A web-based virtual community

Christos Bouras*, Vaggelis Igglesis,
Vaggelis Kapoulas and Thrasyvoulos Tsiatsos

Computer Engineering and Informatics Department,
University of Patras and Research Academic Computer Technology
Institute, Riga Feraiou 61, Patras GR 26221, Greece
E-mail: bouras@cti.gr E-mail: igglesis@cti.gr
E-mail: kapoulas@cti.gr E-mail: tsiatsos@cti.gr
*Corresponding author

Abstract: This paper describes functionality and architectural issues in the design of a web-based virtual community. This virtual community aims to meet the requirements of a virtual collaboration system with both autonomous and collaborative e-learning services, by supporting communities whose members interact and form groups based on their common interests.

Keywords: collaborative virtual community; networked virtual environments; architecture.

Reference to this paper should be made as follows: Bouras, C., Igglesis, V., Kapoulas, V. and Tsiatsos, T. (2005) 'A web-based virtual community', *Int. J. Web Based Communities*, Vol. 1, No. 2, pp.127–139.

Biographical notes: Christos J. Bouras is an associate professor in the Department of Computer Engineering and Informatics of Patras University, Greece, and the Scientific Advisor of Research Unit 6 in Research Academic Computer Technology Institute (CTI), Patras, Greece. His research interests include analysis of performance of networking and computer systems, computer networks and protocols, telematics and new services, QoS and pricing for networks and services, e-learning networked virtual environments and WWW issues. He has published 150 papers in various well-known refereed conferences and journals. He is a co-author of five books in Greek. He has been a PC member and referee in various international journals and conferences. He has participated in R&D projects such as RACE, ESPRIT, TELEMATICS, EDUCATIONAL MULTIMEDIA, ISPO, EMPLOYMENT, ADAPT, STRIDE, EUROFORM, IST, GROWTH, among others. He is also member of several prestigious committees.

Vaggelis Igglesis obtained his diploma from the Department of Computer Engineering and Informatics of the University of Patras, Greece. Currently, he is a postgraduate student of the department and also works as an R&D engineer at Research Unit 6. His research interests include: computer networks, telematics, networked virtual environments and internet technologies.

Vaggelis Kapoulas obtained his diploma from the Computer Science and Engineering Department, School of Engineering, University of Patras, Greece. He is currently a PhD candidate at the same department and an R&D engineer at Research Unit 6. His research interests include networks, telematics and algorithms for distributed systems. He has extended professional experience in the analysis and design of networks, applications of networks, telematic applications, open and distance learning and tele-working.

Thrasyvoulos-Konstantinos Tsiatsos obtained his PhD degree at the Computer Engineering and Informatics Department of Patras University (Greece). He is currently an R&D Computer Engineer at Research Unit 6 of Research Academic Computer Technology Institute, Patras, Greece. His research interests include computer networks, telematics, distributed systems, networked virtual environments, multimedia and hypermedia. More particularly he is engaged in distant education with the use of computer networks, real time protocols and networked virtual environments. He has published four papers in Journals and 18 papers in well-known refereed conferences.

1 Introduction

The constant expanding of the World Wide Web and the familiarisation of the users with the internet have generated all the necessary preconditions for a wide adaptation of the electronic communication. The internet now forms the basis for the constitution of online communities. Software applications allow online 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 they 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 with the ability to learn through an autonomous learning process and 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.

The interrelated trends in the development of an online web-based learning community are three (de Vries and Kommers, 2004). The first one is integration of services. An online meeting place will become more and more a portal that gives access to a wide number of services. This platform aims to combine different interaction services in different spaces. These services will be provided both in 2D and 3D environments and will allow the users to interact with the system and the other community members through various methods.

The second one is practicality of use. The online meeting place is expected to become the networked desktop of the users at home, at the workplace, at seminars, etc. This shows that the usability of the system and the navigation facilities are essential to the function of the system, while their quality will ensure the fulfilment of the platform objectives. For this reason, the interface of the system was designed based on certain criteria.

The last trend is personalisation. The platform is expected to adapt to the preferences and competences of its members. The user must feel at home in an online community and

this is feasible only when he/she is able to customise the spaces of the community that he participates in. The customisation concerns not only the look and feel of the system, but also the personalisation of the content provided by the system or the other members of the community. This has to be also adopted in the 3D spaces of the platform.

