Service-Based Access Control Using Stages for Collaborative Systems

Mario Anzures-García[1], Luz A. Sánchez-Gálvez[2],
Miguel J. Hornos[2] and Patricia Paderewski[2]

[1] Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla,
14 sur y avenida San Claudio, Ciudad Universitaria, San Manuel, 72570 Puebla, Mexico
[2] Universidad de Granada, C/ Periodista Saucedo Aranda, s/n, 18071 Granada, Spain
manzures@siu.buap.mx, luzsg@correo.ugr.es, {mhornos, patricia}@ugr.es

Abstract. The adaptation of collaborative systems includes many dimensions, some of these are: access control, concurrency control, coupling of views, and extensible architectures. In this paper, we focus on access control; for this reason, we present a set of services which are part of a SOA-based architectural model for developing groupware applications and which provide two mechanisms for security management: authentication and access control. The former controls the user access to the shared workspace, and the latter controls the access to the shared resources and the interaction among users, in order to avoid conflicts arising from cooperative and competitive activities. We specify the policy-based management of the groupwork organizational structure by means of an ontology. This allows us to define several group organizational structures and to support the groupwork dynamism, facilitating the management of the security mechanisms mentioned. We consider that in collaborative systems there are several stages, i.e. phases of collaboration. Adequate access to the shared workspace must be controlled at each stage, taking into account the roles that can participate in it. Moreover, we explain by means of an example how interactions are carried out among the services related to the access control process and how these services are sometimes adapted.

1 Introduction

Collaborative systems are focused on investigating how computer-based groupwork can improve the performance of groups of people engaged in a common task or goal [2, 3]. In this paper, we will consider mainly two key aspects of this type of system: groupwork and shared information.

We believe that, with regard to the first aspect, both static and dynamic issues have to be taken into account. The static ones are associated with the groupwork organizational structure, while the dynamic ones should support the dynamism that is inherent in the groupwork. It is very important to provide methods to adequately model these issues, using a set of elements (or concepts) and relations among them. This facilitates adaptation to the dynamic nature of the group and to the changing needs of the groupwork.

Received 09/03/09
Accepted 27/04/09
Final version 06/05/09

We believe that for the second aspect above mentioned, security mechanisms should be considered. Access control to resources and activities is a key element in system security. In most systems, security is achieved through mechanisms such as: authentication, access control, data encryption, digital signature, and so forth. In the collaborative domain, special attention has been paid to authentication and access control mechanisms. The former are mechanisms that allow identification and verification of the user identity, in an attempt to protect the system from unauthorized access. The latter are mechanisms that enable the information to be protected according to security policies, by allowing access to shared resources only to authorized users.

In this paper, an ontological model specifying a policy-based approach to managing the groupwork organizational structure is supplied. This ontology-based policy establishes who authorizes users' registration, how the interaction among users is carried out, and how users' participation is defined (for example, by turns). In addition, we use a set of services that help us to provide authentication and access control to the shared workspace. The concepts related to the access control are designed as services in accordance with the ontological model of the groupwork organizational structure. In this way, the user access to the collaborative system as well as to the shared resources is facilitated.

Generally, the interaction among services is coordinated using Business Process Management (BPM), which is a top-down methodology designed to organize, manage and analyze the processes of an organization, and to undertake reengineering processes. Business processes exist as logical models that can be represented by ontologies. The ontology we propose serves as a logical model for managing the services related to access control on collaborative applications.

We consider long-term groupware applications, where sharing information takes place at various stages. A stage in a coordination model is defined as each of the collaboration moments [6]; for example, a conference management system has several stages: submission, assignment, review, and acceptance of papers. Each stage controls the roles that can participate in it, which facilitates the authentication and interaction among users in the shared workspace. None of the existent access control models for collaborative applications that we have studied takes into account this concept of stage, which facilitates the adaptation process and the access control to the shared workspace in a collaborative system.

This paper is organized as follows. Section 2 gives a brief introduction to the access control. Section 3 explains the ontological model that allows us to define several organizational structures and modify them in runtime. Section 4 describes service-based access control management and presents a real application case based on a well-known collaborative application: a conference management system. Finally, we present conclusions and outline future work.

