# COPSE-Web: An Infrastructure for Developing Web-Based Groupware Applications

José Maria N. David[1] and Marcos R.S. Borges[2]

[1] COPPE-PESC / UFRJ &
Faculdade Ruy Barbosa / BA
david@cos.ufrj.br
[2] NCE & IM
Federal University of Rio de Janeiro
mborges@nce.ufrj.br

**Abstract.** The main approach for the development of groupware applications presented in the literature, has been the use of toolkits. However, at least in the case of groupware applications the main advantages of toolkits have not been fully accomplished. They don't offer the necessary flexibility to address the social aspects related to the interaction of geographically dispersed groups. Although the web is the main application environment for current applications, only recent toolkits have been able to support the groupware development for the web. In this paper, we discuss and describe the work carried out in converting COPSE to COPSE-Web, an infrastructure for the development of web-based groupware applications. We used this work to discuss the requirements, the architectural and implementation issues in the design of an integrated groupware toolkit. We illustrate the advantages of our approach by describing the development of the environment's administration tool.

**Keywords:** World Wide Web, CSCW framework, groupware toolkit, Web-based development.

## 1 Introduction

The main approach for the development of groupware applications, presented in the literature, has been the use of Toolkits. An appropriate toolkit saves time and programming efforts. Moreover, it allows an easy integration of different tools under the same environment. These issues have been the motivating factor for the development of general toolkits, such as Groupkit [12] and specific domain development environments, like Habanero [13], amongst others.

In the case of groupware applications, however, the main advantages of toolkits have not been fully exploited. Besides not offering the necessary flexibility to support cooperative activities, some tools limit the scope of the application [5]. Also, most tools do not address the social aspects related to the interaction of the geographically dispersed groups. Under a different perspective, although the web is the main application environment for current applications, only recent toolkits have been able

J.M. Haake and J.A. Pino (Eds.): CRIWG 2002, LNCS 2440, pp. 275–284, 2002.

to support the groupware development for the web. To sum up, very few toolkits are available which can satisfy the requirements for a complex groupware, without extensive programming.

We have been developing groupware applications using COPSE[1] infrastructure [4, 18]. COPSE is a comprehensive infrastructure based on a client-server architecture. It has some nice features that not only provide facilities for the development of new applications, but also allow their full integration with applications already present in the environment [6]. However, the COPSE infrastructure is not prepared for a web-based application development and lacks many important features, which will be required when developing the complex asynchronous applications.

In this paper, we discuss and describe the work carried out to convert COPSE to COPSE-Web, an infrastructure designed for the development of web-based groupware applications. Although the conversion is our main motivation, we used this work to discuss the requirements, the architectural and implementation issues in the design of an integrated groupware toolkit. We illustrate the advantages of our approach by describing the development of the environment's administration tool.

In section 2 we discuss the requirements for a web-based and comprehensive groupware development environment. Then, in Section 3, the architecture of a groupware infrastructure designed to support these requirements is proposed. In section 4, the implementation issues of the proposed architecture are presented. Section 5 we make a brief analysis of other toolkits. The use of the resulting infrastructure is illustrated by the development of an application, described in Section 6, and finally Section 7 concludes the paper.

## 2 Requirements for Web-Based Groupware Development

There are several toolkits, which support the development of non-web groupware applications, and they very seldom allow these applications to be implemented without an extensive programming effort. In most cases, a specific application solution, such as browsing [14], communication tools, meetings support tools, text edition tools, is implemented. Unfortunately, these applications are built frequently without focusing on the complexities of groupware activities and addressing the inherent needs of a groupware application, for example, integration with other tools. Most toolkits do not offer an open source code, which would permit customization and extension to support different group's needs.

The group's participants usually do not have experience in collaborating. Moreover, new groups of expertise need to be invited to contribute in different locations. That is, the development of these systems demands different roles and competencies to be accomplished. In addition, a high level of flexibility is needed through the extendable components, protocols, services, which could improve the collaboration results among certain heterogeneous groups.

