International Journal of Production Research, Vol. 44, No. 1, 1 January 2006, 181-193 (Not the exact copy of the Journal paper).

Role Mechanisms in Collaborative Systems*

Haibin Zhu

Department of Computer Science and Mathematics, Nipissing University
100 College Drive, North Bay, Ontario P1B 8L7, Canada
Tel: 1-705-474-3461 ext 4434

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. 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. One of the major problems in current CSCW systems is determining how to define and specify roles clearly and rigorously, while maintaining the flexibility for collaboration. Many traditional CSCW systems based on roles have lost their flexibility after introducing roles 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 there is no a special mechanism to express a role. This paper reviews the applications of roles in traditional collaborative systems, clarifies the general meanings of roles, discusses the functions of roles in collaboration, and suggests a role mechanism and a group of principles to support collaboration based on roles. At last, this paper concludes that roles need more comprehensive research and will be applied widely in different areas.

Keywords: Role, Role Mechanisms, Collaboration, Collaborative Systems

1 INTRODUCTION

Group work requires that everyone fulfils his/her obligations and respects everybody else's rights. Individuals should fill specific positions and play specific roles in an organization. A position represents a specific "seat" that entails certain privileges and accompanying responsibilities. A role is a set of prescriptions defining how a position member should behave (Ashforth 2001). Unclear role specification may create dysfunctional ambiguity and conflict in an organization (Bostrom 1980).

A person may be uncertain about many facets of his social or physical environment. In organizations there are several frequently encountered areas of ambiguity which people often find stressful. All too often people are unclear about the scope of their responsibilities; they simply do not know what they are "supposed" to do. Such uncertainties may arise because the expectations defining the role are themselves vague and inconsistent (Kahn 1964).

Role concepts have been applied widely in behavioral science, social science, and psychology for many years (Ashforth 2001, Biddle & Thomas 1966, Bostrom

Email: haibinz@nipissingu.ca

1980, Hellriegel et al. 1983 & Hawkins et al. 1983). Roles have been also successfully applied in many areas such as RBAC (Role-Based Access Control) (Ferraiolo 2001 & Park, Sandhu & Ahn 2001), business management (Kueng 1996, Lupu & Sloman 1997) and software project management (Dafoulas & Macaulay 2001). They also caught much concerns in object and agent system modeling (Cabri, Leibardi & Zambonelli 2002, Depke, Heckel & Kuster 2001, Riehle et al 1998 & VanHilst & Notkin 1996).

The major difficulties to apply roles into applications are among how to define and how to specify a role. It is very difficult to give a commonly accepted definition for roles. Almost every application defines its role concept with its own consideration. The concentration on roles changes a lot from the viewpoints of behavioral sciences, RBAC, modeling methodologies to agent systems.

To support collaboration, we need special methods, tools and techniques (Wognum & Faber 2002). As computer-based tools that support collaborative activities, CSCW (Computer-Supported Cooperative Work) systems 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. 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. However, they did not provide 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 (Edwards 1996). The users have no supporting facility to tune their roles in order to make collaboration more productive and more efficient.

One of the major problems in current CSCW systems is determining how to define and specify roles clearly and rigorously, while maintaining the flexibility for collaboration. Many traditional CSCW systems based on roles have lost flexibilities after introducing roles because they can only provide static role mechanisms based on intuitive role concepts (Winograd 1988). There are no flexible mechanisms for role tuning, changing and transitions because there is a lack of a special mechanism to express a role.

Creativity always implies change (Kahn 1964). If we could keep the flexibility for roles that specify the responsibilities and rights clearly for human users, we can keep the creativeness of the human users by providing flexible role transitions and role tuning.

This paper is arranged as follows. Section 2 reviews the applications of roles in traditional collaborative systems and clarifies the general meanings of roles based on the basic ideas from behavioral and managerial science. Section 3 proposes a new view of roles in collaboration. Section 4 discusses the functions of roles in

collaboration and proposes a framework for role-based collaboration by enforcing object, role and group principles. Section 5 introduces a specification for roles to support the design and implementation of role-based collaborative systems. At last, this paper concludes that roles need more comprehensive research and will be applied widely in different areas.