2 Access Control

Access control models are used to decide how the available resources in the system are managed. In order to implement these models effectively and appropriately into collaborative systems, the following requirements have to be taken into account [4], in such a way that access control must:

- be able to protect any type of information or resources at different levels of granularity.
- facilitate transparent access for authorized users and rigorous exclusion of unauthorized users in a flexible manner that does not constrain groupwork.
- be expressive enough to allow high level specification of access rights, thereby managing better the increased complexity that groupware introduces.
- be dynamic, that is, it should be possible to specify and change policies at runtime.
- support delegation, revocation and management of access policies (meta access control) at runtime.
- grant access control by considering the current context of the user.

There are several access control models for collaborative environments [12], such as *Access Matrix Model* [8], *Role-Based Access Control* (RBAC) [9], *Task-Based Access Control* (TBAC) [11], *Team-Based Access Control* (TMAC) [10], *Spatial Access Control* [4], and *Context-Aware Access Control* [5]. RBAC is very effective and the most important and popular for traditional and collaborative systems, but it has several weaknesses:

- The roles in RBAC lack flexibility and re- ›onsiveness to the environment.
- RBAC supports the notion of role activation within sessions, but it does not go far enough to encompass the overall context associated with any collaborative task.
- RBAC lacks the ability to specify a fine-grained control on individual users playing certain roles and on individual object instances.
- The specification of constraints has not been discussed in the RBAC model. Constraints are an important aspect of role-based access control and a powerful mechanism for laying out higher-level organizational policy.

We take the idea of RBAC that permissions are assigned to roles rather than users. In this way, the policy has not to be changed when users modify their role within the organization.

3 Ontology of the Group Organizational Structure

We present a model that specifies the groupwork organizational structure taking into account all its static and dynamic aspects, and that allows control of the access to the shared workspace and resources. We use an ontology to model this structure. An ontology, according to Gruber, *is a formal and explicit specification of a shared conceptualization* [7]. A domain is specified using the following ontology elements: *concepts*, *relations*, *axioms* and *instances*. In our work, the conceptual modelling of the groupwork organizational structure (see Figure 1) considers the following *concepts*:

