A WEB BASED VIRTUAL COMMUNITY: FUNCTIONALITY AND ARCHITECTURE ISSUES

Christos Bouras
Vaggelis Igglesis
Vaggelis Kapoulas
Thrasyvoulos Tsiatsos
Computer Engineering and Informatics Dept., University of Patras and
Research Academic Computer Technology Institute,
Riga Feraiou 61, GR-26221 Patras, Greece
{bouras, igglesis, kapoulas, tsiatsos}@cti.gr

ABSTRACT

A Web Based Virtual Community is an example of a Collaborative Virtual Environment, which focalizes in providing collaborative functionalities and both synchronous and asynchronous interaction services. This paper describes a virtual community, which aims to meet the requirements of a virtual collaboration space and support autonomous and collaborative e-learning services. In particular, a web-based platform is described, which is able to support communities whose members are able to interact and form groups based on their common interests. The community provides its members with the ability to communicate through enhanced forums and chats, but also, to interact in the social interaction spaces. The users can arrange events, which occur in virtual meeting centers, where 3D avatars represent the users. In these multi-user spaces, extended functionalities are offered, while system agents assist and consult the users.

KEYWORDS

Collaborative Virtual Community, Networked Virtual Environments, Architecture.

1. INTRODUCTION

The constant expanding of the World Wide Web and the familiarization the users with the Internet generated all the necessary preconditions for a wide adaptation of the electronic communication. The Internet now forms the basis for the constitution of on-line communities. Software applications allow on-line communication and interaction among users, which is being expanded by the use of 3D multi-user spaces, simulating real communities. These virtual communities offer a variety of functionalities, while are proven to be efficient for educational and instructional goals [Spector et al, 2002].

Regarding the social and educational aspect of the virtual collaboration spaces [Bouras et al, 2003], this research area is becoming increasingly interesting both from a technological perspective and a social perspective. The technological challenge arises from the need to develop a collaborative environment, which, should be able to offer educational senses to the users. On the pedagogical and social part, this concept raises questions on the models that can be used and how they can contribute in the most efficient way to the distribution of knowledge in a virtual environment. The design of this Web Based Community aims to develop a platform that will provide the users/members of the community with the ability to learn through an autonomous learning process and afterwards the possibility to experience the knowledge offered through a social interaction. In other words, the web-based community described in this paper is a virtual collaboration system that aims to enhance each member on the cognitive and social field.

This paper is structured as follows. Firstly, the basic requirements that arise from the theoretical approach are described, providing the reasoning for the development of the community. Secondly, the functionality of the platform is presented, by introducing the entities that the system ontology consists. Afterwards, the

architectural structure of the system is analyzed, presenting the different layers of the intended model. Finally, some concluding remarks and proposals for future work are provided.

2. BASIC REQUIREMENTS

In this section are provided the basic requirements for the development of a web-based virtual collaboration community. In this platform, the formation of groups is the initiative of the users. Therefore, the groups are spontaneously formed and developed without the system's interference. This is opposed to Formal groups that are created by a parent organization and are intentionally designed to direct members toward some important organizational goal. According to Ward [Ward 2001], a Community is a group of people who are related to some extent to each of the following minimal requirements:

- A common purpose: some form of shared meaning or reason for their co-existence.
- A common cultural context: a collective identity that includes values, beliefs, attitudes, behavioral norms, and accumulated experiences.
- Co-location: all members of the community share a common physical and/or virtual space.
- Voluntary participation: the community chooses who participates in the community and members of the community choose to participate in it.
- Multiple, shifting and overlapping membership and participation: people typically participate in more than one community at a time, tend to join and leave the different communities to which they belong, and more communities may share purpose, location, cultural context or membership.

To maximize these elements, the strength and cohesiveness of the community can be enhanced by the following elements: common history, shared knowledge, common practices, common action, shared vision of the future and co-created future.

The platform described in this paper, functions in two directions, including two different action domains of the users, the Individual and the Social Domain. In the social domain, the user will be able to join a group or to create a new one. Each group will have an objective and a set of rules. The decision on the type of group and the group interaction that will be part of the social domain rests on the theoretical foundation of the Group Theory [Sinclair et al, 2002].

