Intelligent Web Agents for a 3D Virtual Community

Tushar M. DAVE
Yan-Qing ZHANG
G. SCOTT OWEN
Raj SUNDERRAMAN

Keywords

- intelligent agents
- digital library
- graphics and visualisation
- expert systems
- electronic education
- virtual reality
- computational intelligence
- avatars

In this paper, we propose an Avatar-based intelligent agent technique for 3D Web based virtual online communities based on distributed artificial intelligence, intelligent agent techniques, and knowledge bases in a digital library. One of the goals of this joint NSF (IIS-9980130) and ACM SIGGRAPH Education Committee (ASEC) project is to create a virtual community of educators and students who have a common interest in computer graphics, visualisation, and interactive techniques. In this virtual community (ASEC World) using VRML-based technology, Avatars represent the educators, students, and other visitors to the world. Intelligent agents represented as specially dressed Avatars will be available to assist the visitors to ASEC World. The basic Web client/server architecture of the intelligent knowledge-based avatars is given. Importantly, the intelligent Web agent software system for the 3D virtual community is implemented successfully. In the future, useful data and knowledge for a specific education application (or another application) will be input into the avatar database and the avatar rule base, and then the intelligent 3D Web agent software system will have real applications.

1 Introduction

In recent years, intelligent agent techniques and other advanced methods have been widely used in Web-based applications such as E-Commerce, Web search engines, Web mining, 3D Web applications, etc [1, 3–6, 11, 12, 19–22, 31, 33, 34]. Web based virtual communities; including educationally oriented ones is another potentially huge market that can combine intelligent agent technology, distributed systems, and digital libraries. Intelligent 3D agents are a new category of information society tools. We are studying intelligent agents applied to educational virtual communities. This paper focuses on using Avatar intelligent agents in a virtual online community. The major goal is to create a 3D-virtual community (ASEC World), using VRML based technology that provides access to the graphics and visualisation educational materials in the ASEC Digital Library (ASECDL) and common spaces for visitors to congregate and discuss their interests.

Developing a sense of community among Digital Library users and developers is one of the important issues supported by a NSF grant (IIS-9980130). Since the ASECDL contains materials for instruction in computer graphics, visualisation, and interactive techniques, it primarily appeals to a visually oriented audience. As more computer users are exposed to 3D environments, online 3D virtual worlds are going to become more common and expected.

In this paper, a hybrid intelligent agent is developed based on 3D-graphics-based interface agents, information/knowledge agents and Internet agents. The objective of designing ASEC World with the hybrid intelligent agents is to generate a 3D world that will represent the virtual community. Each area or sub-world of ASEC World has agents. These 3D avatar agents with knowledge bases containing both domain-specific questions and relevant answers can move if they want to contact someone, talk when they need and disappear if necessary.

2 Architecture of Intelligent-Agents-Based Virtual World

Our interest in distributed multi-user environment was stimulated by a general interest in VRML and the practical application in education. The VRML approach is based on the following requirements:

i. *Openness*: The system should be open to as many hardware and software platforms as possible. It should not be designed for any specific application area.
ii. *Consistency*: All distributed users are supposed to have the same view. Manipulation by any users should have immediate effect.
iii. *Dynamics*: Users can enter and leave the virtual environment at any moment. New users will have an up-to-date view.

iv. *Persistence*: User manipulations will have effects after the user has left and disconnected from the system. The current state of the world will be saved.
v. *Partitioning*: Complex worlds should be partitioned to minimise the amount of necessary communication.
vi. *Extensibility*: The system should be implemented in a way that allows for easy extensibility concerning new communication protocols, new object behaviors and new functionality like inter-object collision handling.
vii. *Database Interface*: The system should have an interface to a database of objects, avatars and behaviours.
viii. *Complex behaviours*: Object behaviour should not be limited to simple movements. A database of complex behaviours for objects like doors, windows, switches, and any kind of devices, 3D widgets and avatars is required.

Application Architecture

To make asynchronous updates possible a typical server would need a proxy object for each client. Even a not fully distributed implementation still has the asynchrony problem. A call to the server object might block the client until all other clients are notified. This is not desirable for frequent updates of attributes.

