A Portal for Ubiquitous Collaboration

Federico Bergenti+, Socrates Costicoglou * and Agostino Poggi+

+ Dipartimento di Ingegneria dell'Informazione, University of Parma
Parco Area delle Scienze, 181A, 43100, Parma, Italy
(bergenti,poggi}@ce.unipr.it

*SPACE HELLAS S.A – Athens, Greece
Email: scostic@space.gr

Abstract. This paper presents a software framework, called Collaborator, to provide a shared work-space supporting the activities of virtual teams. This system exploits seamless integration of standard Web technologies with agent technologies, enhancing the classic Web communication mechanisms to support synchronous sharing of applications, and its use through emerging technologies such as: third generation of mobile networks and terminals, and new generation of home appliances. The system presented in the paper is the main result of an on-going European research project Collaborator (IST-2000-30045) that aims at specifying and developing a software distributed environment to support efficient synchronous collaborative work between virtual teams, and experiment such an environment in the construction and telecommunication working sectors

INTRODUCTION

Increasingly more and more people need to access enterprise information remotely and need to collaborate with others from different places. The Web is assuming a central role in the way people share the information in local and geographic areas because browsers are available everywhere and they integrate different services into a common, easily accessible, platform-independent user interface. For this reason, the Web has already been adopted as one of the major media for supporting remote collaboration among people (Bentley et al., 1997; Ramduny &. Dix, 1997; Hodes & Katz, 1999). Nevertheless, the basic communication mechanism that the Web offers is not sufficient to support interactive collaboration. In fact, the communication needs that stimulated the development of the Web were about consulting structured documents and were not about supporting an interactive discussion within a virtual team.

The migration of Web technologies toward mobile networks allows porting consolidated Web services and facilities to the mobile user. The widespread availability of high-bandwidth mobile infrastructures and the availability of operating system and network independent Web development technologies allow people sharing informa-

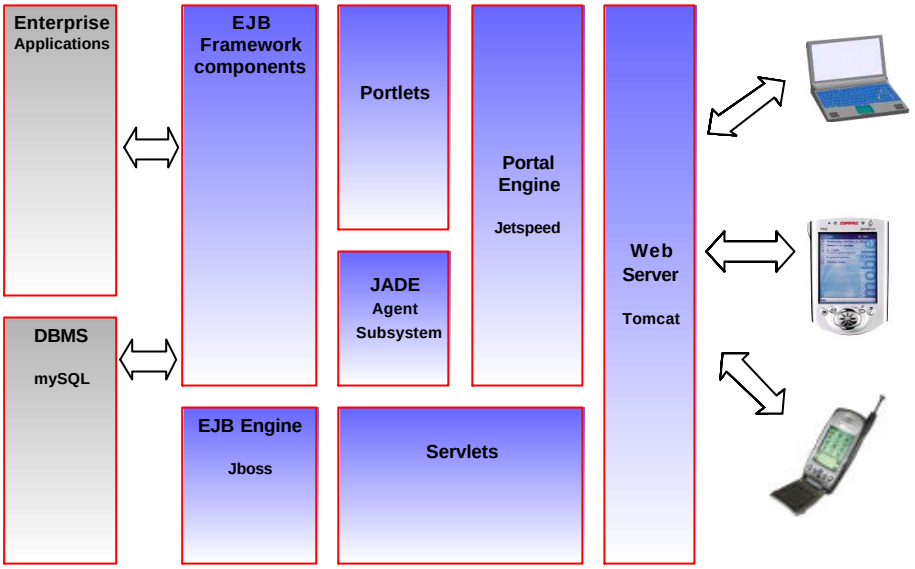

Fig. 1. Collaborator Architecture

tion through heterogeneous hardware, e.g., laptop computers and PDAs, using different operating systems and different kinds of communication protocols.

Nowadays, we have the possibility to provide users with integrated services capable of exploiting the possibilities of both the wired and the wireless network to create flexible and effective virtual teams. Moreover, such integrated services may be more easily integrated and adapted to different users thank to agents. Agents have been used in groupware for a long time because of their social abilities. Ellis and Wainer have recently surveyed (Ellis & Wainer, 1999) the application of agents in groupware and CSCW and they showed the roles that agents may play in such systems. They identified the following roles: i) keepers: agents in charge of the lifecycle of the artefacts produced by the team, ii) coordinators: agents intended to support the coordination of activities within the team, iii) communicators: agents supporting the communication between the individuals in the team, and iv) Team agents: agents acting as artificial participants to meetings.

