Collaborative Social Structures and Task Modelling Integration

Victor M.R. Penichet[1], Fabio Paternò[2], J.A. Gallud[1], and Maria D. Lozano[1]

[1] I3A-UCLM, Av. España s/n, 02007 Albacete, España
{victor.penichet, jose.gallud, maria.lozano}@uclm.es
[2] ISTI-CNR, via G.Moruzzi 1, 56100 Pisa, Italy
fabio.paterno@isti.cnr.it

Abstract. Interdisciplinary work groups have proved to be one of the best practices (in terms of efficiency) in modern organizations. Large applications have many different users who can play different roles with responsibilities and rights depending on such roles. There are so many roles, groups, relationships among them, tasks, and collaborations, that it is very difficult to develop an application without gathering all this information in a proper way. This paper describes a modelling approach supported by a graphical notation, which makes the representation of such information easier to analyse and manage. The goal is to provide a complete and integrated approach to model collaborative interactive systems.

1 Introduction

Software applications are often used by a large number of user groups who have different features and functions. The availability of high-speed network connections has contributed to increasing the number of these applications, where many different types of users participate in a temporally and geographically distributed way.

Such a variety of users, features, tasks, objectives, etc. have to be taken into account carefully when developing large applications. Designers should be provided with techniques and tools to gather all this information, which must be considered to develop multi-user collaborative systems.

This paper presents a modelling approach for collaborative systems to better understand the organization of the different users, the existing collaborations among them, and the individual tasks they perform. Such an approach is based on the role that users play in the system (the role view).

The organizational structure of the users of the system that will be deployed and the relationships among such users are modelled by means of two diagrams: the Organizational Structure Diagram (OSD) and the Collaborative Diagram (CD), respectively. The OSD models such a user structure: groups to which users belong, the roles they play, etc. Several CDs provide a model of the collaborations among users depending on the role they play or the groups to which they belong.

G. Doherty and A. Blandford (Eds.): DSVIS 2006, LNCS 4323, pp. 67 – 80, 2007.

The third diagram making up the role view is the Task Diagram (TD). There is a TD for each role in the system, and it specifies the tasks that a user with such a role performs.

The organizational structure, collaborative relationships, and tasks a user performs are modelled thanks to three graphical representations.

Instead of developing a new graphical notation for the TD, we have adopted an existing one. CTT [10] has been selected for this purpose because it is already widely accepted and consolidated. This paper also presents the integration between the OSD and CD diagrams with CTT. These three diagrams provide designers with an easy way to gather useful information on a system.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 briefly describes the proposed role view to design and analyse collaborative systems. Section 4 is devoted to the integration between the adopted task diagram (CTT) with the rest of the diagrams in the role view. Section 5 presents an example to show the applicability of the proposal. Lastly, Section 6 concludes the work with some final remarks.

2 Related Work

Some coordination and communication problems appear when many users interact with the same system. A system that assists distributed developers in maintaining mutual awareness is presented in [6]. Our proposal is oriented to analysts and designers who develop applications where many users, probably geographically distributed, collaborate with each other. Software developed taking into account the user's organization and collaborations is generally much more usable.

The number of users in collaborative systems and their different features and functions raise specific issues and there is a need to provide designers with specific techniques and methods to model such systems.

In [2], a conceptual model is proposed to characterize groupware [3, 4, 5, 7] systems. This model describes objects and operations on such objects, dynamic aspects, and the interface between the system and the users, and amongst users. This characterization describes a groupware system from its users' point of view. Our approach describes the system from the role point of view, because we want to take advantage of the abstraction of features and functions that roles provide. We also propose a graphical notation to represent the roles, collaborations, etc. in a collaborative system.

Role modelling is used in [12] as an enabling technology for framework design and integration. Class diagrams with some role constraints, which are constraints on object collaborations, are used in such modelling technique. We propose to specify the organizational structure of system users , that is, how roles are grouped and related.

Role modelling is used as a mechanism for separation of concerns to increase the understandability and reusability of business process models [1]. In some modelling techniques for developing groupware systems [10, 11, 14], role or actor concepts are

also considered when modelling the existing collaboration among the users of a system. We propose a notation that provides designers with a flexible way to represent social structures and interactions. It is a view of the system that facilitates the design and the analysis of the users' collaborations and provides a way of classifying, organizing, and representing the roles and the groups to which the users will belong.