This paper is structured as follows. Firstly, the trends in the development of the community 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 analysed, 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 the 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 organisation and are intentionally designed to direct members towards some important organisational goal. 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 the co-existence of the members
- *a common cultural context*: a collective identity that includes values, beliefs, attitudes, behavioural 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 maximise 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 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 and principles that each user adopts when becoming member of this team. The decision on the type of the 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

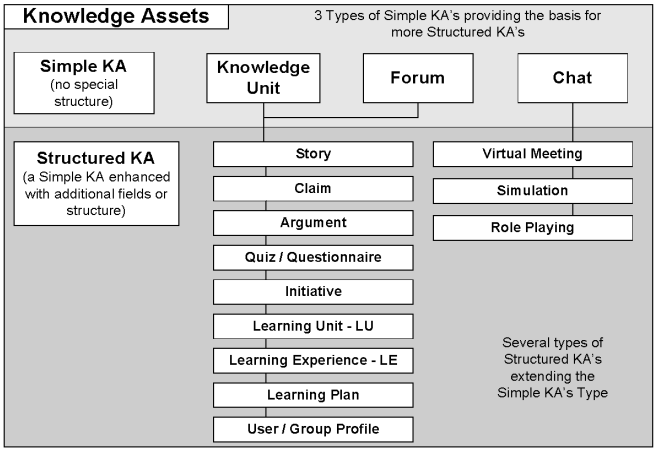
This section describes the design rationale of the platform that also determines the system's functionality. Based on the main concept of the platform presented in the previous section, here the system ontology and the main entities on which the community is based are described. These entities that constitute the platform are the following: Knowledge Assets (KAs), Interaction Places and Actors. The Knowledge Assets (KAs) 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 members of the community with learning experiences through these spaces that will enable them 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 of the platform. The KA types can be divided in two main categories: the Simple KAs and the Structured KAs. The different types of the available Knowledge Assets 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 KAs contain the above information in a structured way, with the addition of specific 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 subplaces, since each place is divided in smaller interaction virtual web spaces. In the following paragraph is the role and the activities supported by each place and sub place of the system is 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, applying 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, the collaborative knowledge exchange system, the 3D social stage, the members' place and the activities' place. All the spaces are enhanced by the presence of the agents that stimulate, help and guide the user.

3.3 Actors

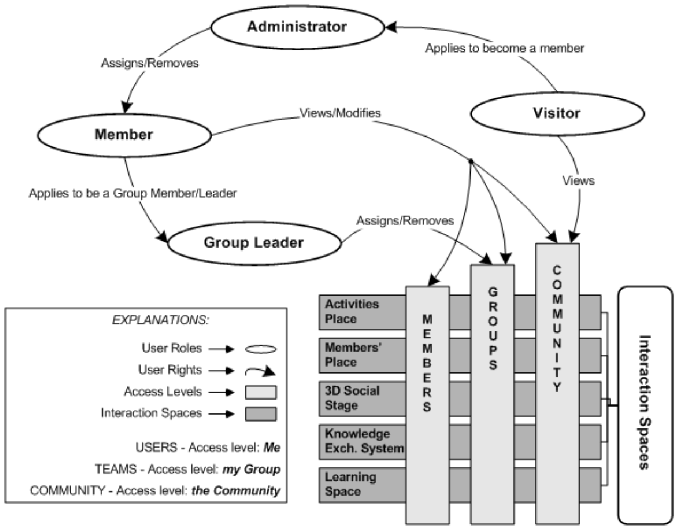
As mentioned earlier, there are three types of actors in the platform. These are the following: individual members, members of a group and the agents of the system. In the following section the roles and rights of each of those actors and their interconnection is presented.

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. 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 possesses certain privileges. 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 and its content. 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 a group of users. He is the representative of this team, and also the one who stimulates the other members of the team, in order to set the group's goals, ideas and initiatives. He is also responsible to communicate the team's principles in the community.

Figure 2 Roles and rights in the platform



As 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 KAs. 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 KAs 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.