This paper presents a software framework, called Collaborator, to provide a shared workspace supporting the activities of virtual teams. The next section introduces COLLABORATOR project, Section three describes system architecture and its components. Finally, section four gives some implementation notes and discusses the main features of the Collaborator system.

COLLABORATOR project

COLLABORATOR (Collaborative Framework for Remote and Mobile Users) is a European Commission funded 5th Framework IST Research project (COLLABORATOR, 2001). The project involves seven partners: Space Hellas (administrative project co-ordinator), Consorzio Nazionale Interuniversitario per le Telecomunicazioni (technical project co-ordinator), Atos Origin Integration, Centre Scientifique et Technique du Bâtiment, Tecnologia Automazione Uomo, Technical University of Madrid

and Telecom Italia Lab. The project started on the 1st of November, 2001 and has a 2 years duration.

The major goal of the project is the realization of a software environment called Collaborator to provide a shared workspace supporting the activities of virtual teams. It exploits seamless integration of standard Web technologies with agent technologies enhancing the classic Web communication mechanisms to support synchronous sharing of applications. Another goal of the project is to set up a trial environment to explore and validate the benefits of integrating Collaborator with emerging technologies such as: third generation of mobile networks and terminals, and new generation of home appliances.

Summarizing, Collaborator is intended to support remote and collaborative work in virtual teams offering the following features:

- *platform-independence and Web integration:* Collaborator is based on the standard technologies of the Web (Java, HTML, TCP/IP, etc.) and it is operating-system and network agnostic;
- *ubiquitous accessibility:* Collaborator can be used seamlessly on desktop computers and on handy devices with sufficient processing power and bandwidth;
- *collaboration transparency:* Collaborator supports synchronous sharing of off-the-self applications, thus enabling the co-operative use of everyday tools;
- *multi-document sharing:* Collaborator is not limited to a single application and the virtual team is provided with a shared desktop capable of containing many applications concurrently;
- *adaptability to network bandwidth and terminal capabilities:* ubiquitous accessibility requires Collaborator to adapt to the capabilities of the various terminals and connections people involved in the virtual meeting use;
- *support for unanticipated sharing and late joining:* people can join and leave the virtual meeting dynamically;
- *floor control:* Collaborator allows moderating the virtual meeting providing integrated floor control policies;
- *flexible user interaction through agents:* people involved in the virtual meeting are associated with personal agents mediating their interactions and providing customisations of the system on the basis of their profiles.

The project started from the analysis of the requirements of different working sectors (i.e. the construction and telecommunication sector), and projects results will be assessed by workers from these sectors through scenarios to be executed during trials. This will insure that the results of the project will fit real needs. The results of the project address enterprise requirements and more particularly they will support new methods of work such as: tele-work, networked co-operative work and collaborative work.

COLLABORATOR SYSTEM

The Collaborator system is modelled over the metaphor of shared workspaces adapted to support ubiquitous and seamless access to collaborative tools and services. A shared workspace is a virtual place where end-users, after becoming participants in a session can share applications, exchange application-specific data, co-ordinate and communicate with each other. Each user is presented with a different view of the shared workspace used during a collaborative session, more sessions can run in parallel and a user may participate to more than one section. A session is characterized by its properties: it is initiated, it is ongoing and finally it is terminated. A session can be initiated and terminated only by an coordinator. A user can join and leave a session at any time.

The architecture of the Collaborator system is a classical three tier architecture and consists of a range of Background server components and a very limited number of Web client components deployed on each device (see Figure 1).

From a functional point of view, we can describe the Collaborator system dividing it in three parts:

- *Framework:* dedicated to manage users, sessions, resources and services.
- *Services:* dedicated to provide the different collaborative services
- *Portal:* dedicated to provide information to the Web clients.

3.1 Framework

The core of the Collaborator system is based on four main subsystems delegated to the management of users, sessions, resources and applications execution.

The Session Management Subsystem (SMS) deals with the management of collaborative sessions and all the related resources. The SMS allows the creation, the release, the participation, the abandoning or destruction of a session, and provides an interface to the rest of the components running on the server to perform these tasks. It maintains the state associated with the users and applications and may interact with an object which encapsulates a defined session management policy. The different functions of the SMS can be grouped into four categories: users management (with access and authorization control), applications management, general session management and collaborative session management. Whenever a user joins a session, the SMS responds by providing the suitable User Session Interface of each user. Moreover, its duty is to start and stop the different collaborative components used during a session; this task is done through the Application Execution Environment.