Other points that should be addressed when designing web-based groupware applications are discussed in [15]. In [14] some issues related to the design of *Collaborative Browsing* systems is presented in a five-dimensions framework, which points out some aspects that should be considered when designing such systems.

[1] COPSE – *COllaborative Project Support Environment*

The increasing demand for network services and also the number of available web-based applications have given evidence to the fact that the Internet is a convenient infrastructure to be accessed in order to look for different information, and deployed mainly in cooperative activities [1]. Thus, we can suggest that web-based technologies can be used not only to support the development of groupware applications, but also the integration of these tools and its interoperability during the group's activities.

The client-server model has already shown advantages when deployed in a groupware architecture design and implementation [10]. However, web applications currently do not make a clear separation between the presentation of the information, and the control and logic of the groupware applications. On the contrary, they use technology that restricts their scalability and portability. Considering the inherent complexities of CSCW systems, the technology deployed by them is not adequate enough to be used by groupware applications.

## 3 The COPSE-Web Architecture

COPSE environment can be considered as the first step in the direction of supplying a collaborative work environment and a tool framework for the cooperative development of software. Its infrastructure supports not only synchronous but also asynchronous activities, which are accomplished in the environment. Furthermore, a tool framework is supplied, allowing the development of groupware applications to be coupled to the environment [5, 6]. This current version is a Java-based application, which does not implement a object-oriented middleware like CORBA, for instance. All integration issues between applications rely on group memory [6]. Aiming at supporting the development of web-based groupware applications this environment was extended to fulfill the requirements to support web technologies.

The proposed architecture for a groupware environment in the web should allow any client running a browser to have the ability to connect to any available application running in the environment. Such applications and other web services do not necessarily need to be running on the same web server. Users need only be authenticated in the environment through a single logon, so that they can execute the available groupware applications at that time.

The architecture we have designed is illustrated in Figure 1. COPSE-Web server is the key component of the designed infrastructure. Each server has an associated user administration module. Besides supplying relative information about users running an application, this module also communicates with a trusted directory service server installed in the network. Thus, the additional functionality related to the applications and participant's information could be managed, such as the set of cooperative tools started by each participant, or interested topics and subjects.

Each application running in a COPSE-Web server has the support of a group memory, which stores specific information related to the application and the activity that is being supported. For example, discussion support tools and elicitation process support tools could provide specific components to measure the participation and contribution rate for each participant [2, 4]. These components are related to each

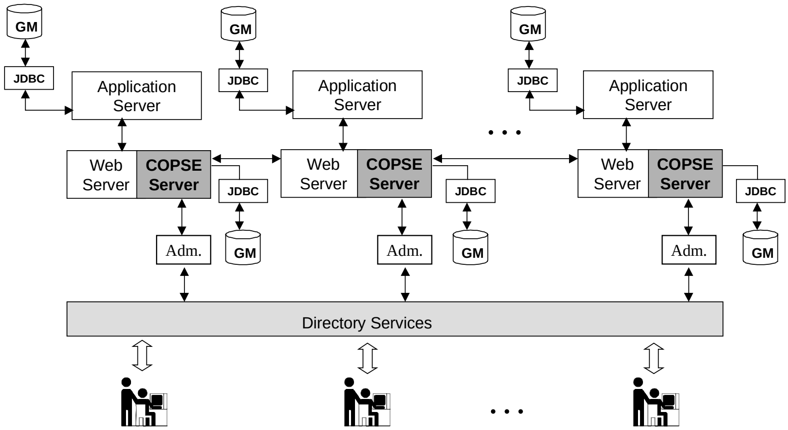

**Fig. 1.** A groupware architecture in the Web

activity carried out by the participants. COPSE servers have also group memories attached to them, which are responsible for the storage of common information to any applications and any user in the environment.