Such methods use these concepts to ease comprehension of the system, and to allow designers to know who does what, or what kind of features or functions a particular user performs. In our work, it is also possible to represent the organizational structure of the system to be built. After modelling the organizational structure of the users, we suggest a graphical notation to represent the person-computer-person interactions, which provides an easy-to-grasp view of the existing collaborations among such users.

We use the ConcurTaskTrees (CTT) notation to model individual tasks instead of a new notation. Therefore, a mapping between different notations is necessary, which is a common technique to achieve a more complete model of a system (e.g. [9]).

3 The Role View

The *Role View* provides a way to model collaborative systems, and accordingly, provides designers with another way to analyse them. The main focus of the Role View is on the actors, the organizational structure, and the relationships among them.

This view is described by means of three diagrams: *Organizational Structure Diagram* (OSD), *Collaborative Diagram* (CD), and *Task Diagram* (TD).

The concepts we use to model collaborative applications by means of the OSD and CD diagrams, as well as the diagrams themselves, are explained in the following subsections, while for the TD, the existing CTT [10] graphical representation has been adopted. A more extensive example of the notation and the way of modelling collaborative systems through the Role View has been introduced in section 5.

3.1 Basic Concepts

We use some concepts in our approach that are going to be briefly explained in this sub-section. Regarding organization, we use three concepts that we call *organizational items*: *actor*, *role*, and *group,* which are described in Table 1. Other concepts are used to express relationships and collaborations between the different classifiers: *instantiation*, *aggregation*, and *cooperative interaction* (Table 2).

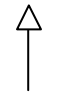
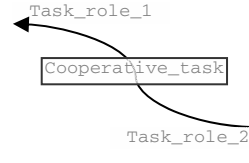
3.2 Organizational Structure Diagram (OSD)

It is possible to model the organizational structure of the users by means of *actors*, *roles*, and *instantiation relationships*, *groups*, and *aggregation relationships*. The main advantage to modelling such structure is the possibility of classifying, organizing, and representing the users of the system.

Table 1. Organizational items

Organizational items	*Description*	*Notation*
Group	A *group* is a set of *roles* that need to interact together and to collaborate in order to reach a common objective. Common objectives would not be reachable without such collaboration.	 GROUP 1
Role	A *role* is a set of *actors* that share the same characteristics and perform the same tasks	 ROLE_1
Actor	An *actor* is an element able to perform a task. We could consider an *actor* as an instance of a *role*.	
User	A *user* is a person who interacts with the system, thus s/he is an *actor*. Some other things (not users) could be actors.	

Table 2. Organizational relationships

Relationship	*Description*	*Notation*
Instantiation (structure)	Between a *role* and an *actor* playing such role there is an *instantiation relationship*, that is to say, this actor is an instance of that role.	
Aggregation (structure)	An *aggregation relationship* is an existing association between the whole and its parts.	
Cooperative Interaction (collaboration)	A *cooperative interaction* means a cooperative task among several *actors*, *roles*, or *groups* in order to reach a common objective.	

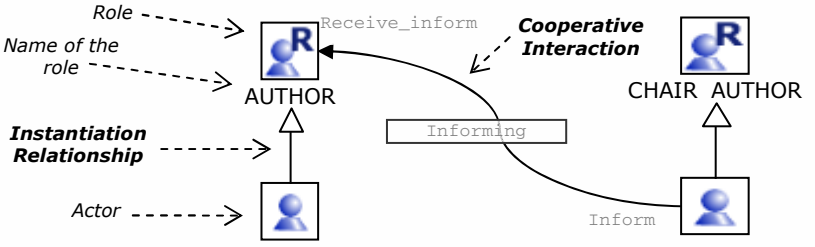


Fig. 1. Instantiation relationships between a role and an actor and cooperative interactions

