., "Extending object-oriented systems with roles", *ACM Transactions on Information Systems (TOIS)*, vol. 14, no. 3, July 1996, pp 268-296
- [12] Greenberg, S., "Personalizable groupware: Accomodating Individual Roles and Group Differences." In *Proceedings of the European Conference of Computer Supported Cooperative Work*, Amsterdam, Sept., 1991, pp17-32
- [13] Greenberg S., Gutwin C., and Cockburn, A. "Sharing fisheye views in relaxed-WYSIWIS groupware applications", *Proceedings of Graphics Interface*, Toronto, Canada, May 22-24, 1995, pp28-38
- [14] Gutwin, C., and Greenburg, S., "Effects of Awareness Support on Groupware Usability", *CHI'98*, April, 1998, pp511-518
- [15] <http://www.akiva.com/products/webboard/index.cfm>
- [16] <http://www.microsoft.com/windows/netmeeting/>
- [17] <http://www.vrs.org.uk/public/journal.html>
- [18] <http://www.webct.com/>
- [19] Leland, M., "Collaborative Document Production Using Quilt", *CSCW'88*, USA, 1988, pp. 206-215
- [20] Marsic, I., "An Architecture for Heterogeneous Groupware Applications", *Proceedings of the 23rd International Conference on Software Engineering*, Toronto, Ontario, Canada, 2001, pp475 - 484
- [21] Roseman, M. and Greenburg, S. "Building Real Time Groupware with GroupKit, A Groupware Toolkit", *Transactions on Computer-Human Interaction*, vol. 3, no. 1, pp66-106
- [22] Schmidt, K., "The Problem with 'Awareness': Introductory Remarks on 'Awareness in CSCW'", *Computer-Supported Cooperative Work*, no. 11, 2002, pp285-298
- [23] Simone, C., Divitini, M. and Schmidt, K., "A notation for malleable and interoperable coordination mechanisms for CSCW systems," in *N. Comstock et al. (Eds.) COOCS '95. Conf. on Organizational Computing Systems*, ACM Press, New York, 1995, pp. 44-54.
- [24] Smith, R. B., Hixon, R. and Horan, B., Supporting Flexible Roles in a Shared Space, *Proceeding of the ACM 1998 Conference on Computer-Supported Cooperative Work (CSCW'98)*, Seattle, Washington, USA, pp197-206
- [25] Stefik, M., Bobrow, D. G., Foster, G., Lanning, S. and Tatar, D., "WYSIWIS Revised: Early Experiences with Multiuser Interfaces", *ACM Transactions on Office Information Systems*, vol. 5, no. 2, April 1987, pp147-167
- [26] Turoff, M., "Computer-Mediated Communication Requirements for Group Support", *Journal of Organizational Computing*, vol. 1, no. 1, 1991, pp. 85-113
- [27] Turoff, M., "Distributed Group Support Systems", *MIS Quarterly*, Dec. 1993, pp. 399-417
- [28] VanHilst, M. and Notkin, D., "Using Role Components to Implement Collaboration-Based Designs", *Proceedings of ACM 1996 Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA'96)*, CA, USA, 1996, pp359-369
- [29] Vertegaal, R., "Designing Awareness with Attention-based Groupware", In *Proceedings of INTERACT'99*, IFIP: Edinburgh, UK, 1999
- [30] Zhu, H., "A Role Agent Model for Collaborative Systems", *Int'l Conference On Information And Knowledge Engineering*, USA, June 2003, pp438-444
- [31] Zhu, H., "A Role-Based Conflict Resolution Method for a Collaborative System", *Int'l Conf. on Systems, Man, and Cybernetics (ICSMC'03)*, USA, October 2003
- [32] Zhu, H., Turoff, M., and Li, Z., "An Object Model for Collaborative Systems", *2000 Americas Conf. on Information Systems*, USA, Aug. 2000, pp590-594
- [33] Zhu, H., Wang, P., and Hu, S., "The Design of a Multimedia Co-authoring Platform based on Client/Server Model", *ACTA Electronica Sinica*, vol 25, no. 5, 1997
- [34] Ziggers, L. and Kozar, K. A., "An Exploratory Study of Roles in Computer-Supported Groups", *MIS Quarterly*, vol. 18, no. 4, pp277-297