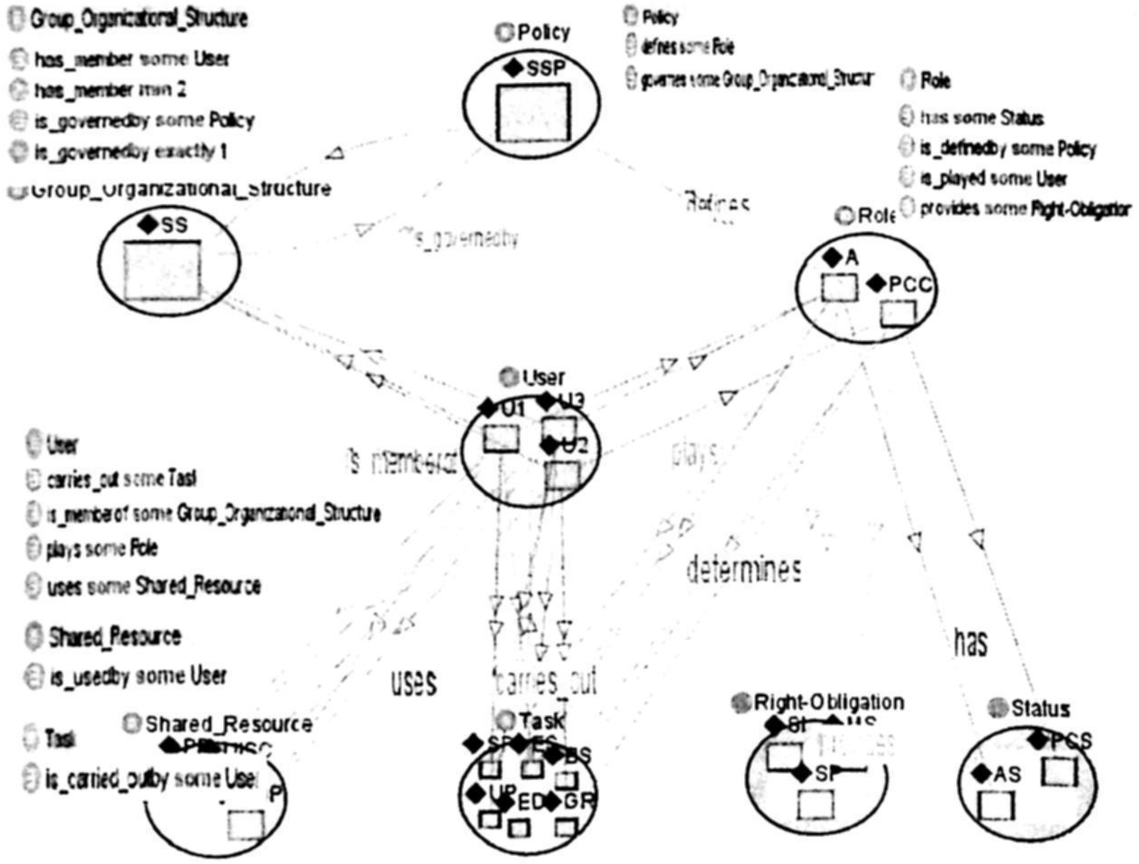

Fig. 1. Ontology of the groupwork organizational structure based on OWL [13] representing the instances for the submission stage. Figure generated by Jambalaya [15] plug-in for Protégé [14].

- *Group_Organizational_Structure*, which is governed at a given stage by a specific policy.
- *Policy*, which configures the above mentioned structure and defines a set of roles.
- *Role*, which must be played at least by one user so that the policy can operate in an appropriate way. Each role has a status and a set of rights/obligations.
- *Status*, which defines the role authority according to the relative position of this inside the organization.
- *Right-Obligation*, which constrains the user actions in the groupwork.
- *User*, which is a person or computational entity that plays one or more roles and carries out collaborative tasks.
- *Task*, which is a set of activities carried out by one or more users to achieve a common goal.
- *Shared_Resource*, which represents the resources used to carry out a task.

The concepts are associated by means of the following *relations* (see Figure 1):

- *is_governedby* (Group_Organizational_Structure, Policy), which specifies that a group organizational structure is governed by a policy at a given stage.

- *is_memberof* (User, Group_Organizational_Structure), which defines that the user is member of the group organizational structure.
- *defines* (Policy, Role), which specifies that the policy defines the roles.
- *plays* (User, Role), which indicates what roles can be played by the user.
- *has* (Role, Status), which determines that the role has a status.
- *provides* (Role, Right/Obligation), which specifies that each role provides a set of rights/obligations.
- *determines* (Role, Task), which indicates that each role determines the tasks that a user playing it can carry out.
- *uses* (User, Shared_Resource), which defines the shared resources used by the user.
- *carries_out* (User, Task), which specifies that the user carries out one or more collaborative tasks in a given stage.

Some of the main *axioms* of our ontology are (see Figure 1):

- A group organizational structure is only governed by a policy in a certain stage.
- A group organizational structure has at least two users.
- Each policy defines at least one role.
- Each role has to be played by at least one user.
- Each task has to be carried out by at least one user.

In order to define the *instances* of our ontology we consider a conference management system, which facilitates the electronic submission, assignment, review, and acceptance of papers (we refer to each of these collaboration moments as a *stage*), along with the management of the whole process. Commonly the roles that can participate are: Author (*A*), Program Committee Chair (*PCC*), and Member of the Review Committee (*MRC*). In accordance with each stage, certain roles can participate, carrying out different tasks. The boxes in Figure 1 show the *instances* corresponding with the submission stage of papers, which are:

- Group organizational structure: Submission Stage (*SS*).
- Policy: Submission Stage Policy (*SSP*).
- Role: The roles that can be played at this stage are:
  a. Author (*A*), whose *status* is *AS* and *rights/obligations* are submitting paper and information (*SP*, *SI*), and has the following tasks associated: submitting paper and information (*SPI*), editing submission (*ES*), and uploading paper (*UP*).
  b. Program Committee Chair (*PCC*), whose *status* is *PCS* and *rights/obligations* are managing the system (*MS*), and can *carry out* the following tasks: browsing submissions (*BS*), extending deadline (*ED*), and generating reports (*GR*).
- User: We define four users: *U1*, *U2*, *U3* and *U4*.
  a. *U1* and *U3* play the *A* role and they use the shared resources: their paper (*PP*) and the user interface that only allows submitting a paper (*UISP*).
  b. *U2* plays the *PCC* role, which uses the user interface without any constraint (*UIWC*).

c. *U3* and *U4* play the *MRC* role. Since this role does not participate in the submission stage, the user *U4* is not shown in Figure 1. *U3* is shown in the figure due to his/her also playing the *A* role.