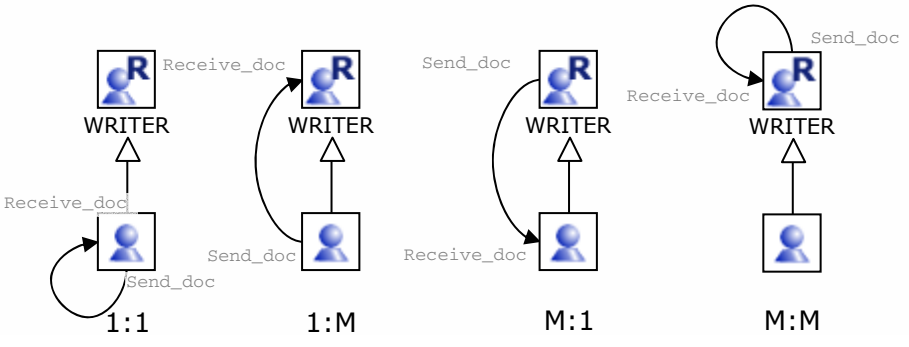


Fig. 2. The four fundamental cooperative relationships

Moreover, it allows designers to gain a much more structured, simple and real view of the role organization and the groups to which the users will belong. It will be the basis for designing user collaborations.

Once the organizational structure of the system users is represented (in terms of group, role, actor, and aggregation), the collaborations existing between the organizational items are established, which describe the different cooperative tasks performed. These tasks are represented in the CD, which is explained in Section 3.3.

The *Instantiation relationship* between role and actor provides a way to represent a role and an actor playing this role in the same diagram. Fig. 1 shows an example of *Instantiation relationship*. It links two organizational items: a role and an actor. This relation means that the source icon is an instance of the destination icon, that is to say, an actor performs such a role. Then, this figure represents a set of actors with the same features and the same functions, and an instance of such a set.

3.3 Collaborative Diagram (CD)

Once the structure of the organization is represented by the OSD, one of the main advantages is the possibility of modelling the collaborations among actors belonging

to different roles and groups. The idea is not to model users' interaction, but to model the interaction between users through computers and networks.

When structuring the whole system according to different primary objectives, a CD is made for each objective. Each CD will explain the existing interaction between the organizational items (groups, roles, actors) that are necessary to achieve such objectives. The whole set of CDs describes all collaboration in the system.

Interaction among actors is symbolized by the *cooperative interaction* relationship, a solid black arrow in the diagram. Such relationships are cooperative tasks, which are performed by several actors within the system.

Solid black arrows representing cooperative tasks have three labels as shown in Fig. 1. The source label is the name of the task performed by the actor who starts the cooperation, and the destination label is the name of the task performed by the actor who cooperates with the first one. The squared label situated in the centre of the arrow is the name of the cooperative task. The arrowhead could be omitted if necessary, for instance, in a concurrent cooperative task where the order of the tasks is not important.

A cooperative interaction relationship representing a cooperative task also has an intrinsic cardinality at the beginning and at the end. This cardinality indicates the number of actors performing the role tasks.

The source and the destination of the arrow representing a cooperative task determine the cardinality. Fig. 2 shows an example with every possible cardinality.

4 Integration of the Role View and ConcurTaskTrees

Tasks models are a useful tool to analyze and design applications from the point of view of the users who are going to interact with such applications.

The role view that we propose provides a new perspective to the designers. Users are organized in roles and related groups. The result of these relationships between roles and groups is what we define as *organizational structure* of the users of an application which is graphically represented in the OSD (see section 3).

Traditionally, in order to discover a way to fix large problems, they are divided into simpler sub-problems which can be modelled separately. We propose the CDs to identify and model the collaborations between users in every sub-problems (see section 3). Different organizational items from the OSD are related in the CDs to represent such collaborations.

Therefore, this method to design collaborative systems allows designers, first, to *identify and analyse the organizational structure* of the system users, and then, it also provides a way to *model the collaborations among the users of such system*.

Mapping different notations is a technique already used to obtain a complete model of the system. For example, [9] shows a mapping approach between ConcurTaskTrees and UML [13] to include one of the most widely used notation for task modelling into the Unified Modelling Language.

The *role view* is composed of three different diagrams as it was mentioned before: OSD, CD, and TD. OSD and CD were detailed in Section 3, while for the TD, the existing CTT graphical representation has been adopted. The organizational items and relationships integration within the CTT notation is explained in the next sub-sections. Such integration is necessary to achieve a coherent model of the system.