Tuplespaces are a type of decentralised distributed data structure. There are at least two readily available implementations in Java, Javaspaces and TSpaces. Tuplespaces in general do not have the notion of a distributed object but just of universally accessible storage. To implement a distributed object would mean to listen to new tuples matching a certain template and write a new tuple.

The model of our system is defined by three concepts. A room, which acts like a locale, defines a group of clients that communicate with each other. Moreover there can be objects that are contained in exactly one room at all times. These typically define VRML content that is rendered at the user clients. An object again may contain network states, which act like attributes of the object the changes of which are distributed to all clients. Moreover should it be possible to render more than one room in an appropriate client. Links should be defined that describe how one room is related to the other. This will happen on a pure spatial basis.

Details of Client Server Model

The architecture of the 3D world multi-user server environment is client/server based. The client/server is web-based with the server being hosted on the web server and the client connects to the server through a web browser through an Internet.

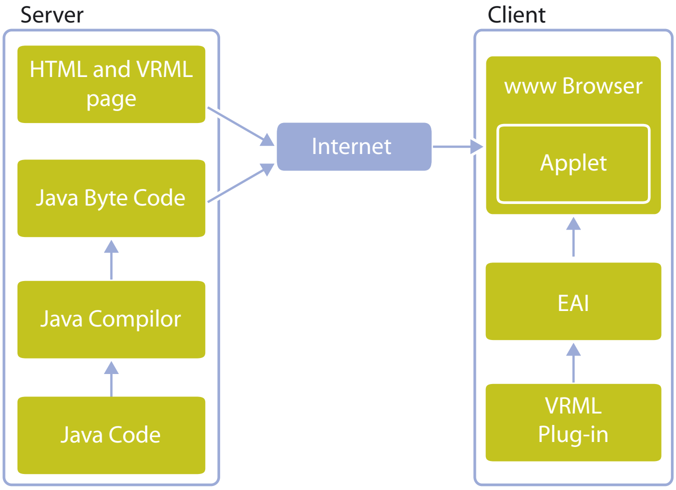
The client/server software architecture is a versatile, message-based and modular infrastructure that is intended to improve usability, flexibility, interoperability, and scalability as compared to centralised, mainframe, and time sharing computing [15]. The basic characteristics of client/server architectures are as follows:

1. Combination of a client or front-end portion that interacts with the user, and a server or back-end portion that interacts with the shared resource. The client process contains solution-specific logic and provides the interface be-

tween the user and the application. The server process acts as a software engine that manages shared resources such as databases, printers, modems, or high powered processors.

2. The front-end and back-end tasks have fundamentally different requirements for computing resources such as processor speeds, memory, disk speeds and capacities, and input/output devices.
3. The environment is typically heterogeneous and multi vendor. The hardware platform and operating system of client and server are not usually the same. Client and server processes communicate through a well-defined set of standard application program interfaces (APIs) and remote procedure calls (RPCs).
4. An important characteristic of client/server systems is scalability. It can be scaled horizontally or vertically. Horizontal scaling means adding or removing client workstations with only a slight performance impact. Vertical scaling means migrating to a larger and faster server machine or multi servers. There is direct communication only between clients and the server; no modifications can be missed by the server.
5. The 3D virtual community requires allowing multiple users to connect to a centralised system as shown in Figure 1 below. The centralised system coordinates the activities of users with similar requests. The agent layer controls the communication between the user and centralised server. Thus the system can be divided into three architectures: Client Architecture, Server Architecture, and Agent Architecture.