2 THE ROLE COCEPTS IN COLLABORATIVE SYSTEMS

As a common word, “role” is easily understood. However, as a concept, there is no fundamental discussion and clear definition of roles. There are many different role concepts applied in different systems, to be reviewed in this section.

2.1 *Roles in current collaborative systems*

Although many have applied roles in their collaborative systems, they apply intuitively the role concept without defining it clearly and exactly (Edwards 1996, Leland et al. 1988, Simone et al. 1995, Smith et al. 1998 & & Turoff 1991).

In Quilt (Leland 1988), there are roles for writers (who are allowed only to change their own work), readers (who are not allowed to modify the document), and commentators (who can only add “margin notes” to the document). In EIES (Electronic Information Exchange System) (Turoff 1991), roles are built out of a subset of the primitive privileges (such as append, link, assign and use) that are crucial to the human communication process. Simone et al. (1995) define a role as a set of responsibilities of resources and tasks, and a set of rules that are established in the organizational context.

Edwards (1996) introduces access control policies and roles to avoid chaos in collaborative applications. It is good experience for him to propose a role-based model for collaboration. He describes a role as a category of users within the user population of a given application; and all users in a certain role inherit a set of access control rights to objects within the application. By describing role memberships, users can be relieved of some of the burden of tracking, updating, and anticipating role membership explicitly. In the work of Edwards, dynamic roles are supported by binding a predicate function to a symbolic role name and mapping from roles to policies. In fact, the essential characteristics of roles are determined by policies in this way. The dynamic property of roles is shown in the form of changing the mapping between a role name and a policy by associating a predicate function. The question here is: should the method require going further in this direction? Or, how can a policy be tuned, changed or transferred? Edwards (1996) actually applied into the policies the methodology of access control, i.e., to rely on an administrator to tune, change, or restrict access control policies.

Lupu & Sloman (1997) proposed a role-based framework for distributed system management. They mentioned a role template specification that was applied in

RBAC. They take the role as a set of policies relating to a particular manager position and the notation to specify policies.

Smith et al. (1998) emphasize the importance of roles in their Kansas system. They state that people in a group play various roles. They believe that the physics underlying reality does not define roles though it supports them. Therefore, there is no concrete role component in Kansas. They consider their Kansas system being able to support roles by special treatments in multi-user interfaces. In Kansas, roles are in general supported by a system's treatment of output and user inputs, which are similar to this paper's view of incoming messages and outgoing messages in a more abstract level.

Smith et al. (1998) emphasized that the Kansas system can support flexible roles by managing inputs and outputs, however, there is only one role facilitator that can manage other users' inputs and outputs.

Guzdial et al. (2000) recognize the usefulness of roles in a collaborative tool CoWeb. They give the description of roles as specific concerns and activities associated with individuals. They also define a role as a human construct created and sustained by the interaction of minds, and their roles extended on and off-line and are not associated with capabilities within the shared space.

From the above discussion, we know that the roles used in most traditional collaborative systems are considered as labels for specific objects (Edwards 1996, Leland et al. 1988, Simone et al. 1995, Smith et al. 1998, Turoff 1991 & Zhu 2003c). Based on these labels, the system designers use switch/case like structures to define different working processes for different roles. With this view, once a collaborative system with roles is completed, it is very difficult to adjust the roles in the system, although the requirement of tuning roles is very common in collaboration. Papers (Edwards 1996 & Smith 1998) emphasized the importance of flexible or dynamic roles in collaboration. They have provided successful methods to support flexible or dynamic roles by a capability mechanism and interface design. Their practices do demonstrate that there is a requirement to develop an underlining role mechanism to support more flexible collaborative systems with roles.

2.2 *Roles in behavioral and managerial science*

From Oxford English Dictionary (OED 2004), a role is "The part or character which one has to play, undertakes, or assumes"; "The part played by a person in society or life"; or "The typical or characteristic function performed by someone or something". A role is also defined in a behavioral and psychological view "the behavior that an individual feels it appropriate to assume in adapting to any form of social interaction; the behavior considered appropriate to the interaction demanded by a particular kind of work or social position."

In our common sense, the term "role" actually derives from the theater and refers to the part played by an actor. It was defined by Thomas and Biddle as a set of prescriptions defining what the behavior of a position member should be (Biddle 1966).

