Cooperative Administration of a Public CU-SeeMe Reflector

Hugo Fuks
e-mail: hugo@inf.puc-rio.br

Carlos Laufer
e-mail: laufer@inf.puc-rio.br

PUC-RioInf.MCC20/97 June, 1997

Abstract: This article proposes a groupware architecture for WWW applications, independent of environment or platform, that can integrate diverse services. We present a WWW site prototype for cooperative administration of a public CU-SeeMe reflector. Internet software components are put together to offer a set of services.

Keywords: CU-SeeMe, Services Integration, WWW, Videoconference, Groupware, Digital Communication Design

Resumo: Este artigo propõe uma arquitetura que integra os diversos serviços oferecidos por uma aplicação *groupware* independente de plataforma e de ambiente de programação. É apresentado um protótipo para administração cooperativa de um refletor CU-SeeMe público, onde um conjunto de serviços é oferecido utilizando-se componentes de *software* disponíveis na Internet. São levantadas características cooperativas em aplicações na WWW.

Palavras-chave: CU-SeeMe, Integração de Serviços, WWW, Videoconferência, Groupware, Design de Comunicação Digital

---

This paper was partially supported by CNPq: Hugo Fuks grant n° 524557/96-9 and Carlos Laufer grant n° 143296/96-5.

Cooperative Administration of a Public CU-SeeMe Reflector

Abstract

This article proposes a groupware architecture for WWW applications, independent of environment or platform, that can integrate diverse services. WE present a WWW site prototype for cooperative administration of a public CU-SeeMe reflector. Internet software components are put together to offer a set of services.

Keywords

CU-SeeMe, Services Integration, WWW, Videoconference, Groupware, Digital Communication Design

1. Introduction

The proliferation of personal computers together with the growing popularity of the Internet has increased communication among diverse users and has created a variety of services offered through this enormous heterogenous network of environments, systems and platforms. This use of technology is very recent and various research projects aim to find out what are the relevant aspects for groupwork using computers (Grudin 94). These aspects can be seen in the structures of communication, coordination and cooperation that the groups use to work with (Medina-Mora 92, Sproull & Kiesler 91). The services and software offered on the Internet, due to their use and diffusion, are natural candidates for components of environments for groupwork.

The variety of services offered on the browsers of the World Wide Web ranges from sending electronic mail to ordering a pizza. Each of these services has its own interface, with its own individual characteristics, making them difficult to be learnt and used.

If one were to consider a specific application that required services of a different nature, it would be desirable that the user had at his disposal an interface that could integrate those services and offer them in a simpler and more standardized way. At the moment, there are various applications on the WWW, with increasingly complex architectures, based on the client-server model.

Gellersen et al. (96) argues that, under the perspective of software engineering, the WWW is a new domain of applications, and he suggests that a new area should be created within software engineering, entitled web engineering, to deal with it. A typical web engineering problem would be, for example, maintenance difficulty, due to the reproduction of the html code of various pages, relating to the design of a given site. In the case of pages generated on the fly, Gellersen says care should be taken to check that the CGI script generates consistent information.

The majority of WWW sites use few programming resources. Even when they are used, the programming is restricted to the use of CGI script which is part of the model of the http protocol (Lee et al. 96). Applications using WWW have been growing rapidly, largely due to the simplicity of this stateless protocol (Smurlo & Madelaine 96). The majority of the current applications are of the one shot type—when a client requests a resource, it either already exists and is transferred to him, or it is dynamically generated through a CGI script.

Greater use of the WWW has allowed for an increase in the quantity and diversity of applications. The sophistication of these applications has also increased, generating a series of models that aim to create more controlled environments than the one provided by the http protocol. The introduction of Java (Campione & Walrath 96) has also made it possible for the client to operate his code, increasing the complexity of the applications.

