AN APPROACH FOR A CAMPUS VISUALIZATION SYSTEM

Bernd Hetze, Gerald Hübsch, Horst Kohlschmidt

University Computer Centre
Dresden University of Technology,
01062 Dresden
Germany

{bernd.hetze|horst.kohlschmidt}@urz.tu-dresden.de http://www.tu-dresden.de/~hetze/visu.html
gh7@inf.tu-dresden.de

ABSTRACT

Intense work is spent especially in the fields of city planning and tourist information to examine possibilities for combining generic information with graphic data to enhance computer based information desk systems. This article presents an approach towards the creation of a campus information system. This campus information system shall allow the user to navigate and select information with the help of a user interface that integrates possibilities to use 2D and 3D interaction concepts simultaneously. The VR entities are accessed by user interaction via the render window.

Keywords: visualization, desktop VR, interaction techniques, 3D Modelling, VR walk through, computer graphics

1. PROJECT OUTLINE

The Dresden University of Technology is running a project to establish a campus information system. This system is being developed by the Visualization working group in co-operation with the computing centre of the Department of Civil Engineering and the Institute of Cartography.

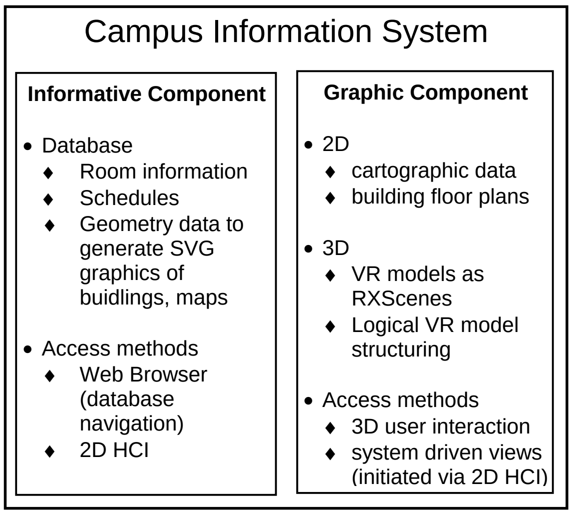

figure 1 – System Overview

The final stage of the project is intended to provide mutual access to data from both components, informative and graphic, in one user interface (fig 1). Textual and 2D information – such as schedules etc. – retrieved by accessing an ORACLE database with the help of the KOPERNIKUS Facility Management System, is displayed as HTML content in a commercial web browser [2]. The web browser is configured to start up automatically when textual information is requested.

Any graphic information is displayed in a 2D respectively 3D model of the campus area. The 2D information is based on auto-generated W3C Scalable Vector Graphics (SVG). The 3D model format as well as the C++ based graphical VR API is provided by the Realax AG [8]. The project centres at the development of 3D interaction concepts.

2. THE DATA MODEL

This section describes the data model defined by the Realax API and its adaptation to the requirements of the campus information system.
Any 3D model created with Realax is based on a structure which is represented by a n-ary tree. This tree comprises all elements and parts of the model. The constituents of the tree, their semantics and the valid structure of the tree are defined by the Realax

API. An instance of this tree is defined as the Object Hierarchy of a Realax model.

The constituents of the Object Hierarchy are

- **Group Nodes,** used to structure the Hierarchy tree (similar to the directories in file systems)
- **Polygon Nodes,** containing sets of polygons that form logical groups in the model and providing access to the geometry of the polygons it contains
- **Dynamic Coordinate System (DCS) Nodes,** allowing the definition of animations in the model
- **Level of Detail Nodes (LOD),** defining the complexity of the model rendering depending on the viewer's position
- **Spline Nodes** to predefine the movements of objects in the model
- **Light Nodes** to allow the definition of lighting effects
- **Switch Nodes** to dynamically manipulate the visibility of subtrees in the Object Hierarchy
- **Camera Nodes** to predefine views of the model
- **Acoustic Nodes** to include audio information as part of the model