4.1 Group and Aggregation Relationship

Group and aggregation concepts do not have a direct mapping from the OSDs and CDs to the CTT notation. They are concepts used to classify the roles that the users of the system are going to play.

An OSD represents the set of all the users of the system. These users play roles, and such roles belong to one or more groups. A user, by himself, cannot directly belong to any group.

The group concept was defined in Section 3 as a set of *roles* whose actors need to interact together and to collaborate in order to reach a common objective. Hereby, this common objective could be one of these sub-problems in which the main problem is decomposed to make it simpler to manage, as mentioned in the introduction of this section.

Each sub-problem is modelled by means of a CD and several TDs. The CD represents the existing collaborations among some organizational items of the OSD, while there is a TD, which is represented by the CTT graphical notation, for each role that participates in the sub-problem.

4.2 Role

The *role* organizational item of the Role View (see Table 1) is the one which has the most direct correspondence in the CTT task model, because the role concept is also considered in the CTT notation in the same way.

In a CTT cooperative model, every role has an associated task diagram with all the tasks that are performed by such role. That is, if there is a role item in the Role View, there will be an associated CTT task diagram for this role, which shows all the tasks performed by a user playing such role.

4.3 Actor and Instantiation Relationship

Actor and instantiation relationship concepts do not have a direct mapping between the OSD and the TD. User interactions with the system and collaborations between users with different roles are considered in CTT. However, collaborations among users with the same role are not taken into account. Therefore, the introduction of these concepts provides a way to represent such collaborations.

Actors introduce a new concurrent situation in the typical CTT models because of this type of collaboration. Fig. 4 and Fig. 5 show the representation of two actors with the same role who are collaborating with each other. This kind of collaboration is further explained in the following sub-section.

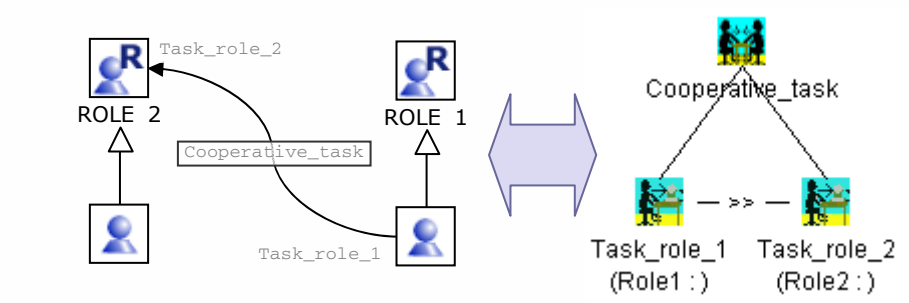


Fig. 3. The basic correspondence between the CD and TD diagrams

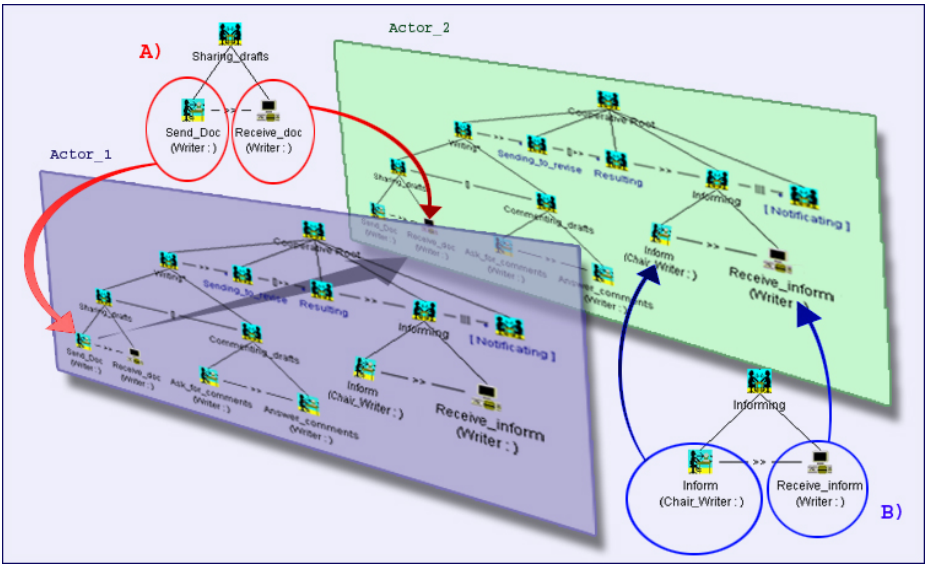
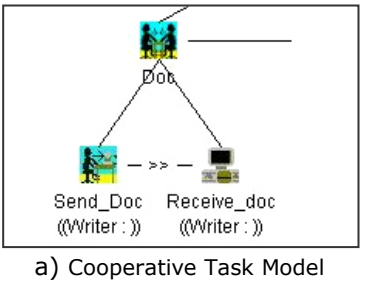
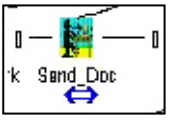


