

## IMPACT ON 3D MODELLING FROM THE DESIGN OF A WEB-BASED 3D INFORMATION SYSTEM

Steffen LANDES  
Institute for Photogrammetry and Remote Sensing  
University of Karlsruhe  
Englerstr. 7, 76135 Karlsruhe  
E-mail: landes@ipf.bau-verm.uni-karlsruhe.de  
GERMANY

Commission V, Working Group V/3

**KEY WORDS:** 3D user interface, VRML, virtual environments, 3D modeling, information systems

### ABSTRACT

By the immense spreading and frequent use of the World Wide Web a powerful new medium has been created. Accompanying this development integration of new technologies like Java and VRML made it possible to build distributed client/server applications for the WWW.

These features had a vital influence on the design of the 3D campus information system of the Karlsruhe University. For this purpose VRML is used to create a graphical front end. It's handling the dynamic display and exploration of 3D data in real-time. Java is used for the communication between Web browser and VRML scene, and for the connection to the database of the information system.

The 3D model of the physical scene – including buildings, streets, etc. of the campus – is created by the fusion of different data sources. In context of this scene geometric representations of abstract data are also embedded. This is done by using physical, real-world metaphors like signs, blackboards, etc. Thereby a significant aid to the user is given in terms of comprehension, navigation, and exploration.

To use this scene as a 3D user interface new concepts – also considering cognitive aspects – are developed to model and represent both types of 3D data. It is demonstrated that the modelling process is not a straight forward mapping of real-world objects to 3D models and should consider the guidelines mentioned in the paper.

### 1 MOTIVATION

In the field of information systems a new type of user interface will be established: it is the 3D user interface, which allows the user to explore interactively a 3D data set in real time.

This development has a vital influence on the design of the 3D campus information system of the Karlsruhe University. The aim of this information system is to

- present the university world-wide,
- provide information for students, staff and visitors and for
- the use as a planning tool.

For this purpose it is obvious to use common Internet standards. Therefore the information system is accessible in the World Wide Web by a Web browser. In this Web browser VRML (Virtual Reality Modeling Language) is used to provide an interactive real-time 3D interface. The connection between a database management system and the user interface is implemented in Java. Java (Arnold and Gosling, 1998) is also utilised to implement system-specific extensions to the user interface. With these standards it is possible to realise a platform-independent system which is supporting distributed client/server applications.

By fusion of different data sources the 3D model of the physical scene (including buildings, streets, etc.) is created. Geometric representations of abstract data are also

embedded in this scene by using physical, real-world metaphors like signs, blackboards, etc. This should give appropriate assistance to the user.

To support an easy and intuitive access to the information inside the 3D scene, this graphical 3D user interface is following the object-action paradigm. This means that a point-and-click interaction with the model of a real-world object should first select this object and then perform an action on it with an immediate visual feedback to the user. This is why an object is caught in a dualism where it is used as

- a real-world object and
- a widget

at the same time. In the modelling process of a 3D scene this dualism has an impact on the representation of the objects, which is leading to new design aspects with an great influence on the modelling process.

### 2 COMPONENTS OF THE 3D CAMPUS INFORMATION SYSTEM

The World Wide Web a powerful new medium. But the Web is not only a great source of information, it is also integrating standards of the underlying delivery mechanism (Rohrer and Swing, 1997). In this field the development of new technologies like Java and VRML are supporting distributed client/server applications for the Web. The advantage of the Web is also the interaction and navigation

style which lets the user following associated hyperlinks and drills down to underlying information by a intuitive point-and-click paradigm.

This easy to use user interface principles and the established new protocols had lead to the implementation of the 3D campus information system for the Karlsruhe University on the base of Web technologies.

## 2.1 User Interface

The design of the user interface for the 3D information system of the university campus is considered as the most important part of the whole information system. If the user is accepting the way how data and information is presented by the information system, the system will be used in a proper way. This problem especially occurs if a user is just working occasionally with the system and not day by day.