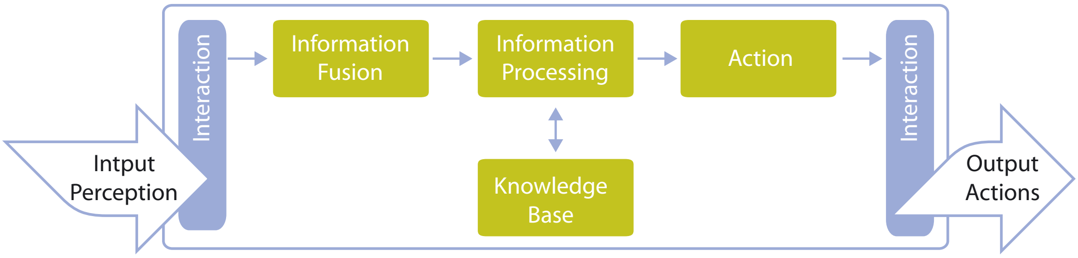
Figure 1 Client/Server Architecture



Agent Architecture

An agent can be assumed as a black box that receives a range of inputs. It uses its intelligence to process these inputs and produces an output. The major task of an agent is to use interaction modules that specifically match to the capabilities and particularities of the corresponding interaction partner. The agent uses interaction modules to obtain information and changes within its environment it also initiates its own actions. The central task of agents is not to interact with the environment but rather process and interpret the perceived information and to achieve its own goals. The incoming information must be first integrated in an appropriate manner and accepted in agent's knowledge base. This process is called information fusion. The next step is to process this external information. This is the central component of the agent because it reflects the agent's true functionality. The agent architecture is shown in Figure 2 below.

Figure 2 Agent Architecture



The agent interacts with the users using its knowledge base for information processing. The users input are used in the information fusion task. The knowledge base resides on a relational database management system (RDBMS). The access to this RDBMS is through Java's open database connectivity JDBC. The agent stores and retrieves information for processing the information using this knowledge base.

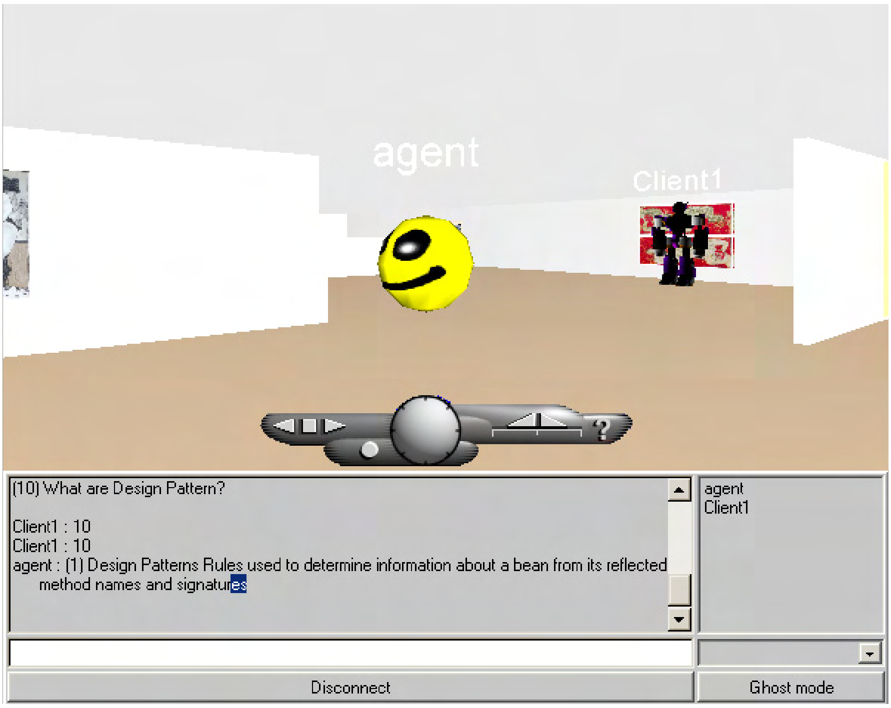
The agent puts forth some questions in front of the user. These are the most frequently asked questions in the environment. The client replies to these frequently asked questions (FAQs) by selecting the appropriate questions. Or if he/she has some other questions he can put forth a completely new question related to the topic of the subject. If the client has requested for FAQ's then the agent refers to its knowledge base and based on this information he gives back the response. If the agent has been bombarded with a new question then the agent uses language processing to parse the keywords out of the question. The keywords are list of words to which lots of answers are related. Once the keywords are parsed the knowledge base is queried for results using the logical OR operation. The knowledge base returns relevant information to the agent, which in turn forwards to the client who requested this information. If the client agrees on that an email is posted to the administrator about this question. The administrator replies back to the user the answer and this updates the agent's knowledge base.