Bostrom (1980) defined a role as a set of expectations about behavior for a particular position within a work system. Hawkins et al. (1983) defined a role as a prescribed pattern of behavior expected of a person in a given situation by virtue of the person's position in that situation. Ashforth & Mahwah (2001) defined a role simply as a position in a social structure. A position means a more or less institutionalized or commonly expected and understood designation in a given social structure such as accountant (work organization), mother (family), and church member (religious organization). Generally, a role is a position occupied by a person in a social relationship. At this position, the person possesses special rights and takes special responsibilities.

Generally, we describe rights and responsibilities in natural languages. As a matter of fact, it is difficult to describe roles clearly and strictly because natural languages are ambiguous to some extent. That is why different persons at the same position can make different contributions.

In an information or software system, we may specify roles clearly with some special considerations. The possibility comes from the exactness of computer languages and the limited type and number of resources a person can control at a time.

Because there is no unified role definition, roles are difficult to be applied in collaborative systems for the ambiguity of a discourse.

Collaborative systems aim to support people working in groups. Therefore, we can accept the basic ideas about roles from social, behavioral and managerial science as our basis to form our role concepts in CSCW applications. However, with only these ideas, we cannot build a real practical collaborative system.

In summary, roles specify not only what the system may request the user but also what the user may ask for the system. Consideration of social sciences reveals that roles include two aspects such as rights and responsibilities, but does not make it clear how to build real practical systems. We need to find a more practical mechanism to support roles in collaborative systems.

3 A NEW VIEW OF ROLES IN COLLABORATION

From the development of computer software systems, we have learned that many good managerial methodologies are well adopted in designing a software system. It is also true in building a collaborative system.

In business management, role modeling (RM) was proposed as a business engineering technique (RM 2001) to provide a model of an organization in terms of roles, responsibilities and collaboration among individuals and teams, and a discovery and transformation process for the enterprise, applicable at small or large scale and a vehicle for reengineering and process improvement. In RM, a role has a cohesive set of responsibilities and a purpose for its existence. A role may be played by a person, group, organization, team, or automated system. Each of these entities may fill many roles. RM draws a major part of its power by separating "what" of a role from "who" is filling the role at any particular point in time. That

separation allows teams to come together to discover the nature of the work that must be done to meet the demands placed upon them by their sponsors, and then to figure out how the roles must be filled.

For example, an application development group and data management group are faced with complaints that the results presented to the customer are not based on the correct source data the customer assume it to be. Each group thinks that the other group is responsible for quality assurance. In building their role model they can easily agree on the need for a quality assurance role. As they flesh out the responsibilities of that role it becomes clear that it has to be filled by a team with representation from both organizations. The team is then further described in terms of the roles (and responsibilities) of its members.

RM acclaims the importance of roles and shows how roles collaborate to fulfill their responsibilities. However, RM only provides a business management method but no a role structure that supports the design of a collaborative system.

Based on the basic ideas about roles in behavioral and managerial science, it is possible to map such ideas to a software solution domain. We need a specification that is easily applied in a software system.

- At the viewpoint of collaboration, a role is tag, a cap, a costume, or a uniform. An object can hang a tag, put on a cap or a uniform, etc. So does a human user.
- A role is a view of a person to the object world. It consists of a list of messages to a subset of the object set and also provides a group of services. Roles restrict the abilities or rights and the services of a human user.
- A role is an entity that expresses both a request interface including outgoing messages a human user can send out and a service interface including incoming messages a human user can accept. The outgoing messages may be sent to classes, objects, agents and groups and the incoming messages to the agent.

Patterson (1991) introduces a role concept with the idea of an interface between objects. He states that a user-interface is simply an elaborate mechanism for moving messages back and forth between the user object and other objects. A role is an abstract user type. It is an object class of which a user is an instance. Given the roles of users, the messages understood by them are known. In the RENDEZVOUS system, the role was used to enable and disable an object's visibility and to act as a filter on the input events. Unfortunately, he does not go far enough to provide a real abstract structure to support more flexible role application. No follow-up work and application has been observed along this direction.