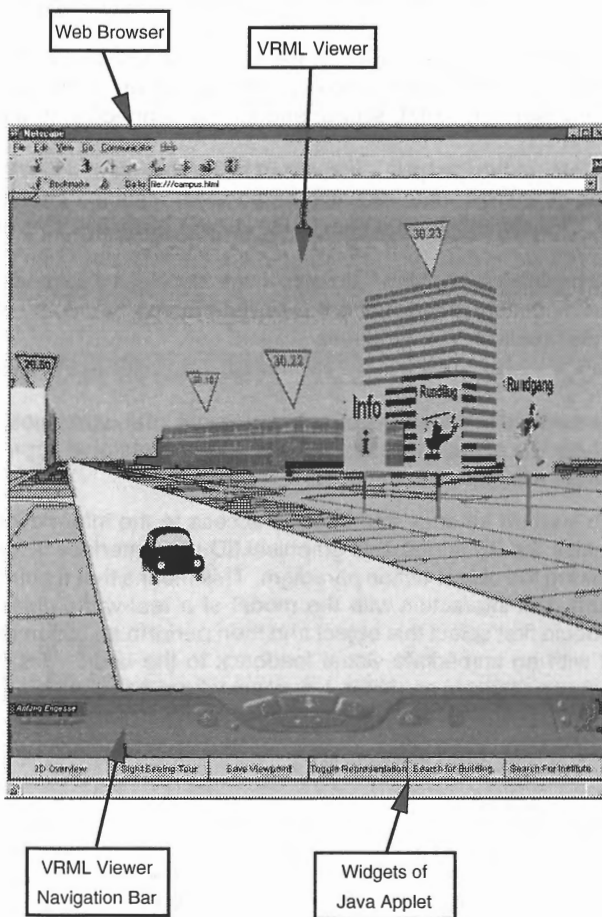


Figure 1: User interface for the 3D campus information system

Therefore the 3D scene has to be offered to the user in a intuitive way. Images or movies can't do this, because they only offer fixed viewpoints or fixed paths and normally no possibilities of interaction. This is why we have to share the whole scene with the user.

**2.1.1 Virtual Reality Modeling Language** On the Internet the Virtual Reality Modeling Language (VRML) is the file format standard for 3D multimedia and shared virtual worlds. It's a simple language for describing the shape of 3D objects and interactive environments. In 1997 it has

been established as a Web standard by the ISO (ISO, 1997). This abstraction by a standardised description language gives us the possibility of sharing scenes in a common ASCII file and provides a high degree of portability. VRML scenes are displayed by a special VRML viewer which is normally running inside a Web browser.

**2.1.2 Classes of Geometric Objects Inside the VRML Scene** For the representation of the university campus several classes of objects are modelled in 3D. The classes for this real-world objects are:

- buildings
- streets
- footpaths
- parking places
- trees

Beside of the models of these "physical" objects we also include geometric representations of abstract data in the scene by using real-world metaphors like signs and blackboards. In terms of comprehension, navigation, and exploration it has been necessary to give a significant aid to the user. With the help of such artificial signs we can easily provide the user with the names of the buildings, streets, etc. In addition to this advantage these signs may be used as a widget to establish links to information about the specific object.

**2.1.3 Object-Action Paradigm** The 3D user interface can be considered as the evolution of the 2D graphical user interface (GUI). This is why some of the properties of 2D GUIs are very useful if they are utilised for the design of 3D user interfaces.

One of these properties is the object-action paradigm. For example, in a 2D GUI we have real-world analogies like windows and icons which can be directly manipulated by a input device (e.g. the mouse). This is put into practice by pointing to an object on the screen and then clicking on this object to select it (point-and-click interaction style). After the object is selected a predefined action is performed with a visual feedback to the user. In a 2D GUI all the objects are on one plane. This leads to the restriction that only objects which are not hidden by other objects can be activated. In a 3D VRML scene this restriction doesn't exist any more. Here the user is able to navigate inside the scene and to move to the desired object for doing a selection.

Although this kind of interaction style has in an heuristic sense just little firm evidence, we should consider design guidelines (Hix and Hartson, 1993), which are valid for 2D and 3D user interfaces.

**2.1.4 Extensions to the 3D User Interface** There are two different classes of interactive queries inside an information system:

- Getting information about a selected object.
- Searching for objects which satisfy specific requirements.

The first class of queries is already implemented inside the VRML specification and is an integral part of the VRML language. As described in section 2.1.3 the object-action paradigm is implemented and can be used by the assignment of an action to a geometric object. Some of the possible actions are: (a) hyperlinks to HTML pages, (b) portals to other VRML scenes or (c) the execution of a Java Applet in the Web browser.

