From Virtual Storytelling. Berlin, Springer, 2001

VIRTUAL HOUSE OF EUROPEAN CULTURE: e-AGORA

(**E**lectronic **A**rts for **G**eographically **O**pen **R**eal **A**udience)

Jaroslav Adamec[1], Jiri Cizek[2], Michal Masa[1], Pascal Silondi, Pavel Smetana[3], Jiri Zara[1]

[1] Czech Technical University, Prague, Czech Republic
[2]Charles University, Prague, Czech Republic
[3]CIANT, Prague, Czech Republic

Abstract. This co-operative project links up important European centres of art and culture and creates an open network with an innovative application of the new information technologies: the creation of an unconventional and highly efficient communication base founded on an interactive, multimedia and trans-disciplinary approach to the production and presentation of contemporary forms of performing arts. The main channel of this project is the revolutionary communication and navigation system e-AGORA, which enables visitors to move intuitively in a virtual 3D environment on the Internet. In the VIRTUAL HOUSE OF EUROPEAN CULTURE a broad spectrum of the public can investigate contemporary Euro-regional artistic programs in real time and communicate interactively. At the same time, it is an instrument of individual and collective artwork in the domain of contemporary performing arts as it opens up a new horizon for a multidisciplinary form of artistic expression and presentation of artworks. The implementation of the project VIRTUAL HOUSE OF EUROPEAN CULTURE also has important theoretical and educational dimensions: a series of practical workshops, an international academic conference, thematic exhibitions, the production of the e-AGORA CD-ROM and a printed publication.

1 Objectives

The complex project **VIRTUAL HOUSE OF EUROPEAN CULTURE** has arisen as an active and open network of several important cultural centres; their previous co-operation and experiences have resulted in a revision of the old communication models. In the future, new types of co-operation and new instruments of communication will answer the need for a more efficient regional exchange; these will be used for the first time in the project **VIRTUAL HOUSE OF EUROPEAN CULTURE**. This common house stands on the firm foundation of actual artistic exchange between cultural centres of similar orientation.

It will introduce, however, the revolutionary feature of a common virtual space: **e-AGORA**. This will be shared not only by the connected cultural centres, but also by the artists and the general European public. The internet platform is open practically to everyone, without distinction: if offers **multi-participant** and **multi-lingual communication** and **interactive cultural entertainment**.

The multimedia navigation system **e-AGORA** will introduce everyday actual information (sound, music, image, video) from the existing spaces of five European cultural centres using the most modern information technologies. This information will be presented in real time in the virtual spaces of e-AGORA on the Internet. Visitors to this virtual 3D space will have the opportunity to select their own individual avatars; these avatars will represent them and enable them to communicate interactively with other avatars. The passive viewers of the general public (contemporary TV) will become individual members of a virtual European community, which is a revolutionary alternative to the local and mass-media limited approach to culture, art and information. **e-AGORA** will make use of the actual interiors of the connected cultural centres (e.g. DE WAAG in The Netherlands, Palace Akropolis in Prague etc.) for the **modelling of the virtual 3D space**.

When it is fully operative, the **VIRTUAL HOUSE OF EUROPEAN CULTURE** will be a unique site for artwork and the reception of artistic programs across the European continent. At the same time, however, it will remain an **open structure** with possibilities for further **expansion** and **the integration of cultural centres in other European countries**.

The transfer of artistic and technological information on a European scale will be so inventive, thanks to the key channel **e-AGORA**, that it will significantly influence the actual regional cultural exchange and prepare a new instrument for independent multidisciplinary artwork for creative artists in the field of the performing arts.

2 E-Agora architecture

E-Agora is a multi-user virtual environment aimed mainly at social interaction. For such a system two main decisions have to be made prior to the implementation: how to render the 3D scene and how to communicate between participants' machines.

Since we wanted to spare time and develop the system at minimum cost, we decided to exploit existing technologies to the most extent possible. Thus, instead of implementing our own rendering engine, we chose to base our system on VRML and to adopt one of the existing and freely available rendering engines – VRML browsers.

For the same reasons, to support communication between participants' machines, we used an existing Java library (DILEWA/GV [1, 2]) that is being developed by our research group. The library deals with distribution of messages among several machines connected to the Internet and solves the problem of bringing later connected users (latecomers) up-to-date on the current state. The communication pattern is based on the client-server model.

A typical implementation of a networked virtual environment has to consider the following issues [3]: a shared sense of space (participants have the illusion of being located in the same space), a shared sense of presence (participants perceive each other by the help of avatars), a shared sense of time (real time interaction with the world), a way to communicate (chat, gestures, voice, video), a way to share (the environment is shared, every change is visible to all participants).