3 Real Implementation

This virtual community is implemented on the client/server architecture on the web (see Figure 3 below). The multi-user environment is hosted on an HTTP server. The client accesses the server through the Internet using a Java applet. When the knowledge base is inadequate to answer the visitor's questions, the agent will contact other agents, or a member of the ASEC who can answer it and the answer is passed on to the visitor.

The agent works on its local knowledge base and in order to access other agents' knowledge it has to go through the central moderator that keeps track of the other agents' specific interests [13]. Thus the major functions of this agent would be Reactivity, Goal Orientation, Communication/co-operation, Reasoning, Autonomy, and Character.

Figure 3 The Intelligent-agents-based 3D Virtual Community



The Virtual Environment

In order to allow multiple users in the single environment, information has to be shared between clients. Since these clients may be distributed over a heterogeneous network, any implementation has to be platform independent [17].

The virtual environments try to mimic the real world. In the virtual environment you create visual representation of a synthetic environment, acoustic and other sensual simulation is equally important.

In the shared environment the users has to be able to interact with the environment, and the objects in the environment. Thus the application has to provide the functionality, which permits the manipulation of the multimedia information [1, 36].

Relevant Technologies

Open technologies are optimal for satisfying the above requirements. The technologies include Java, VRML, EAI of the VRML, RDBMS. The implementation and authoring of multi-user environments requires thorough knowledge of VRML, Java, JavaScript, and the EAI.

Java is an object-oriented programming language developed by the Sun Microsystems. Java is a web programming language, which supports platform independence. Java code can be executed in HTML browsers and used as the scripting language in VRML. Java applets can open network connections to servers as well as present graphical user interface. Furthermore with the help of External Authoring Interface, it can access the VRML plug-in functionality [4] to display and control visual simulation. In traditional programming languages a compiler or a runtime interpreter is used to convert program source code into system specific binary code. The Java compiler does not directly translate the Java source code into binary code but Java byte code. This byte code is platform independent and can be executed without modification on all platforms that support Java. A Java interpreter developed for a particular platform is used to execute the byte code on the target platform. Thus Java uses both compiler (to create the byte code) and an interpreter (to execute the byte code). A virtual machine is used to execute the Java byte-code on the target platform. A virtual machine is added to the existing operating system of the target computer and provides a simulated runtime environment. Irrespective of the actual system platform Java virtual machine always provides a Java program with standardised runtime environment.

The Virtual Reality Modelling Language (VRML) is a file format for the description of dynamic scene graphs containing 3D objects with their visual appearance, multimedia content, event model, and scripting capabilities. VRML is designed to be used on the Internet and local client systems and to be used as an exchange file format. VRML is the universal language for integrated 3D graphics and multimedia [13]. VRML is designed based on following criteria:

i. *Author ability*: Enable the development of computer programs to create, edit and maintain VRML files as well as automatic translation programs for converting other commonly used 3D file formats into VRML files

ii. *Composability*: Provide the ability to use and combine dynamic 3D objects in the VRML and thus allow re-usability
iii. *Extensibility*: Provide ability to add new object types not explicitly defined in VRML
iv. *Implementation*: Capable of implementation on a wide range of systems
v. *Performance*: interactive performance on wide range of computing platforms
vi. *Scalability*: Enable arbitrarily large dynamic VRML worlds.

VRML browsers as well as authoring tools for the creation of VRML files are widely available. VRML supports extensibility model that allows new dynamic 3D-object and application communities to develop interoperable extensions [18].