## 4 Implementation Issues

One of the main reasons for the selection of COPSE as the infrastructure to be extended for this project was the available open source code and existing documentation. Other characteristics also contributed to the extension of this infrastructure, for instance: (i) it is a tool that allows the integration of other CSCW applications and allows customizations in other functionalities of these applications; (ii) it offers an open code making some possible adaptations and extensions to the group needs, whilst keeping portability, scalability and interoperability of the applications; (iii) it was originally developed in Java programming language.

In this context, we also needed an architecture model that could be applied so that the system's functionalities could be adequately separated. Moreover, we should moreover build a cohesive and significant set of functionalities necessary to support the group activities. Through this model we should be able to provide the flexibility in software development activities.

Analyzing the *Model-View-Controller* (MVC) model, it was possible to visualize some benefits that the implementation of this model would bring to this project. In the view layer, the pages for information exhibition in the client browser are found. These pages can be codified in HTML, XML, JSP, or another presentation language can be used on web. How the information is obtained and processed is not its function. In the control and model layers, mechanisms capable of controlling and modeling data flow are found respectively. The model layer is responsible for activities such as, data management and database queries. The control layer is responsible for the interaction between the front-end (View) and the back-end (Model). The decisions process related to different presentation modes are kept in this layer [8].

However, it would be necessary to choose a technology that could be used together with the previously designed COPSE architecture [5, 6]. Only WWW and pure Java-based applications do not offer the necessary support for the implementation of this architecture and the designed mechanisms. One example of this is the COPSE-Web servers' initialization task. There was a need to use technology capable of supporting the remote start and stop of different COPSE-Web application servers.

Through the analysis of the current web technologies, we decided to use Java Server Pages (JSP) technology to provide not only a user session management service, but also a dynamic functionality to the system providing personalized web page contents [8]. While this technology provides a more substantial support for asynchronous groupware applications, it is not suitable for synchronous applications when changes in any information (model) are made and have to be sent to client browser (view). Furthermore, it would require frequent pages reload by each group participant. Then, applets were designed in COPSE-Web to be used in synchronous groupware applications.

For the development of COPSE-Web architecture, we used a servlets centered approach for the design of applications running in the environment. In the extended architecture a set of servlets can accomplish some activities drafted initially for the *User Session Management* module (Figure 2), in which each user can establish a connection with the project server through environment authentication. Thus, important user information, such as information about the tools started by them, is directed to the *Web Server GUI Servlet*, and soon after to the *Web Server GUI JSP* in order to constructs the web page.

Tools launched by each user establish a connection to secondary servers (special servers or document server). Different tools share the same document set in the Document Server. The Process Server is responsible for the execution of collaboration process and for the association to a project [5, 6, 17].

COPSE-Web environment implements not only a Project Server – responsible also for the database maintenance of relative documents to the projects – but also other servers, namely *Special Servers*, which can be found in the data layer (Figure 2). Dias and Borges [6] present some servers coupled to the environment, and likewise in [17] Santoro and Borges present some adaptations both in the COPSE conception and implementation.

In this architecture, we propose the development of another server: the *Profiles and Awareness Server*. This server is directly associated with the profile mechanisms shown in the Figure 2. It also offers generic awareness components for the environment, or specific awareness components for the development of groupware applications. The implementation of this server and the mechanisms associated to each *User Session Management* is part of an awareness service, which is under development and is to be coupled to the environment at a later date.