Based on the observation of our society and the ideas we discussed in Section 2, we find that in collaboration, persons generally have two kinds of existence, one is a server and another a client. When they play a role, they provide specific services and have specific rights to ask for services. With this common sense, a role can be considered as a view of persons to the world. When they play a specific role, they have a special view to the surroundings. Their role in a collaborative environment

is actually a wrapper with the service interface and the request interface as shown in Figure 1. We can separate a person's role into two parts: one is the service interface including incoming messages and the other the request interface including outgoing messages.

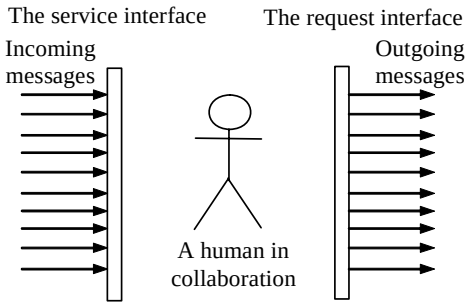

Figure 1. A role as a wrapper of a person in collaboration

With an object-oriented paradigm, we can easily express a role's incoming messages to express its services with class mechanisms. Therefore, we can use a special object as a representative of a human user to provide services in a collaborative system. In object modeling and systems, many have discussed this view (Gottlab 1996, Kendall 1999, Pernici 1990, & Riehle 2000). In other words, when an object plays a role, it provides specific services relevant to this role.

However, we really have no facilities to define the outgoing messages to express a role's right until now. RBAC has done much in dealing with human users' permissions to definite operations in order to simplify the tasks of a security administrator (Ferraiolo 2001& Park, Sandhu, & Ahn 2001). It ignores the human users' incoming messages when they serve others.

In summary, we should consider a role as two special interfaces: one is called the request interface from a human user to the system and the other the service interface from the system to the human users. At this time, the system is taken as a medium to transfer other human user's requests. In this view, roles are the real entities to let human users communicate with the system.

4 RINCIPLES OF ROLE-BASED COLLABORATION

Role-based collaboration means that human users apply a role-based collaborative system to cooperate and to obtain an ideal collaboration result. Based on the discussion above, we need to comply three sets of principles, i.e., object principles, role principles and group principles to build a role-based collaborative system.

4.1 *Object principles*

Object-oriented methodology is widely used in system modeling and software engineering. A collaborative system can also be considered at an object-oriented

viewpoint. We should emphasize the object principles as follows (Meyer 1988, Kay 1993, Zhu 2003b, Zhu & Zhou 2003).

- Everything in the world is an object. An object can be used to express everything in a collaborative system.
- Every system is composed of objects and a system is also an object.
- The evolution and development of a system is caused by the interactions among the objects inside or outside the system.
- A human user is a person who is involved in collaboration.
- A message is a way to activate services of a human user or an object.
- An interface is a list of message patterns.
- The interactions among objects are expressed by sending messages that are requests to invoke objects' actions.
- Each object is an instance of a class which shows the commonality of a group of objects.
- Each class might inherit another class which is called a superclass while it is called a subclass.
- Classes can be taken as templates of objects.
- An agent is a special object that represents the existence of a human user in a collaborative system.

4.2 *Role principles*

A role can show the specialties of human users. It provides human users not only message patterns to serve others but also message patterns to access objects, classes or groups (See 4.4). We should emphasize the principles as follows.

- A role is an independent object in a system. We can define it separately.
- A role can be played by one or more human users at the same time. A role can be played by an agent based on the relevant human user's requirement.
- A role can be created, changed and deleted by a human user with a special role.
- A role should consider both responsibilities when taken as a server and rights when taken as a client.
- A role is mainly concerned with two interfaces, the request interface from a human user to the system, and the service interface from the system to a human user.
- As for the service interface, a role is actually a filter of messages sent to a human user.
- As for the request interface, a role expresses or restricts the accessibility of a human user to the system.
- A role is enacted in groups.

4.3 *Group principles*

In reality, human users work in a group and hold roles. Every work setting involves groups of individuals. In a group, to accomplish a task, the group members (human users) will interact with each other. We should emphasize the principles as follows.

- A group is a fundamental structure in a collaborative system.
- A group can be created, changed and deleted.
- A group can be embedded, i.e., one group may be an object in another group.
- A group can be overlapped with other groups, i.e., the members may belong to two or more groups.
- The state of a group is dynamically changed.
- To specify a group is actually to specify all the roles and objects in this group.
- To form a group is letting human users join the group, play the roles and access the objects. We can call the human users the members of this group.
- A group can be public or private. Public means that all the human users using the system can join the group, private means that the joining is controlled by a special human user who plays a special role.
- A group can be open or close. Open means that newly-coming human users can still join the group and close means that no new human user will be accepted to join the group.

