



A co-operative environment for local government: an Internet–Intranet approach

Christos Bouras ^{a,*}, Panagiotis Destounis ^b, John Garofalakis ^a,
Vassilis Triantafillou ^a, Giannis Tzimas ^a, Panagiotis Zarafidis ^b

^a *Computer Technology Institute (CTI), 3 Kolokotroni St., 262 21 Patras, Greece*

^b *Computer Engineering and Informatics Department, University Patras, 26500 Patras, Greece*

Abstract

In this paper we present a Web-based, database oriented CSCW system to support information dispersal between different Local Government authorities (Region, Prefecture and Municipalities) and the citizens. The system is implemented around the ‘shared workspace’ notion, a group of directories and files controlled by the users of the workspace. Intranet and Extranet technologies are exploited to enable asynchronous and synchronous co-operation, 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). © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: WWW; Intranets; Extranets; CSCW; Shared workspace; Dynamic web pages

* Corresponding author. Tel.: +30-61-960375; fax: +30-61-993973.

E-mail addresses: bouras@cti.gr (C. Bouras), destoun@ccid.upatras.gr (P. Destounis), garofala@cti.gr (J. Garofalakis), triantaf@cti.gr (V. Triantafillou), tzimas@cti.gr (G. Tzimas), zarafidi@ccid.upatras.gr (P. Zarafidis)

1. Introduction

This decade is witnessing the forging of a link of unprecedented magnitude and significance between the technological process and economic and social organization. A new ‘information society’ is emerging in which the services provided by information and communication technologies (ICT) underpin human activities. Where the general public is concerned, the penetration of ICT-dependent products and services into everyday activities is also striking. This generates new forms of economic and social organization the structure of which is no longer subject to geographical constraints but depends on telecommunications networks. Authorities concerned with the management of public funds and wishing to provide their constituents with better quality services also call upon ICTs. Relationships between the general public and the authorities are changing.

Today’s public and private organizations have recognized as a major issue that there is a need for efficient, quick and error-free information exchange between Citizens and the Administration and within the organization itself.

Internet and especially WWW offer an attractive platform for developing collaborative applications (groupware). Although that the WWW technology is insufficient in several areas (stateless server/client architecture, insufficient support for synchronous communication, etc.) (Chiu and Griffin, 1995) it 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 collaboration¹. 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 (Bentley and Horstmann, 1997). The workspace comprises asynchronous and synchronous co-operation 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:

1. 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

¹ <http://www.w3.org/Collabortion/Overview.html>.

folder structure, a notion that all users are familiar with. The repository is equipped with the basic functions of delete, search, annotation, versioning and MIME recognition.

2. A discussion forum (bulletin board). A shared location for posting announcements supporting follow-up messages (i.e., replies to existing messages).
3. 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. Section 3 provides details of the system architecture, specifically presenting analysis, design and implementation issues. In Section 4 the functionality offered by the system is outlined. Section 5 summarizes conclusions elicited from our work and presents future work issues.

2. Description of the project

2.1. General overview

The first organizations that adopted the use of Internet and Intranet technologies in the working environment were organizations that spent a significant percentage of the daily working time in distribution of documents (Bernard, 1998). Public organizations and especially Local Government authorities such as Municipalities, Prefectures or Regions circulate a vast amount of information.

The information dispersal is characterized by the following drawbacks:

- The citizens often get tired and confused by the complexity of the procedures, the time consuming operations and even the geographic dispersion of the offices and are forced to understand the structure and operation of the administration mechanism in order to keep track of their affair.
- The lack of efficient classification, the inability to provide valid and on-time information renders its retrieval, which is considered as a crucial issue.
- The aim to minimize transportation costs and financial and human resources can be achieved through a process avoiding the immediate contact between the service and the citizen (e.g. distribution of application forms).

Since a major source of information are the citizens the communication and distribution of information to and from the citizen in an immediate, time and cost effective way is particularly important.

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. This will result in

reducing the workload of the employees and potential users will be able to reach the information needed whenever one wishes, without delay.


- The presentation of complex information in a user-friendly way.
- In brief, the services provided by the System are (Fig. 1):
- 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 and bulletin board.
 - E-mail for the employees of the Prefecture of Corfu.
 - Access to the Internet, and specifically WWW, to support information retrieval, easier and equal 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.