Fig. 4. Interaction between actors and roles



a) Cooperative Task Model



b) Task model. Role: Writer

c) Task model. Role: Writer

Fig. 5. Collaboration between actors with the same role

4.4 Cooperative Interaction Relationship

Cooperative interaction relationships between the organizational items of a CD have a direct mapping in the TD.

Every role within CTT has an associated task diagram as mentioned before, but there is also a cooperative part to structure the cooperative tasks, which are decomposed until tasks performed by a single user are reached [10]. These single user tasks also appear in the corresponding role task diagram as connected tasks (see Fig. 5).

The right part in Fig. 3 shows a *cooperative task* represented in CTT. The left part in Fig. 3 shows *cooperative interaction relationship* between two roles in a CD. As shown, the mapping of *cooperative task*, *task role*, and *role* concepts between the two diagrams is a simple matter.

Each diagram shows a different view of the system, hence some information appears in one but not in the other. In a CD, every *cooperative interaction relationship* has an intrinsic cardinality at the beginning and at the end, which indicates the number of actors performing the role tasks. This aspect is not shown in the TD. On the other hand, a TD has some other elements that are not in the CD such as: (1) the *CTT task category* of each task performed by each role, that is, if they are *interaction*, *application*, or *user* tasks; and (2) the *CTT temporal operator* necessary to link the two role tasks which will constitute the cooperative one, that is, if it is an *enabling*, *enabling with information passing*, *etc*. operator.

Considering that CDs represent actors and roles in the same diagram, two sorts of collaborations could occur: collaborations among users with different roles, and collaborations among users with the same role. Up to now, CTT has focused on relationships among users in a system playing different roles (Fig. 4). The notation we propose in this paper (see Fig. 3) allows designers to model relationships among users playing the same role as well (Fig. 4).

Collaborations among users with the same role can be appreciated more clearly in CDs. Although the CTT notation could represent this semantic by writing the same role name in the individual role tasks below the cooperative one (Fig. 4), for the sake of clarity, we have added double brackets and arrows, as shown in Fig. 5.

5 Example

We show a simple example of an application for internal publication of documents in an organization in order to better explain the approach proposed.

5.1 Brief Problem Description

Some employees elaborate together documents to be published in their organization. There is interaction among them in order to get a candidate document to be published. A supervisor (writers' chair) can send a document to be revised by other members of the organization. The candidate document is received by a reviewers' chair who decides what kind of review to apply. There are two possibilities: (AAO) all at once,

where all the reviewers receive the candidate document at the same time and the chair waits for their answers to continue; and (OAA) one after another, where the chair selects an order for review. In the latter case, if any reviewer decides that the candidate document is not ready to be published, then is not necessary to continue with the process. If the candidate document is finally published, then the authors will be informed. Published documents can also be read by readers: a group of people who can only read and comment documents, not modify them.

5.2 Designing the Collaborative System

When designing a collaborative system, our approach uses the three diagrams introduced in this work. Such diagrams provide designers with a way to gather information about the organizational structure of the users of the system (OSD), relationships among them (CD), and the tasks they are going to perform (TD).

The diagram in Fig. 6 shows the OSD of the example considered. Such diagram represents the organizational structure of the users of the application for internal publication of documents. The first decision is to make a logical division of the users into two groups: those who are able to modify, create, etc. (internal), and those who only have the possibility of viewing the products generated by the members of the first group (external). The "external" group is only composed of users playing the "reader" role.

