From Proceedings of the 3rd Workshop on Virtual Reality. Rio de Janeiro: Brazilian Computer Society - SBC, 2000.

On Design of 3D and Multimedia Extension of Information System Using VRML

Jiří Žára
Daniel Černohorský

Department of Computer Science & Engineering – Czech Technical University
Karlovo nam 13 – 121 35 – Praha 2 – Czech Republic
zara@fel.cvut.cz, xcernoho@fel.cvut.cz

Abstract. *This paper describes an experimental approach to information system based on virtual reality. The basic part of the system consists of 3D model of the Department of Computer Science & Engineering at the Czech Technical University. The model has been created in VRML and stored in a standalone database. The information system provides three main functions: retrieving information from the model, automatic navigation, and interactive editing of 3D model. The last function allows changing the appearance of classrooms and offices as well as manipulating furniture and other objects within those rooms. The system is fully controlled via Internet.*

1. Introduction

The interfaces to information systems on the web are mostly based on HTML documents containing textual data, images, forms and other 2D elements. In some cases it is advantageous to present additional information in 3D form. This is the case of the information system (IS) used at our university. We have decided to enhance the functionality of the system by utilizing 3D space, i.e. by presenting the model of the Department in VRML. We have focused on combination of real 3D data (building, rooms) and multimedia objects retrieved from the standard IS.

The building, in which the department is located, contains more than forty rooms. There are offices of teachers, lecture and seminar halls, computer laboratories and a lot of corridors and staircases (see Fig. 1). Since this historical building has a complicated internal structure, we have also taken into account a support for the automatic navigation. This task allows the user to find the shortest path from current position to arbitrary destination. Similar but simpler approach can be found in [Pittarello et al. 1998, Schaerf and Tessicini 1999]. All rooms are equipped with furniture corresponding with the real world, and virtual objects with special functionality. Interactive objects are connected with information obtained from IS, e.g. timetables of lectures in a lecture room, or hyperlinks to teacher‘s home page. The system can be maintained and changed easily, because all 3D information is stored in a database. It enables to remove, edit, and add virtual objects into the model as well as to set up the look of rooms using textures and colors. It is also possible to create/import new types of virtual objects. The virtual assembly is similar to approach in [Nousch and Jung 1999].

The paper consists of the following parts: Section 2 describes how the 3D model is stored in the database, Section 3 introduces the structure of the web client/server communication, Section 4 is given to automatic navigation, Sections 5 and 6 describe editing the model. Acquired experience and ideas for the future works in Section 7 conclude the paper.

2. Database structure

The description of the whole 3D model, VRML objects, links to HTML documents and other information are stored in the database. This approach is quite common for web based applications [Schaerf and Tessicini 1999] and brings several advantages:

Flexibility: The VRML model can be easily extended and/or changed.
Multi-user edit mode: Several users can edit the model simultaneously.
Security: Individual rights to the database are assigned to different users.

