ISSN 1064-2307, Journal of Computer and Systems Sciences International, 2017, Vol. 56, No. 6, pp. 937–944. © Pleiades Publishing, Ltd., 2017.
Original Russian Text © V.V. Gribova, L.A. Fedorishchev, 2017, published in Izvestiya Akademii Nauk, Teoriya i Sistemy Upravleniya, 2017, No. 6, pp. 41–49.

COMPUTER METHODS

Software Tool for Virtual Environment Development Based on Cloud Platform

V. V. Gribova* and L. A. Fedorishchev**

Institute of Automation and Control Processes, Far Eastern Branch, Russian Academy of Sciences, Vladivostok, Russia

**e-mail: gribova@iacp.dvo.ru*

***e-mail: fleo1987@mail.ru*

Received May 13, 2015; in final form, July 11, 2017

Abstract—This paper considers a concept for developing and managing professional virtual environments using a software tool. We propose managing the logical model of a virtual environment by a new extended ontology. We also introduce a control technology for 3*D* scenes based on a virtual environment model representation. Finally, we describe a mechanism to create and manage specific handling functions of virtual environments and also to interpret and launch a virtual environment with integrity checking.

DOI: 10.1134/S1064230717060065

INTRODUCTION

Nowadays, intelligent systems are being actively used around the world to solve different professional tasks. A special place among them belongs to intelligent systems with virtual reality (virtual environments), which are primarily intended for designing simulators and computer-based trainers; such complexes allow us to train motor skills and assimilate knowledge in virtual reality, to conduct secure virtual experiments, etc. However, different difficulties are encountered [1] in the course of developing virtual environments using systems of this class. Forming a considerable part of such a system, expert knowledge is organized as a separate component in accordance with the technology of designing intelligent systems, developed, and maintained by domain experts. In the virtual reality systems designed using the virtual reality implementation tools currently available in the market, experts have limited authority and act merely as consultants. We sign out or note the following main reasons for this: development tools are programmer-oriented; knowledge is described in a programming language (i.e., the description depends on development tools) and is embedded in the program code. In addition, note that the development teams of such systems include a wide range of specialists who are often geographically dispersed, which also imposes constraints and causes difficulties for the development process. Due to the growing need for systems with virtual environments intended for solving professional tasks in a certain subject area and training specialists, we created an internet-based development of virtual interactive environments (IRVIS) [2], a software tool to design intelligent virtual reality systems. This tool has the following main features: (1) expert knowledge is organized as a separate component developed and maintained by domain experts in a given subject area using a structural editor managed by a virtual environment ontology; (2) collective development is supported by implementing the software tool as a set of cloud services; (3) all components of the software tool have a declarative description with further interpretation of the declarative model; (4) additional agents can be connected to the interpreter to extend its functionality; (5) the intelligent virtual reality system itself is also implemented as a cloud service.

The aim of this paper is to analyze the suggested implementation solutions of the software tool and compare it with the existing analogs, as well as to describe the applications developed using this tool.

1. EXISTING SOLUTIONS

Presently, numerous algorithms and models are available to develop and maintain virtual environments [3, 4]. There exist different special-purpose and universal development tools, application software packages, and libraries to develop general-purpose virtual environments, such as Del'fin, ToolBook, Lectora, CAVE, WorldToolKit, Avango, Lightning, Juggler, Unity3D, Virtools, Alternativa3D, and Flare3D. The market also offers special-purpose game projects (Minecraft, BlockByBlock, and Sims, to name a few), which cannot be considered as development tools for professional virtual environments: they are limited to the game space and a finite set of game objects. However, in such game projects, users design their own virtual reality directly inside it.

Several factors affect the value and applicability of the development tools and technologies for virtual environments. Almost all the development tools considered demonstrate good results in computer-generated graphics, hardware device usage, and interfacing. Besides the above-mentioned factors, the specifics of professional virtual environments (PVEs) impose the following fundamental requirements on development tools: orientation to domain experts in a given subject area; simplified development, maintenance, and deployment; cloud organization (for a wider applicability of PVEs, the absence of special demands on the technical characteristics of computers, operating systems, and the installation process). In turn, expert-oriented development tools have to support a technology to organize expert knowledge as a separate component; i.e., experts are provided with a convenient and intuitively clear development environment. Here we comprehend a development environment as an environment that renders terminological and tools-oriented support for the creation, modification, and extension of expert knowledge.