The Resource Repository Manager (RRM) is in charge to provide access to resources information for service access as well as management of resources. These resources or part of them can be located in the Collaborator system, therefore RRM will be in charge to store such information; others can be located at other places (they are defined from URI) in this case RRM need to be able to retrieve such information. This is done in a transparent way, therefore other Collaborator system components will not need to know about location of the information.

The Application Execution Environment (AEE) provides service adaptation as well as mechanisms enabling the execution and an easy integration of applications in the Collaborator system. When a user requests the starting of a collaborative component, the AEE creates a Component Manager whose purpose is to start and control such a collaborative component. In particular, a Component Manager represents the delegate of a shared application for the system to provide a consistent view of the workspace adapted to the capabilities of each user device.

Two different kinds of agents: Personal Agents (PA)s and the Session Manager Agents (SMA)s help users in negotiation and notification activities.

PAs assist to generic users and coordinators in their activities within Collaborator and will be the interface between the user and the agents realizing some Collaborator services. The aim of PAs in Collaborator is mainly facilitating the work of their users avoiding, as much as possible, their direct participation. PAs substitute users in process as negotiation or decision making, which can be automated using the knowledge about user behaviour that personal agents have got (from the programmer or acquired by learning). In particular, PAs are involved in the meeting scheduler process with the goal to determine, for each meeting request, a meeting date that most of intended participants will effectively participate. In this task, PAs facilitates the meeting scheduling substituting, when possible, users in the automatic process of negotiation of dates for the meetings.

Moreover, a PA has the duty of notify events to its user when the user is not connected to the system. This task is done by the PA thank to its possibility of sending messages through different means (email, SMS and voice phone messages) and its knowledge about user's preferences and agenda.

A SMA is an intelligent agent, associated with each collaborative session, that facilitate some of the activities of a coordinator along the life cycle of a particular collaborative work, from the phase of meeting preparation until all the activities finish. Moreover, a SMA allows the access to the non agentized components of the Collaborator framework to the agents realizing some Collaborator services.

During the meeting, the SMA interacts with the personal agents of participants when some conflict arises and coordinator asks for its mediation. When the meeting finishes, the SMA assists to coordinator in the generation of final report.

3.2 Services

The services offered by Collaborator system have been defined to provide support to the following collaborative activities:

Virtual Meeting. It's a collaborative environment providing a shared space where mobile end-users, after becoming participants in a given "virtual" meeting, can communicate application-specific data to other participants in the same meeting until a goal is achieved.

Group Presentation. Group Presentations are multicast, enabling passive viewing from offices or rooms. This increases awareness and interaction among geographically distributed participants, and allows remote viewers to interact with each other and the speaker. Group presentations make use of real-time audio and video for both

formal and informal interactions and they may be enhanced by adding web navigation as well.

Group Calendar. Group Calendar enables information to be centrally posted by authorised users and accessed by the general public. Also, calendars can be used for private purposes requiring user log in. Users browse, filter and search for events or bookings that meet their needs, then display the details with a simple click, following hypertext links embedded in the details, or displaying a multimedia insert such as a video clip of a course session. Changes, additions or cancellation of events cause email notification to all interested parties.

Application sharing. Application sharing is based on Virtual Network Computing (Richardson, 1998) technologies and allows both the sharing of Java and Windows "off the shelf" applications (Bergenti et al., 2002). Users interact with the shared applications through their terminals. The client application that runs on each terminal (i.e., applets) receives the graphic output of the shared applications and captures all the inputs, i.e., mouse and keyboard, events. Such events are then redirected to the server through dedicated channels. Once at the server, input events are regenerated locally and passed to the shared applications. Only the events arriving from the user, that currently holds the modification floor, are passed to the shared applications, while the others are ignored. The result of this mechanism is that only the user holding the floor is allowed to work actively on the shared workspace. Whenever a new member joins the running session, the server synchronizes the state of the session asking to all shared applications to completely redraw their user interface.

Chat service. The chat service is based on the broadcasting of the chat messages through the pushlets engine software (Pushlet, 2000) which encodes the messages in events received by Web clients that registered their interest to some of such events and managed through java (applets) or javascript code. Moreover, the synchronization of new comers also becomes through pushlet events; in this case, when a new comer opens her/his chat connection, the pushlet engine broadcast a chat history event captured by clients registered to such an event (i.e., new comers clients opening the chat connection).

Document management service. The Document Management allows the uploading and use of shared documents. The document management service is responsible for: who is uploading a document, who is using/reusing the information, who is updating the information and when. The Document Management service works with several document formats (.doc, .pdf, etc….). It is adaptable to different terminal capabilities, particularly it has to take care of handheld devices, with small screens.