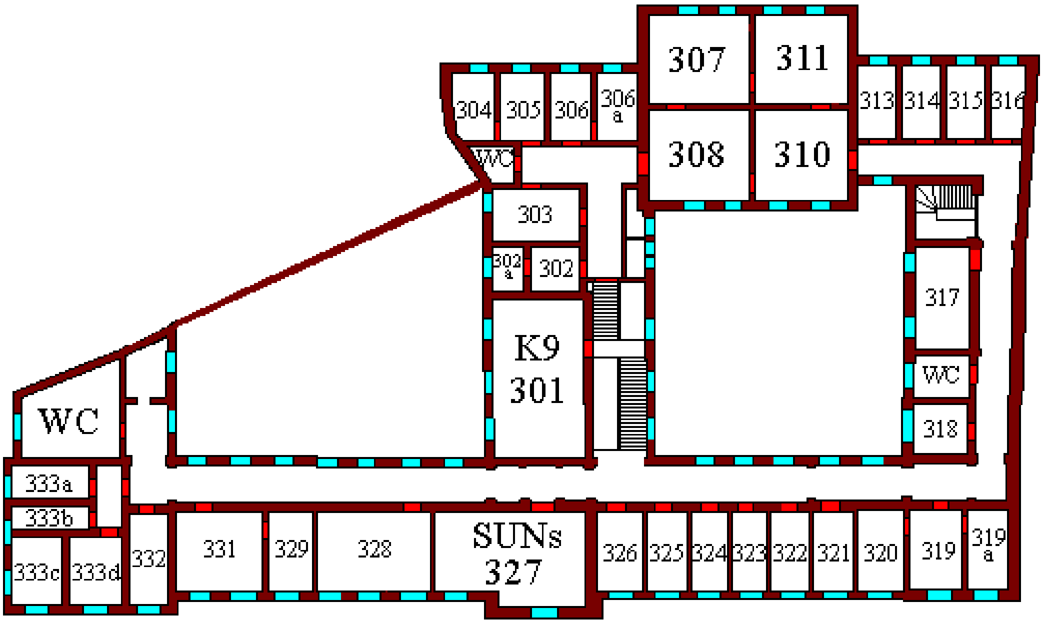


Figure 1. 2D layout of the 3rd floor of the Department

We have used publicly available MySQL database [MySQL] that is fully appropriate for the experimental purposes. The database contains information about:

- geometry (dimensions and coordinates of surfaces), window and door positions, interior object specifications, links among the rooms (neighborhood)
- appearances of surfaces, materials, textures
- positions of furniture and multimedia objects in rooms
- viewpoints, lights, other VRML stuff
- links to multimedia files (sounds, movies)
- users and their rights

3. WWW client/server communication

Since the database does not contain pure VRML code, the spatial model has to be generated by CGI script that reads raw data from the database. The mechanism of creating VRML source code for one room is as follows (see also Fig. 2):

1. The user activates a link from HTML document to a CGI script.
2. The link always contains input parameters for the script: number of required room and the name of the desired viewpoint in this room.
3. The CGI script creates SQL queries upon the input data.
4. The script connects to the database, sends queries and receives raw data.
5. The same CGI script converts raw data into the VRML hierarchy.
6. WWW browser receives VRML source code and interprets it.

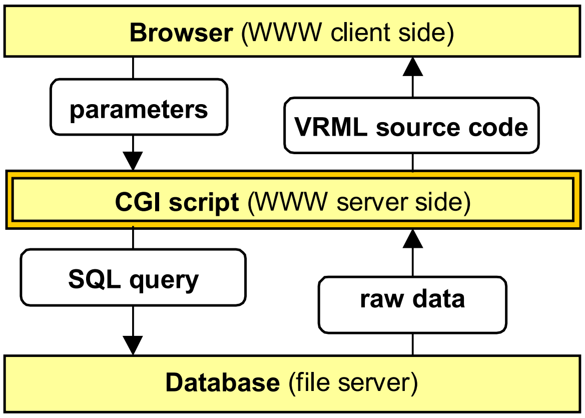


Figure 2. Mechanism of generating VRML source code

The coordinates used in VRML model correspond precisely with the dimensions of the real building. On the other hand, all rooms are modeled with minimal number of faces, because the aim is to achieve a high rendering speed even on older PCs. The walls are combined with windows and doors. Doors are always interactive and contain links to CGI scripts (with the neighboring room number). One VRML room or a corridor is the basic unit presented to a user.

Rooms often contain furniture and objects, either interactive or non-interactive. Interactive objects are mostly VRML Anchor nodes with a hyperlink to HTML documents. A few textual data can be found directly in VRML model too, e.g. information tables located on walls near entrances to lecture and seminar rooms or computer laboratories. Such VRML Text nodes are again hyperlinked to rich timetables written in HTML. Besides the virtual object, couple of viewpoints located near doors helps the user to navigate. Those viewpoints are used also for automatic navigation.