The second class of queries searches for a specified object in the VRML scene. This is not implemented inside of VRML and is one of the disadvantages of the VRML standard (Coors and Jung, 1998). Therefore we have extended the 3D user interface with 2D user interface elements (see Figure 1). In the Web browser not only a VRML viewer but also a Java Applet is running which is able to communicate with the VRML scene via the external authoring interface (EAI, see section 2.2.1). Inside of this Java Applet we have implemented this kind of queries, which can be accessed by common 2D user interface widgets. On first sight these additions seem to be a violation to the 3D approach. But on the other hand, most VRML browsers are not only displaying the pure 3D scene. They also have a navigation bar in 2D. By embedding the VRML viewer and the Java Applets inside a HTML page – which is possible in popular browsers like Netscape and Internet-Explorer – the additional widgets can be arranged around the navigation bar of the viewer to construct a homogenous interface.

Another problem for users in 3D scenes is the so-called “lost-in-space” syndrome. Especially inexperienced users have problems to navigate in 3D space. And sometimes it’s just a undesignedly movement which lets you lose your orientation. This is why a user interface should provide some features to give an assistance to users. One important help is the determination of viewpoints. A viewpoint defines a specific location in the local co-ordinate system from which the user may view the scene. If the designer of a 3D scene provides enough viewpoints for the most important locations in the scene, then the user is able to restart his navigation from one of these well know points. For the user interface of the university campus there is also a widget implemented to save the actual position of a user as a viewpoint. Hence the user is able to save his own points of interests.

The second addition to the user interface is a 2D overview window where the user is able to see his actual position in the campus. Therefore we are displaying the same scene in another VRML viewer form a top view (see Figure 2). In this viewer the same scene is used, but for a necessary simplification only 2D representations of the buildings are used. In this scene a marker is indicating the actual location of the user.

## 2.2 Data Management

As discussed in section 2.1 pure VRML is not enough to implement an entire 3D geographic information system. We have to add more functionality to the system. The main part in this system is a Java Applet. This Java Applet is loaded together with an initial VRML scene by the Web client from the Web server. Inside the Web client the Applet is communicating with the VRML viewer by the EAI (see section 2.2.1). The Applet also handles the connection to a database server via the Web server (see section 2.2.2). With this architecture – which is used for the campus information system of the Karlsruhe University – we have implemented additional functionality to the user interface:

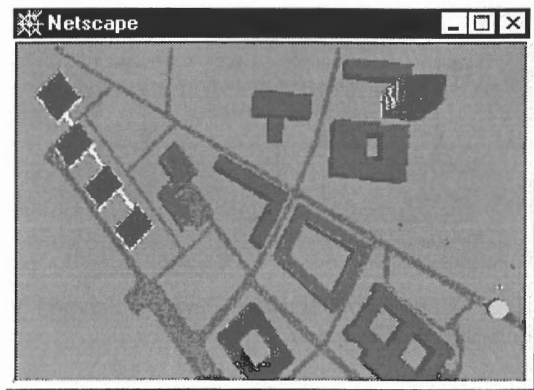


Figure 2: 2D overview window including the location of the user indicated by a marker

*2D overview:* a second VRML viewer is started showing the campus from a top view including the location of the user indicated by a marker (see Figure 2).

*Save actual viewpoint:* the Java Applet is asking the VRML viewer for the actual position of the user inside the 3D scene and is saving this viewpoint in a viewpoint list which can be accessed via the navigation bar.

*Searching for buildings:* the user is selecting a building by its attributes. The result is shown by changing the viewpoint and an animated marker above the building.

*Searching for institutes:* searches for the buildings where the institute is located.

*Sightseeing tour:* offers a guided tour through the university campus.

*Toggle representation:* is a switch to change the representation of buildings from 2D to 3D, with or without textures.

*Show section:* the campus of Karlsruhe University is split into different sections, respectively areas. With this button the user is able to switch on/off the objects inside of these areas.

These functions can be access by the user through the widgets of the Java Applet. The widgets are located next to the navigation bar of the VRML viewer to build a uniform user interface (see Figure 1).

**2.2.1 External Authoring Interface** The Web client is loading an initial VRML scene and a Java Applet. For the communication between the VRML scene and the Java Applet an interface between the two components is needed. For this purpose the External Authoring Interface (Marrin, 1997) is used. It defines the set of functionality of the VRML browser that the external environment can access.

The External Authoring Interface allows 4 types of access into the VRML scene:

1. Accessing the functionality of the Browser Script Interface.
2. Sending events to objects inside the scene.
3. Reading the last value sent from objects inside the scene.