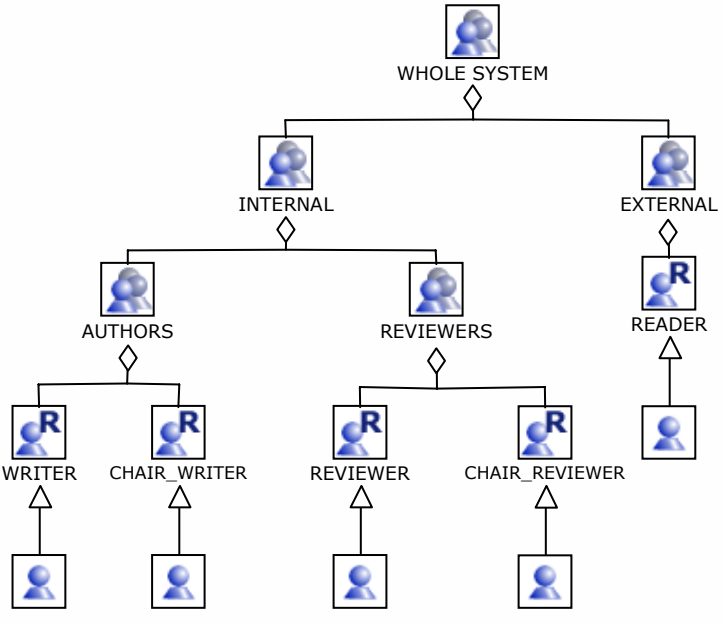


Fig. 6. Organizational Structure Diagram (OSD) of the example

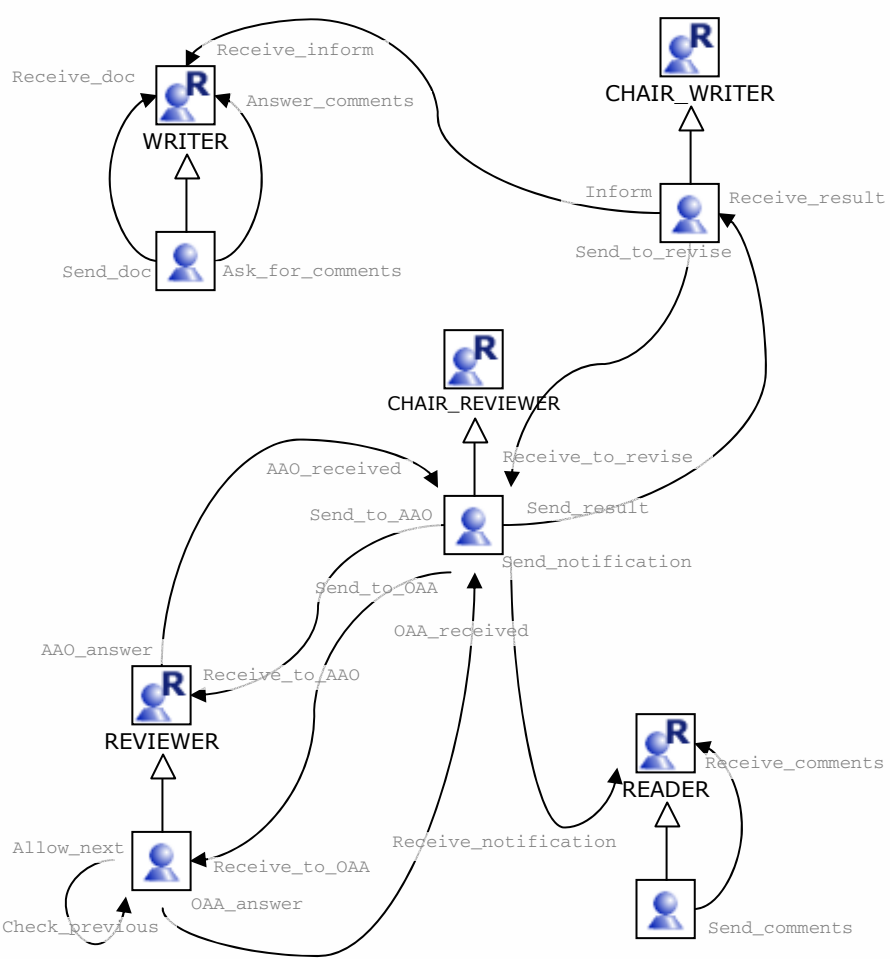


Fig. 7. Actor interactions in the internal publication system example: Role View. Note that the name of the cooperative tasks have been omitted for the sake of simplicity.