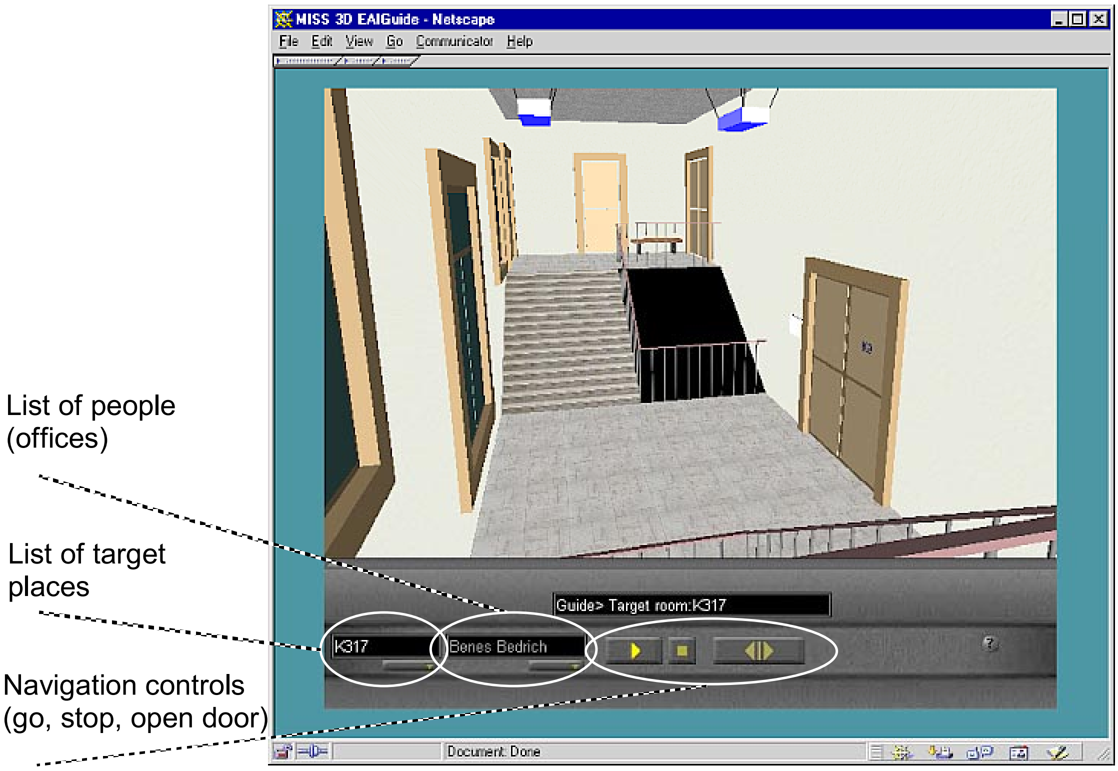
4. Automatic navigation

This service enables simple and quick searching of path from current room to another one. The resulted path is the shortest possible connection between two specified locations. When the path is found a set of viewpoints/corridors is created and sent to a browser thus allowing the user to walk automatically through the model to the destination room.

The service has been implemented in Java and is executed as an applet using External Authoring Interface – EAI [Marin 1997] to control a VRML browser. GUI of the applet has been designed in the same style as the user interface of popular Cosmo Player VRML browser for Windows (see Fig. 4). A user can select the destination from two lists – either by room numbers or by names of people. The first list is useful when searching for lecture halls and laboratories while the second is good for finding offices of the staff.

The next part of applet GUI ensures automatic walking through the model. Two buttons allow starting and stopping pre-computed interpolation of user's position among viewpoints. The third button opens a door between two neighboring rooms/corridors. In fact, this button sends a request to a server for a new part of the virtual model.

The navigation support is based on Dijkstra algorithm for finding the shortest path in a graph. All viewpoints are initially stored in the form of a graph encoded in a field of invisible VRML Text node. The Java applet reads these data from VRML scene and creates internal graph structure. Whenever the shortest path is evaluated, a user is navigated from his/her current position to the nearest viewpoint in the current room. Then simple position interpolation to the next VRML Viewpoint from the computed list of viewpoints is repeatedly performed.