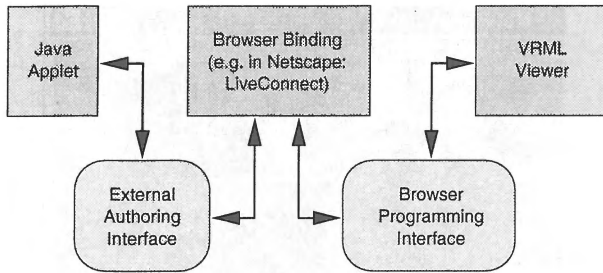


Figure 3: Connection between Java Applet and VRML viewer by the Web browser

4. Getting notified when events are sent from objects inside the scene.

With this set of functionality special actions are implemented like changing the attributes of geometric objects or deleting or inserting objects of the scene.

**2.2.2 Java Database Connectivity** To make a query in the database we need access to a database server which is connected via Internet/Intranet to the Web server. Since Java is designed to work in a secure environment it is only allowed to communicate with the Web server from which it was loaded. Therefore a three tier architecture is implemented. On the client side the Java Applet is using the Java Database Connectivity which is a standard SQL database access interface, providing uniform access to a wide range of relational databases (Hamilton et al., 1997). In combination with the Java Remote Method Invocation (Arnold and Gosling, 1998) the Web server is providing a set of functions that allows the Java Applet to access the DBMS server (see Figure 4).

This is just an interim solution and should be replaced by a professional approach like CORBA (Koschel et al., 1996).

**2.2.3 Data Base Management System** PostgreSQL (Stonebraker et al., 1990) is a free object-relational database management system (DBMS). In opposite to pure relational DBMS it has some additional concepts which extend the system in an object-oriented way. Beside of these concepts it is offering a lot of data types. Especially the geometric data types are useful for geographic information systems. Geometric data types represent spatial objects like point, line, etc. Postgres is also offering a set of functions and operators to perform geometric operations such as translation, determining intersections and distances between objects.

By the use of Postgres as the DBMS for the information system of the university campus, different data types can be managed in one single system. All the different data types of a class of objects are held in a consistent way. For example: the object class "building" is joining descriptive data like ownership, year of construction, etc. with geometric data like height of the building and bounding box. The advantage of this method is the homogeneous way of performing a query on this data set. Therefore it is no difference to search a building by its location or by its ownership. Beside of managing these kind of data the Postgres DBMS is also used to manage the VRML geometry of the objects. Hence the Java Applet – which is running together with the VRML viewer in the WWW client – is able to perform a query in the DBMS via JDBC and will get as a result the geometry

of the selected objects. These objects can be inserted in the VRML scene by the Java Applet, because the Applet is also communicating via EAI with the VRML viewer.

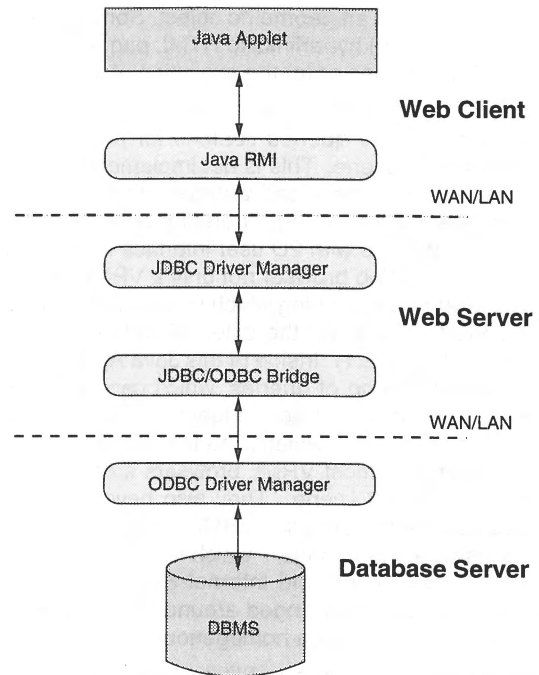


Figure 4: Connection between Java Applet and database via JDBC in a three tier architecture

### 2.3 Data Modelling

In the campus information system several types of data are integrated:

- geometric data
  - models of real-world objects
  - geometric representations of abstract data
- descriptive data

This data is managed in the DBMS described in section 2.2.3.