The External Authoring Interface (EAI) connects the Java Virtual Machine running in the web browser to execute applets and plug in used to display VRML content as shown in Figure 3 below. It is accessed with a set of Java classes defined in the EAI Specification [29]. The client uses the EAI to control the visual simulation in the VRML plug-in. Thus EAI allows the external environment to access the nodes in the VRML scene using the existing VRML event model. In this model an eventOut of a given node can be routed to an eventIn of another node. When the eventOut generates an event the eventIn is notified and its node processes that event. Additionally, if the Script node has a reference to a given node it can send events directly to any eventIn of that node and it can read the last value sent from any of its eventOuts.

Navigation is possible by using plug-in controls therefore the plug-ins have to update the applet about the users current position and orientation, and pass events from the VRML scene. The External Authoring Interface allows four types of access into the VRML scene:

- Accessing the functionality of the Browser script interface,
- Sending events to eventIns of nodes inside the scene,
- Reading the last value sent from event outs of nodes inside the scene, and
- Getting notified when events change values of node fields inside the scene.

There are three main items in the VRML browser that can be accessed from an external application: The browser, nodes within the scene graph and fields within nodes. The VRML browser exposes a set of services, which allow the external application the JVM of applet to interact with it.

When multiple applications make request of the browser the request can be serviced in the order of arrival time. The arrival time is used to sort out conflicting request from the multiple applications to ensure consistent results.

4 Applications of ASEC World

The 3D gives an actual feeling of physical space to the virtual meeting space. It is easy for visitors to move around. The knowledge base is quite large and hence the visitors can get information about different subjects and for a particular subject, detailed knowledge is available. Other advantages include the following:

- reduced resource usage of the client,
- reduced network load, and
- a decentralised structure.

Many possible applications can be seen for the future. The information agents are used for information retrieval and filtering, news watcher, advising, focusing, and traffic. The co-operation agents can be used in Entertainment, Groupware, and Network Management/telecommunication. Transaction agents can be used in E-commerce, manufacturing and management of business process [13].

Case Study 1: A computer science professor from California logs in to ASEC World. As she walks into one of the common spaces she encounters another avatar that is an art professor from Georgia and also a professor from the University of Leeds, in the U.K., who teaches visualisation. They engage in a 3-way discussion about the possibilities of incorporating aspects of each other's disciplines into their own courses. The art professor mentions a book on drawing that illustrates how certain techniques can help a person focus in on what is important in an image and the Leeds professor decides he will look at it and incorporate parts of it into his class. They then ask an SV avatar if there are any examples of using art techniques directly in scientific visualisations. The SV responds by taking them to another area where a researcher has published such a paper and they can all look at the Van Gogh like images that are actually representations of data. After further discussion the three of them decide to get together later and submit proposals to their respective Universities for interdisciplinary courses that they will jointly develop. From this example, we can see advantages of using a virtual world based on 3D graphics rather than plain text-based chats.

More and more applications are enjoying distributed 3D-computer Graphics and multi user environment, e.g. military simulations, virtual surgery, engineering, architecture, CBT, gaming, product presentation and virtual shopping malls. The development of MUSE has been driven forward in following fields:

i. Military Applications: The benefits of virtual environment for military purposes have been realised early: no danger of life or destruction, any real damage, strategic simulations in arbitrary terrain and landscapes, and simulation of vehicle prototypes. More possibilities are added if the environment is distributed: training of teams, scalability of the number of participants, installation in different separated locations, simulation with a combination of several military forces and semi-automated forces (SAFs).

ii. Entertainment: The entertainment sector offers a potentially large marketplace either for home-based or location-based entertainment. In the 1970s games like Adventure or Dungeon & Dragon spawned a new genre of role-playing games. MUD (Multi-User Dungeon) and their object-oriented versions became the generic descriptions for multi-user games. Home-entertainment devices fall in two categories: game consoles and PCs. Game consoles have little support for multi-user playing. The history of multi-player games and teaching systems goes back to the 1960s and culminated 1993 in Doom. More and more games with support for TCP/IP and protocol tunnelling reach the market. The notion of location-based entertainment describes multi-player gaming with specialised equipment in BattleTech centres or amusement parks.
iii. Research and commercial systems: The widespread use of the WWW makes computer supported collaborative work and virtual shopping applications more interesting for research and commerce.
iv. Cyber Shopping: The multi-user server environment is suited for creating virtual communities. These communities allowed shopping by actually allowing online shoppers to view the 3D objects and can get assistance in how to use system and shop in the cyberworld.
v. Virtual Teaching Environment: The virtual environment can be used to conduct online courses with students. There can be a good interaction between the students and no need to be physically present in the classroom.
vi. E-commerce: E-commerce can be business to client or client to business. The virtual environment combined with agent architecture can well fit into buying agents or selling agents that assist the users about the various products available over the Internet.