MISS 3D EAIGuide - Netscape
File Edit View Go Communicator Help
List of people (offices)
List of target places
Navigation controls (go, stop, open door)

Guide> Target room:K317

K317

Benes Bedrich

Document: Done

Figure 3. VRML browser and Java applet for automatic navigation

One of the crucial problems solved during implementation of the automatic navigation was a smooth transition among adjacent viewpoints. Each viewpoints has just one orientation assigned in VRML, but it is used in different directions during walk-through the model. Imagine one viewpoint in front of a door. When entering the room (opening a door), this viewpoint is oriented straightforward. When leaving the room, an avatar has to be moved to this viewpoint, but with opposite orientation.

We have solved this by installation of pairs of viewpoints as shown on Fig. 4. As the result, we have obtained an oriented graph of viewpoints in which each pair has the same position but different orientation. It is interesting that this approach allows "to drive" an avatar similarly like in the car – in our case always on the right side of corridors.

5. Editing room properties

Users with proper rights (typically staff) can change the appearance of rooms by setting colors and textures of walls, a ceiling, and a floor. These functions are password protected. A dedicated CGI script has been developed for this task. Three separate windows are used for editing rooms:

- *Main window* is subdivided into several forms. They allow creating relations between various parts (walls, floor, ceiling) and materials and/or textures.
- *Info window* contains a preview of currently selected material color and texture in image form. The window is also used for creating, changing and removing materials and textures.
- *VRML window* presents current state of virtual room depending on properties selected in the main window. When satisfied a user can update the appearance of the room in the database.

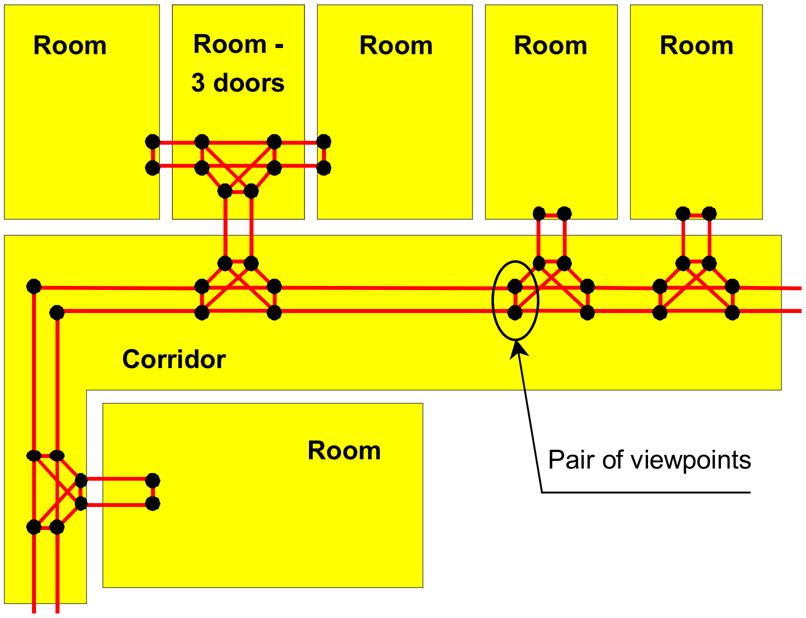


Figure 4. Pairs of viewpoints help to smoothly change the orientation of avatar during walk-through different places in a model

Both materials and textures have symbolic names assigned, because users are not familiar with parameters of Phong shading model used in Material node. All textures are stored in the database. Upload is currently not allowed, since we want to preserve reasonable number of textures, their quality and size within the model.

CGI scripts invoked from forms communicate with the database by analogous mechanism as CGI script described in Section 3.

6. Editing furniture and multimedia objects

The set of functions for manipulating with virtual objects has been designed not only for initial creation of the model but also as a test-bed application for assembling virtual objects by non-experienced users – teachers and other staff. Such people are now able to change the look of their virtual offices, to add new objects, to tune the positions of furniture and to improve the appearance of their working place before a manipulation with real and possibly heavy objects would occur. The subsystem for editing interiors is password protected. The user interface is based on one HTML page subdivided into four frames:

- *"Room editor" frame* contains a Java applet – the main program for interior editing. The applet works with one room only (see Fig. 5).
- *VRML object frame* is used as a preview window for selected VRML object that is going to be placed into a room.
- *VRML room frame* presents the whole VRML model of the current room.
- *Control frame* helps to edit, remove, and add objects already existing in the database.

