7. International Immersive Projection Technologies Workshop
9. Eurographics Workshop on Virtual Environments (2003)
J. Deisinger, A. Kunz (Editors)

Open ActiveWrl - A Middleware Based Software Development Toolkit for Immersive VE Systems

C. Winkelholz T. Alexander and M. Weiß

Research Institute for Communication, Information Processing, and Ergonomics,Wachtberg, GERMANY

Abstract

A Virtual Environment (VE) is not an application in itself. It should further be seen as a complex, multi modal human-computer interface. Immersive VE systems must be able to couple resources, most likely distributed in a cluster, into a parallel, real-time simulation. The shortcoming of most VE software libraries is the inability to provide an environment that satisfies the three main requirements in the implementation of a VE system, namely the design of the VE interface, the implementation of the actual application and the distribution of parallel tasks to appropriate resources within a cluster system. This paper presents Open ActiveWrl, an open source software project that tackles these problems by a generic software development approach. It is tailored for heterogenous cluster systems. Open ActiveWrl uses a general-purpose middleware that interprets VRML/X3D as an interface definition language.

Categories and Subject Descriptors (according to ACM CCS): D.1.3 [Programming Techniques]: Concurrent Programming D.2.11 [Software Engineering]: Software Architectures D.2.12 [Software Engineering]: Interoperability D.2.13 [Software Engineering]: Reusable Software I.3.3 [Computer Graphics]: Graphics Systems

1. Introduction

Using cluster systems for the realization of immersive VE systems has become common. For immersive VE systems it is essential that all the rendering and interaction tasks use the very same model for the virtual environment. Because of limited bandwidth and latency in the network the concept of a shared scene-graph is widely used for VE cluster systems. Local copies of the model are available in each component, and each frame the models are synchronized.

Open ActiveWrl can be seen as a software development toolkit that allows implementing parallel immersive VRML/X3D [2] browsers. Each VRML/X3D browser consists of different components that are reusable in different setups. The components are linked together via the Cluster Event Broker Architecture (CEBA) [1] middleware, which we designed and implemented for this purpose. The computation of any script-node of a VE can be assigned to a dedicated workstation in the cluster. This helps to increase system performance and facilitates a generic software development approach for the integration of foreign software components into the parallel simulation. The middleware allows embedding script-nodes into a foreign framework similar to CORBA-objects.

2. Middleware Based Approach

The challenge of parallel and distributed systems is to design an appropriate development environment. To achieve cross-platform interoperability of distributed software-components, the separation of interface and implementation of an object proved to be useful. The interfaces are defined by a specific interface definition language (IDL). An IDL-compiler generates skeleton-code for the implementation of the object itself and stub-code for proxy-objects that can be used in other applications to have access on the services provided by the implemented objects. Instances of the implemented objects can reside in arbitrary applications. A proxy can be assigned to the instance of an object with an appropriate interface at runtime. In addition to enabling interoperability, this generic software development approach produces components, which easily can be reused in different applications. Microsoft ActiveX and CORBA are well known implementations of this concept.

The aim of Open ActiveWrl is to transfer the advantages

of these established approaches to parallel real time virtual environment systems on heterogenous clusters. In contrast to these established approaches VE cluster systems demand low latency and needs to distribute the same data from one component to multiple components. ActiveX or middleware based on CORBA uses point-to-point junctions. This make them unsuitable for data broadcasts needed in VE systems. Further they do not include buffering mechanisms to reduce number of high latency communication calls. Therefore, one main part of the Open ActiveWrl project is the implementation of the Cluster Event Broker Architecture middleware.

3. Event Execution Model

A useful characteristic of VE systems is that the behavior of the VE regarded as an advanced graphical human-computer interface can be implemented in an event execution model. In this case the synchronization of the VE models can be achieved by distributing the events appropriately. The easiest way is to distribute the resulting events of the physical input devices like it is done by the current implementations of most VE cluster systems. However, this is very inflexible and inappropriate in the following situations:

1. A task has to compute a more complex behavior.
2. A task needs resources that only a dedicated computer can provide.
3. A task links the virtual environment to an external application.

In the ladder case it would be inappropriate to execute such a task on all computers of the cluster, because the external application would have to communicate with all components in the clusters.

The requirement to execute some tasks of event processing on a dedicated workstation in the cluster, under the condition to keep the replicated models synchronized, suggests to distinguish three different roles of an object: active, passive, and neutral. The computations of the methods of neutral objects are performed on each component. The computation of an active object is performed only in a single dedicated component. The corresponding objects of an active object in the other components are passive. A passive object serves as a proxy to receive the resulting events of the corresponding active object to behave as the corresponding active object. In this way the several distributed instantiated models are kept synchronized throughout the simulation. The fact that a VE simulation is scheduled in frames is exploited to amalgamate events in each component and to optimize exchange of data.

The decision, which object is active on a dedicated workstation, does not need to be made at time of compilation. This allows easy adaptation of an implementation of a VE to different runtime architectures.

4. Levels of System Development

Open ActiveWrl supports implementing VE systems on different levels of the development process. At the base level a class library provides the developer with classes facilitating the implementation of different kinds of components like display components, physical sensor components or application specific components. Several instances of a component can be used in one single system setup.

At the application level of the implementation the content of the virtual environment is described in VRML/X3D. In general, once the system has been setup any VRML/X3D-file can be browsed and the system serves as an immersive VRML/X3D-browser. Proto libraries including menus and controls facilitate the design of a appropriate interface. In terms of a generic software development approach an IDL compiler can be used to generate C++-skeleton code from the declaration of VRML/X3D script nodes, that allows to embed the implemented script-node in a foreign component similar to a CORBA-object.

Further, Open ActiveWrl provides generic creation of skeleton code for the implementation of native nodes. Native nodes are unavoidable if special treatment in the traversals is needed and therefore these nodes can't be implemented as VRML/X3D-prototypes. In this way the framework can easily be extended with new kinds of, e.g. appearance nodes, like environment mapping, etc.

Open ActiveWrl allows implementing different interaction techniques in the form of VRML/X3D-PROTOs by extending the interface of script-nodes to the runtime environment by some intersection services and the facility to trigger events of sensor nodes. The advantage of this approach is that the behavior of the interaction device can be influenced on the application level of implementation.

5. Conclusion

In this short paper the main concepts of Open ActiveWrl, a development toolkit for the implementation of immersive parallel VE cluster systems, was sketched. At our institute Open ActiveWrl is used in several different setups, like Workbenches, parallel rendered projection-screens, and multi-user collaborative augmented reality systems. Open ActiveWrl is published under GPL and is downloadable at http://open-activewrl.sourceforge.net/. It is implemented for Windows NT/2000/XP and Linux platforms.

References

1. C. Winkelholz and T. Alexander. Approach for software development of parallel real-time VE systems on heterogenous cluster. In *Eurographics Workshop Proceedings on Parallel Graphics and Visualisation 2002*, pp. 23-32, ACM Press.
2. http://www.web3d.org