The "authors" group is made up of roles with writing features, whereas the "reviewers" group is composed of roles with document editing features.

Once the organizational structure of the users has been represented by means of the OSD, a CD is generated for each main objective in the problem. As the problem we are studying in this example is not very large, all its collaborations will be represented in the same CD. Such diagram is shown in Fig. 7.

Lastly, a TD is specified by using the CTT notation for each role in the system. Fig. 8 shows the cooperative model, while Fig. 9 shows an example of a TD which represents every individual task that a user playing the role "Chair_reviewer" could perform. That is, an actor playing such role also has to perform the tasks "Select_doc", "Read_doc",

"User_decision", "Select_AAO_or_OAA", "Answer_analysis", among other tasks in addition to those tasks that will be connected to constitute a cooperative one.

5.3 Analysis of a Collaborative System

The graphical notation presented in this paper could also be used to analyse an existing collaborative application in order to improve the way in which users work together to achieve common objectives.

Groups and roles can be represented and organized in an OSD to study if the current organizational structure of the users of the system is the best or, on the contrary, could be improved by restructuring such organization.

Likewise, CDs facilitate the study of the cooperative tasks performed in the system. As our proposed graphical notation provides analysts with additional information about collaborations among users, and such information is represented in an intuitive and easy way, then it is possible to analyse whether collaborations are adequate, or they should be redesigned in some other way.

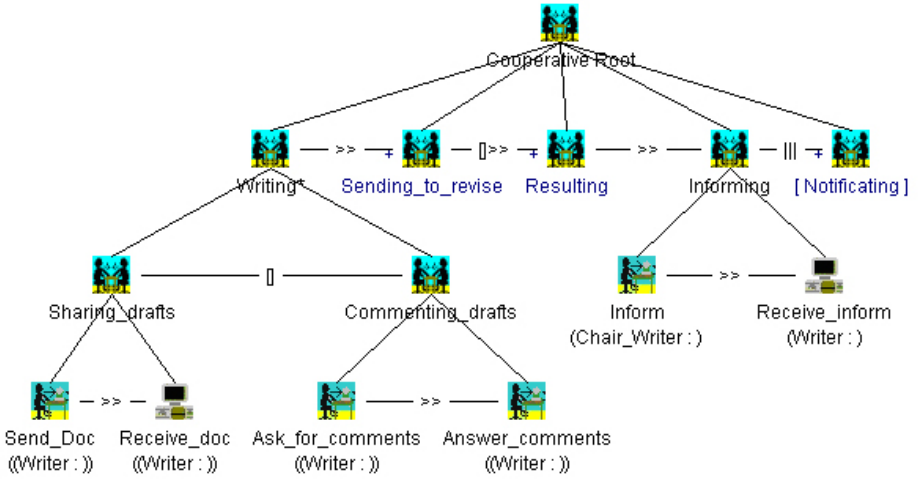


Fig. 8. Actor interactions in the internal publication system example: Task View

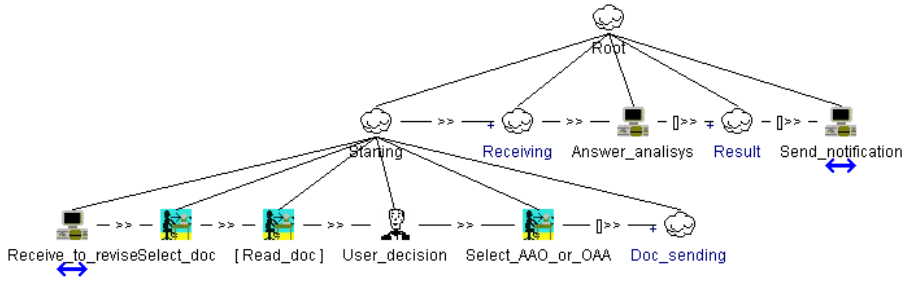


Fig. 9. Role task diagram example: "Chair_reviewer"

For instance, when the “Chair_reviewer” knows the result of the review, a notification is sent to the “Chair_writer”, and then the “Chair_writer” informs authors of the document about the decision. By analyzing the diagram, analysts could detect that a “Chair_reviewer” could inform actors with the role “Writer” directly, which would avoid an unnecessary step

6 Conclusions

A new modelling approach for collaborative systems has been presented in this paper. By means of the proposed graphical notation, the organizational structure of the users of the system is specified on the basis of the roles they play and the groups to which they belong.