Audio and Video conferencing service The audio and video conference service allow a more real interaction among users, however, it requires more bandwidth and more computational power that sometimes cannot be available. Moreover, in some kind of collaborative work may be not necessary and in some cases (e.g., users of different countries with different language and no strong knowledge of a common language) the use of chat interaction may be better. The audio and video conference service is based off the self components supporting H323 and SIP protocols.

Agenda Management Service. The Agenda Management service is offered by the Agent Subsystem and its objective is to facilitate the management (create, configure,

consult, edit, save and destroy) of the different appointments that a Personal User Agent has.

Meeting Scheduler Service. The meeting scheduler service is offered by the Agent Subsystem and its objective is to find in a negotiated way the hour and date that allows the maximum possible number of users to attend a certain collaborative session considering the restrictions that each user can have in his own agenda.

3.3 Portal

Information is provided to the users thank to the use of Jetspeed portal software (Jetspeed, 1999). Jetspeed is an open source implementation of an enterprise Information portal, using Java and XML. A portal makes network resources (applications, databases and so forth) available to end-users.

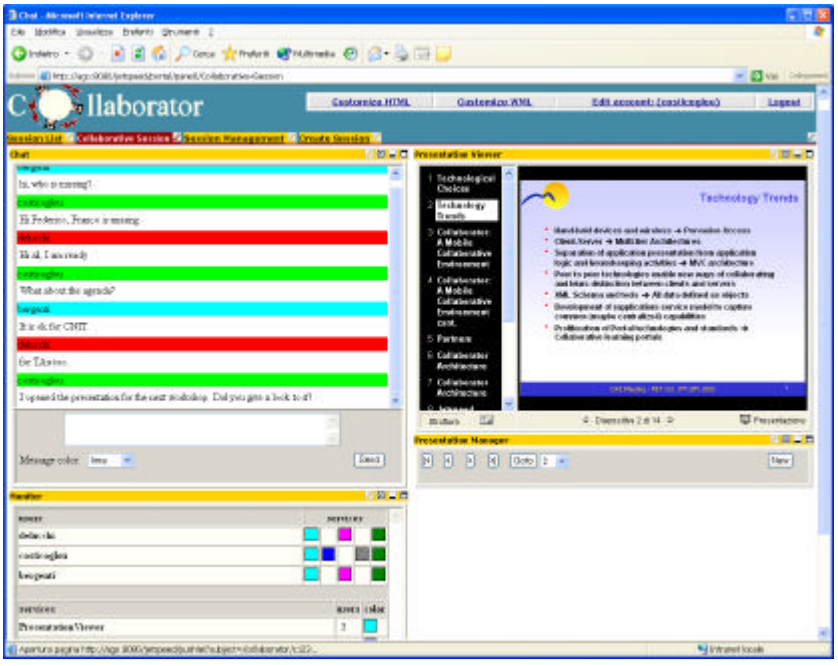

Fig. 2. Portal view of a collaborative session.

The user can access the portal via a web browser, WAP-phone, pager or any other device. Jetspeed acts as the central hub where information from multiple sources is made available in an easy to use manner. Moreover, Jetspeed provides portal view customization and access to information on the basis of user capabilities. Therefore, each user can access to a subset of the information / applications available and select the part which she/he is interested in.

Information about your interests are maintained in her/his user profile that will be used to build her/his portal view when she/he will go in again.

Therefore, the use of Jetspeed is fundamental for the Collaborator system, because it allows an easy implementation of the software components providing:

- support to different kinds of devices (PC, PDA, smart telephone, etc.);
- different portal views to provide both multiple sessions and user personalization;
- presentation of the interface of different applications in the same portal view.

CONCLUSIONS

This paper has presented, Collaborator, a system to provide a shared workspace supporting the activities of virtual teams. A first implementation of the system has been realized supporting the interaction among users with desktops and laptops on wired and IEEE 802.11 wireless networks. The work in realizing the second (and final) system implementation is already started, and has the goal of extending the first system providing access via wireless WANs (GPRS and UMTS) and providing collaboration through PDAs and smart phones.

At the end of the development phase, Collaborator will be experimented in a real working environment, using both static and mobile devices. In fact, there will be two separate field trials. In the case of the first field trial, the working domain is the construction industry and the scenario for the trial is the realisation of a construction project. The working domain of the second field trial is the telecommunication industry and is represented by a telecommunication company with field technicians performing daily activities as network maintenance and equipment installation both in inbound and outbound environments.

The most innovative aspects of the Collaborator system are:

Collaborator is the next step in the direction of the mobile office. Mobile office is not only offering to users the possibility of taking their desktops with them while on the move, but it is also giving users the possibility of taking their colleagues with them while on the move.