## 5 Related Work

Groupkit [12] was designed to support the development of synchronous CSCW applications. However, we observed that several works like this don't contemplate the

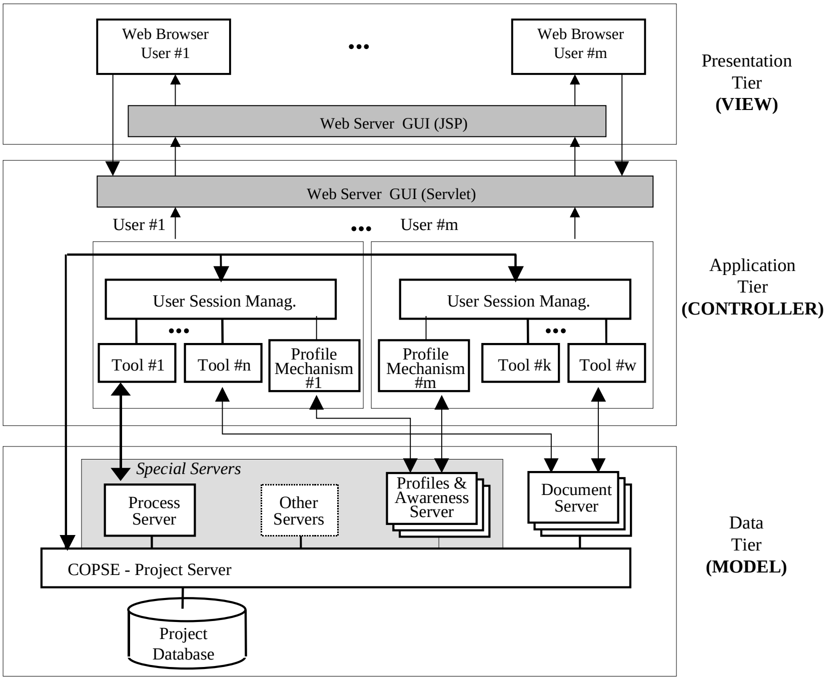

**Fig. 2.** Architecture of COPSE-Web (adapted from [6])

users' fundamental needs of groupware application. They support neither the transitions between synchronous and asynchronous activities, nor the dynamic support to group awareness [3, 4, 5, 11].

MetaWeb [21] toolkit provides support for synchronous web-based groupware applications. The development of applications are based on three concepts: 'users' interact with the toolkit, 'locations' are places where the group interaction takes place and 'sessions' is a medium through which better group communication can take place. Furthermore, it provides support for access control but a single server approach does not allow interoperability between different needed applications to support cooperative work.

Promondia [10] is a Java-based framework that provides real-time group communication for groupware applications in the web. This functionality was implemented as Java applets, which are loaded from any Promondia server. Several communications tools such as chat, shared white board, voting and surveys are provided. This framework does not support other group activities and the infrastructure is proposed only to support group communication for WWW.

Using the metaphor of rooms, TeamWave Workplace [11] allows tools to be placed on a shared space and customized according to the group's needs. However, all participants share the same information and events. COPSE-Web provides resources to configure shared workspaces with mandatory awareness components. Each group's participant can accomplish an individual workspace configuration.

Although considering integration issues of web technologies, ESCOT [7] project supports group activities in order to compose lessons in a specific domain. As in this project, the approach proposed in COPSE-Web suggests that a multidisciplinary activity can be supported through components reuse and their interoperability.

NESSIE [16], PoliAwaC [20] and AREA [9] also give us some directions about some aspects to be included in the software requirements specification of the awareness service in COPSE-Web. Our proposal is focused on ‘profile of interest’ to filter awareness components, and as a result, to determine coordination actions.

WWG [15] is a distributed and decentralized infrastructure, which supports distributed group learning and team work activities. This infrastructure is based on event distribution mechanisms that provide an information exchange with many geographically dispersed groups. WWG does not describe how information is presented to users, but describes how different interfaces are provided in order to collect and to propagate events, and to notify the participants about their activities. This functionality has inspired us to design a more adequate event mechanism, which is to be used in the COPSE-Web awareness service.

## 6 Applications

Aiming at evaluating issues that have been discussed previously, an application was developed according to the concepts and to the architecture described above. This application provides administration services to the environment. It also communicates with the administration server that centralizes the users’ data. Through a directory service, this server allows a single logon in the environment, from the moment the user is registered by the use of administration module. Each user has access to the information and the user’s properties (user name, password, email address, user projects and profiles, for example), for reading purposes only, depending on the access control rights assigned by the administrator of the COPSE-Web environment.