Realax only defines the way the Hierarchy tree has to be structured physically. Structuring the tree logically is solely up to the creator of the particular model. This includes the definition of the nodes' names and other annotations that can be set for the components of the Hierarchy.
The Realax API facilitates access to the Object Hierarchy by defining methods to traverse and scan parts of the tree. Therefore it implements an event driven approach. The actions that are to be taken on the triggering of an event can be defined by the programmer by implementing a defined interface. The implementation called is passed the instance of the node that triggered the event and that is hence available for manipulation.
Using these possibilities, we defined the following structure for our model of the campus area of Dresden University of Technology.
Polygon Nodes are used to group polygons having a logical relationship. This means they are components of one building in the model. The names of the nodes are chosen to be unique within the model. They correspond to the name of the "real" building.
Group Nodes are used to represent the position of the building as it may be defined by the postal address of the building or on a city map. The children of these nodes are Polygon Nodes aggregating the geometry of the buildings found at the address/street represented by the Group Node.
This structure defined for our model builds the virtual bridge from the campus model to the user of the information system, keeping the user as far as possible away from the difficult task of complex navigation through the virtual environment. The user may select a building in the model and is given its name and address by mapping the user selection to the polygon node containing the selected Polygon.
If the user wants to view a building knowing only the name of it, he must only enter the name and is given a 3D view of the building which is calculated from the geometry information of the node with the name the user entered.
Moreover, expanding and deleting parts of the model is simple. Only another Polygon Node must be added to the Object Hierarchy to integrate a new building. As the structure of the information is known to the information system, it will be able to include the new data without changes.

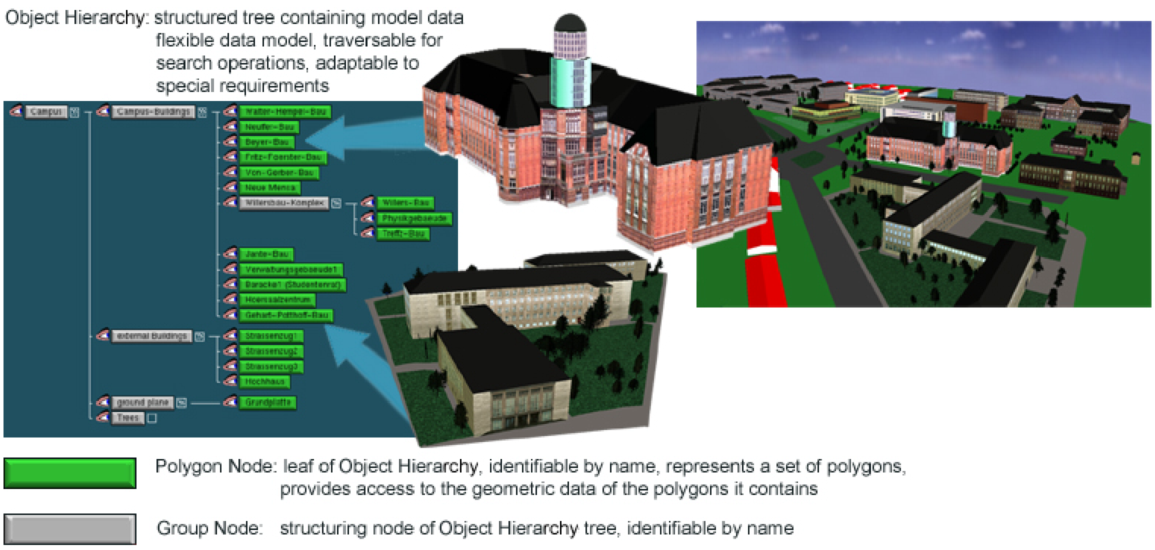
Object Hierarchy: structured tree containing model data
flexible data model, traversable for search operations, adaptable to special requirements
Campus
Campus-Buildings
Walter-Hempel-Bau
Neuffer-Bau
Beyer-Bau
Fritz-Foerster-Bau
Von-Gerber-Bau
Neue Mensa
Willersbau-Komplex
Willers-Bau
Physikgebaeude
Trefftz-Bau
Jante-Bau
Verwaltungsgebaeude1
Baracke1 (Studentenrat)
Hoersaalzentrum
Gehart-Potthoff-Bau
external Buildings
Strassenzug1
Strassenzug2
Strassenzug3
Hochhaus
ground plane
Grundplatte
Trees
Polygon Node: leaf of Object Hierarchy, identifiable by name, represents a set of polygons, provides access to the geometric data of the polygons it contains
Group Node: structuring node of Object Hierarchy tree, identifiable by name