Engagement of experts in the development process. The development tools available in the market are not oriented toward domain experts in subject areas. For example, VRSpace, Vrui VR Toolkit, ViSTA, dVISE, and Alternativa3D are intended for programmers only, since these development tools contain script editors (console editors); as a result, experts and designers (potential consultants of programmers) do not fully participate in the development process. In other development tools with powerful integrated editors (such as Vizard Virtual Reality Toolkit, Unity3D, and Flare3D), it is possible to separate the work by type: designers can independently edit scene representations, the models of objects, etc. However, in these tools, the models (the graphical images of objects in a scene), scripts used to control them, and scenario logic (implemented by a programmer) for the events occurring in the scene all have strong relations with each other and are often inseparable from each other. Language tools have no generally accepted standards, and therefore development tools use different languages for the scripts of program logic. For example, in the Vizard Virtual Reality Toolkit, the scripts are written in Python; in VRSpace, in Java and JavaScript; in the Vrui VR Toolkit and ViSTA, in C++; in Unity3D, in C# and JavaScript; and in Alternativa3D and Flare3D, in AS3. In other words, all these development tools are programmer-oriented, and users without the required IT background cannot create appropriate logic independently.

Graphics and Internet. Work with 3D graphics is an important element of any virtual environment. In this context, a new crucial requirement is to organize work through the Internet, allowing virtual environments to operate as cloud services. At the same time, good computer-generated graphics involve powerful hardware resources of a computer that are available at the local level only (i.e., directly on the user's computer). The existing development tools do not reproduce the same quality of 3D graphics on the Internet, since they were mostly designed as desktop development tools. Thus, as of today, there are no development tools for virtual environments with the feature of cloud work. At the time of our analysis, only a few technologies (Flare3D, Unity3D, and some others) implemented the necessary 3D graphic functionality, without, however, representing full-fledged development tools for virtual environments.

Consequently, we may draw the following conclusion on the strength of a thorough analysis of the existing development tools. The modern market does not offer full-fledged development tools for virtual environments that satisfy the new requirements.

2. METHODS AND TECHNOLOGIES OF IMPLEMENTATION

As a solution of the first posed problem (an expert-oriented development), we proposed a declarative approach based on ontologies. According to this approach, virtual environments are first described by domain experts in the given subject areas as declarative models in terms of the original formalism, i.e., the universal ontology of virtual environments [5] (see Fig. 1). Therefore, the development process is distributed among different specialists, namely, experts, designers, and programmers. The main description of the logic of virtual environments is given by domain experts in a given subject area (not programmers), which is natural as the former have complete knowledge of the tasks to be solved using virtual reality means.

As a solution of the second problem (cloud organization, both for development tools and PVE itself), we suggested and implemented a software tool based on the IACPaaS cloud platform [6].

To provide the PVE with high-quality 3D graphics on a cloud platform, we considered different technologies available in the market, such as DirectX, OpenGL, Unity3D, Flash, and Silverlight.

