WebTalk: a 3D collaborative environment to access the Web

P. Paolini, T. Barbieri, P. Loiudice, F. Alonzo, M. Arru, M. Zanti and G. Gaia

Politecnico di Milano & Museo della Scienza e della Tecnica, Milano, Italy

Webtalk@elet.polimi.it, paolini@elet.polimi.it, barbieri@hoc.elet.polimi.it

Abstract

The notion of cooperative visit and interaction within a web site, both at 2D and 3D level, could add decisive motivation to pursue exploitation of the third dimension in networked environments. A prototype developed at the HOC laboratory in Politecnico of Milan, WEBTALK, allows development of custom 3D environments in which actions and situations can be shared throughout a network. The current prototype makes use of standard web-based Internet/Intranet technology: it is composed of a VRML (Virtual Reality Modeling Language) graphical engine and a Java-based TCP/IP (the Internet Protocol) communication layer. The technique has proven itself successful and has given rise to several interesting applications, such as the creation of Virtual 3D Museum Spaces at the National Science Museum (www.museoscienza.org), in which visitors can visit the museum interacting with the exhibits themselves and sharing their experiences with other visitors - even joining a guided tour with the virtual museum guide. A new prototype will make use of a completely Java-based environment, using Sun's Java 2 and Java 3D. The new WEBTALK will stress even further the issues of third party external authoring of virtual worlds and of cooperation patterns within the shared environments, with a distinction between shared object geometries, shared object behaviors, shared object data (stored in an on-line DBMS), and cooperation metaphors within the virtual world. Linking to 2D-based web sites resources will lead cooperation patterns to be extended also to regular web-space.

Keywords: virtual reality, cooperation, multi-user, third party authoring, web, databases, VRML, Java, Java 3D

1. The VRML version of the WebTalk prototype

WebTalk is a MultiUser technology engine for distributed Virtual Reality worlds. Under the view of the WebTalk system, the notion of Cooperative Environment implies the sharing of actions on objects component of the worlds and distribution of navigation events of each user in the world. Combining these features with the notion of engine, which provides for easy integration with existing worlds (plug-and-distribute), offering in the meantime a transparent interface to the user, the system makes possible to write the geometry and design the behavior of the worlds, plug the engine in, setup the distribution server, and publish the result on the Web as a Shared Virtual Space.

Imagine that Client A and Client B have connected to the same WebTalk web page. This connects them automatically to the main server, which runs on a dedicated machine. In each page a VRML world is available, made of local and shared objects. The shared objects reside in the Zone. When the shared object 1 in Zone A generates an action (e.g. the user has clicked on it), it becomes the *pilot* of its related *drone* in Zone B. The actions are filtered via the Client, which sends the events along the network connection to the Server, which in turn is appointed to re-broadcast this event to all other Clients (that host the drones to the pilot which triggered the action). Avatars work in the very same way: whenever a user loads the web page in, a new connection is detected by the Server. This triggers the creation of a new avatar in each world sharing the connection. Every time an avatar is moved, information about its new position is distributed by the Server.

Third party customization is allowed only on the VRML portion of the system. Authoring is performed creating the geometries and the dynamics of the virtual space, and assigning to predefined objects in the world the "Shared" property. It is also possible to decide which of a predefined set of behaviors must be shared for an object: for example, any translation of the object must be shared, or any single click event of the object must be shared. The first step is generating the geometries by means of any off-the-shelf 3D design program, and export the final design to VRML code. It is on the code itself that modifications have to be performed in order to plug the WebTalk engine in. This is at present still done manually, by declaring the interested node as

Published by the Eurographics Association ISSN 1017-4656

Shared Object or as Zone to host the Avatar Movement.

2. Moving on to a new approach

The current prototype has been successfully utilized to deploy an interesting application, a 3D Virtual Museum which allows visitors to cooperate on the museum's exhibits (http://www.museoscienza.org). The use of VRML allowed for easy integration with current web technologies, via Cosmo Player, a standard VRML Viewer plug-in for Netscape and Explorer. While it is true that the use of Cosmo Player is spreading in the Cyber community, the external plugin gives no control in respect to the way the world is navigated and interacted with. The plugin is also an additional burden on the already heavy browser environment, and with future releases of Netscape and Explorer this issue is not getting any better. Moreover, the designer is currently forced to go hands-deep in the VRML code, which contains all geometry and behaviors definitions, and to define the shared nodes, potentially damaging the structure of the virtual world. Geometry and Behaviors are not well-separated concepts within this architecture.

By using the SharedObject notion, the designer is given a powerful but simple to grasp concept, which allows designing of 'active' areas in contrast with 'background' or 'local' areas, but with a scope limited to only two possible cases: perform the default behavior or do not perform it. This binds the system to a set of cooperation metaphors which are all slight variations to the same cooperation pattern: broadcast groups. Since the ability of the system is only to broadcast or not to broadcast, the only control given to the designer is limiting the scope of the broadcasting by creating different subgroups with different group-gathering mechanisms. The possibility of binding object behaviors and object distribution to time, context, data types, characteristics of the approaching user, and more in general to custom object properties would allow to implement and design more complex cooperation patterns and metaphors. This in turn implies a completely object-oriented structure both for active objects and avatars, which cannot be reached in a satisfactory way with the VRML strictly-hierarchical approach.