In the main COPSE-Web options menu some items were implemented, others are still under development. Among the implemented items we can find the *‘Servers’* menu item, which allows to start/stop different legacy application servers available in the environment. By selecting the *‘Tools’* menu item, each participant can start any COPSE-Web enabled groupware applications. The *‘Awareness’* item allows any user to load generic awareness components in the shared workspace, such as the active user list and active cooperative tools in each group participant workspace. Other specific awareness components related to the groupware application are only shown in each application menu toolbar. For example, in a pre-meeting application, each participant can start an awareness component showing the invited members and their position about the coordinator’s invitation. In addition, they can launch the *participameter* component [2]. These awareness components are still under development as part of another groupware application to support pre-meeting activities in the web. In the *‘Communication’* menu item, chat and message tool for example, can be started in the common workspace. In the *‘Documents’* menu item, which is at present still under development, is a document database administration tool, this can be started in order to manage all information produced by the group in the form of documents in the environment.

Legacy applications have already been integrated to the COPSE-Web environment and can be launched by the use of main option menu. In the earlier versions some of these applications were developed by the use of COPSE tool framework and were not extended to the COPSE environment. Others have already been developed using available tools in COPSE environment and they have already been integrated to the environment, such as Editex [18] – Cooperative Text Editor -, and the CEPE[2] [19] - Cooperative Editor of Process Elicitation. To be fully integrated to the environment and web browsers, they should be extended to the COPSE-Web environment according to the model.

## 7 Conclusions

When we planned to design this architecture, we tried mainly to minimize the client's complex processing procedure, just leaving the activities related to the information presentation in the front-end. On the server's side, we can find the application control functions and the data manipulation associated to the applications. We also noticed that by using a servlets centered approach, groupware applications could be developed in an evolutionary way, while preserving the inherent dynamic behavior of web-based applications. That is, new interface components and new web technologies could be developed, and become available to the user. At the same time, control and data management related mechanisms could evolve without changing applications. In these applications, sequences of actions can frequently be interrupted, demanding appropriate treatment by the technology used. This was one of the reasons why we chose JSP technology.

As this technology provides a dynamic functionality to COPSE-Web server, applets technology provides interactive functionality to a special-purpose synchronous groupware application. Furthermore, by using this architecture, we could preserve legacy groupware applications, which were developed according to the concepts of previous COPSE versions, and it was not possible to change these applications at that time. In the future, we are planning to extend their functionalities to web browsers.

Some issues related to the distributed groupware architecture were not specified and we are also investigating how to integrate different applications in order to establish adequate communication between them, without an intensive programming effort.

**Acknowledgments.** This work has been partially supported by a grant from Faculdade Ruy Barbosa (Salvador / BA – Brazil).

[2] In a second version of the system CEPE, new functionalities are being built, but according to the previous COPSE version.

## References

[1] Bentley, R., Horstmann, T., Trevor, J., "The World Wide Web as enabling technology for CSCW: The case of BSCW". In: *Computer-Supported Cooperative Work: Special Issue on CSCW and the Web*, vol. 6., 1997, pp. 111–134.

[2] Borges, M. R. S. and Pino, J. A. , "Awareness Mechanisms for Coordination in Asynchronous CSCW". In: *Proc. 9th Workshop on Information Technologies and Systems (WITS 1999),* Charlotte, North Carolina, Dec. 1999, pp. 69–74.

[3] David, J. M. N., Borges, M. R. S., "Improving the Selectivity of Awareness Information in Groupware Applications". In: *Proc. of the Sixth International Conference on CSCW in Design*, London, Ontario, Canada, Jul. 2001, pp. 41–46.

[4] David, J. M. N., Borges, M. R. S., "Selectivity of Awareness Components in Asynchronous CSCW Environments". In: *Proc. of Seventh International Workshop on Groupware (CRIWG'01),* Darmstadt, Germany, Sept. 2001, pp. 115–124.