The Individual Domain includes the virtual environment plus its content, main features and characteristics whose design aims to encourage the user to cultivate an autonomous learning process. In the Social Domain, the system provides the user with the ability to create social groups. Beyond the dialogical interaction processes that will be grown among the members of each group, these groups will be stimulated by the system to produce and apply ideas, which will serve their needs. The system's support to all these capabilities will be felt both in an individual and a social level. We should underline that although there is a distinction between the users' activity both in the Individual and the Social Domain, these domains will be not presented separately in the platform as two different parts. User's navigation in the system will be based on the content and the activities provided, which are related to the two different domains. For example, a user is able to perform an individual activity, such as studying some courses on how to participate in interviews and then practice on what he learned by searching a partner in the social domain.

3. DESIGN RATIONALE

This section describes the design rationale of the platform that also determines the system's functionality. Based on the main concept of the platform presented in the previous section, here the system ontology and the main entities on which the community is based are described. These entities that constitute the platform are the following: Knowledge Assets (KA's), Interaction Places and Actors. The Knowledge Assets (KA's) represent all the entities that exist on the system. These entities are objects that become resource of knowledge for the users of the platform. Such an object can be a mathematical theorem, a personal experience, and a set of learning activities or even a personal profile of a user, since each one of those can offer knowledge to a member of the community. The Interaction Places are different spaces within the platform where the different Knowledge Assets can be found. In these spaces, users are able to retrieve

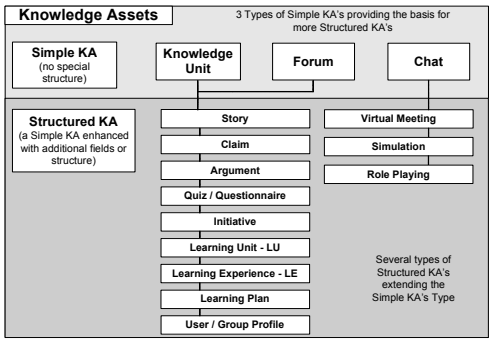
information stored in the system or communicate and interact with other members of the platform. The Actors can be individual members, members of a group or the Agents of the system.

The main target of the platform is to provide the users with learning experiences and with these spaces, which will enable the members of the community to interact and participate in various knowledge exchange processes. This platform is a complex, distributed, virtual environment, whose members' orientation is to learn through the use of the variety of Knowledge Assets, to share knowledge and to experience the Interaction Places.

3.1 Knowledge Assets

The Knowledge Asset (KA) is an essential concept in the platform. Every object in the system is a potential Knowledge Asset. According to the structure, every component can be described and represented as a KA. For this reason, the system consists of all the necessary Knowledge Asset Types, which are suitable to describe all the components. The KA types can be divided in two main categories: the Simple KA's and the Structured KA's. The different types of KA's are shown in Figure 1.

Figure 1: Knowledge Asset Types



In the first category, the content is not structured in a particular way, thus containing information such as texts, images or multimedia applications. On the contrary, the Structured KA's contain the above information in a structured way, with the addition of metadata that help the users and the system to classify and use this information efficiently.

3.2 Interaction Places

The Interaction Places represent the way in which a user is able to navigate through the system. They provide all the possible activities for a community member, which are supported by the system. The set of the system's functionality consists of these activities, which are embedded in different Places and Sub Places, since each Place is divided in smaller interaction virtual web spaces. In this paragraph the role and the activities supported by each Place and Sub Place of the system will be analytically described.

The environment is divided into five Interaction Places. In these virtual spaces the users experience different Knowledge Assets. These web spaces cover the different aspects of the Learning Experience. Their title had to indicate the activities supported in each space, and point the dynamically created content they contain. Therefore, in a similar rationale to the Five Stages of Group Development [Tuckman and Jensen 1977], the five Interaction Places, in which each member of the community experiences a different set of Knowledge Assets that contribute to his personal learning activity, are the following: the Learning Space (*LEARN*), the Collaborative Knowledge Exchange System (*SHARE*), the 3D Social Stage (*PRACTICE*), the Member's Place (*CONNECT*) and the Activities Subspace (*JOIN*). All the spaces are enhanced by the presence of the Agents that stimulate, help and guide the user.