2.2. Infrastructure


The distribution of information and the exchange of documents within, from and towards (region, municipalities, citizens, various groups, etc.) the Prefecture is supported by the development of Web servers in the Municipalities of Corfu and in the Prefecture. The Web server of the prefecture will provide the communication



Prefecture of Corfu
Intranet


[Help](#)


[Comments](#)


[Back](#)

Direction:	Administration		Last Logon:
Department:	Informatics		6/7/1998 4:13:00 μμ
Employee:	Panagiotis Zarafidis		Remote IP:
			150.140.121.36

Services Provided

- ◆ Shared Workspaces
- ◆ Electronic Mail
- ◆ Multimedia Announcements
- ◆ Citizens' Applications - Comments
- ◆ Communication with the Citizens
- ◆ Teleconference
- ◆ Useful Information

Computer Technology Institute & GSRT
6/7/1998 5:52:29 μμ

Fig. 1. First page of the prefecture intranet.

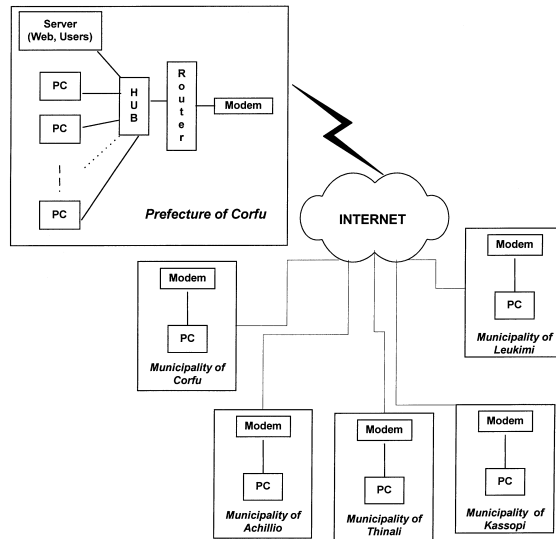


Fig. 2. Connection of the prefecture with the municipalities.

with outside organizations (ministries, EU information nodes, etc.) as displayed in Fig. 2.

3. System architecture

3.1. System analysis

Our architectural and design decisions concerning the structure, the functionality and the development of the shared workspaces were mainly dictated by the analysis of the user needs (Bouras et al., 1998a). The analysis showed that in order 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 and that fall to the following categories:

Performance

- 24 h availability of the system.
- When the system data change, the updated data will be immediately available to the users.

Safety

- If a sub-system fails it must not affect the other sub-systems.
- The system must be able to recover from a system failure within 10 min.
- No information is to be lost if the system fails.
- System availability is ensured during degraded mode operation.

Security

- All databases must be able to validate the authority for any changes that are made and for the access rights (read, write, etc.) of the users.
- The consistency of the data must be ensured.
- There will be constant validation of the user rights during the viewing and updating of the system contents (web pages, files, shared workspaces).
- The database administrators can only modify the databases of the system with data provided by authorized persons (only).
- The option for encryption of the data transferred must be provided.
- Detailed records and logs must be kept so those potential problems can be traced back.

Maintenance

- Small changes of enhancement, extension or adaptation can be performed without necessitating re-development of the system and without the need for an interruption in its operation.
- A strict backup procedure must be followed.
- The whole system architecture must be open in terms of smooth integration with new products or new services.

Environment

- The inexperienced user must be able to use the system within a short time period.
- The system administrators will be able to fully operate the system after a day's training.

3.2. System design

The architecture/design of the system follows the strong movement towards three-tiered, server-centric architectures (Bouras et al., 1998b). 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, maintainability, security and code reusability.

In Fig. 3 the three levels of our system is depicted.

In the first level, 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 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 and the user groups 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 to 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.

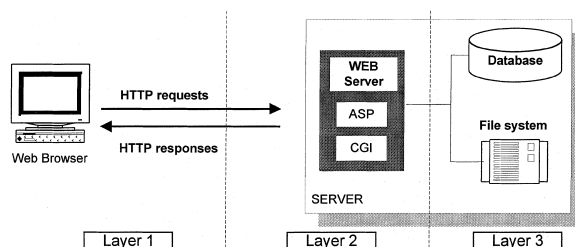


Fig. 3. The three tiered architecture.

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 system, 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.

3.3. Implementation issues

3.3.1. General

The web-clients send their requests and receive 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 – as shown in Fig. 2 – every time the user makes the request. Thus, the use of CGI and web server scripting is essential in building such systems. However, executing scripts and components on the server, has not the performance limitations and development difficulties of CGI, while allowing 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?² 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 com-

² <http://www.microsoft.com> white papers: web server scripting.

ponents 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 (Middleton et al., 1997). 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 depicted in Fig. 3. 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.

3.3.2. 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 lockers 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 (see also Microsoft IIS 4.0 documentation³). 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 *Global.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. Every ASP page for the connection to the database uses the DSN.

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 the entire ASP page that the user will request throughout his visit.

3.3.3. 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

³ <http://www.microsoft.com/iis>

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 match 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.

4. System functionality

4.1. 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.

4.2. File exchange

Various features of the system facilitate the exchange of files among the users.