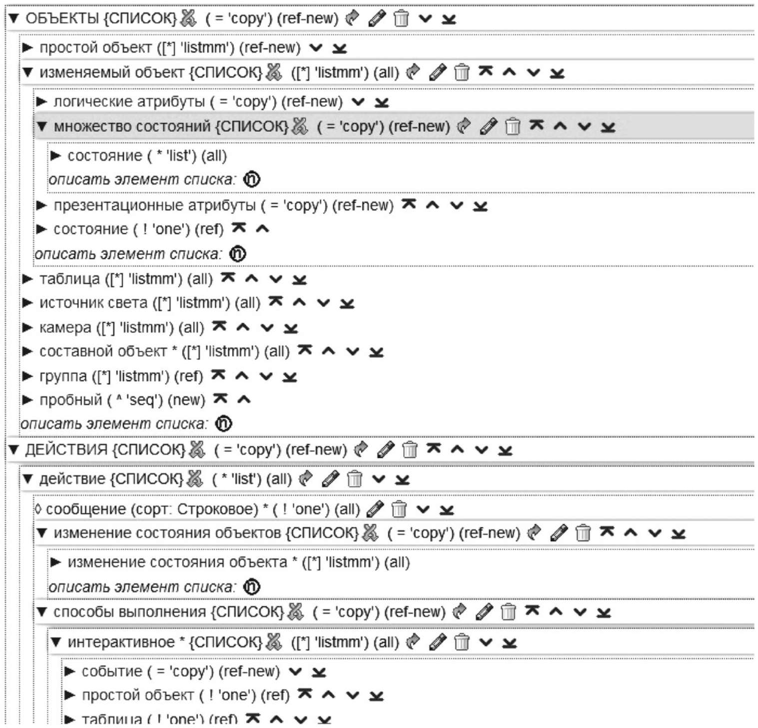

Fig. 1. Ontology of virtual environments.

The technologies based on DirectX and OpenGL have a good performance level and graphics quality, as well as good support and documentation. However, despite the existing mechanisms to use them in a browser (particularly, through NaCl—Native Client), these technologies are mostly intended for developing standard desktop clients in C++, C#, and Java (i.e., not for web clients). In addition, they have complex coding in the above-mentioned languages.

Unity3D also demonstrates a good performance level and quality of graphics and, moreover, is dedicated to developing web applications (it can be a web client). However, Unity3D requires modular programming oriented to computer games.

The WebGL standard, also based on OpenGL, which is a software library for JavaScript to create interactive 3D graphics, also exists. Due to the low-level OpenGL means of support, a part of the WebGL code can be executed directly on graphics cards, which results in good performance. However, this standard is now being developed and, at the time when IRVIS appeared, was in embryo without rich (software and hardware) functionality. Recently, WebGL has been considerably improved. Perhaps, in future it will become a leading technology in the field.

Flash is a multimedia development platform for web applications created by Adobe. The operation of Flash requires a Flash Player, a virtual machine that executes a Flash code from the Internet.

The Flash technology has the following advantages.

First, the Flash plug-in is installed by an overwhelming majority of web users (over 90%). This means Flash applications are easily available and expandable.

Second, the technologies of raster and vector graphics are combined, which appreciably reduces the graphical memory required for Flash applications.

Third, Flash provides a powerful platform, a programming language, and rich functionality to develop web applications.

Fourth, this technology fully supports hardware-accelerated computer-generated 3D graphics.

And fifth, there are different freeware development tools for Flash applications, including libraries, editors, and frameworks.

Consequently, using the above analysis we chose Flash to implement the web client of the system. Note that the Flash-based implementation is independent of the main system of virtual environments and therefore can be replaced by another technology in future (if necessary).

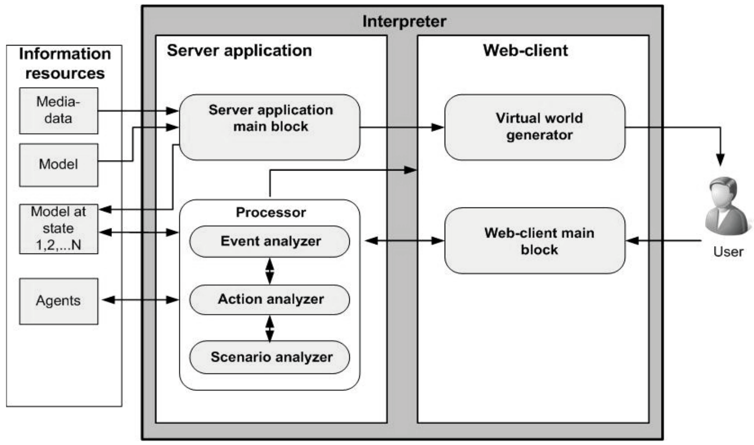

Fig. 2. Client-server interaction model of virtual environment.

3. IMPLEMENTATION OF SOFTWARE TOOL

The software suite is implemented based on the client-server architecture. The client can be a computer at any location worldwide with a preinstalled Web browser and a Flash plug-in. The server is implemented using the multiagent technology and placed in the IACPaaS cloud platform.