Learn: The LEARN place is the virtual place where the users are able to retrieve pure knowledge. The Knowledge Assets provided in this space are the following: Learning Experiences, Learning Units, and Learning Plans. Every Learning Experience or Learning Unit is related to a life domain, in order to indicate the content of each KA to the user. The life domains can be family, work, science, entertainment etc.

Share: This system area introduces the users to all the provided services that support the members of the community to act individually or as a team, in order to create, share, store and retrieve information.

Practice: The Practice Interaction Place is the web space where the users of the platform are able to investigate the system's learning possibilities, through virtual environments (worlds). It is the Social Interaction area of the platform, where the members of the community are able to meet, interact and experience a great number of activities that take place in a real-time, multi-user, 3D environment.

Connect: In the Connect Interaction Place, all the community members' profiles are presented. These profiles concern the individual users/members of the community, the configured teams and the Agents. These profiles are concerned to be Knowledge Assets.

Join: The Join Interaction Place provides the users with a summary of the most interesting ongoing events (for each user) within the platform's community. Other critical information is also provided in this space (such as community search, who is on-line, help and FAQs).

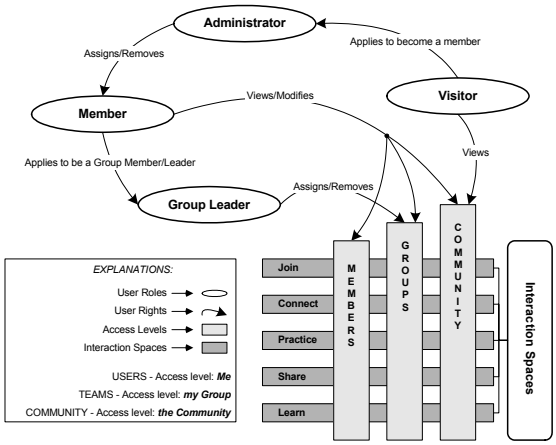
3.3 Actors

As mentioned above, there are three types of actors in the platform: the individual members, the members of a group and the Agents of the system. In this section the Roles and Rights of each of those actors will be presented.

3.3.1 Roles and Rights

A very important factor to be taken into consideration in the design of such a platform is the definition of the users' roles within the community, as well as the levels of access that each role involves. The community we describe consists of three different entities called Actors. The Actors are: the Individual Members, the User Groups and the Agents. Each Actor has its own profile, which is accessible from the community members. Apart from these entities, there are four, well defined, user roles each of which possess certain privileges.

Figure 2: Roles and Rights in the platform



The user roles (pictured in Figure 2, along with the user rights) are as follows:

- Visitor: This type refers to the users who have not yet been registered to the system. The Visitors may view and access information about the community in general, but they are not able to navigate through the system's functionality. In order to achieve this, they have to register to the system, by completing a registration form, which will be approved by the system administrator.
- Administrator: The administrator is the person responsible for the efficient and smooth functioning of the community. Hierarchically, the administrator has full access to the system. He is responsible to approve or reject applications from new users, and is able to add, remove and modify functionalities or fix possible feeblenesses of the system.
- Member: The members of the community constitute the majority of the system users. They are able to browse through the content, interact with other users and experience all the system's capabilities. The

individual members are also capable of forming new user groups. The member who is the founder of a group becomes automatically the Group Leader.

- Group Leader: The Group Leader is the member who is responsible for the team. He is the representative of the team, but also the one who stimulates other members of the team, in order to set the group's goals, ideas and initiatives.

As we already mentioned, each component of the system can be represented as a Knowledge Asset. Thus, the rules that exist in the system are related to the KA's. These general rules concern the permissions a user holds to create, delete or modify Knowledge Assets. There are three permission levels for each KA. The options for the author of a KA are the following: include the accessibility to the user himself (level: *me*), his group (level: *my group*) or the entire community (level: *the community*). These levels are valid for the access and modification of the specific KA's by the community members.

3.3.2 Agents

The Agents can be seen within the platform as special, non-human members of the users' community, whose main activity is to intervene dynamically and proactively (or upon request) with value-adding suggestions formulated to each user individually and in a situated way.