Many studies are in the process of being undertaken regarding models that integrate services and applications. Vinoski (97) presents a model for the integration of diverse applications in distributed heterogenous environments. Gall & Hauck (96) present an environment for group communication in real time through the WWW. Patterson et al. (96) proposes a notification service for use in synchronous groupware. Fernández et al. (96) presents experience in the integration of a distributed cooperative application on the WWW. Licea & Favela (96) present an environment for the development of collaborative software based on the WWW.

The WWW with its two basic components, the web server and the web browser, created a universal user interface, independent of environment or platform. This type of interface allows concepts like operational systems, languages and others to remain abstract, being more appropriate to the world of the developers, which facilitated its acceptance by the users.

Our research aims to develop a model that integrates various services offered by a groupware application, independent of platform or programming environment (Rice at all 96).

This paper presents a prototype for distributed administration of a public CU-SeeMe reflector, where a group of services is offered using software components available on the Internet.

The document is organized in the following way: section two introduces the CU-SeeMe reflector and client; section three deals with the Rio Internet

TV reflector—the first public CU-SeeMe reflector in Brazil—and the services used in its administration; section four presents the WWW Rio Internet TV administration site and aspects of its implementation and section five describes the cooperative aspects of this work. Finally, the conclusion and future work are discussed.

2. The CU-SeeMe Reflector and Client

The appearance of CU-SeeMe software has enabled videoconferences to become more popular, previously only available to corporations that could invest large sums of money in equipment and high bandwith networks.

Figure 1 - CU-SeeMe Window

The development of CU-SeeMe began in 1992 in Cornell University's Information Technology Organization. It was originally written for Macintosh, but now also runs on PC platforms (Dorcey 95).

At the moment, White Pine is CU-SeeMe's master licensee. It developed a product called Enhanced CU-SeeMe that uses color and improved codecs for image and audio. The CU-SeeMe reflector is a software that works as an audio and video server, where all the clients connected may receive any of the other clients' video and audio. The reflector is available for different platforms.

3. Rio Internet TV Reflector - RITV

The Rio Internet TV Reflector of PUC-Rio was the first public reflector created in Brazil. It began operating in 1994 and receives hundreds of visitors every day. It has already been mentioned in many magazines and newspapers (Ediouro 96, Mandarim 96 & Informática etc 96) and has a companion WWW site (http://www.inf.pur-rio.br/~refletor) which is a major source of information for the CU-SeeMe community in Brazil.

The CU-SeeMe reflector is a server that allows various people to connect to a videoconference. It is possible to connect to the RITV and take part in a multimedia conference—video, audio and text—through a client connection—which could be White Pine Software's commercial version—Enhanced CU-SeeMe (http://www.wpine.com/), or the University of Cornell's academic version (http://cu-seeme.cornell.edu/).

The RITV possesses two videoconference suites: one for public use with a capacity of 30 participants and one for private use with a capacity of 10 participants. It runs on a Sul Ultra 1 machine with Solaris Operational System 2.5.1. This machine (DNS=refletor.inf.puc-rio.br) is part of the network of the Post-Graduate Laboratory of the Department of Computer Science at PUC-Rio.

The RITV remains on air 24 hours a day. As it is a public place with uncontrolled access, it needs to be monitored to avoid any undesirable behavior. Monitoring may include the disconnection of certain visitors or maybe activation of the RITV in case of failure due to problems on the network. These procedures demand specific knowledge, like for example: knowledge of Unix, TELNET and reflector port command language. Furthermore, access to determined procedures implies in the use of passwords, because it is necessary to access the account where the program reflector is located. We were faced with the problem of organizing a team of administrators—all absolutely trustworthy and with the necessary technical knowledge—for a 24-hour administration task.

The RITV appears on lists throughout the world, including the phone book distributed by White Pine, and receives a considerable number of visitors daily. A new medium generates unprecedent behavior and a research reflector needs constant monitoring. For that reason we projected a WWW site specifically for administration of the RITV, that could minimize the security problem and dispense with the need for the technical knowledge mentioned above. The new interface did simplify the administration, and we were therefore able to invite some of the regular visitors to take part in the team, thus creating a distributed administration of the RITV.