5 Conclusion

One of the goals of this joint NSF (IIS-9980130) and ACM SIGGRAPH Education Committee (ASEC) project is to create a virtual community of educators and students who have a common interest in computer graphics, visualisation, and interactive techniques. In this virtual community (ASEC World) Avatars will represent the educators, students, and other visitors to the world. Intelligent agents represented as specially dressed Avatars will be available to assist the visitors to ASEC World. ASEC World provides a prototype of the 3D virtual communities of tomorrow. It can simulate a virtual meeting space with avatars representing both visitors and intelligent agents. Students, faculty, and other visitors are able to communicate more comfortably and more efficiently and use agents for quick access to information. This system is a multi-user VRML application. It enables several users to share a single VRML world. VRML events are distributed between the several instances of the VRML world on the different user computers. The system is split into a server program running on the web server where the VRML worlds and supporting HTML files. The agent is virtual helper who provides information for the users.

In the future, useful data and knowledge for a specific education application (or another application) will be input into the avatar database and the avatar rule base, and then the intelligent 3D Web agent software system will have real applications.

ACKNOWLEDGMENTS

The project detailed in this paper is supported by NSF under Grant IIS-9980130 and the ACM SIGGRAPH Education Committee. The authors would like to thank the anonymous reviewers very much for their careful comments.

REFERENCES

1. Hermann A. C., Gaspary L. P. and Almeida J. B. (1999). "Design and execution of adaptive multimedia applications in the Internet," Proceedings of the 1999 ACM symposium on Applied computing, pp. 624–629.
2. Yang A. and Bachar Y. (1999). "Using Java and the socket interface in teaching client/server programming," Proceedings of the 4th annual SIGCSE/SIGCUE on Innovation and technology in computer science education, pp. 206.
3. Adelson B. (1992). Evocative agents and multi-media interface design, Conference proceedings on Human factors in computing systems, pp. 351–356.
4. Jung B. and Milde J.-T. (1999). "An open virtual environment for autonomous agents using VRML and Java," Proceedings of the fourth symposium on the virtual reality modeling language, pp. 7–11.
5. Blaxxun Corporation http://www.blaxxun.com, (accessed date: 8 Dec. 2000).
6. Martinho C. and Paiva A. (1999). "Pathematic Agents: Rapid Development of Believable Emotional Agents in Intelligent Virtual Environments," Proceedings of the third annual conference on Autonomous Agents, pp. 1–8.
7. Ancona D. (1998). "3DXML Authoring, Publishing and Viewing Structured Information," http://www.vrml.org/WorkingGroups/dbwork/ancona/home.html.
8. Lipkin D. "Integrating XML and VRML: A Technical Discussion", http://www.vrml.org/WorkingGroups/dbwork/vrmlxml.html, (accessed date: 8 Dec. 2000).
9. Rabenhorst D. A., Wang C. P., Norton A., Sweeney P. and Turk G. (1992). "Interactive simulation in a multi-person virtual world," Proceedings on Human factors in computing systems, pp. 329–334.
10. Waller D. and Miller J. (1998). "A desktop virtual environment trainer provides superior retention of a spatial assembly skill," Proceedings of the conference on CHI 98 summary: human factors in computing systems, pp. 339.
11. Chorafas D. N. (1997) Agent Technology Handbook, McGraw Hill.
12. Menczer F. and Belew R. K. (1998). "Adaptive information agents in distributed textual environments," Proceedings of the second international conference on Autonomous agents, pp. 157–164.
13. Goddard T., Sunderam V. S., (1999). "ToolSpace: Web Based 3D Collaboration," Proceedings of the fourth symposium on the virtual reality modeling language, pp. 161–166.
14. Grand S., Cliff D. and Malhotra A. (1997). "Creatures: Artificial Autonomous Software Agents for Home Entertainment," Proceedings of Autonomous Agents 97 Conference, ACM Press.
15. Berghel H., (1995). "CYBER-SURFING: the State-of-the-Art in Client Server Browsing and Navigation," Proceedings of the 1995 ACM symposium on Applied computing, pp. 1–4.
16. Doppke J. C., Heimbigner D. and Wolf A. L. (1998). "Software process modeling and execution within virtual environments," ACM Trans. Softw. Eng. Methodol. 7 (1), pp. 1–40.
17. Saar K., (1999). "VIRTUS: A Collaborative Multi-User Platform," VRML 99 Paderbom Germany, pp. 141–152.
18. Taniguchi M. (1998). "Event processing for complicated routes in VRML 2.0," Proceedings of the third symposium on virtual reality modeling language, pp. 83–88.