**2.3.1 Geometric Data of Real-World Objects** We consider the area of the university campus as flat because the city of Karlsruhe is located in the Upper Rhine Valley. This is why we are digitising the contours of the streets, footpaths, parking places directly from an existing map of the campus. From this map we also get the ground plans of the buildings. The height of each building is gained from the analysis of aerial images and/or from the digital elevation model of a laser scanner flight. The solid 3D models of the buildings are created by an orthogonal projection of the ground plan using the height of the building as the distance of the projection. After some refinements in a CAD system the models are converted to VRML. In VRML the façades of the buildings are mapped with textures. These textures are gained from images of the buildings after a projective transformation (Landes, 1996).

**2.3.2 Geometric Representations of Abstract Data** In terms of comprehending a 3D scene it should be extended by geometric objects which have no physical equivalence in the real-world scene. The instances of this class of objects are signs, blackboard, etc. Most of these items are generated in a 3D modeller and were integrated into the VRML scene. In Figure 1 one can see the building numbers levitating above the corresponding buildings. On the right hand side of the scene there are several signs which can be activated to perform actions like a sightseeing tour.

**2.3.3 Descriptive Data** Descriptive data already exists and is collected by the university administration. This data is stored in a MS-Access database and can therefore easily imported into the Postgres database.

### 3 DESIGN ASPECTS

The pure photorealistic 3D model of a real-world scene might be a nice representation but if you want to use it as a 3D user interface for a GIS it has to be modified. There exists a dualism for geometric objects. On the one hand they are representing physical objects and on the other they are representing widgets. This is why they have to be recognised in both ways. Another problem is to offer each user adequate help for the navigation inside a virtual environment. Therefore additional elements have to be integrated in the model of the scene.

Every designer of 3D worlds should also consider the cognitive abilities of users exploring 3D worlds. This is why an object doesn't have to be modelled absolute realistically. In most cases it is enough to represent the characteristic elements of the object.

This aspects are leading to the following empirical guidelines:

- Use real-world analogies as much as possible (to convey real-world objects and real-world information to the user).
- Keep the visual representation as simple as possible (scene not more complex as necessary).
- Show different views of the object (user-selectable complexity of the scene).
- Substitute missing object details by the use of textures.

And by the way, these guidelines help to avoid performance problems.

### 4 CONCLUSION

Designing and implementing a 3D information system is a creative task. It requires a lot of experience because it is not a straight forward mapping of real-world objects to 3D models. The essential part inside this virtual environment is the user, because it is important for a 3D user interface that the user is still able to apply his cognitive abilities which he had learned outside in the real world. Accuracy and completeness is not the superior goal in the modelling process. It is more important that the user can interact in a intuitive way inside the artificial scene.

Especially the architectural photogrammetry has to consider these problems to give proper solutions to the VR market.

### REFERENCES

- Arnold, K. and Gosling, J., 1998. Java Programming Language. JavaTM Series, Addison Wesley Longman.
- Coors, V. and Jung, V., 1998. Using VRML as an interface to the 3d data warehouse. In: VRML'98 Symposium, Monterey, Calif., USA.
- Hamilton, G., Cattell, R. and Fisher, M., 1997. JDBC Database Access with Java: A Tutorial and Annotated Reference. JavaTM Series, Addison Wesley Longman.
- Hix, D. and Hartson, H. R., 1993. Developing User Interfaces. John Wiley and Sons, Inc., NY.
- ISO, 1997. Information technology – computer graphics and image processing – the virtual reality modeling language. International Organization of Standardization. ISO/IEC 14772:1997.
- Koschel, A., Kramer, R., Nikolai, R., Hagg, W. and Wiesel, J., 1996. A federation architecture for an environmental information system incorporating GIS, the World Wide Web, and CORBA. In: Proceedings Third International Conference/Workshop on Integrating GIS and Environmental Modeling, NCGIA, Santa Fe, New Mexico, USA.
- Landes, S., 1996. Architectural photogrammetry and picture processing for acquisition and documentation of a brazilian town ensemble. In: International Archives of Photogrammetry and Remote Sensing, Vol. XXXI, ISPRS, Wien, pp. 309–312.
- Marrin, C., 1997. Proposal for a VRML 2.0 informative annex: External authoring interface reference. <http://www.vrml.org/WorkingGroups/vrml-eai/ExternalInterface.html>.
- Rohrer, R. M. and Swing, E., 1997. Web-based information visualization. IEEE computer graphics and applications 17(4), pp. 52–59.
- Stonebraker, M., Rowe, L. A. and Hirohama, M., 1990. The implementation of POSTGRES 2(1). In: IEEE Transactions on Knowledge and Data Engineering, IEEE.