Web-Based 3D Real Time Experimentation

C. C. Ko
National University of Singapore, Singapore

Ben M. Chen
National University of Singapore, Singapore

C. D. Cheng
NDI Automation Pte Ltd, Singapore

INTRODUCTION

Spurred by development in computer science and network technology, the use of the Internet has been expanding exponentially. It is now extensively used as a connectivity and reference tool for numerous commercial, personal, and educational purposes. In education, the Internet opens a variety of new avenues and methodologies for enhancing the experience of learning as well as expanding educational opportunities for a larger pool of students. Specifically, distance education and non-traditional classrooms have the capability to reach more students using specialized instruction and self-paced learning.

In the area of distance education, many **Web-based real time experimentation** systems have been reported in the literature (Ando, Graziani, & Pitrone, 2003; Daponte, Grimaldi, & Marinov, 2002; Ko, Chen, Chen et al., 2000; Ko et al., 2001; Kumar, Sridharan, & Srinivasan, 2002; Yeung & Huang, 2003). These Internet-based remote laboratories allow users or students to carry out physical experimental work at their own pace anytime anywhere. They generally require very little physical space and minimal manpower to maintain, and are ideal for the sharing of expensive equipment. However, all these experimental systems can only provide 2D operation panels. Due to this limitation, the actual shapes of 3D instruments and equipment, some of which may have controls or display components on different sides, may not be possible to be reflected on the remote user's client display window.

BACKGROUND

Although many 3D visualization schemes on the client side have been presented (Geroimenko & Geroimenko, 2000; Hobona, James, & Fairbairn, 2006; Nakano, Sato, Matsuo, & Ishimasa, 2000; Oellien, Ihlenfeldt, & Gasteiger, 2005; Osawa, Asai, Takase, & Saito, 2001; Ueda, 2006; Vormoor, 2001) and some additional collaborative functions have been proposed for communication amongst multiple remote users or between client and server (Bender, Klein, Disch, & Ebert, 2000; Engel, Hastreiter, Tomandl, et al., 2000; Nielsen, 2006; Zhuang, Chen, & Venter, 2000), applications and issues such as Web-based real time control and 3D-based monitoring have not been addressed. We present in this article the development of **Web-based 3D real time experimentation** using Java 3D visualization tools.

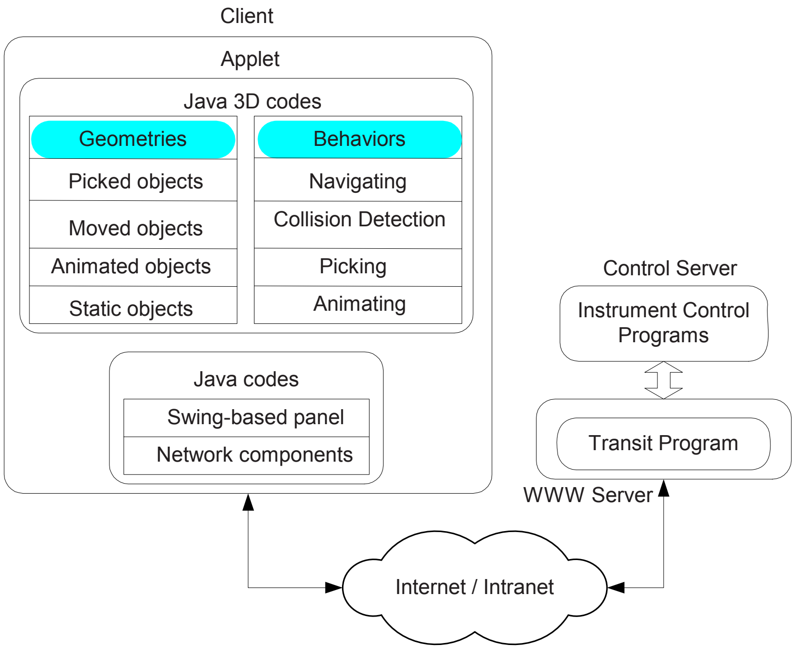
Among the various tools available, **Java 3D** is ideal from certain perspectives. Specifically, **Java 3D** is an efficient tool that provides a very flexible platform for building a wide range of Web-based three-dimensional graphics applications, and is becoming one of the most attractive tools for creating 3D user interfaces, **3D visualizations** and virtual environments. It provides not only strong **3D programming** but also excellent integration with previous version of Java components.

In comparison with other 3D virtual experimental systems, this chapter attempts to address all the important issues with an emphasis to provide a complete solution. Specifically, issues on connecting actual experimental instruments, real time data transmission, three-dimensional virtual scene and three-dimensional behaviors are addressed. These ensure that the user will get a more realistic feeling when operating and controlling three-dimensioanl experimental instruments as well as monitoring actual experimental results without any significant delay.

PROPOSED SYSTEM REFERENCE MODEL

Figure 1 shows our reference model for the creation of **Web-based, 3D, real-time experimentation** using Java 3D visualization tool. Note that on the client side, the combination of Java 3D API and Java realizes 3D visualization and network connection. Usually, 3D visualization consists of geometry and **behavior objects**. The former includes the picked, moved, animated and static objects, and the lat-

Figure 1. System reference model for creating Web-based, real-time experimentation



ter consists of navigating, collision detection, picking and animating behaviors.

The picked objects cover all controls such as buttons, knobs, sliders and connectors of the virtual equipment and experiment, while the moved objects include curve, text and screen displays on the **virtual instruments**. The animated objects cover all active and periodically moved objects. The other visual objects, such as walls, windows, tables and certain components of some **virtual instruments**, are taken to be static ones. These **geometry objects** designed using Java 3D helps to promote the rendering efficiency of the 3D virtual scene. In the relevant behavior objects, only the animating behaviors provided by Java 3D API are used without modifications.

PROPOSED HARDWARE ARCHITECTURE

The system reference model of Figure 1 can be supported by the double-server-client distributed hardware architecture of Figure 2. The whole system includes a user's computer on the client side, the Internet and/or an intranet to transmit command and data, a Web server to host the Web site of the remote experimentation, and a control server with control cards attached to programmable instruments together with some circuit boards.

In particular, programmable instruments have to be connected to the control server through control cards and cables in a 3D remote experimentation system. For example, two separate TCP/IP interface modules are used for real time control and retrieval. The commands coming in through the TCP/IP control interface are converted into the format required before being sent to the programmable instrument to be controlled. Experimental data for the generation of real time curve or text for the user is transmitted to the client through the TCP/IP retrieval module.

3D INSTRUMENTS AND SCENE

To be as realistic as possible, and to overcome certain limitation posed by 2-D operation panels, using which the actual shapes of **3D instruments** and equipment cannot be shown, the use of 3D visualization tools in real time Web-based experimentation may be considered. Figure 3 shows a typical example GUI realization on the client computer developed based on Java 3D.

Ideally, anyone conducting an experiment through the Internet should be able to do it in the same manner as in a real laboratory. This can be accomplished in a 3D environment through three behavior modules on navigating, collision detection and picking in the GUI interface.