4.2.1. File download

The collaborator can easily fetch the files located in the server. A user can save the file in his local disk, or open it automatically with the application of his preference, provided that his Web browser is appropriately configured to do so. Besides, in case of the most common file types, the file type’s icon is displayed on the browser, next to the file name, so that the user can recognize the application it was created with.

4.2.2. Directories

In order to achieve a more clear and efficient organization of files in the shared workspace, the users can create and modify directories within the workspace. Thus, the task of grouping all relevant, to some issue, files is accomplished in an easy and understandable, for the user, way.

4.2.3. Versioning and asynchronous writing

The application supports asynchronous file writing, that is, a user creates a file, saves it in the server, another user downloads it, comments on it or corrects it or adds to it and uploads the new version on the server so that the other users can view the updated version.

When the users uploads a file with the same name with a file that already exists in the server, the new file is automatically considered the next version.

4.2.4. Access rights

All the users of the workspace are allowed to see (download) the files or view the contents of a directory available but only the creator (and of course the administrator) of a file or a directory is allowed to modify its information or delete it.

4.2.5. File search

We have included in our system the capability to search for a file based on some information related to it. This is very useful in case a user needs to find a document, of which he remembers partial information, such as the title or part of the description, etc. but not the exact location or the folder it belongs to.

When the user completes and submits the search form, the contents of the shared workspace are searched and the results are displayed as a file list where the user has the standard options (download, delete, etc.).

4.2.6. Additional info

In order to assist the procedure of file exchange and to avoid loss of precious time and confusion, additional information about the files and directories is provided. For every file and directory there is a short description (entered by the creator during the upload procedure) along with other info such as creation date, version, creator, etc.

4.3. Bulletin board

A bulletin board has a topic and everybody can read and send his own 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.

A bulletin board is an ‘off-line’ way of communicating among the collaborators. The membership in a shared workspace is considered as an automatic way of subscription its bulletin board.

A message in the Bulletin Board contains the collaborator’s name, the date, the subject and the body of the message. A user can see all the messages posted so far to the shared workspace’s bulletin board, send new messages, send follow-up messages (that is, replies to existing messages), follow-ups to follow-ups and so on.

4.4. Chat

Chat is an on-line way to conduct a conversation among the collaborators of the same-shared workspace. The collaborators that are simultaneously using the shared workspace, to send messages or other textual content to other users use it. 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.

4.5. 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.

4.6. Changing password

The IIS 4.0 enables users to change their password whenever they wish through the web. One can change ones password remotely from the following location <http://servername/IISADMPWD/aexp.htr>. The secure socket layer (SSL) is used to encrypt the user's password and to ensure data integrity.

For security reasons user passwords must change periodically. The administrator of the system has the ability to enforce compulsory change of password to users via the *user manager for domains/account policy*, an administration tool provided by the Windows NT 4.0 operating system. The administration can set the maximum password age for all passwords or set for a particular user the property *user must change password at next logon*. This, of course, is not an innovation. The innovation lies on the fact that when a user whose password has been expired tries to entered a restricted object through the web, e.g. to enter to a shared workspace, the web server will automatically redirect the user to a new page. This page prompts the user to enter a new password. More details about these properties can be found in <http://support.microsoft.com/support/kb/articles/q184/6/19.asp>.

5. Conclusions and future work

The design of the system as well as the platform adopted for the development ensures 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 logs on the system, 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 (Griera, 1998).
- Exploitation of the use of digital signatures to support the authentication during exchange of documents between the different organizations and employees.

Acknowledgements

This work is supported by the project “An Intranet and Internet based Information System for Administration and Information purposes for the prefecture of Corfu” funded by the Greek GSRT of the Ministry of Research and Technology.

References

- Bentley, R., 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. Zoura, A. 1998a. Web-Enabled Distributed Collaborative Learning Environment. In: Sixth International Conference on Computers in Education-ICCE 98, Beijing, China, 14–17 October.
- Bouras, C., Destounis, P., Garofalakis, J., Spirakis, P., Tzimas, J., Triantafillou, V., Zarafidis, P., 1998b. An Intranet and Internet-based information system for administration and information purposes. In: Fourth European Digital Cities Conference: Changing patterns of urban life, Salzburg (AT), 29–30 October.
- Chiu, D.M., Griffin, D., 1995. Tools and applications for the WWW-based group collaboration. In: Workshop on WWW and Collaboration, Cambridge, MA, 11–12 September.
- Griera, J.I., 1998. Managing CSCW. In: Proceedings of the ERCIM workshop on the CSCW and the Web.
- Middleton Deng Kemp, 1997. Web Programming with PERL 5: Samsnet-http://www.sampublishing.com.
- Rernard R., 1998. The Corporate Intranet. Wiley, New York.