[5] Dias, M. S., "COPSE – Um Ambiente de Suporte ao Projeto Cooperativo de Software". Master Dissertation in Portuguese. COPPE/UFRJ, Rio de Janeiro, RJ, Brazil, Jul. 1998.

[6] Dias, M. S. and Borges, M. R. S., "Development of Groupware Systems with the COPSE Infrastructure". In: *Proc. of 5th International Workshop on Groupware (CRIWG'99),* Cancún, México, Sept. 1999, pp.278–285.

[7] ESCOT, 2002, URL http://www.escot.org (web page as of May 2002).

[8] Fields, D. K. and Kolb, M. A., "Web Development with JavaServer Pages", Manning Publications Co., 2000.

[9] Fuchs, L., "AREA: A Cross-Application Notification Service for Groupware". In: *Proceedings of the Sixth European Conference on Computer-Supported Cooperative Work (ECSCW'99),* Copenhagen, Denmark, Kluwer Academic Publishers, Sept. 1999, pp. 61–80.

[10] Gall, U. and Hauck, F., "Promondia: A Java-Based Framework for Real-time Group Communication in the Web". In: *Proc. of Sixth International World Wide Web Conference*, Santa Clara, California, USA, Apr., 1997.

[11] Greenberg, S. and Roseman, M., "Using a Room Metaphor to Ease Transitions in Groupware". Research report 98/611/02, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, Jan., 1998.

[12] Groupkit, 2000, URL http://www.cpsc.ucalgary.ca/redirect/grouplab/projects/groupkit/ (web page as of July 2000)

[13] Habanero, 2001, URL: http://habanero.ncsa.uiuc.edu/habanero/ (web page as of Jan. 2001).

[14] Rivera, G. J. H., Courtiat, J. and Villemur, "A Design Framework for Collaborative Browsing". In: *IEEE 10th Int. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'01),* 2001.

[15] Marquès, J. M. and Navarro, L., "WWG: a Wide-Area Infrastructure for Group Work". In: *IEEE 10th Int. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'01),* 2001.

[16] Prinz, W., "NESSIE: An Awareness Environment for Cooperative Settings", In: *Proc. of the Sixth European Conference on Computer Supported Cooperative Work (ECSCW'99),* Copenhagen, Denmark, Kluwer Academic Publishers, Sept. 1999, pp. 391–410.

[17] Santoro, F. M., Borges, M. R. S. and dos Santos, N., "An Infrastructure to Support the Development of Collaborative Project-Based Learning Environments". In: *Proc. of Sixth International Workshop on Groupware (CRIWG'00),* Madeira, Portugal, Oct. 2000, pp.78-85.

[18] Santoro, F. M., "Um Modelo de Cooperação para Aprendizagem Baseada em Projetos". Phd Thesis, In Portuguese, COPPE/UFRJ, Rio de Janeiro, RJ, Brazil, Dec. 2001.

[19] Santoro, F., Borges, M. R. S., Pino, J. A., “CEPE: Cooperative Editor for Processes Elicitation”. In: *Proc. of the Collaboration Systems and Technology Track of the Thirty-Third Hawaii Int. Conference on System Sciences (HICSS-33*), Maui Island, Hawaii, USA, Jan. 2000 (in CD-ROM).

[20] Sohlenkamp, M., Prinz, W., Fuchs, L., “POLIAwaC: Design and Evaluation of an Awareness Enhanced Groupware Client”. *AI & Society*, vol. 14, 2000, pp. 31–47.

[21] Trevor, J., Koch, T. and Woetzel, G., “MetaWeb: Bringing synchronous groupware to the World Wide Web”. In: *Proceedings of the Fifth European Conference on Computer-Supported Cooperative Work (ECSCW’97)*, Lancaster, UK, Kluwer Academic Publishers, Sept. 1997.