4 Service-Based Access Control

The success of collaborative systems mainly depends on their capability to be reused and adapted to different and dynamic collaborative scenarios. A change in the groupwork objectives, the participants involved, the group structure, etc. of a collaborative scenario can make a previously successful collaborative system unsuitable for the new situation. The potential solution to these adaptation and reuse problems is nowadays a recognized benefit of Service-Oriented Architecture (SOA), since it is easier to reuse a service that supports the common functionality of several applications than reusing complete applications across different scenarios. The adaptation can be achieved by replacing or even only modifying one or several application services in order to change solely that part of the application that did not fit the characteristics of the new scenario. For this reason, we have proposed a SOA-based layered architecture that facilitates the development of adaptive and adaptable collaborative applications [1], along with authentication and access control to the shared environment. This proposal presents the following elements (see Figure 2):

1. **Application Layer**, which contains applications (such as a conference management system, a chat, etc.). Each collaborative application provides a user interface (web page) that the user uses to carry out a common goal with other users. This layer makes use of different services of the lower layer in order to supply users with all the necessary aspects to perform groupwork.
2. **Group Layer**, which establishes shared workspaces that are appropriate and adaptable to the needs and dynamics of the groupwork. It includes several services: Session Management, Session Management Policy, Registration, and Shared Management. We will devote a subsection to explaining in detail each of them. The interaction among these services is controlled and managed by the lower layer.
3. **Control Layer**, which contains two services: Services Control and Adaptation Control. The former, which models the business logic using the concepts and relations of the ontology proposed in Section 3, also describes the interaction flow of the services related to access control. The latter manages the adaptation process to provide adaptive or adaptable collaborative applications.

In the next subsections, devoted to explaining in detail each of the services related to access control, we will focus on the actions that allow control of interactions among users and avoid inconsistencies in the application due to cooperative and competitive activities.