The following text explains our approach to implement these features in E-Agora system.

2.1 E-Agora client

The client consists of VRML browser (responsible for rendering the scene) and Java applet (responsible for communication issues and the scene control). The browser runs as a plug-in of Internet browser, which accomplishes the delivery of VRML files to the plug-in.

The VRML browser renders the scene composed of a shared environment, participants' avatars and control components. We have decided to incorporate control components (for example gesture selection panel or chat-board) to the scene for two reasons. First, we wanted to provide the users with a pure 3D interface, making the view of the application consistent. Second, we wanted to stay within VRML to ensure easy portability of the system.

The connection to the server is maintained by a Java applet encapsulated in a VRML Script node. To support basic features of the MUDVR, following information is distributed among clients: notifications when a user enters/leaves the system, specifications of the users' avatars (URL of the VRML file), positions and orientations of the avatars, identifications of the gestures being performed, chat strings and environment changes. The DILEWA/GV library has been used to represent and distribute these data and the details will be discussed in the next section.

To control and receive the response from the scene, the applet is connected via VRML routes with the dynamic entities in the scene (avatars, control components and dynamic parts of the environment). These entities can generate events as a response to user's interaction (events are passed to the applet) and/or their state should be modified by the applet accordingly to the information received from the server (events are passed to the entities).

For example, clicking on another user's avatar brings up a chat-board and the user can type a message. In the background the avatar generates an event, which is handled by the applet. The applet determines the recipient and brings up the chat-board by sending another event to chat-board component. When the user clicks OK button on the chat-board, the message is sent to the recipient. Again, in the background, the chat-board generates an event containing the message and closes itself. The event is processed by the applet that communicates the message to the recipient (through the server). When the recipient's client receives the message, it sends an event containing the message to the chat-board component – the chat-board on the recipient's client is brought up with the message shown.

2.2 E-Agora server

As we have seen in the preceding section, various information has to be exchanged among clients (notifications, gesture identifications, avatar specifications, positions, orientations…). Moreover, since clients always load the original VRML scene that is unaffected by later changes, the system should also bring latecomers (users connected later to the system) up-to-date on the current state. For example, later connected client should receive information concerning all previously connected clients to display their avatars in appropriate positions.

We chose to exploit DILEWA/GV server, which was designed especially for such purposes. It allows creation and distribution of so called *general variables*.

General Variable consists of a name for its unique identification and a list of *commands* performed on the variable. A typical command sets the variable to an arbitrary *value*. The flexibility of the concept is based on the fact that the value can be compounded of any number of any primitive data types. It can be a simple value as well as a heterogeneous structure. When a user attempts to interact with the world (navigate through the world, click on another avatar, perform gestures…), the client application creates adequate variable and adds a specific command containing a value representing the user's action. The variable is then sent to the server, which is responsible for broadcasting the variable to other clients. Finally, the receiving client should decode the meaning of the variable and replay the original action locally. Additionally, the server stores all variables in a journal, which could be sent to latecomers to update their state. A set of flags associated with every variable controls its distribution and storage; it determines whether the variable should be sent to all connected clients or to subset only and whether the variable should be stored and how. Three storage methods are provided: not stored variables (distributed only), persistent variables and temporary variables (deleted from the server as their creator disconnects).

Let us illustrate the use of general variables in the E-Agora system with three examples. In the first example there is a variable containing information about the user (avatar URL and nickname). This variable is sent to all clients and stored temporarily at the server until the user disconnects from the system. In the second example the variable represents user's gestures. It is also sent to all clients, but it is not stored at the server, since latecomers are typically not interested in gestures performed prior to their connection. In the last example a variable represents a chat string. For the same reason, the variable is not stored at the server, too. In contrast of the previous example, the variable is not sent to all connected clients, but to the recipient only.

3 Future work

Our future effort will be aimed at making the system more stable and scalable by implementing UDP protocol in addition to TCP. This can be accomplished by an additional variable flag that will determine the reliability of the distribution. Next, we plan to add more shared dynamics to the environment (light switches, doors, desk games…). Since the client provides limited support of *NetworkNodes* as proposed in [4], this can be done by an integration of specially designed VRML objects with the environment.

References

1. Masa M., Zara J.: The Distributed Learning Environment Without Avatars, Proceedings of 2000 IEEE, London (2000)
2. Masa M., Zara J.: The General Variables Concept: A Simple Step from Single- to Multi-user Environment, Proceedings of 2001 IEEE, London (2001)
3. Singhal S., Zyda M.:S Networked Virtual Environments, ACM Press, New York (2000)
4. Living Worlds, http://www.vrml.org/WorkingGroups/living-worlds/