The load is distributed between the client and server according to the tasks they perform. Complicated hardware calculations required to display high-quality computer-generated 3D graphics are implemented on the client. The server executes logic, i.e., defines the behavioral scenarios of users in a virtual environment according to the interactive information acquired from the client and modifies the current declarative information about the virtual environment.

The interaction between the client and server is implemented through permanent asynchronous messaging, see Fig. 2.

The main interactive functionality on the communication with the virtual environment is available to a user on the client, since scene rendering and manipulator event handling (mouse, keyboard) are implemented for the client. Such an organization provides almost synchronous interaction of a user with the environment in real time. Information about any action of a user comes to the server in the asynchronous mode without “freezing” the actions of other users. In most cases, the server’s reply does need to be synchronized.

Data upload optimization. An important task that arises in the course of developing a virtual environment for the client using the information acquired from the server concerns uploading bulky graphical information, which is stored on the server of the platform. We adopted the following solution to optimize this task. During the primary processing of information about the scene, all the data about the required objects are accumulated in a special array. Using these arranged data, a unified array is then constructed on the server that contains the above-mentioned graphical data in the form of binary files. Finally, the unified array is archived and sent to the client as a unique reply.

Safety. The IRVIS software tool is placed in a cloud platform, thereby enjoying all the advantages of safety, authorization (identification), and reliability provided by this platform. The IACPaaS platform is designed so that all users and developers have different rights depending on their authority in a given subject area. In the issue of server programming, the platform also takes the responsibility for safety by (1) providing a special API that encapsulates unsafe functions and (2) by filtering the connected external library scripts. The reliability of IRVIS is also guaranteed by the reliability of the cloud platform itself, as the latter backs up data (at least on a regular basis).

4. COMPARISON WITH ANALOGS

We compared the software tool with similar solutions in terms of the following criteria, see Table 1.

Adjustment and deployment. The software suite is based on the IACPaaS cloud platform and available through the Internet. Thus, the system does not require complicated adjustment and deployment, in contrast to the majority of the development tools for virtual environments.

Table 1. Design of professional virtual environments: comparison of existing approaches

Criterion	IRVIS software suite	Other development tools of general-purpose virtual environments
Adjustment and deployment	Virtual environments act as services of IACPaaS platform. No need to adjust or install any components.	Virtual environments are represented by separate systems or exe-files, which require adjustment and deployment by users.
Maintenance and management	Cloud organization of platform and interpretation of virtual environments guarantee simple maintenance and management.	Any modification of virtual environments requires recompilation and system reinstallation by users.
Engagement of experts into development process	Experts in given subject area directly participate in development process, creating declarative model based on expert knowledge without assistance from programmers.	Experts in domain experts subject area may act merely as consultants for programmers.
Separation of works by specialists	Development process is distributed between experts and designers using appropriate two-level organization of ontology. Experts work in structural editor, while designers in graphics editor.	Experts can work only together with other specialists. Designers cannot create graphical representation of scene without programmers, who construct scene by coding using models developed by designers.
Availability through Internet	Due to cloud organization of platform, professional virtual environments are available through Internet as Web services.	Availability through Internet requires additional work with web servers, corresponding scripts, sessions, etc.
Technologies of modern computer-generated graphics	Client module, graphics editor and interpreter are implemented using Flash technology. This technology outputs 3D graphics through graphics card with hardware acceleration, thereby providing modern level of computer-generated graphics.	High-quality computer-generated graphics is achieved using modern graphical libraries based on DirectX and OpenGL.

Maintenance and management. Due to the organization of the cloud and the development means in the form of web services, programmers, designers, and experts can maintain virtual environments remotely regardless of their location. The declarative approach realized in the system allows us to manage the development process by introducing permanent modifications and refinements in the model of a virtual environment without the need for recompiling and uploading to the server. Other systems are highly complex to maintain due to traditional software (instead of web services), the need for recompilation, and complicated software updates for users.

Engagement of experts in development process. The software tool engages domain experts in the given subject areas in the development process of virtual environments. Without the assistance of knowledge engineers and programmers, experts are able to define the logical content of the program, in contrast to the majority of the existing systems, where experts act merely as consultants.

Separation of work by specialists. The software suite separates developers and is mostly oriented toward experts in the given subject areas and designers. At the same time, most of the existing analogs are programmer-oriented.