4.4 *The collaboration scenario*

With the above principles, we can build a system.

- A system is built with primitive objects that may be classified by classes, primitive roles and primitive groups.
- Agents are created for those human users who log in the system.
- Roles are created by some human users who play special primitive roles.

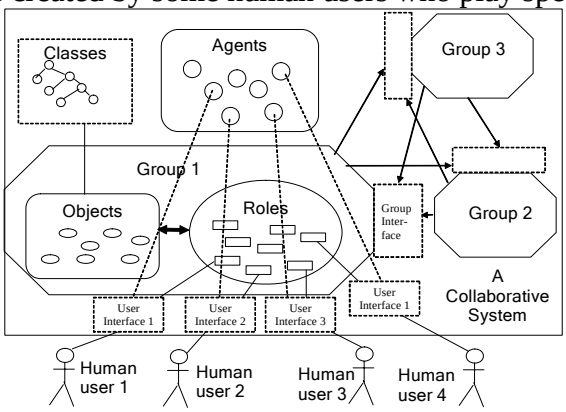

Figure 2. A conceptual model of role-based collaboration.

A collaborative system built with the role mechanisms is shown as in Figure 2. The octagons in the figure demonstrate that the three sets of principles are the key points in role-based collaborative systems. We could form the scenario of role-based collaboration as follows.

- Each human user uses an ordinary interface like a web browser to log into the system on his/her client computer with a role.
- Logging in sets the human user's agent a state of online and attaches a default role in a default group to the human user.
- Some human users create classes, objects, agents and roles in the system.
- Human users may play many roles but they can only play one role at one time.
- Human users may change their roles based on the requirement in collaboration.
- Human users can directly use the request interfaces relevant to their roles while their agents accept messages relevant to their roles.
- Agents represent human users who temporarily log out of the system by collecting the messages sent to the human user.
- Roles can be modified or tuned by some human users with specific roles based on the requirement in collaboration.
- Through the roles, human users access objects and their agents, contact other agents, join/leave groups, provide services and contribute to the collaboration.
- The result of the collaboration is reflected by the states of objects and agents in the system and the human users who use the system.

5 SPECIFY ROLES

Based on discussed above, we need to specify an underlined mechanism role to support the design and implementation of collaborative systems.
We can define a role a template of different kinds of messages, such that: $\mathcal{R} ::= <\mathcal{N}, \mathcal{M}_i, \mathcal{M}_o>$ where,

- $\mathcal{N}$ is the identification or the name of the role; and
- $\mathcal{M}_i$ and $\mathcal{M}_o$ denote sets of message patterns, wherein, $\mathcal{M}_i$ expresses the incoming message patterns to the relevant agent or the human user; $\mathcal{M}_o$ express different sets of outgoing message patterns to the objects.

These message patterns can be extracted from the interfaces of classes specified by an object-oriented programming language such as Smalltalk, C++, or Java. In implementation, $\mathcal{M}_o$ might be a list of links or dialogue boxes that a human user can send messages and $\mathcal{M}_i$ is a subset of the services the agent can provide. To simplify the message sending, we may classify messages into three categories. One is any-message, the second is some-message and the third is all-message. By any-

message we mean that the message may be sent to any member of a group or any instance of a class; by some-messages we mean the messages should be sent to a definite number of instances of a class or members of a group; and by all-message we mean that the messages should be sent to all the members of a group or all the instances of a class.

A role defined this way can be used as a wrapper for an agent or a human user. A human user who is playing a role can only send out messages restricted by the set $\mathcal{M}_o$, and the messages sent to the human user must match the message patterns in the set $\mathcal{M}_i$.

This specification is completely consistent with the basic idea we discussed in section 3, $\mathcal{M}_i$ expresses the server aspect and $\mathcal{M}_o$ expresses the client aspect.

