COOPERATIVE INFORMATION SYSTEMS OVER THE INTERNET AND THE WWW

Christos Bouras[1,2], Panagiotis Destounis[1,2],
John Garofalakis[1,2], Giannis Tzimas[1,2],
Vassilis Triantafillou[1,2], Panagiotis Zarafidis[2]
[1]Computer Technology Institute,
P.O. Box 1122, 26110 Patras,Greece
[2]Department of Computer Engineering and Informatics,
University of Patras, 26500 Patras, Greece
E-mail: bouras@cti.gr

KEYWORDS

Intranets, Extranets, CSCW, Shared Workspace, Dynamic Web Pages

ABSTRACT

In this paper we present a WWW-based approach for the development of a system supporting information sharing within an organizational environment. The system developed is exploiting Intranet and Extranet technologies enabling asynchronous and synchronous cooperation, manipulation of information from heterogeneous sources, security and easy administration, providing in parallel advanced communication facilities. We propose an innovative approach for the development of shared workspaces over WWW, based on state-of-the-art software platforms, integrating WWW and database technologies. The whole system was developed under a project funded by the Greek General Secretariat of Research and Technology and aims in the efficient distribution and management of information between and within different types of local authorities (Municipalities, Prefectures).

INTRODUCTION

Nowadays the need for efficient, quick and error-free information exchange within an organization is recognized as a necessity for competitiviness and continuous adaptation to the quickly changing business environment. Data must be distributed among several departments of an enterprice or between different companies and sites. In the past, heterogeneous networks, data formats or communication channels, mostly private, were developed and used in order to solve the communication requirements of data sharing and provision of status information.

The development of a system that provides an environment for efficient data exchange and retrieval, and offers real-time information when needed, can make an organization more efficient, competitive and effective.

Internet and especially WWW offer an attractive platform for developing collaborative applications (groupware). Among the primary advantages of the WWW are (Chiu and Griffin 1995):

- the wide availability of web browsers on a large number of platforms
- the core technology being based on a set of widely accepted standards such as HTML, MIME types and Internet naming
- it is the relatively inexpensive
- friendly coverage and extensibility of the resource naming conventions (URL)
- extensibility via CGI interface
- user friendly interface
- a wide variety of public domain software exists

On the other hand, the WWW technology is insufficient in several areas, most notably (Chiu and Griffin 1995):

- Primarily oriented for information distribution. Since the server supports mostly read access, there is no concept of write access control (for object creation, modification, deletion)
- Stateless server/client architecture, meaning that the standard version of the protocol does not support methods for a client to keep state information associated with a sequence of client server interactions. This forces some applications to “store” the state information in URLs and as a result they are less portable.
- Protocol (HTTP) insufficient for synchronous communication (bandwidth, response time, etc)
- No client/client communication
- The standard http server’s information repository is based on the file system, whereas groupware typically require a database as the repository for potentially a large set of information objects
- The browsers are usually single-window-based, and lack in capabilities for customizing toolbar and other graphical user interface support often found with contemporary desktops applications (drag-and-drop etc)

Despite these limitations, WWW technology has the potential to become the primary infrastructure for network computing. This fact is advocated by the plethora of several tools, commercial or not, implementing various features needed by an environment for collaboration. Well-known tools are Microsoft Netmeeting, Netscape Conference, Lotus Notes, Novell Groupwise etc. However, the tools mentioned above do not sufficiently combine the centralized control, the wide variety of files in exchange, the broad use of the tool over any operating system and the simplified access to the application through a web browser.

In this paper, we present a Web-based, database-oriented CSCW system implemented around the 'shared workspace' notion. The workspace comprises asynchronous and synchronous cooperation and communication tools. The use of these tools is restricted (password controlled) and can be accessed via standard, unmodified web-clients using the login/password authentication scheme. After a successful log-on, only the shared workspaces that the user has access privileges are presented to him. Within the boundaries of a workspace the user can manage and share with other members of the workspace different kinds of information, including:

- A folder-file repository, in which the users may transfer various types of files from their local store into a shared repository (uploading) or retrieve files from the workspace to their local system (downloading). The repository is organized in a folder structure, a notion that all users are familiar with. The repository is equipped with basic functions of delete, search, annotation, versioning and MIME recognition.
- A discussion forum (bulletin board). A shared location for posting announcements supporting follow-up messages (that is, replies to existing messages).
- A real-time Chat session. A synchronous communication, Java-based, client/server application integrated in the web-browser. The members of the workspaces can communicate in real time via a chat session belonging exclusively to the workspace, a session similar to the well-known IRC chat sessions.