In the following paragraphs we explain some of the procedures necessary to administrate the RITV, and the services employed to put those procedures into effect.

Activation of the RITV

The reflector is activated through a script that is executed in a specific account on the network of the Post-Graduate Laboratory. Therefore, in order to activate the RITV it is necessary to enter the reflector

account and run the activation script. In order to enter the account one must know the account's name, login and password. For security reasons the password is kept under guard, therefore, activation of the RITV, in this way, is quite restricted.

Monitoring Participants

Once activated, the reflector offers a services port through which it is possible to make a series of commands; for example, listing the participants connected to the reflector, temporarily disconnecting a participant from the reflector, or permanently disconnecting a participant from the reflector. In order to connect to this services port one must execute a TELNET.

Once connected to the port there is a specific command language that must be used in order to obtain the services desired. For example, the **who** command lists all the participants connected to the reflector. Certain commands need a password for their execution to be accepted, like the **kill** command, that temporarily disconnects a participant. Once again, for security reasons, the password is not freely divulged.

```
[obaluae] telnet 139.82.17.17 7640
Trying 139.82.17.17...
Connected to 139.82.17.17.
Escape character is '^]'.
White Pine Reflector on obaluae.
wpreflect@obaluae>who
UC_CLIENT: [0] CRV 05:12:13  Rio TV@139.82.35.125
UC_CLIENT: [0]          00:03:25 melange@131.130.170.20
UC_CLIENT: [0] CRV 00:17:52  Joel@150.176.108.17
UC_CLIENT: [0] CRV 01:00:05 Yuji@203.180.244.147
UC_CLIENT: [0] CRV 00:47:31  LC@200.255.215.26
UC_CLIENT: [0] CRV 00:23:48  DUDA BOY@194.97.78.38
UC_CLIENT: [0] CR   00:07:25  PC@200.255.48.67
UC_PEER_CLIENT: [0] C     00:01:09  Raquel@143.106.37.97
UC_CLIENT: [0] CRV 00:15:16  Depto Tecnico@200.245.3.10
UC_PEER_CLIENT: [0] CR   00:01:20  MOR@200.245.9.27
UC_CLIENT: [0] CRV 03:42:04  Lab-DI@139.82.17.61
UC_PEER_REF: 07:01:22 Group Uni REF@200.132.0.21

Total # of clients 11  senders 7  lurkers 4
```

Figure 2 - Output of a **who** Command

4. The RITV WWW Administration Site

The objective of the RITV WWW administration site is to integrate services of a different nature that are used to control the RITV and to allow for a distributed administration. These services include activation of the RITV, monitoring the connected participants and communication between the administrators. There is also a link with the RITV WWW information site where one could find out who is taking part in the videoconference at that time. From this site it is possible to connect to the RITV videoconference suite through automatic activation of the Enhanced CU-SeeMe.

As previously mentioned, there are a variety of distinct services that are used for administration of the RITV. The administrators must have specific knowledge in order to be able to execute each one of these services with its own characteristics, however that knowledge is not necessary for the administration of the RITV in itself. Ideally, these services should be integrated into one interface that could make the services' characteristics clear to the administrators, thus only demanding knowledge of the actual interface. Furthermore, it would be desirable that the administrators could take control regardless of the platform or environment in which they work. We have named this type of application—services agglutinant—Services Manager.

The philosophy of the WWW where a web server together with web browsers make information available to clients distributed on different platforms, was used as a model for the development of a site where the administration services of the RITV are offered.