figure 2 – demonstrative view of the data model used

3. SOFTWARE ARCHITECTURE

The software architecture used for the implementation of the Campus Information System (CIS) is based on the *layer architecture* design pattern (figure 3).
This architecture allows us to clearly separate the user interface, the implementation of the core functionality of our Campus Information System and the graphic and network components of our system.
The different layers communicate using the well-defined interfaces of the Realax Scene API to access and navigate the 3D model data and the network API provided by the operating system.

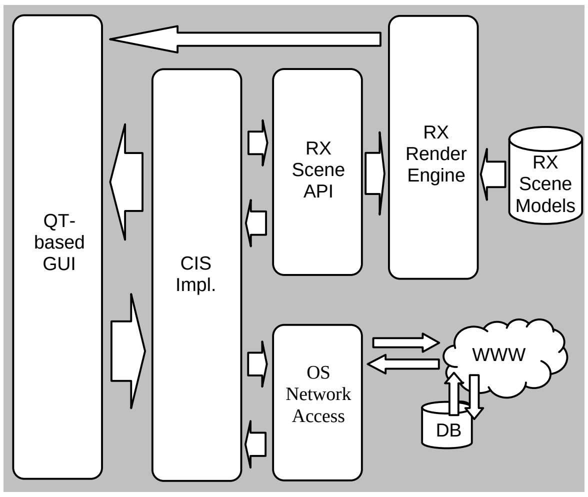

figure 3 – overview of the Campus Orientation System's software architecture

The *Realax Scene API* layer provides functionality to access and manipulate the VR components of the system. This comprises the transformation of the viewer's position in the model to a position which was calculated before and the determination of user's activities related to the 3D window, such as the mapping of a user selected polygon to the name of the Polygon Node containing the selected polygon.
The *network access* layer is used to retrieve information from database servers and other sources that might be required to satisfy a user's request for information.
The *CIS* layer implements the core functionality of the information system. It listens to user activities on the user interface and converts them into calls on the designated functions. It is further responsible for handling and converting results of network queries into a user readable form and to propagate instructions received via the network to the Realax Scene API layer. Such instructions may be commands for highlighting buildings that are referred to in the data received from the network or setting the user's viewpoint to a position calculated by the CIS layer.

The *Realax Render Engine* is OpenGL based and responsible to render the Realax VR models. There is no direct access to this layer through the Realax Scene API. We only mention it to show all components involved.

4. USERINTERFACE/INTERACTION

Many cities and universities offer virtual "walk throughs" in form of a mostly VRML based 3D model on their web sites [3,4,5,6,7]. The possibilities of interaction are determined to a great extent by the capabilities of the VRML Viewer used. The approach described here is based on the utilisation of a VR system that allows direct access to the Object Hierarchy of the model it displays.