The type of suggestions produced by the agents depends mainly on the characteristics of the agent itself, but always aims 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:

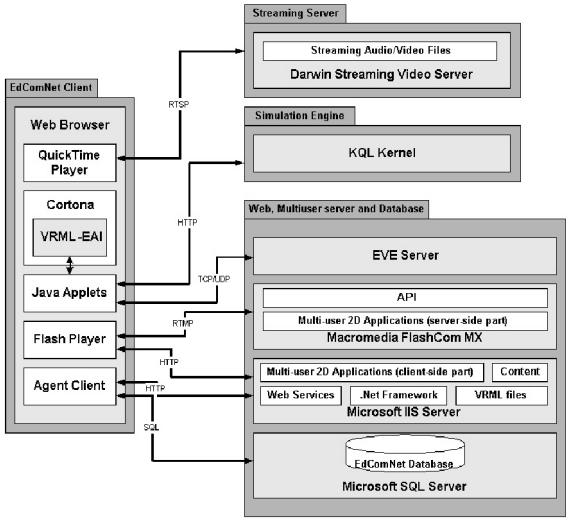
- contribute to gradually increase the level of familiarity of each user with a specific subsection (space) of the overall system (agents are allocated to spaces)
- contribute to increase the value perceived by each user from using the system by locating and suggesting the exploration of valuables KAs 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 behaviour 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. These components are the following: 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 the following 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, utilised 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, 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 Social Stage (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

- storing the content KAs
- storing and executing the scripts/modules of the web services
- 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 the database through the modules/objects (web services) in order to implement specific functionality.

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 and tracking users' actions for the agents. They also support the functionality of the learning space, the collaborative knowledge exchange system, the members' place and the activities' place, while they provide the internationalisation 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 3D social stage as well as the text chat interaction in the knowledge exchange system. 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).

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 3D social stage), text chat (in the knowledge exchange space). Each of these applications consists of two parts:

- a flash movie, which provides the user interface and runs on the client side (flash player)
- server-side scripting (action script), 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 Inc., 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, the Cortona VRML client and 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 quick time player. In order to embed streaming media in a web page, the EMBED tag is used in conjunction with reference to the movie that contains a streaming track with an RTSP URL, which points to the media on the streaming server.

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 interact 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:

- storing of users' data (roles, access rights, profiles, etc.)
- storing of KAs/Learning content data (accessibility rules, editability rules, metadata, etc.)
- 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 organised in groups of them. Therefore, two different tables exist in the database, where individual users' and groups' data are 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 (KAs), Spaces, Subspaces and Tools concept of the platform. The KAs are the basic 'cells' of the system. There are many types of KAs that have different characteristics as well as common fields. Therefore, in the database there is a common table for all KAs, where all the common characteristics of all KAs are stored. In order to support the different types of KAs (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. The database will store this information, and the statistical analysis of the data will create records for individual users and Knowledge Assets. These records will feed the agents with information for their proper functionality. For this reason, 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 and 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 socialising 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 messages. The upgraded development of the multi-user environment platform (EVE) will support the existence of the agents in the virtual worlds of the platform. This will enable the agents to exist and function both during the navigation in the website and in the 3D spaces of the community, providing the users with a completed way of interacting with the intelligent components of the platform.

Acknowledgements

This work is partially funded by 'EdComNet (Humanistic Urban Communal Educational Net)', IST Project (EDCOMNET / IST-2000-26037). Many thanks to everyone involved in the project for their contributions, fruitful discussions and hard work.

References

Bouras, C., Giannaka, E. and Tsiatsos, T. (2003) 'Virtual collaboration spaces: the eve community', *The 2003 Symposium on Applications & the Internet- SAINT-2003*, Orland, Florida, January 27–31, pp.48–55.

Bouras, C., Psaltoulis, D., Psaroudis, C. and Tsiatsos T. (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, Vol. 2, October 9–12, pp.658–665.

Bouras, C., Psaltoulis, D., Psaroudis, C. and Tsiatsos, T. (2002) 'An educational community using collaborative virtual environments', *1st International Conference on Web-based Learning*, Hong Kong, pp.180–191.

de Vries, S. and Kommers, P. (2004) 'Online knowledge communities: future trends and research issues', *Int. J. Web Based Communities*, Vol. 1, No. 1, pp.115–123.

Macromedia Inc. (2002) 'Client-side communication, server-side communication, actionscript dictionary', *Macromedia Flash Communication Server MX*, May.

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

Sinclair, G., Cuttell, D., Vandeveer, R. and Menefee, M. (2002) *Human Behavior in Organizations*, 4th ed., Pearson Custom Publishing, Boston.

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*, Vol. 2, pp.419–427.