This paper is organized as follows: in section 2 there is a short description of the project context's. The third section provides details of the system architecture, specifically presenting analysis, design and implementation issues. In section 4 the functionality offered by the system are outlined. The last section summarizes conclusions elicited from our work and presents future work issues.

DESCRIPTION OF THE PROJECT

The whole system was developed under a project funded by the Greek GSRT. The main concept of the project is the design and implementation of an Intranet and Internet based Information System for Administration and Information purposes. The system developed deals with the distribution of information and the exchange of documents within, from and towards (region, municipalities, citizens, various groups etc.) the prefecture of Corfu.

The main objectives of the project are:

- The efficient distribution and management of information and documents from the Prefecture to the municipalities and vice versa.
- The minimization of the number of citizens' visits to the administration centers by decentralizing / increasing the accessibility to the information, which will result not only in the reduction of the workload of the employees but also in the potential user being able to reach the information needed whenever one wishes and without delay.

In brief, the services provided by the System are:

- Exchange of Information between the authorities of the Prefecture, Municipalities and other organizations through Web servers.
- Co-operation among the members of Working Groups (e.g. Committees) and electronic exchange of documents through shared workspaces.
- Newsgroups.
- Videoconference.
- On-line chat or bulletin board.
- E-mail for the employees of the Prefecture of Corfu.
- Access to the Internet, and specifically WWW, to support information retrieval, access to information sources by citizens and capability of communication between the citizens and the authorities of the Prefecture.
- Adaptation of existent Databases in the Intranet.
- Web pages for the Prefecture and the Municipalities.

SYSTEM ARCHITECTURE

System Analysis

Our architectural and design decisions taken conserning the structure, the functionality and the development of the shared workspaces were been mainly dictated by the analysis of the user needs. The analysis showed that in order for the system to satisfy the demands and needs of the potential users the following functional requirements must be implemented taken into account.

- Closed groups of employees need to use shared workspaces through the Intranet where they can exchange documents, notes etc. while the members of these groups will be able to communicate with each other (point to point or multipoint to multipoint connections) through videoconference tools (NetMeeting), on-line chat or bulletin board. These types of communication will also be accessible by users outside the limits of the workgroup.
- Information retrieval (announces, proceedings) must be based on alphanumeric keys of several types (e.g. creator, date, title, description).
- Efficient and complete administration of the system that includes the creation, deletion and modification of shared workspaces and user information.

Additionally, there are some non-functional requirements that should be met in order to ensure the quality of the system regarding performance and security issues.

System Design

The architecture/design of the system follows the strong movement towards three-tiered, server-centric architectures. A three tiered architecture enables the presentation, business logic, and data elements of applications to be cleanly separated and run on different machines connected by a network. What makes this architecture so important in Web environments is that applications can have browser-based user interfaces that access data components that reside safely behind corporate firewalls. Generally, a move towards three-tiered approaches for Web-centric applications stems from the realization of broader benefits such as enhanced scalability, security, and code reusability.

In the first level of our architecture, there is the Internet Browser through which the user gains access to the system. The second level consists of the Web Server that serves as a Web Glue between the other two levels. Particularly, the web server through ASP and Perl scripts provides both the interface with the user, specifying the way the data are to be displayed in the browser, and the connection with the actual data which are stored in the database and the file system. Finally, the third level includes the server file system where all the documents are stored and the database server where all the information regarding the documents is stored.

The types of information stored into the system can be categorized according to its storage location, database or file system.

The Database contains all the necessary information about the members of the shared workspaces and the data exchanged between them. This information is distributed in the tables described below:

Services: Contains information regarding the services of the prefecture.