Likewise, this approach provides designers with a way of representing cooperative relationships that users must perform to achieve a common objective. Organizational structure and collaborations among users are easily and intuitively represented by way of an OSD and one or several CDs.

We have adopted ConcurTaskTrees to represent individual tasks that each user performs because it is one of the more widely accepted notations in task modelling. Therefore, an integration of the organizational and collaboration models and CTT is also presented in this work.

Representing tasks, collaborations, and the organizational structure of the users of a system not only makes its design easier, but also facilitates the study and the analysis of existing systems to propose a new re-design if necessary.

Acknowledgements

We would like to thank the European ADVISES TMR for funding this work, which was also supported by the Spanish CYCIT project TIN2004-08000-C03-01 and the grant PCC-05-005-1 from JCCM.

References

1. Caetano, A., Silva, A. R., and Tribolet, J. 2005. *Using roles and business objects to model and understand business processes*. In Proceedings of the 2005 ACM Symposium on Applied Computing.L. M. Liebrock, Ed. SAC '05. ACM Press, New York, NY, 1308-1313
2. Ellis, C., Wainer, J.: *A Conceptual Model of Groupware*, in Proceeding of CSCW’94, 1994, p. 79-88, ACM Press.
3. Greenberg, S.: *The 1988 conference on computer-supported cooperative work: Trip report*. ACM SIGCHI Bulletin, 21(1), pp. 49-55, July 1989.
4. Greif, I.; *Computer-Supported Cooperative Work: A Book of Readings*. Morgan Kaufmann, San Mateo CA, 1988.
5. Grudin, J. *CSCW: History and Focus*. University of California. IEEE Computer, 27, 5, 19-26. 1994

6. Gutwin, C.; Schneider, K.; Penner, R.; Paquette, D.; *Supporting Group Awareness in Distributed Software Development*. EHCI/DSVIS 2004, 9th IFIP Working Conference on Engineering for Human-Computer Interaction, Tremsbuttel Castle, Hamburg, Germany, July 11-13, 2004, 14 pages. Also to be published in Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2004
7. Johansen, R. (1988): *Groupware: Computer support for business teams*. New York: The Free Press.
8. Mori, G., Paternò, F. ,Santoro, C. "CTTE: Support for Developing and Analysing Task Models for Interactive System Design", IEEE Transactions on Software Engineering, pp.797-813, August 2002 (Vol. 28, No. 8), IEEE Press.
9. Nóbrega, L., Nunes, N., Coelho, H.; *Mapping ConcurTaskTrees into UML 2.0*, 12th International Workshop on Design, Specification, and Verification of Interactive System (DSV-IS'2005), July 13–15, 2005, Newcastle upon Tyne, England, Springer-Verlag, 2005
10. Paterno', F.; *Model-based Design and Evaluation of Interactive Applications*. F.Paternò, Springer Verlag, November 1999, ISBN 1-85233-155-0
11. Pinelle, D., Gutwin, C., Greenberg, S.: *Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration*. ACM Transactions on Computer-Human Interaction (TOCHI) Volume 10 , Issue 4, Pages: 281 - 311. (2003) ISSN:1073-0516
12. Riehle D, Gross T.; *Role model based framework design and integration*. Proceedings of OOPSLA '98. ACM Press, 1998; 117-133
13. Rumbaugh, J.; Jacobson, I.; Booch, G.: *The Unified Modeling Language. Reference Manual*. Addison-Wesley. 1999
14. Van der Veer, G. C.; Van Welie, M. 2000. Task based groupware design: Putting theory into practice. In Proceedings of the 2000 Symposium on Designing Interactive Systems. New York, ACM Press, 326–337.