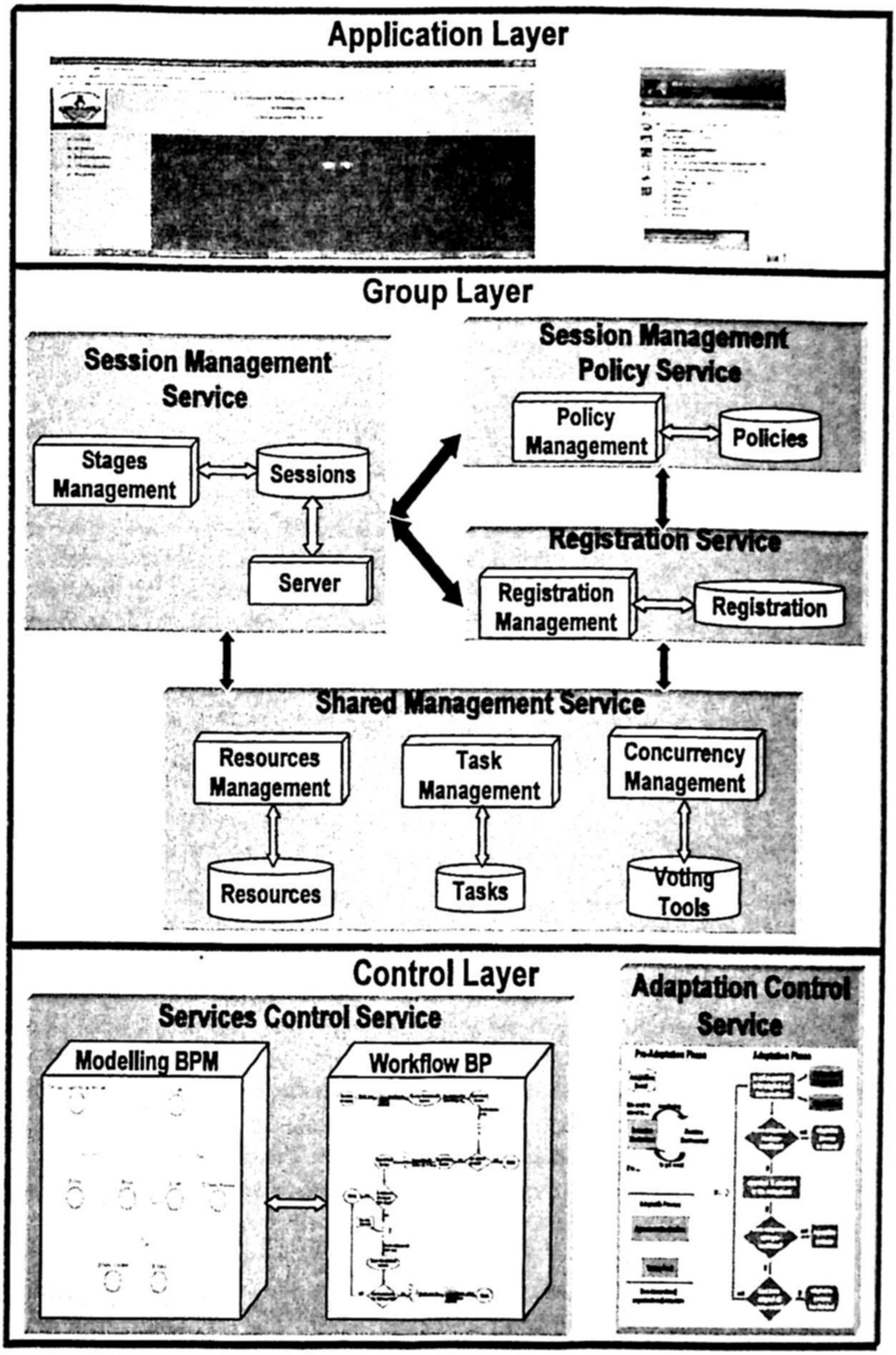

Fig. 2. General schema of the SOA-based layered architecture for collaborative systems.

4.1 Session Management Service

This service allows asynchronous and synchronous sessions to support complex and large-scale collaborative applications, and provides a mechanism that allows users to connect to sessions, as well as to join, leave, invite someone to, and excludes someone from a session. In order to facilitate the access control, this service is able to:

- Manage and control the session and the connections of users to it; in this way collaborative work is allowed.
- Store information about a user and his/her work session; with this it is possible to identify the users that are connected to each session (a user can participate in more than one session).
- Manage the collaboration moments, by establishing when a stage begins and ends. When a stage ends, it is possible to change the current policy to another more appropriate.
- Grant to the Session Management Policy Service the control of the groupwork organizational structure in order to adapt the application to the group changes and to the new needs of the groupwork.
- Send the information of the stages to the Registration Service, which associates this information with the roles that each user will play at a specific stage.
- Allow the Shared Management Service to manage all the aspects related to the shared resources in accordance with the organizational structure defined by Session Management Policy Service.
- Notify all the Group Layer services of the moment in which a user joins or leaves a session so that each service can carry out the necessary actions to avoid inconsistencies in the application.

4.2 Session Management Policy Service

This service uses the ontology presented in Section 3 to establish the groupwork organizational structure (see Figure 1) and define the access control to the shared workspace as well as to the shared resources. Consequently:

- It stores the information related to the ontology: concepts, relations, axioms and instances. This allows us to modify individually the instances of each concept without having to change the others.
- It associates roles with the current stage defined in the Session Management Service. This establishes the authorized users that can participate in a specific stage; therefore, it facilitates the access control to the shared workspace.
- It checks the constraints to be fulfilled so that the current policy always is valid; for example that each role always has to be played by at least one user.
- It sends the information about the roles that can be played at each stage to the Registration Service, in order to control access to the collaborative system.
- It transmits the tasks that must be carried out in this stage and the shared resources that can be used for these tasks to the Shared Management Service.

4.3 Registration Service

This service allows the registration of new users in an existing session through the user interface (web page) of the Application Layer so that they can participate in the groupwork. To control access to sessions, this service carries out the following operations:

- It authenticates the access to the session when the user inputs a login and a password, and stores these data, which are used by the system to corroborate that he/she is an authorized user.
- It stores the information that users submit at the moment of their registration in the system (in the case of our example, this service stores personal data and information about the paper).
- It assigns the adequate role to the registered user taking into account the valid roles at the current stage and the data submitted by him/her.
- It associates the role with its corresponding rights/obligations and its status, which determine the user behaviour in the collaborative application.
- It sends the information on roles, rights-obligations and status to the Shared Management Service so that this service can manage the interactions among users and the uses of shared resources by users.