Departments: Holds information on the services' departments of the Prefecture.

Employees: Information about the employees of the prefecture and in general the users of the system.

Files: Information regarding the files uploaded by the users in the shared workspaces.

Folders: Information about the folders created by the users in the shared workspaces.

Parent_Child: Contains the information for the folder hierarchy and in particular the association of each folder with its sub-folders.

Workspaces: Information regarding the shared workspaces of the system.

Workspaces_Employees: The relationships between employees and shared workspaces of which they are members.

Announcements: In this table there is information on the multimedia announcements created and uploaded by the users.

All the files and folders uploaded by the users are stored in the file system. It should be noted however that only the file or the folder is actually stored in the file sytem, all other relative information, including its name, is stored in the database. Multimedia Announcements are files, of special format, uploaded by the users and are stored in the file system.

Separate from this Database/File System scheme is the Bulletin Board. All messages posted by the users are directly stored in separate files in a special folder in the workspace which is not part of the directory structure where the file exchange takes place.

Implementation Issues

General

The web-clients send their requests and receive the answers via the hypertext transfer protocol (HTTP). The majority of the HTML document responses are produced 'on the fly'. Since the contents of a shared workspace and especially a file repository continuously change, e.g. the contents of a folder, the HTML page has to be generated from the second level of the architecture every time the user makes the request. Thus, the use of CGI and web server scripting is essential in building such systems. Executing scripts and components on the server, without the performance limitations and development difficulties of CGI, allows developers to easily create Web-based applications and dynamic content. Currently, the two major vendors, Netscape and Microsoft, in the web-server industry offer web server scripting in their products. For the development of this system the Microsoft Internet Information Server v4 for Windows NT 4 was used.

But why is web server scripting so essential? First of all, Web server scripting enables the "backend" for Web applications. HTML and HTTP do not by themselves provide a way to access databases or carry information about users from page to page. Server-side scripting accesses programs on the server that provide this necessary functionality behind the scenes to deliver Web applications and customized HTML for each user. Second, web server scripting separates the content from the presentation for easier design and data management. Server scripting allows the use of templates for creating HTML documents on the fly. The contents of a page can come from anywhere –databases, plain text files, searches, calculations– and be dynamically inserted before it is sent to the user. Information can be managed in the most appropriate manner, and does not have to be stored in HTML pages that must be changed by hand whenever the data changes.

Active Server Pages (ASP) is the technology that enables the use of web server scripting with the MS IIS 4 in virtually any language. ASP pages are files that contain HTML tags, text and script commands. ASP pages can call ActiveX components to perform tasks, such as connecting to a database or performing a business calculation.

Besides the ASP pages in the second level there are also some conventional CGI scripts written in Perl. These, as far as the file repository is concerned, execute only the uploading and downloading while they comprise the core module of the discussion forum.

The ASP pages and the Perl scripts are responsible for the manipulation of the information stored either in the file system or the database, the third layer of our architecture. For the ASP pages the connection to the database is provided from the ActiveX Data Objects (ADO) through the OLE DB and ODBC. The Perl scripts are connected through the ODBC extension for Win32 Perl.

The second level, namely the Web Server, except the execution of the scripts, is also responsible to enforce the authentication and security. Authentication involves prompting users for unique user name and password information, which must correspond to a valid Windows NT user account, governed by Windows NT File System (NTFS) file and directory permissions that define the account's level of access.

The MS IIS supports Basic and Windows NT Challenge/Response authentication. However, only the MS Internet Explorer supports Windows NT Challenge/Response, so, in order our system to be compatible with as many browsers as possible the Basic authentication was selected. Naturally, when a user wants enhanced security he can use the SSL.

Basic concepts

Two basic concepts involved in the implementation of the system are : the *application* and the *session*.

An *application* is any file that is executed within a defined set of directories in the Web site. When an application is created, Internet Service Manager is used to designate the application's *starting-point directory* in the Web site. Every file and directory under the starting-point directory in the Web site is considered as a part of the application until another starting-point directory is found. Thus directory boundaries are used to define the scope of an application.