The problem left in building a system based on roles is a mechanism to collect these message patterns. The basic way to set up an $\mathcal{M}_o$ is to search all the messages of all the objects and select some from them. The way to set up an $\mathcal{M}_i$ is relatively easy because we only need to collect all the messages of a special object, i.e., an agent. A key mechanism to specify roles can be built based on this definition.

6 CONCLUSION

Role is one of the foundation concepts for building collaborative systems (Zhu 2003a). Making it a fundamental mechanism is really helpful to build a more practical collaborative system. Such roles mechanisms can be applied to a broad area relevant to collaboration, such as CSCW, GSS (Group Support System), GDSS (Group Decision Support System), CMC (Computer-Mediated Communication), personalized interface (Greenberg 1991), business management (Lupu & Sloman 1997), adaptive interfaces, virtual organizations (Wognum & Faber 2002), emergence response systems (Turoff et al. 2004), software engineering, concurrent engineering, system modeling such as UML (Unified Modeling Language), object systems, agent systems and security such as RBAC. Current research about roles in different areas has not taken roles as an underlining mechanism.

Therefore, research on role mechanisms in collaborative systems may lead to revolutions in the development and applications of collaborative systems.

We anticipate innovations on role applications in all kinds of collaborative systems.

7 ACKNOWLEDGMENTS

This research is supported by the research funding of Nipissing University (No. 10-3287-42195) and the IBM Eclipse Innovation Grant Funding. Pierre Seguin is now making programs to demonstrate our ideas on role-based collaboration. His questions always activate me to think more.

PREFERENCES

Ashforth, B.E. and Mahwah, N.J., 2001. Role Transitions in Organizational Life: An Identity-based Perspective, Lawrence Erlbaum Associates, Inc.

Biddle, B.J. and Thomas, E.J.(Ed), 1966. Role Theory: Concepts and Research, John Willey & Sons, Inc.

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, pp. 88-115

Cabri, G., Leibardi, L. and Zambonelli, F., 2002. Separation of Concerns in Agent Applications by Roles, Proc. of the 22nd International Conference on Distributed Computing Systems Workshops

Dafoulas, G.A. and Macaulay, L.A., 2001. Facilitating Group Formation and Role Allocation in Software Engineering Groups", Proc. of the ACS/IEEE International Conference on Computer Systems and Applications, pp0352-0359

Depke, R., Heckel, R. and Kuster, J.M., 2001. Improving the Agent-Oriented Modeling Process by Roles, The Fifth International Conference on Autonomous Agents, Montreal, Quebec, Canada, pp. 640-647

Edwards, W.K., 1996. Policies and Roles in Collaborative Applications, Proc. of ACM Conference on Computer-Supported Cooperative Work, Cambridge, USA, pp. 11-20

Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R. and Chandramouli, R., 2001. Proposed NIST Standard: Role-Based Access Control, ACM Transactions on Information and system Security, vol. 4, no. 2, pp. 224-274

Gottlob, G., Schrefl, M. and Röck, B., 1996. Extending Object-Oriented Systems with Roles, ACM Transactions on Information Systems, vol. 14, no. 3, , pp. 268-296

Greenberg, S., 1991. Personalizable groupware: Accomodating Individual Roles and Group Differences", Proc. of the European Conference of Computer Supported Cooperative Work, Amsterdam, pp. 17-32

Guzdial, M., Rick, J., and Kerimbaev, B., 2000. Recognizing and Supporting Roles in CSCW, Proc. of the ACM Conference on Computer-Supported Cooperative Work, Philadelphia, Pennsylvania, USA, pp. 261-268

Hawkins, D. I., Best, R. J. and Coney, K. A., 1983. Consumer Behavior, Business Publications, Inc., Plano, Texas

Hellriegel, D., Slocum, J. W. Jr. and Woodman, R. W., 1983. Organizational Behavior, West Publishing Co. St. Paul, Minnesota

Kahn, R. L., Wolfe, D. M., Quinn, R. P., Snoek, J. D., and Rosenthal, R. A., 1964. Organizational Stress: studies in Role Conflict and Ambiguity, John Wiley & Sons, Inc., New York

Kay, A.C., The Early History of Smalltalk, in *ACM SIGPLAN Notice, Proceedings of the second ACM SIGPLAN History of Programming Languages Conference*, 1993, 28(3), 69-75.