4.4 Shared Management Service

This service enables management of the shared context, since it takes into account the session management policy in order to determine who can carry out a task and which shared resources can be used for this task. It uses the concurrency mechanism (should this be necessary) to avoid conflicts due to cooperative and competitive activities. The Shared Management Service carries out the following operations:

- It stores the tasks that must be carried out in the shared workspace.
- It connects the tasks with the roles that can perform them.
- It checks that each task is carried out by authorized users, i.e., users playing a role which allows them to perform the corresponding task.
- It stores the existing shared resources in the collaborative application.
- It associates the shared resources with the tasks to be carried out.
- It corroborates that the collaborative work is suitably carried out in accordance with the workflow of the Services Control Service.

4.5. Services Control Service

This service controls the interactions among the services related to the access control by modelling the business logic and specifying a workflow in accordance with the model. In this work, we use the ontological model specified in Section 3. The workflow establishes the interrelations of each user with the shared resources and other users as well as the different services related to the access control. Figure 3 shows the workflow for submitting a paper using the conference management system.

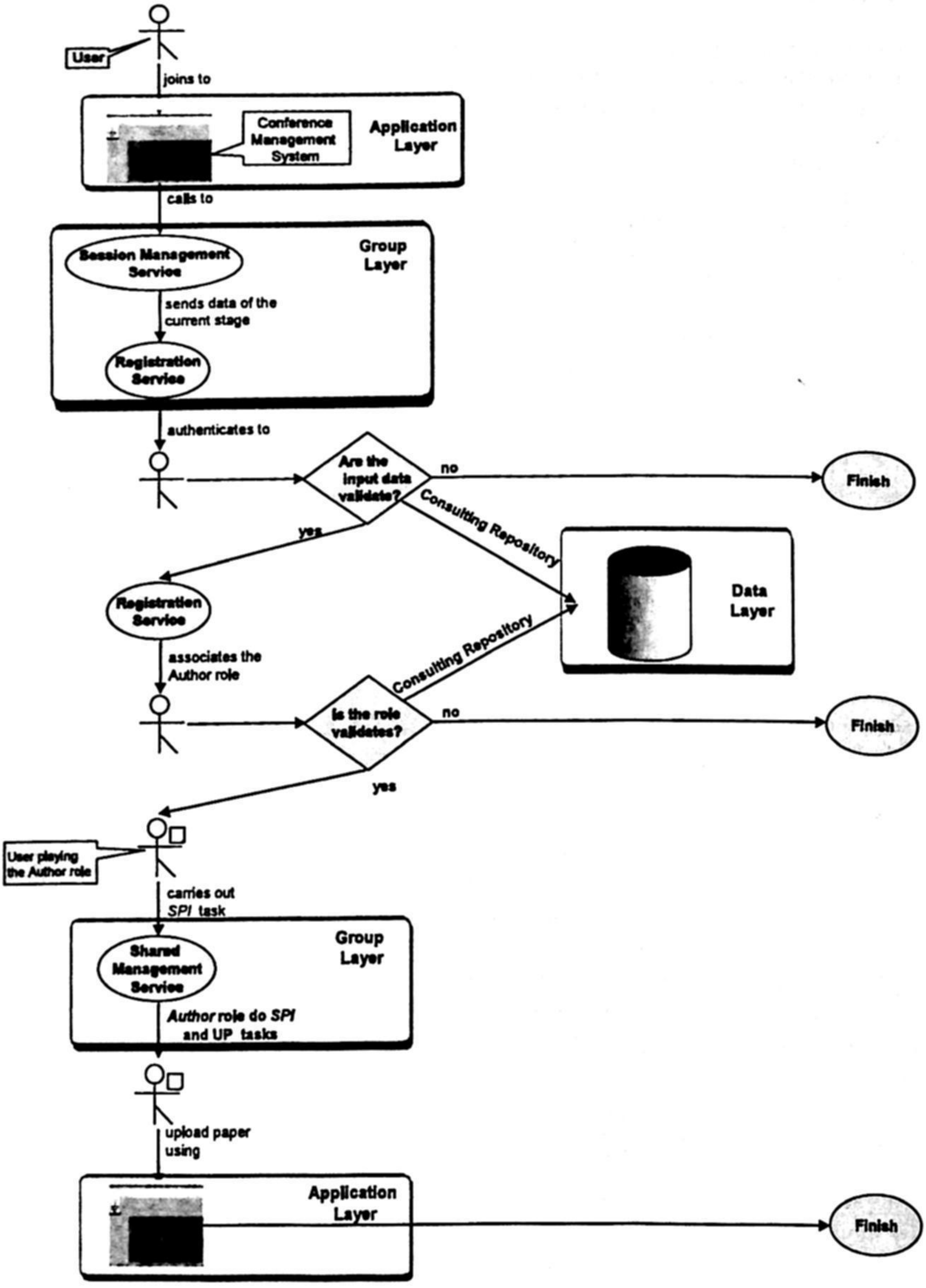

Fig. 3. Workflow for submitting a paper using a conference management system.

5 Conclusions and Future Work

In this paper, we have presented a service-based access control, which facilitates the access of users to the shared workspace. This provides all the related aspects to the security of collaborative systems, and enables groupwork with more sophisticated and appropriate access policies, as well as supporting the dynamic nature of the groupwork, in such a way that permissions, roles and constrains are part of the model; therefore, they can be changed in a just-in-time fashion. The access control uses as business logic the ontology of the groupwork organizational structure. This ontology enables adjustment of the organizational structure (for example, changing the role that a user can play in a session and/or the rights/obligations of a role), and the transformation of the model (for example, adding a new role or changing the policy established). Moreover, we have also used the concept of stage to facilitate the access control.