The "Room editor" applet enables to select a global type of furniture (like chairs, tables, etc.) and to choose a particular piece of furniture from those lists (see two pull-down menus on the right top of Fig. 5). Selected object is then added to the room and positioned. A user can change the position and orientation of all already existing objects as well as remove them by dragging outside the room borderlines. Small objects are automatically elevated when moved to such a position where they overlap another object. Each time a new interactive/multimedia object is added, the application asks user for *url* used as a hyperlink, name of audio clip, etc. The editor presents only ground plans of rooms and uses simple 2D bounding boxes for positioning 3D objects. This simple approach is sufficient for most common cases.

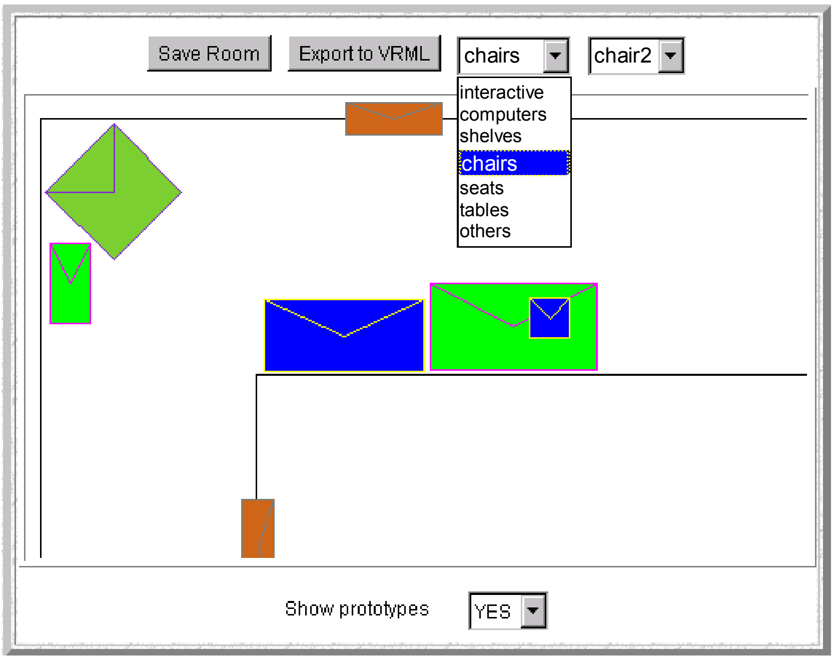


Figure 5. "Room editor" applet shows a part of a corridor, a door (on top), a window (bottom), and five objects. One smaller object is placed on the top of the bigger one.

Two control buttons on the top of applet allow storing current state to the database, and executing CGI script that generates VRML source code for "VRML room frame".

The mechanism of collaboration between Java applet and CGI scripts is depicted on Fig. 6:

1. When initialized, the applet activates a CGI script responsible for the communication with the database.
2. The applet receives raw database data from the script and creates a 2D room layout with objects.
3. Whenever a new object is chosen from a list of candidates, the applet calls the script, which generates one VRML model and sends it to "VRML object frame".
4. Control button "Export to VRML" calls the script, which generates model of the whole room presented in "VRML room frame".
5. "Save room" button initiates the script storing the current state of a room to the database.

Arbitrary VRML object can be uploaded to the database. The only requirement is that the object has to be defined as a prototype containing *bounding box* description and optionally a *color*. These parameters are stored in the database and provided to the "Room editor".

The applet for maintaining VRML objects in the database has been developed using CGI script and HTML forms.