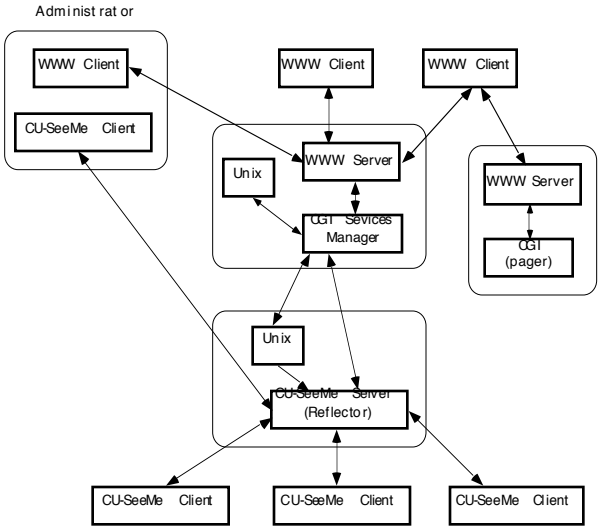

Figure 3 - Architecture of RITV Administration

In the architecture proposed in Figure 3, the Services Manager possesses three components: the web server; the web browser and a CGI script developed to activate many of the necessary services. We named this CGI script, CGI Services Manager. We decided to use a CGI script that runs all functions from a specification of the desired command.

The web server and the web browser are responsible for the exchange of information through the WWW while the CGI Services Manager is the one that effectively takes care of the administration of the RITV. The web server used was a Unix NCSA 1.4 in the www.inf.puc-rio.br machine running SunOS 4.1.4.

As well as these three modules there also exists an initial page in html that allows access to the RITV administration site.

The CGI Services Manager was programed in language C in one module only and runs six functions:
· access validation;
· RITV activation;
· listing of the participants in conference;
· execution of administrative commands;
· obtaining data to send email to administrators;
· sending mail to administrators.

Each of these functions uses basic services on the network that are clear to the administrators.

As well as these functions the RITV administration site has a link to the pager site that allows messages to be sent to the administrator on duty.

In the following paragraphs we will present the RITV administration site interface and some aspects of its implementation.

Access to the RITV Administration Site

The RITV WWW site can be accessed through a page developed in html, which is the only page on the site that is not generated on the fly. This page asks the user to type in a password that allows the user to enter in the condition of administrator of the RITV.

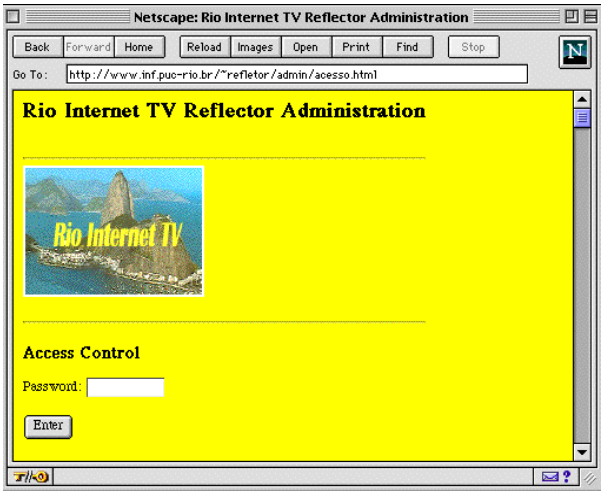

Figure 4 - Access Page to the Administration of RITV

The CGI Services Manager tests if the password supplied by the user is valid, and if so shows the screen that monitors participants as an exit (see Monitoring Participants below).

If the RITV is not active when the administrator accesses the site, an activation screen is shown so that the administrator can activate the RITV.

The CGI Services Manager runs a TELNET in the refletor.inf.puc-rio.br machine where the reflector operates. As soon as the connection is established a login is triggered in the RITV account and the account name and password are entered. Once the login has been accepted the Unix script activates the RITV. Then the screen that monitors participants is shown to the administrator.

Monitoring Participants

If the RITV is activated after the entrance of the administrator has been validated, a page is shown listing the users connected to the RITV at that time. This page also gives the administrator the possibility to make commands that affect the participants taking part in the conference on the RITV at that moment. The administrator also has the ability to disactivate the RITV.

The CGI Services Manager runs a TELNET to the RITV services port and executes a **who** command that returns the participants connected to the RITV at that moment.