Availability through Internet. The software suite is available to anyone (developer and users) through the IACPaaS cloud platform on the Internet. The majority of other systems are implemented as traditional desktop software suites, which complicates the access to programs.

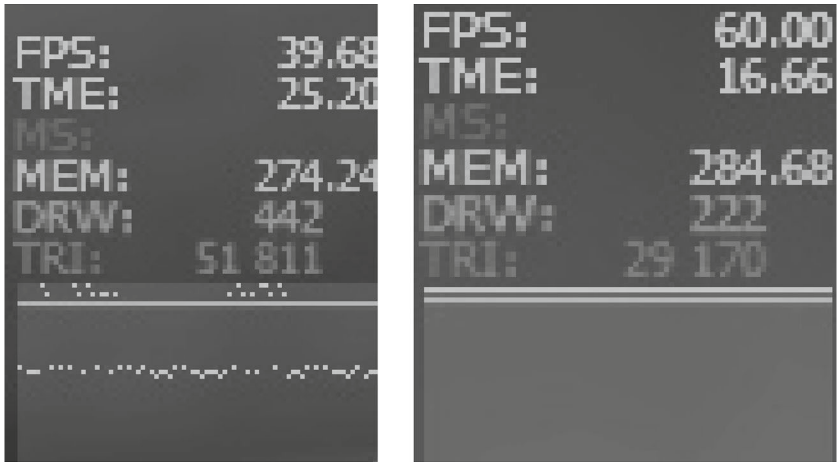

Fig. 3. Imaging speed measurements.

Technologies of modern computer-generated graphics. The software suite provides high-quality computer-generated 3D graphics (available through a web browser) being as good as the graphics of the complex desktop software solutions based on DirectX and OpenGL.

5. TECHNICAL SPECIFICATIONS OF SOFTWARE SUITE

We tested the software suite on a computer with an NVIDIA GeForce 210 graphics card and obtained the following results. The imaging speed of the graphical data in the cloud environment is highly competitive with that of traditional imaging due to the direct use of a graphics card through the Flash technology. As established, it is possible to image up to 100000 polygons at a rate of 30 FPS and higher. In practice, the graphical engine demonstrated a rate of 30–60 FPS in an interactive scene that varies each 30 ms, with lighting, shades, physics, programmed logic, and objects containing about 50000 polygons in a frame (and many more polygons in the whole scene). The RAM space required for such a scene varied within the range 250–300 Mb. Figure 3 shows the corresponding measurements for the given example of a complex interactive scene.

This figure has the following abbreviations: FPS as the imaging speed in frames per second; RAM as random access memory; and TRI as the number of polygons (triangles) imaged in a frame at the current time.

The upload rate of the graphical data and the speed of their use for virtual scene visualization were increased so that the internet connection ceased to be a restricting factor. This was achieved by optimizing web requests and a massive load of resources with minimal flows. As demonstrated by the testing, large amounts of data are uploaded through the network much faster if sent by a single large flow instead of several small ones. In addition, a large amount of data is compressed at higher rates. As a result, data archiving is more efficient and hence the same data requires less space during transmission through the network. Nevertheless, placing all possible resources in a virtual environment is unreasonable, since many resources are used on request or may not be used at all; in other words, the demand for resources depends on the interaction between a user and the environment. Therefore, at the initialization phase of the graphical system, the software suite defines the necessary resources by a declarative description at the start time, sending a unified request for them to the server. The resources required at the subsequent times are provided asynchronously using small requests.

The virtual environments developed in the IRVIS software tool allow uploading and visualizing the modern 3D models created in professional graphics editors (3dsmax, Maya, and Blender, to name a few). The graphics editor embedded in IRVIS allows designing complex dynamical scenes with the uploaded professional 3D models, and using and adjusting animation in them by the key frames for the corresponding logical states of the scene objects. The built-in services of the software suite (editor, declarative model interpreter) visualize a set of complex objects consisting of many polygons. Thus, these services can be used to design scenes of different levels and dimensions, from small virtual rooms to large urban open spaces. It is possible to upload 3D models of almost arbitrary complexity containing from 1 to 100000 polygons. The geometrical dimensions of the models are limited only by the real number range of a computer.