The type of suggestions produced by the Agents depend mainly on the characteristics of the Agent itself, but always aim at gradually increasing the member's familiarity, interest and involvement in the community dynamics, as well as to stimulate their reflective thinking. The key functions of Agents are hence to:

- Contribute to gradually increase the level of familiarity of each user with a specific subsection (Space) of the overall system (Agents are allocated to Spaces).
- Contribute to increase the value perceived by each user from using the system by locating and suggesting the exploration of valuables KA's and Members (reduced search costs for value creation at the individual level).

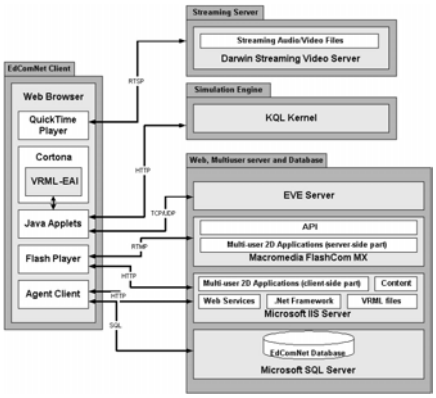
Contribute to gradually increase the level of participation/involvement of each user (provide stimulus to knowledge sharing and proactive behavior with consequent value creation at the collective as well as individual level).

4. SYSTEM ARCHITECTURE

This section describes the system architecture in the form of functional blocks and their interrelations. The main aspects of the system described here is the architectural structure, which is the static structure of the architecture in terms of its components, their interconnections, and the interfaces and operations offered by the components.

The platform's architecture is based on several components that provide all the needed functionality as follows: the web server along with the web services, the multi-user server, the streaming video server, the simulation engine, the database and the client (which is a web browser along with the respective plug-ins for accessing specific functionality such as streaming video, 3D multi-user spaces, voice chat, text chat and interactions with the agents). An overview of this architecture is shown in Figure 3.

Figure 3: Architectural Overview



The main system components are described in this paragraph. For each component, there is a description of its responsibilities, interfaces it offers, and other components with which it will interact. Furthermore, a description of the component interfaces and their operations is presented.

4.1 Client

As a web-based platform, the client of the system is a typical web browser, utilized by the users in order to access the learning content and the functionality of the virtual environments. The only requirement of the Web browser is that it must be able to embed and interact with the Flash player and the Parallel Graphics Cortona VRML plug in.

Flash Player: The Flash Player interacts with the web-server in order to download the client-side part of the above applications and then with the multi-user server in order to communicate with the other participants in the above multi-user applications. The interaction between the Flash player and the Flash communication server is used in order to support multi-user applications such as Voice and Text Chat, Shared Whiteboard etc. This interaction is achieved through the exploitation of the client-side application program interface (API) of the Flash Communication Server MX.

The Flash Player is a plug-in or an ActiveX control, which is embedded into the web-browser. The embedded player interacts with other HTML-based elements of the web page through ASP.NET and JavaScript functions. This interaction will be used in the system in order to deliver the user data (e.g. name, assigned roles, avatars etc.) to the Flash movies. There are two ways to achieve this interaction:

- The first one comprises a set of Flash methods (JavaScript functions) that are specific to Flash movies. These methods are used in order to send JavaScript calls to Flash movies from a scripting environment.
- The second way is used for sending and loading variables to and from a remote source. It comprises a set of Flash actions and object methods that are used in order to send information to and receive information from, server-side scripts, text files, and XML files. It can also be used for loading JPEG and MP3 files from a remote source into a Flash movie while the movie plays. In addition, server-side scripts can request specific information from a database and relay it to a Flash movie. These actions and object methods uses a protocol to transfer information, and require information to be formatted in a certain way.

Parallel Graphics Cortona VRML client: Cortona is used for accessing the 3D Practice Space (Virtual Plaza, Virtual Meeting Rooms etc.). Furthermore it is used in order to access streaming video and embedded flash applications (such as whiteboard) in a 3D space.

It interacts with the web browser and external applications (such java applets) through VRML-EAI [Marrin, 1997]. Furthermore, it interacts with EVE multi-user server through Java applets in order to send/receive shared VRML events to/from EVE server. In addition, Cortona can exploit various VRML extensions (EXTERNPROTOS) in order to embed flash applications and streaming media in a VRML file.