Kendall, E. A., 1999. Role Model Designs and Implementations with Aspect Oriented Programming, Proc. of ACM Conference on Object-Oriented Programming, Systems, Languages and Applications, CO, USA, pp. 353-369

Kueng, P., Bichler, P., Kawalek, P. and Schrefl, M., 1996. How to Compose an Object-Oriented Business Process Model, Proc. of the IFIP TC8, WG8.1/WG8.2 Working Conference, Atlanta, pp. 94-110

Leland, M. D. P., Fish R. S. and Kraut, R. E., 1988. Collaborative Document Production Using Quilt, Proc. of the Conference on Computer-Supported Cooperative Work, Portland, OR., USA, pp. 206-215

Lupu, E. and Sloman M., 1997. Towards a Role Based Framework for Distributed Systems Management, Journal of Network and Systems Management, vol. 5, no. 1, pp5-30

Meyer, B., 1988. Object-Oriented Software Construction, Prentice Hall

OED, 2004. http://dictionary.oed.com

Park, J. S., Sandhu, R. and Ahn, G. J., 2001. Role-Based Access Control on the Web, ACM Transactions on Information and System Security, vol. 4, no. 1, pp. 37-71

Patterson, J. F., Comparing the Programming Demands of Single-User and Multi-User Application, *Proceedings of the fourth Symposium on User Interface Software and Technology*, SC, USA, 1991, 87-94.

Pernici, B., 1990. Objects with Roles, ACM SIGOIS Bulletin, Proc. of the Conference on Office Information Systems, vol. 11, no. 2-3, pp. 205-215

Riehle, D., Brudermann, R., Gross, T. and Mätzel, K. U., 2000. Pattern Density and Role Modeling of an Object Transport Service, ACM Computing Surveys, vol. 32, no. 1, pp. 1-6

RM, 2001. http://www.rolemodeling.com

Simone, C., Divitini, M. and Schmidt, K., 1995. A Notation for Malleable and Interoperable Coordination Mechanisms for CSCW Systems" , Proc. of the Conference on Organizational Computing Systems, Milpitas, California, USA, pp. 44-54

Smith, R. B., Hixon, R. and Horan, B., 1998. Supporting Flexible Roles in a Shared Space, Proc. of the ACM Conference on Computer-Supported Cooperative Work, Seattle, Washington, USA, pp. 197-206

Tripathi, A. R., Ahmed, T. and Kumar, R., 2003. Specification of Secure Distributed Collaboration Systems", Proc. of International Symposium on Autonomous Distributed Systems, Pisa, Italy, pp. 149-156

Turoff, M., 1991. Computer-Mediated Communication Requirements for Group Support, Journal of Organizational Computing, vol. 1, no. 1, pp. 85-113

Turoff, M., Chumer, M., Van de Walle, B. Yao, X., 2004. The Design of a Dynamic Emergency Response Management Information System (DERMIS), Journal of Information Technology Theory & Application, 2004, 5(4), 1-36.

VanHilst, M. and Notkin, D., 1996. Using Role Components to Implement Collaboration-Based Designs, Proc. of ACM 1996 Conference on Object-Oriented Programming, Systems, Languages and Applications, CA, USA, pp. 359-369

Winograd, T., 1988. A Language/Action Perspective on the Design of Cooperative Work, Human-Computer Interaction, vol 3, no. 1, pp. 3-30

Wognum, P.M. and Faber, E.C.C., 2002. Infrastructures for collaboration in virtual organizations, Int. J. Networking and Virtual Organisations, vol. 1, no. 1, pp. 32-54

Zhu, H. 2003a, Some Issues on Role-Based Collaboration, Proc. of 2003 Canadian Conference on Electrical and Computer Engineering, Montreal, Canada

Zhu, H. 2003b. A Role Agent Model for Collaborative Systems", Proc. of Int'l Conference on Information and Knowledge Engineering, USA, pp 438-444

Zhu, H. 2003c. A Role-Based Conflict Resolution Method for a Collaborative System, Proc. of Int'l Conf. on Systems, Man, and Cybernetics, USA

Zhu, H. and Zhou, M. C., 2003. Methodology First and Language Second: A Way to Teach Object-Oriented Programming", Proceedings of the 2003 Educator's Symposium on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA'03), California, USA, pp. 140-147