19. Klusch M. and Weib G. (1998). "Cooperative Information agents II, learning, Mobility and Electronic Commerce for Information Discovery on the Internet," Second International Workshop '98 Paris, France.
20. Nwana H. S. and Azarmi N. (1997). Software Agents and Soft Computing: Towards Enhancing Machine Intelligence, Springer-Verlag.
21. Doyle P. and Hayes-Roth B. (1998). "Agents in annotated worlds," Proceedings of the second international conference on Autonomous agents, pp. 173–180.
22. Ossowski S., (1999). Coordination in Artificial Agent Societies, Springer.
23. Owen G. S., Sunderraman R. and Zhang Y.-Q. (1999). "The Development of a Digital Library to Support the Teaching of Computer Graphics and Visualization," Proc. of GVE'99, pp. 62–67.
24. Park S. and Han T. (1997). "Object-Oriented VRML for Multi-user Environments," Proceedings of the VRML '97 Symposium, pp. 25–32.
25. Funkhouser T. A. (1995). "RING: a client/server system for multi-user virtual environments," Proceedings of the 1995 symposium on Interactive 3D graphics, pp. 85–ff.
26. Mey V. D. and Gibbs S. (1993). "A multimedia component kit: experiences with visual composition of applications," Proceedings of the first ACM international conference on Multimedia, pp. 291–300.
27. VRML Standard Version 2.0, ISO/IEC CD 14772, http://vrml.org/VRML2.0/, (accessed date: 8 Dec. 2000).
28. VRML Repository, http://www.web3d.org/vrml/, (accessed date: 8 Dec. 2000).
29. VRML External Authoring Interface specifications, http://vrml.org/WorkingGroups/vrml-eai/, (accessed date: 8 Dec. 2000).
30. VRML Living Worlds Working Group proposal draft 2, http://vrml.org/WorkingGroups/living-worlds/, (accessed date: 8 Dec. 2000).
31. Brenner W., Zarnekow R. and Witting H. (1998). Intelligent Software Agents: foundations and applications, Springer.
32. Web3D Consortium http://www.vrml.org/fs_specifications.htm, (accessed date: 8 Dec. 2000).
33. Lamersdorf W. and Merz M. (1998). Trends in Distributed Systems for E-Commerce, International IFIP/GI Working Conference Trec'98, Hamburg Germany.
34. Broll W. and England D. (1995). "Bringing worlds together: adding multi-user support to VRML," Proceedings of the 1995 symposium on Virtual reality modeling language, pp. 87–94.
35. Broll W. (1997). "Populating the Internet: supporting multiple users and shared applications with VRML," Proceedings of the second symposium on Virtual reality modeling language, pp. 33–ff.
36. Honda Y., Matsuda K., Rekimoto J. and Lea R. (1995). "Virtual Society: Extending the WWW to Support a Multi-user Interactive Shared 3D Environment," Proceedings of VRML'95, pp. 109–116.