With regard to future work we think that it is necessary to analyze and model the access control management from the first step of the development of collaborative systems at both architectural and organizational level, in order to facilitate their dynamism and greatly decrease their final cost. In addition, we are studying how the changes in the group organizational structure affect in order to consequently adapt the access control.

References

1. Anzures-García, M., Hornos, M.J., Paderewski, P.: Architecture for Developing Adaptive and Adaptable Collaborative Applications. Lecture Notes in Computer Science (LNCS), Vol. 4758, pp. 271-274. Springer, Heidelberg (2007)
2. Beaudouin-Lafon, M., et al.: Computer Supported Cooperative Work. Trends in Software, John Wiley & Sons (1999)
3. Beaudouin-Lafon, M.: Beyond the Workstation: Media Spaces and Augmented Reality. In Proceedings of the Conference on People and Computers IX. Vol. 9, pp. 9-18 (1994)
4. Bullock, A., and Benford, S. An Access Control Framework for Multi-User Collaborative Environments. ACM GROUP (1999)
5. Covington, M., Long, W., Srinivasan, S., Dey, A., Ahamad, M., Abowd, G.D.: Securing Context-Aware Applications Using Environment Roles. In ACM Symposium on Access Control Model and Technology (2001)
6. Ellis, C., Wainer, J.: A Conceptual Model of Groupware. Proceedings of the ACM conference on CSCW, pp. 79-88 (1994)
7. Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human Computer Studies. Vol. 43(5/6), pp. 907–928 (1995)
8. Lampson, B.: Protection. In Princeton Symposium on Information Science and Systems, pp. 437-443. Reprinted in ACM Operatives Systems Rev. Vol. 8-1, pp. 18-24 (1974)
9. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based Access Control Models. IEEE Computer. Vol. 29-2, pp. 38–47 (1996)
10. Thomas, R.: Team-based Access Control (TMAC). In Proceedings of 2nd ACM Workshop on Role-Based Access Control, pp. 13–19 (1997)

11. Thomas, R., and Sandhu, R.: Task-based Authorization Controls (TBAC): Models for Active and Enterprise-Oriented Authorization Management. In Database Security XI: Status and Prospects. Lin, T. Y., and Qian, X. (eds.) North-Holland (1997)
12. Tolone, W., Ahn, G., Pai, T., Hong, S.: Access Control in Collaborative Systems, ACM Computing Surveys. Vol. 37-1, pp. 29–41 (2004)
13. OWL Web Ontology Language Guide, http://www.w3.org/2004/OWL/
14. Protegé Platform, http://protege.stanford.edu/
15. Jambalaya Plug-in, http://www.thechiselgroup.org/jambalaya