Collaborator can accommodate seamlessly wired and wireless terminals and it can adapt the users' experiences to the capabilities of the terminal and to the QoS of the network.

Collaborator can adapt to the variety of different terminals (PDAs, tablet PCs, laptop PCs, desktop PCs) and networks that users may choose to take part of a collaborative session.

Collaborator moves portals to active portals. Collaborator enhances the basic portal technology available today with the integration of the push paradigm: users are supplied with information as soon as it is available.

Collaborator is both a platform and a system. Collaborator is a platform because it can be used as a framework to integrate third-party software and to turn almost any piece of software into a group oriented system. Collaborator is a system because the rich set of basic services is sufficient in many circumstances to deal with the needs of the collaborative session.

Collaborator is based on off-the-shelf (Jetspeed), standard-based (Portlet) portal technologies .Personalization is one of the key features of Collaborator, and it exploits off-the-shelf, standard-based technologies that was meant solely for the purpose of making web applications personalizable.

Collaborator is based on off-the-shelf, standard-based server-side technologies. Collaborator exploits the Java 2 Enterprise Architecture and a number of off-the-shelf technologies (e.g., JBoss, mySQL, Tomcat and Jetspeed) related to it to implement a high-performance, robust, scalable infrastructure for supporting large collaborative sessions.

ACKNOWLEDGEMENTS

We like to thank all the partners of the Collaborator project. This work is partially supported by the European Commission through the contract IST-2000-30045, COLLABORATOR - Collaborative framework for Remote and Mobile Users.

REFERENCES

J. Begole, C.A. Struble and C.A. Shaffer. Collaboration transparency in Java through event broadcasting. In Proceeding of the 1997 Symposium on User Interface Software and Technology, pages 55–64, 1997.

F. Bellifemine, A. Poggi, G. Rimassa. Developing multi agent systems with a FIPA-compliant agent framework. Software Practice & Experience, 31:103-128, 2001.

R. Bentley, W. Appelt, U. Busbach, E. Hinrichs, D. Kerr, K. Sikkel, J. Trevor and G. Woetzel, "Basic support for cooperative work on the World Wide Web", International Journal of Human Computer Studies, 1997.

F. Bergenti, A. Poggi, M. Somacher. A collaborative platform for fixed and mobile networks. Communications of ACM, 45(11):39-44, 2002.

COLLABORATOR Project Home Page. Available at http://www.ist-collaborator.net, 2001.

CVW. Collaborative Virtual Workspace Home Page. Available at http://cvw.sourceforge.net. 2000.

C. Ellis, J. Wainer,. Groupware and Computer Supported Cooperative Work. In Weiss, G. (Ed.) Multiagent Systems, MIT Press, 1999.

Foundation for Intelligent Physical Agents - FIPA Home Page. Available at http://www.fipa.org, 1997.

B. Grosz et al. Planning and acting together. AI Magazine, 20(4):23-34, 1999.

H323. H323 Home Page. Available at http://www.h323.org. 1999.

T. Hodes, R.H. Katz. A Document-based Framework for Internet Application Control. Proceedings of the Second USENIX Symposium on Internet Technologies and Systems (USITS '99), Boulder, CO, 1999.

Java Agent Development Framework - JADE Home Page. Available at http://jade.cselt.it. 1998.

JBoss. JBoss Home Page. Available at http://www.jboss.org. 1999.

Jetspeed. Jetspeed Home Page. Available at http://jakarta.apache.org/jetspeed. 1999.

MySQL. MySQL Home Page. Available at http://www.mysql.com. 2000.

Pushlets. Pushlets Home Page. Available at http://sourceforge.net/projects/pushlets. 2000.

D. Ramduny, A. Dix. Why, What, Where, When: Architectures for cooperative work on the World Wide Web. In Thimbley, H., O'Connail, B., and Thomas, P., (Eds.) Proceedings of the Conference on Human Factors in Computing Systems, pp. 283–301, 1997.

T. Richardson, Q. Stafford-Fraser, K.R. Wood, A. Hopper. Virtual Network Computing", IEEE Internet Computing, .2(1):33-38, 1998.

Tomcat. Tomcat Home Page. Available at http://jakarta.apache.org/tomcat. 1999.

Y. Ye, S. Boies, P.Y. Huanf, J.K. Tsotsos. Agent-Supported Adaptive Group Awareness: Smart Distance and WWWaware. IEEE Trans. Systems, Man and Cybernetics, 31(5):369:380, 2001.