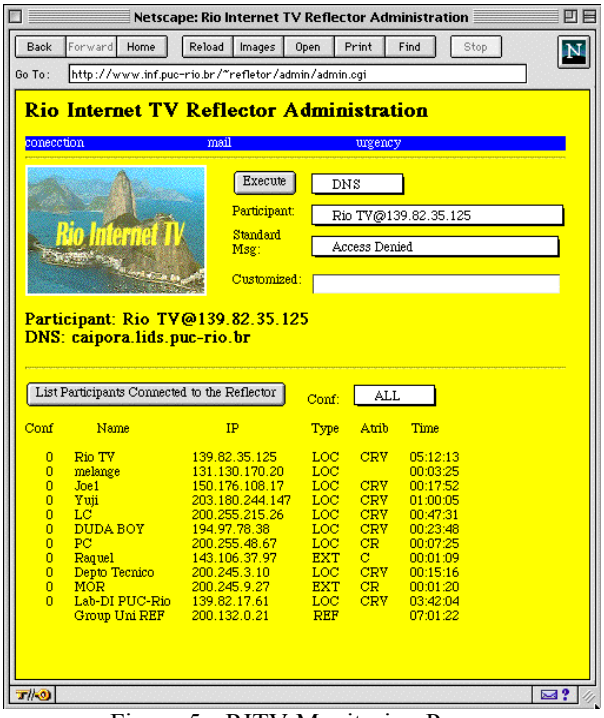

Figure 5 - RITV Monitoring Page

This information is shown on the screen together with a html form that allows the administrator to request execution of one of the following commands:
· information about a participant's DNS (**DNS** command);
· temporary disconnection of a participant (**Kill** command);
· permanent disconnection of a participant (**Deny** command);

· termination of the RITV (**Term** command);
· exhibition of the administrators command log (**Log** command).

The **Kill**, **Deny** and **Term** commands are executed by the CGI Services Manager through a TELNET to the RITV services port. Before these three commands can be accepted the RITV protection password must be entered.

The **DNS** command is executed by the CGI Services Manager through a UNIX **gethostbyaddr** command where one can try to obtain a participant's DNS from an IP address.

The emission of the **Kill**, **Deny** and **Term** commands is registered in a log archive where the following information is stored: the IP address of the administrator who made the command, the command itself, and the date and time the command was made. The log archive also registers errors made when entering the password to access the site, as well as any possible activation or disactivation of the RITV. The **Log** command shows the information recorded in the log archive on the screen,

It is fitting to point out here one criticism to our site's interface. Despite being semantically different, the five commands previously mentioned are all activated from one same command button. The **DNS**, **Kill** and **Deny** commands monitor the participants. The **Term** command controls the RITV, while the **Log** command helps the administration. As there was a certain urgency in the development of the site, we opted for a more simple interface—grouping the five commands into one same command button.

5. Cooperation between Administrators

As there are various administrators, there is a necessity for communication between them, in order to, for example, report situations of use of the reflector and to inform on any new procedures. One of the most common forms of communication through the Internet is electronic mail. Email can be sent in many ways: two examples are the Unix application or a program like the Eudora. In order to use either of these applications, one must know their particular commands, which are both quite distinct.

Another popular form of communication, in cases of emergency, is the use of a pager. It is possible to send messages to a pager through the telephone or through a site on the Internet.

The RITV WWW administration site allows mail to be sent to the administrators as well as messages being sent by pager to the administrator on duty, in cases of emergency. Thus, the administrators can make reports of situations that happened during their periods of control and such information can help the general administration of the RITV.

Figure 6 - Log Page of the Administration Commands

Regarding implementation, when an administrator asks to send an item of mail the CGI Services Manager generates a html form that can be filled in with information which will be sent through email to the other administrators.

Another form of increasing cooperation between the administrators and improving administration of the RITV is to create a log of commands executed by the administrators. In this way it is possible to analyse the performance of each of the administrators, as well as checking information about undesirable participants.