4.2 Web Server

The web server is used for (a) storing the content KA's, (b) storing and executing the scripts/modules of the web services, (c) storing the client-side files of multi-user applications (Voice Chat, Text Chat, Shared Screen, 3D Games). It interacts with the web browser using the HTTP protocol and with the database through the modules/objects (web services) in order to implement specific functionality (such as forum, KA creation tools, etc).

Web Services: The Web Services represent one of the basic components of the system architecture. More specifically the scripting environment supports the operations such as the interaction with the database, the administration of the system, users' authentication, manipulation/extraction of access rights of each user in each KA, tracking users actions for the Agents, supporting the functionality for the LEARN, SHARE, CONNECT, JOIN Spaces and providing internationalization to the web interface.

The web services are designed to be implemented using Microsoft .NET Framework, the ASP.NET scripting languages in conjunction with MS SQL Server 2000 as the database management system. Due to this fact, the web server that has to be adopted for the development of the system is Microsoft IIS server.

4.3 Multi-User Server

The multi-user server is exploited for achieving the multi-user interaction in the Practice Space of the system's environment as well as the text chat interaction in the Share Space. It supports the collaboration and communication of multiple users in the same subspace of the above system spaces. The platform will exploit the Macromedia Flash Communication Server MX in conjunction with EVE platform [Bouras et al, 2001] as a multi-user server in order to support the above services.

The Macromedia Flash Communication Server MX will host and run the following applications in order to provide the desired communication services: whiteboard, shared screen brainstorming board, agenda tool (in the Practice Space), text chat (in the Share Space). Each of these applications consists of two parts: (a) a Flash movie, which provides the user interface and runs on the client-side (Flash player), (b) Server side scripting (ActionScript), which enables flexible control of shared state information and provides the logic for mediating real-time interactions among multiple users. The Macromedia Flash Communication Server MX communicates with the respective clients (Shockwave and Flash) via the Macromedia Real-Time Messaging Protocol (RTMP). For the implementation of the above referred functionality, two Application Program Interfaces (APIs) are provided along with the Flash Communication Server: the client-side API, and the server-side API [Macromedia, 2002].

The EVE platform will provide the multi-user substance in the 3D practice space in terms of users' representation by 3D avatars, chat communication as well as user interaction through avatar gestures [Bouras et al, 2002]. It interacts with specific Java applets and the Cortona VRML client. Also, it will interact with the simulation engine through HTTP protocol.

4.4 Streaming Server

The streaming server will be used in order to host streaming audio and video files. The streaming Server that will be exploited in the System is Darwin Streaming media server. The streaming client is used in order to play the streaming media and it is embedded into a web browser. The streaming client that will be used in the platform is QuickTime player. In order to embed streaming media in a web page, the HTML EMBED tag is used in conjunction with reference movie that contains a streaming track with an RTSP URL, which points to the media on the streaming server [Apple, 2002].

4.5 Database

The system's database is the most basic module of the system. It keeps the main part of the information needed for the operation of the virtual environment. Almost all modules of the system are interacting directly (Web services) or indirectly (e.g. multi-user server) with the database. More specifically, the database of the system is responsible for the following tasks: (a) storing of users' data (roles, access rights, profiles, etc.), (b) storing of KA's/Learning content data (accessibility rules, editability rules, metadata, etc.), (c) storing of auto-logging parameters users' actions for evaluation purposes as well as for facilitating agents' tasks.

Database structure for users' data: The main entity in the platform is the Actor. There are three types of Actors (Individual Users, Groups and Agents) from which the two of them are external users. The platform enables system users to exist as individual community members as well as organized in groups of them. Therefore, two different tables exist in the database, where individual users' and groups' data is stored. A third table is used, in order to mark the Group Members of each group. Finally, the Group Profile table is related to the Individual Profile table, to indicate the user who is the Group Leader.