One can isolate an application so that it runs in a separate memory space from the Web server. Isolating an application ensures that other applications, and the Web server itself, keep running even if the isolated application stops responding or fails.

A *session* is the equivalent of a visit from the moment the user initially connects to the system and until the user terminates the connection. The first time a user requests an .asp file within a given application, ASP generates a *SessionID*. A number produced by a complex algorithm, the SessionID uniquely identifies each user's session. At the beginning of a new session, the server stores the Session ID in the user's Web browser as a cookie.

The SessionID cookie is similar to a locker key in that, as the user interacts with an application during a session, ASP can store information for the user in a "locker" on the server. The user's SessionID cookie, transmitted in the HTTP request header, enables access to this information in the way that a locker key enables access to a locker's contents. Each time that ASP receives a request for a page, it checks the HTTP request header for a SessionID cookie.

After storing the SessionID cookie in the user's browser, ASP reuses the same cookie to track the session, even if the user requests another .asp file, or requests an .asp file running in other application. Likewise, if the user deliberately abandons or lets the session timeout, and then proceeds to request another .asp file, ASP begins a new session using the same cookie.

Our system has been defined as an application in the Microsoft Management Console of the IIS 4.0. In order to enter the CSCW system the user must provide a valid, unique login name and a password in the dialogue box that the browser displays when the user requests a URL that has prefix http://servername/intranet. This URL is the starting point of our intranet application. The moment the user has successfully logged on the system, the G*lobal.asa* file located in the application's root directory is executed. The Global.asa file is an optional file in which event scripts are specified and objects are declared that have session or application scope. It is not a content file displayed to the users; instead it stores event information and objects used globally by the application. An application can only have one Global.asa file. In our case the global.asa file contains two functions: Application_OnStart, Session_OnStart. The first one is executed only at the beginning of the application and only once when the first user enters the application. It contains declarations of objects with application scope e.g. application path, absolute paths used to make the system easily portable. The second one is executed each time any user enters the system and contains user specific session information such as maximum idle time, the Data Source Name (DSN) used to locate and identify the particular ODBC compliant database, which in our case is SQL Server. The DSN is used by every ASP page for the connection to the database.

The first page is default.asp located at the application's root. Using the environment variable that contains the user name that has just logged in, the database is queried and the information about the user is returned. This information is stored in the session object, the aforementioned user's locker so as to be available from all the ASP page that the user will request throughout his visit.

Shared Workspaces

Only the shared workspaces to which the user has access are displayed in the browser. This information has been stored into the database by the administrator. When a user selects a workspace the workspace.asp is executed. The user sees the contents of the workspace root folder and all available functions associated with the workspace.

The contents of each folder that the user accesses are generated "on the fly" by queries made in the database requesting the sub-folders and files of a particular folder. The database maintains a tree structure that reflects the actual tree structure on the file system. Instead of their real name, the files or folders are stored in the file system named after their IDs which are contained in the respective tables of the database. This way we avoid the limitations and problems imposed by the naming conventions of the file system.

For a specific folder the user can:

Create an new sub-folder: When the user completes and submits the new folder form a new database entry is created in the Folders and Parent_Child table and the ID, the primary key of the Folders table, of the new folder is used to name the directory created in the file system.

Delete a folder: When a folder is deleted all the relevant database entries, all sub-folders and their files, are removed from the appropriate tables and the file system directory is erased from the file system.

Modify folder info: The user can see all the available information about the folder and only the creator can submit a new name or a new description of the folder. In that case only the appropriate database entries need to be updated.

Upload file: The user selects a file from his local file system and submits the new file form. The action of this form is a Perl script that performs the uploading of the file by inserting a new entry in the Files table of the database and creating the new file in the file system. The Perl script also recognizes the MIME type of the file by examining its filename suffix. The MIME type is necessary so as to display the appropriate icon in the file list. If the name file already exists in the database the file is inserted with a version number equal to the maximum version number of all the instances of the file increased by one.

The rest of the user options found in a workspace are:

Search: When the user completes and submits the search form, a query is formed according user's input, committed to the database and all the records that much the search criteria are returned to the user's browser in a way similar to the file list of some folder.