Sometimes, people who connect to the RITV write to us complaining about certain behavior taken by the invited administrators. In these situations, the log can be a valuable source for resolving conflicts. If one consults the log together with a recorded chat window session, it is possible to reconstruct the conversation exchanged between the offended party and the administrator, and either confirm the administrator's attitude or warn him about his prepotence.

The existence of conflicts between participants and administrators created the necessity for a standard procedure for dealing with participants who broke the rules of etiquette of the RITV. Some participants opposed the rules, arguing that they were being denied freedom of expression. After discussing the matter with the administrators and studying the reports of specific cases, a procedure was defined that reduced the conflicts during conferences and established that

complaints be made through email to us at this university, taking the responsibility away from the invited administrators.

In the case of our work, CU-SeeMe serves as a tool of cooperation between our administrators. Through this program, it is possible to exchange private messages between participants, which allows administrators to conduct private conversations, when two or more are connected at the same time. Another form of establishing a private conversation is to use a special suite protected by a password, that can only be accessed by the administrators.

Administrators' Feedback

After using the RITV administration site for about three months we asked the invited administrators to give us feedback of their impressions. In their reports we found criticisms to the interface, suggestions of new features, information about the participants behaviour, etc.

The administrators told us they had some difficulty in imposing their authority over the regular participants.In some situations when they used the private chat to advise participants they did not obtain positive results. The participants did not believe they were the administrators and this fact prompted some heated discussions. Initially some administrators questioned the validity of identifying themselves as administrators to the videoconference participants. Later they concluded that that was the best procedure, because the regular participants themselves started telling the new participants about the administrators' identity, making it easier for their authority over the reflector to be accepted.

The constant presence of an administrator in the reflector, helped to establish the RITV's participants' profile. Those that broke the RITV rules of etiquette were disconnected with the other participants' agreement.

A low level of interaction between administrators was also observed, which created repetitive situations regarding unwanted participants. More interaction could produce a homogeneous administrative posture towards dealing with specific problems. One possible solution could be regular meetings in the administrators' conference suite. However, a simultaneous meeting of all the administrators is an impossible task, giving that they live in different time zones. The solution was the creation of an asynchronous way of discussion, in the form of a discussion list named ritv-adm.

6. Conclusion and Future Work

Digital communication design studies the migration of traditional communication media to digital technology, aiming to develop digital groupwork environments. In our research we aim to develop a model that integrates the diverse services offered by a groupware application, that can be run on any platform or in any environment, regardless of the program. This study of distributed administration of a reflector also aims to highlight cooperative characteristics in applications on the WWW and to give information regarding the way the model is implemented in terms of integration of components—tools—existing on the market.

Version 2.1 of the White Pine reflector offers a software that allows monitoring the reflector; called: Reflector Manager. This program presents a more elaborate interface than our WWW site, with more possibilities for configuration than the RITV. However, it is dependent on the machine code, in this case, PC platforms. One of our premises in the construction of our experiment was the independence of platform and environment.

For greater protection regarding undesired visitors, it is possible to configure not only the activation password, but also the IP address of the machines from where access to the command port is permitted. In the RITV it is only necessary to authorise access to the machine where the CGI Services Manager is located. In the case of the Reflector Manager all the machines from where one wants to manage the reflector must be authorised.

As an extension of this work we intend to introduce the connection of the CGI Services Manager with a database. Currently all information is stored in archives. Working with a database we will be able to make more in-depth inquiries, like searching for some particular information; at the moment we can only read the administration log.

The part relating to awareness in this paper was actually obtained through CU-SeeMe. However, it is not possible to know if more than one administrator is accessing the WWW administration site at the same time. The pages loaded on the web browser are passive, in other words, they do not establish communication with any other process, other than triggering the CGI Services Manager to request a service. The introduction of Java applets will allow the web browser to have a more active posture. A more complex extension could be thought of in terms of creating a server that could run in the web server machine and communicate with the web browser applets.

It is our intention to make the architecture of the Services Manager more sophisticated, using object oriented technology, in such a way that its modules could be defined as components of an application permitting an easier form of adaptation and extension.