Database structure for Knowledge Assets: The structure of the database is strongly connected with the Knowledge Assets (KA's), Spaces, Subspaces and Tools concept of the platform. The KA's are the basic "cells" of the system. There are many types of KA's that have different characteristics as well as common fields. Therefore, in the Database there is a common table for all KA's, where all the common characteristics of all KA's are stored. In order to support the different types of KA's (namely: Knowledge Units, Structured Knowledge Units, Synchronous and Asynchronous Exchange), additional database tables are exploited to store the specific information of each KA type.

Database structure for auto-logging parameters: One of the main goals of the platform is to track user actions. This will provide statistics about all Knowledge Assets and how the users behave in each of the

system Spaces and Subspaces. The critical information that has to be stored is the number of each user actions in each Knowledge Asset, and the time between every action. Concerning Spaces, Subspaces and Life Domains, it is important to store which Spaces / Subspaces and Life Domains are being visited by the users, and how much time the users spend in each Space or Subspace. All this information has to be accessed by the Agents, in order to provide them with the desirable information to implement their functionality. Every user action during a visit in a Knowledge Asset is stored in a table that relates the Individual profile's table, the Knowledge Asset's table and the KA Actions table. Moreover, it is essential to design a table that relates the profile of each individual user with the Space (or Subspace) and the Life Domain he is currently visiting. The number of those visits, and the time that every user remains in the Space/Subspace, is stored in the User Visits table. Finally, the overall time that each user spends in the system is stored in his individual profile.

5. CONCLUSIONS – FUTURE WORK

This paper presented the technological functionality and architecture issues of a web-based collaborative learning environment. The features of this environment include Interaction Spaces, Knowledge Assets, and Intelligent Agents, which are guided by the roles and rights of the platform. Its n-tier architecture combines various technologies that enable a rich browsing experience in the environments' spaces. The client side includes a browser extended with Macromedia's Flash Player and the Parallel Graphics Cortona VRML client. Multiple server-side products are used that enable the provision of the services offered to the users of the system: IIS web server, SQL Server, Macromedia Flash Communication Server (in conjunction with the prototype of the EVE educational virtual environment) and the Darwin Streaming Media Server.

The interconnection of so many diverse technologies in a single platform will enhance users' interaction and assist them in discovering new possibilities for socializing through the Web. The learning process gains new opportunities, since the web-based services such as forums, instant messaging, chat and educational content presentation is being empowered by the 3D interaction and simulation, which occurs in the Social Domain of the community.

A future development step of the described platform could be the integration of the Agents in the 3D multi-user spaces. According to the presented web-based community, the Agents exist only in the 2D spaces and interact with the users through text. An expansion of the multi-user environment platform (EVE) will enable the Agents to exist and function both during the navigation in the website and in the 3D spaces of the Community.

REFERENCES

Apple Computer 2002, *Administrator's Guide, QuickTime Streaming Server, Darwin Streaming Server*, Inc. 2002 Apple Computer, Inc.

Bouras C. et al, 2003, *Virtual Collaboration Spaces: The EVE Community*, The 2003 Symposium on Applications & the Internet- SAINT-2003, Orland, Florida, January 27-31 2003, pp. 48-55

Bouras C. et al, 2002, *An Educational Community Using Collaborative Virtual Environments*. 1st International Conference on Web-based Learning, Hong Kong, pp. 180-191

Bouras C. et al, 2001, *A Platform for Sharing Educational Virtual Environments*, 9th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2001) Split, Dubrovnik (Croatia) Ancona, Bari (Italy), 9-12 October 2001, Vol. II, pp. 658-665

Macromedia Inc, 2002, *Client-Side Communication, Server-Side Communication, ActionScript Dictionary*, Macromedia Flash Communication Server MX, May 2002

Marrin C., 1997, *Proposal for a VRML 2.0 Informative Annex, External Authoring Interface Reference*, Silicon Graphics, Inc. January 21, 1997

Sinclair G. et al, 2002, *Human Behavior in Organizations* (4th ed). Boston: Pearson Custom Publishing

Spector M. et al, 2002, *Networked Learning: Perspectives and Issues*, Springer-Verlag, ISBN 1-85233-471-1, pp. 93-109.

Tuckman B. and Jensen M., 1977, *Stages of Small Group Development*. Group and Organizational Studies, 2, 419-427.

Ward A. 2001. *What is a Community*, Work Frontiers International