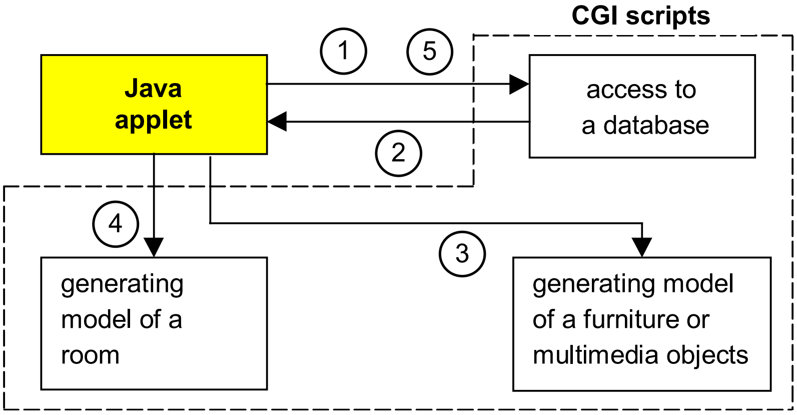

CGI scripts
1
5
Java applet
access to a database
2
4
3
generating model of a room
generating model of a furniture or multimedia objects

Figure 6. Collaboration of „Room editor“ applet and CGI scripts

7. Conclusion

The goal of the presented experimental system was to extend already existing information system by spatial and multimedia information with respect to needs and requirements of students, teachers, and visitors. The important condition was the possibility to change a model according to changes of the real state. The pilot version of the system has been successfully implemented and it is being tested now.

The first experience shows that limited amount of information can be directly retrieved from standard IS, while 3D data are stored in specific database. The virtual 3D model can serve as an additional source of information but not as a general user interface to standard IS, since for many types of information it is better to provide them in form of HTML pages rather than in VRML worlds. However automatic navigation is the feature that has been naturally incorporated into 3D virtual space and this service can help users significantly. The same is true for virtual manipulation with furniture and other objects.

There are some problems remaining, mostly on the software side. The combination of WWW browser, VRML browser, Java Applet, and CGI scripts is not fully stable. We have found that the bottleneck is probably the EAI interface between a VRML browser and other components.

The system is still under development. We are going to improve the following topics:

- *More multimedia.* Currently we have incorporated just a few Sound nodes with audio-clips consisting of ‘welcome speech’ of teachers. More sounds would increase a quality of the virtual environment. Knocking on keyboards in computer classrooms is funny example of such approach. The size of audio-clips is limiting factor, so we prefer simple sounds played in a loop.
- *More interactivity.* Advanced users (staff) should be able to upload their own VRML models into the system as well as images and photos later used as textures. Since we want to ensure high frame rate of rendering, we have to automatically check the complexity of new objects and size of image data.
- *On-line and real-time information.* The model will contain information depending on the current time and real world activities. The example is the information about current lecture in a lecture hall.

Acknowledgment

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under research program No. Y04/98: 212300014 (Research in the area of information technologies and communications).

References

ISO/IEC 14772-1:1997, Information technology – Computer graphics and image processing – The Virtual Reality Modeling Language (VRML) – Part 1: Functional specification and UTF-8 encoding.
http://www.vrml.org/Specifications/VRML97/

Marin, Ch.: Proposal for a VRML 2.0 informative annex, External Authoring Interface. Silicon Graphics Inc., 1997
http://www.cosmosoftware.com/products/player/developer/eai/

MySQL home page
http://www.mysql.org/

Nousch, M., Jung, B.: CAD on the World Wide Web: Virtual Assembly of Furniture with BEAVER. In Proceedings of VRML 99 – Fourth Symposium on the VRML, Paderborn, Germany, 1999, ACM Press, pp. 113-119

Pittarello, F., Pittarello, M., Italiano, G.F.: Architecture and Digital Exhibitions The Einstein Tower World, In Proceedings of the Eurographics Workshop on Virtual Environments ’98, 1998, Springer Verlag, ISBN 3-211-83233-5, pp.162-171.

Schaerf, M., Tessicini, A.: JubilEasy: Build a personalized 3D visit of Rome. In Proceedings of VRML 99 – Fourth Symposium on the VRML, Paderborn, Germany, 1999, ACM Press, pp. 105-112.