7. Acknowledgements

This paper was partially supported by CNPq: Hugo Fuks grant n° 524557/96-9 and Carlos Laufer grant n° 143296/96-5.

We would like to thank the invited reflector administrators.

The paper was translated by Julann Smyth.

8. References

Campione, M. & Walrath, K.; The Java Tutorial: Object-Oriented Programming for the Internet; The Java Series, Addison-Wesley, 1996 <http://www.javasoft.com:80/nav/read/Tutorial/index.html>

Dorcey, T.; CU-SeeMe Desktop Videoconferencing Software; CONNEXIONS, pp 42-45, March 1995

Ediouro; Vejo você, você me vê - Um tutorial completo do CU-SeeMe; Guia da Internet.br, Ano I, Nº 3, Ediouro S.A, June 1996 (Portuguese)

Fernández, D., Pastor, E. & Bellido K.; Experiences Integrating Cooperative Multimedia Applications into WWW - Proceedings of CRIWG'96, pp 43-59, Puerto Varas, Chile, September 1996

Gall, U. & Hauck, F. J. ; Promondia: A Java-Based Framework for Real-Time Group Communication in the Web; Hyperproceedings of Sixth International World Wide Web Conference, Santa Clara, California, April 1997 <http://www6.nttlabs.com/HyperNews/get/PAPER100.html>

Gellersen, H., Wicke, R. & Gaedke, M.; WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle; Hyperproceedings of Sixth International World Wide Web Conference, Santa Clara, California, April 1997 <http://www6.nttlabs.com/HyperNews/get/PAPER232.html>

Grudin, J.; Computer Supported Collaborative Work: History and Focus, IEEE Computer Vol 27 No.5, pp 19-26, May 1994

Informática *etc* ; Dá uma passada no meu monitor qualquer hora; O GLOBO 23 September 1996 (Portuguese)

Lee,T. B., Fielding ,R. & Frystyk, H.; Hypertext Transfer Protocol–HTTP/1.0–Internet Request For Comments rfc1945; May 1996 <http://www.cis.ohio-state.edu/htbin/rfc/rfc1945.html>

Licea, G. & Favela, J.; ICARO: A Web-Based Environment for Collaborative Software Development - Proceedings of CRIWG'96, pp 23-30, Puerto Varas, Chile, September 1996

Mandarim 96; Janelas Indiscretas - BBS Magazine Ano 2 Nº 13 Editora Mandarim August 1996 (Portuguese)

Medina-Mora, R., Winograd T., Flores, R. & Fernando Flores, F.; The ActionWorkflow approach to workflow management technology; Proceedings of the Conference On Computer-Supported Coooperative Work, Toronto, November,1992.

Palfreyman, K. & Rodden, T.; A Protocol for User Awareness on The World Wide Web; Proceedings of ACM 1996 Conference on Computer Supported Cooperative Work, pp 130-139, Boston, November 1996

Patterson, J. F., Day, M. & Kucan, J.; Notification Servers for Synchronous Groupware; Proceedings of ACM 1996 Conference on Computer Supported Cooperative Work, pp 122-129, Boston, November 1996

Rettinger, L; Desktop Videoconferencing: Technology And Use For Remote Seminar Delivery; M. Sc. Thesis in Computer Engineering, North Carolina State University, Raleigh, North Carolina July 1995

Rice, J., Farquhar, A., Piernot, P. & Gruber, T.; Using the Web Instead of a Window System; Proceedings of Computer-Human Interface, pp 103-110, April 1996

Sproull, L. & Kiesler, S.; Connections; The MIT Press, 1991

Szmurlo, M. & Madelaine, J.; A Network of Asynchronous Micro-Server as a Framework for Server Development; Hyperproceedings of Sixth International World Wide Web Conference, Santa Clara, California, April 1997 <http://www6.nttlabs.com/HyperNews/get/PAPER106.html>

Vinoski, S.; CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments; IEEE Communications Magazine, Vol 14, No 2, February 1997