Fig. 4. Virtual simulator based on Golovin–Sivtsev table.

6. SOFTWARE TOOL

The IRVIS software tool was used to develop the following virtual environments: a computer-based simulator for ophthalmology, a virtual chemical laboratory, a model of transportation flows in a developing urban infrastructure.

The computer-based simulator [7] includes tasks on classical ophthalmological analysis methods such as eidoptometry using Golovin–Sivtsev tables and Polyak's optotypes; clinical refraction by the objective method (retinoscopy); and visual field analysis by campimetry (see Fig. 4). In each training task, the student has to demonstrate knowledge of an appropriate analysis method and its technology, identifying a required attribute by the behavior of a virtual patient.

The virtual chemical laboratory is intended for safe chemical experiments on school courses in chemistry. This educational service visualizes a set of standard chemical reactions on mixing and heating chemical elements, both with the correct and incorrect actions of a schoolchild. Mixing different chemical elements in the virtual environment, a schoolchild observes the result. In the case of mistake, e.g., an explosion occurs. A real mistake would cause traumas but the virtual environment guarantees a safe training process.

The main purpose of the interactive modeling system of transportation flows is a timely load estimation for the elements of an urban road network under different modifications and variations in the arrangement and purpose of urban objects. Transportation flows are predicted using a mathematical model constructed by integrating a gravitational descriptive model of the correspondence matrix and an equilibrium distribution model of the transportation flows. The virtual environment is intended for the interactive modeling of different characteristics of the urban road network and infrastructure, namely, varying capacities and traffic directions of the existing roads, the creation of park and ride facilities, prevents entry into a certain area, and the arrangement of infrastructural objects (construction and demolition of residential complexes, the objects of industrial, commercial, financial, cultural and social spheres of activity, etc. (see Fig. 5)). A prototype of this system has been developed and its further implementation is in progress.

Fig. 5. Model of transportation flows in urban district.

CONCLUSIONS

In this paper, we have described a fundamentally new approach to develop virtual environments running in the Internet. A special place in such software systems has been assigned to domain experts in a given subject area, who represent full-fledged developers of the systems. Computer-generated graphics provides a visualization of the virtual environment in the Internet. The approach proposed in the paper solves the two key visualization problems of cloud virtual environments: it reduces the excessive complexity of the development process and provides support in the web space.

ACKNOWLEDGMENTS

This work was supported in part by the Russian Foundation for Basic Research (project nos. 16-07-00340, 15-07-03193).

REFERENCES

1. A. V. Trukhin, "The analysis of computer training systems developed in Russian Federation," Otkryt. Distants. Obrazov., No. 1, 32–39 (2008).
2. V. V. Gribova and L. A. Fedorishchev, "Software package for cloud virtual environments development," Program. Produkty Sist., No. 2, 60–64 (2015).
3. M. E. Zubov, "Mathematical and software of new technologies of virtual training systems design," Extended Abstract of Cand. Sci. (Tech. Sci.) Dissertation (Moscow, 2003).
4. A. M. Il'in, "Mathematical and software design technology of environment generation for training systems," Extended Abstract of Cand. Sci. (Tech. Sci.) Dissertation (Tula, 2008).
5. V. V. Gribova and L. A. Fedorishchev, "Ontologies for design and generation of professional virtual environment," in *Proceedings of the All-Russia Conference with International Participation on Knowledge—Ontologies—Theories (ZONT-2015), Novosibirisk, 2015,* Vol. 1, pp. 68–76.
6. V. V. Gribova, A. S. Kleshchev, D. A. Krylov, F. M. Moskalenko, S. V. Smagin, V. A. Timchenko, M. B. Tyutyunnik, and E. A. Shalfeeva, "IACPaaS Project. Complex for intelligent systems based on cloud calculations," Iskusstv. Intellekt Prinyat. Reshen., No. 1, 27–35 (2011).
7. V. V. Gribova, M. V. Petryaeva, and L. A. Fedorischev, "Using artificial intelligence methods and 3D graphics for implementation a computer simulator for ophthalmology," in *Proceedings of the International Conference on Advanced Educational Technology and Information Engineering AETIE-2015* (Destech Publ., Beijing, 2015), pp. 567–573.

Translated by A. Mazurov