The possibility to open conventional HTML pages from the Virtual Space has given rise to another topic: the use of a 3D Virtual interface for information consumption. At present the dynamics of this information browsing is using the third dimension as a shell to 'flat information' (2D pages). While a design effort to create effective 3D shells to flat information can bring to interesting results with the current technology, a flaw is still present, that is cooperation stops as soon as flat information is accessed. Cooperation has to integrate seamlessly to 2D navigation, either expanding the current white-boarding cooperation metaphor, or integrating flat information within the 3D environment, by exploiting new metaphors which will turn information from 'flat' (2D) to 'spatial' (3D). A crystal example is a page listing of related links, which lay on the same 2D plane. By making spatial this flat information, it is possible at a glance to tell that more pertinent links are nearer to the avatar and fully illuminated, while less pertinent links are farther away and in the dark. This will imply a seamless interface to database engines, in which records are object oriented contents, provided with a geometry, a behavior, a default cooperation pattern, and flat content. This kind of deep integration between flat and spatial information, and the extension of cooperation patterns to conventional flat information browsing are issues not targeted by the current architecture.

By dropping the VRML paradigm, the system will lose an extremely consolidated and simple approach (the VRML modeling language), to gain in depth, complexity, and authoring flexibility. A Java 3D solution, while allowing for seamless integration between the graphical engine and the communications layer (thus avoiding the use of a third party plugin), will open the door to a truly object-oriented architecture, which will make possible providing the third-party designer to specially created API's for 3D management of shared objects and avatars, control on general communication, control of the navigation system within the world, and so forth.

3. Database-Oriented Dynamic Scene Creation

The key to a full-blown cooperative 3D engine is using a complete object-oriented paradigm laying its foundations into the previous client/server architecture. The new WEBTALK project, currently under development at the HOC Laboratory of Politecnico di Milano, Italy, envisions the possibility of integrating conventional web 2D and 3D information, with the possibility of creating dynamic environments, which are database and user aware, and in the meanwhile providing full flexibility for creation of third-party applications.

The first step is dividing the *Space Scene* from the objects. The *Space Scene* determines the spatial environment in which action takes place. For example, a cloister which mimics the interior of an italian museum. The objects are placed within the *Space Scene*, and can be kept local (interaction and effects of interaction are kept locally) or shared, as in the early WebTalk paradigm. However, thinking of a 3D object through the OO paradigm, allows to attach to each object *Properties* and *Methods*.

Standard properties of each object are: its *geometry*; this defines the shape, the color, the material of the object, and can be a portion of VRML code, a Java 3D geometry class, a Lightwave object, and so forth; its *behavior*, this defines the interface method between user input device and the object itself. For example links a particular action to the left-click event, and another one to the right-click event; its *animation pattern*; defines the set of movements the object might perform when appropriate behavior is triggered; its *internal logic;* a source code (a Java or TCL/TK code segment) describing conditions, and if-then branches which can control the appearance or effects that user interaction has on the object

The object is also linked to a *Data Set*. The *Data Set* contains: the *internal memory* of the object. This describes the current history or status of the object; the *flat data*. This contains any kind of flat information, e.g. a text, an image, a hyperlink to an html resource; the *3D link.* This contains a direct link to another 3D object. This link can change in time or upon certain conditions, and thus it is treated like a member of the *Data Set*.

Properties and Data Set can be accessed and modified by means of *Methods.* Standard methods will be provided along with the objects, but it could be possible to make use of the inheritance concept of OO programming, allowing the creation of third party methods based on the core ones. The whole object will be stored inside a Data Base System, which will be dynamically accessed during navigation of the environment.

Under this architecture, authoring of a world will go far beyond geometry drawing with 3D Studio Max. The creation of a WebTalk Virtual Environment will consist of several different steps: designing of the 3D and 2D contents, with the definition of the link and access structures between 2D objects, 3D objects and 2D to 3D objects; the conventional drawing of the objects; the choice of the correct behaviors for each object; the programming of the logic of each object; the creation of object animations when applicable.

Once the virtual hyperbase is created, the *Space Scene* geometry will have to be built, and then linked to an *Access Paradigm* for the *Scene Builder*. The *Scene Builder* takes care of merging the elements stored in the database with the *Space Scene*. The most straightforward way to do this is by defining fixed spatial coordinates to position each object in relation to the space scene. Besides, not always physical space is the more efficient *access paradigm* to a set of data. One could devise a completely different approach for providing access to virtual content, even dramatically different than the notion of a predefined collection of rooms, linked one to the other, containing objects. In this case, it could be possible to write an *Access Paradigm*, a chunk of Java or TCL/TK code which describes the criteria with which objects are to be dynamically loaded and accessed within the *Space Scene*. One immediate application of the availability of an *access paradigm* is to build and decide on the fly the positioning and disposition of 3D objects within the *Space Scene*, for example by dynamically placing in full light and near the avatar the objects belonging to the same conceptual area the user is exploring, and removing objects not related to the criteria currently in use. An example could be a virtual museum which dynamically changes its floorplan to provide the virtual user with the ideal touring path in its contents, according to the user's interests (which could be inferred by the choices made during navigation). In this example, the exhibits will be loaded from the 3D Resources via a JDBC layer (Java's Data Base management APIs), and the Scene Builder will add the geometries to the *Space Scene* according to the program instructions detailed in the *Access Paradigm* block. Each object loaded via the access paradigm is therefore offered for information consumption: this will happen via the interfaces offered to the users by means of the *behavior*, under the conditions set by the object *logic*, and giving as result the information stored in the *data set*. If the *Data Set* points to 2D HTML information, navigation in the web page will be performed in a *cooperative way*, that is a common access to the flat information that will allow for common reading, discussion and activity within the web page.

References

1. Rohel B., et al., *Late Night VRML 2.0 with Java*., Ziff-Davis Press, Emeryville, 1997.
2. Paolini P., Barbieri T., et al. - *Visiting a Museum Together - How To Share a Visit to a Virtual World* in Proc. MUSEUMS & THE WEB 99, New Orleans, 1999
3. Garzotto F., Paolini P., et al. *HDM - A Model Based Approach to Hypertext Application Design* in ACM Trans. On Information Systems, Vol. 11, 1993