Bulletin Board: The bulletin board is mainly based on Perl scripts. The user can see all the messages posted by the others in the same shared workspace. When the user completes and

submits a new message, a new message file is created in the file system and a new link is added in the message list. The Perl script connects to the database so as to authorize user access.

Chat: There is the chat server that runs as a service in the WinNT server and waits for connections from the Java applet clients. The appropriate parameters are passed to the Java applet in order for the client to recognize the identity of the user and to ensure that only the members of the same workspace can participate in a chat session.

SYSTEM FUNCTIONALITY

Functional Overview

The main functionality of the shared workspace service is the management of a number of files of any format that are stored in the server. This functionality is enhanced by means of interactive communication between the users, so as to fully satisfy the real-time needs of a group. The user of the shared workspace may exploit its communication features independently of the file exchange, rendering it a powerful way of co-operation, adjustable to many distance working scenarios.

As mentioned earlier, the application is built upon the idea of a "shared workspace", a group of directories and files controlled by the users of the workspace. Standard procedures are the downloading of files from the server to the local station of the user, the viewing and modification of files, the insertion of new files or new versions of existing files.

These functions can be seen in detail below.

File Exchange

Various features of the system facilitate the exchange of files among the users, including:

- *File Downloading.*
- *Directory Strcucture.*
- *Versioning and Asynchronous Writing.*
- *File and Directory Access rights.*
- *File Search.*
- *File and Directory Annotation.*

Bulletin Board

A bulletin board has a topic and everybody can read and send his own announcements or answers to existing announcements. The existing bulletin boards in the Internet support electronic announcements in a simple text format, which is chosen as the most appropriate format to be used for the bulletin boards in the shared workspace environment. The membership in a shared workspace is considered as an automatic way of subscription its bulletin board.

Chat

Chat is an on-line way to conduct a conversation among the collaborators of the same shared workspace. It is used by the collaborators that are simultaneously using the shared workspace, to send messages or other textual content to other users. Unlike an audio conference, the Chat service consumes few network and computer resources. When the user enters the chat session his identity is automatically displayed in the other users' screens, and he can begin talking with them. It should be noted that the discussion that takes place among the users of the particular workspace is restricted only to them and cannot be attended by external users.

Administration

A complete set of administration scripts are available to the administrator so that he can perform all administrative tasks from the browser, in an easy, user-friendly and transparent way. Specifically, there are forms for creating new workspaces, deleting the existing ones, giving and removing user access to several shared workspaces, creating new Services, Departments or any other organizational unit of the prefecture.

CONCLUSIONS AND FUTURE WORK

The design of the system as well as the platform adopted for the development ensure a relatively easy and quick enrichment of the interface and the services. This remark is advocated by the fact that the information is stored in a database that can be easily expanded to include additional information stored in new tables and relationships. Hereby we introduce some features that would contribute to a more efficient environment of collaboration.

Session Monitoring: every time a user enters Synergy, he will see whether the sessions contain announces or files altered later than the last time he used the application

The application could use *Drag and Drop Utilities*, so that the entities (e.g. files) could be easily copied or moved.

Wastebasket: The deleted files will be stored in a temporary location from which the user can retrieve deleted files or erase them permanently.

The system can also be modified to support workflow concepts such as definition of roles and rules by actors or a subset of actors of the members of a workspace and definition of tasks or activities within a shared workspace (Giera 1998).

Exploitation of the use of digital signatures to support the authentication during exchange of documents between the different organizations and employees.

REFERENCES

Bentley R. and Horstmann T. 1997: "Distributed Authoring on the Web with the BSCW Shared Workspace system": ACM Standards View ACM Press.

Bouras C, Lampsas P., Zarafidis P., and Zoura A. 1998 "Web-Enabled Distributed Collaborative Learning Environment": 6th International Conference on Computers in Education-ICCE 98, Beijing, China.

Chiu D.M. and Griffin D. 1995. "Workgroup forum: Tools and applications for the WWW-based group collaboration". http://www.w3.org/Collaboration/Workshop/Proceedings /P3.html

Giera J.I. 1998. "Managing CSCW. Proceedings of the ERCIM workshop on the CSCW and the Web".