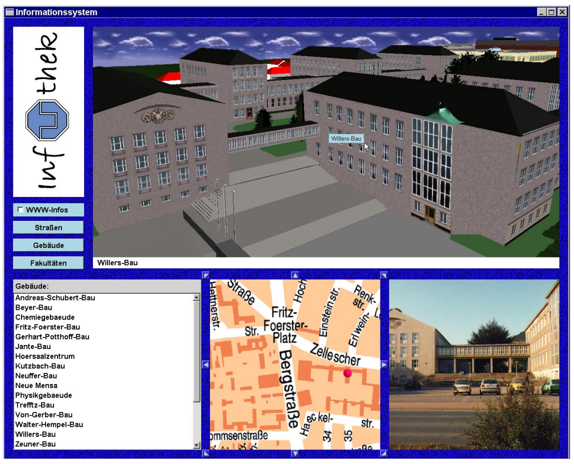

figure 4 – User Interface

Figure 4 shows the present state of the user interface. The classical usage of such a model is supporting the search for a specific building or institute.
A building is selected by the user with the help of the buttons and the listbox shown in figure 4. The first step made by the system is to locate the building in the model by a search operation for the related Polygon Node in the Object Hierarchy. The view in the render window, the content of the map window and the photography displayed is changed. All windows will then display information about the building or the building where the selected institute is located respectively. This requires setting the viewer's position as well as his perspective in the render window, creating a representative view of the building in the window's centre as well as of some parts of the area surrounding it. Therefore, the position of the viewpoint (its distance from the balance point of the building) is calculated with respect to the geometric parameters of the building. The perspective, however, is stored as an attribute of the building in the model, allowing it to be displayed in a characteristic front view. The algorithm used

also allows the generation of views from arbitrary positions. In the map window, a marker is set to show the building on the map. The position of this marker is calculated from the building's coordinates in the model. If necessary, another clipping of the map containing the building is shown.
The photo window shows a digital photograph of the selected building. The status line displays the name of the building displayed.
If the checkbox 'www-infos' is activated, a browser window displays additional information about the building or institute retrieved from a database or from the institute's home page.
Access to the interactive graphics functions is provided by the 3D render window and the 2D map window.
The render window and the map window implement the identification of a building or street by the mouse-pointer. This information is retrieved using a selector function integrated in the Realax Scene API. This capability is used for

- implementing the "mouse over" function, that is the name of the building or street the mouse pointer is positioned at is shown in a separate textbox next to the mouse pointer
- selecting buildings by double-click to acquire related information
- initiate the presentation of the routes linking the selected buildings.

5. RESULTS AND FUTURE WORK

The functionality of the first prototype was tested using the Virtual-Reality-Software-Interface based on REALAX 5.0 for Windows NT 4.0, the relational database Oracle 8.1.5i Win NT 4.0/Solaris 2.7, an Apache 1.3.14 web server, the Servlet Engine Tomcat Version 3.2.1 (Apache extension), the Oracle XSQL-Servlet 1.0.4.1 for dynamic database access and the CAFM-System Kopernikus.
The partial VR model currently contains 20 of 90 buildings on the Campus in total. The VR model has a complexity of 9k polygons. 30 MB of texture in PNG format are used. The model used for our Campus Information System is an adapted part of a complex model containing all buildings of the TU Dresden that was developed for the use with animation software. In the present state, the model used is a fixed, non-dynamic representation of the campus area. It is planned to dynamically generate 3D model data using an extended version of the database currently used for the 2D components [1].
The identification of buildings in the 3D scene was tested successfully. It forms the basis for the mouse-over function and the selection of buildings. Preparatory work for the integration of routes connecting two buildings was successfully finished and led to the desired results. Realtime movements in the model are possible and will be used to present the suggested route to the user.
The next step will be the integration of rooms inside buildings and of the ways to reach them. For this task, the model must be extended by the rooms' geometry. This will cause a sharp increase in the amount of polygons.
The interprocess communication is planned to be migrated from basic Win-Socket implementation to a CORBA distributed object environment.
It is the authors' opinion that the integration and interaction of 2D information which is available as web content and complex 3D data into one information desk system will help to increase the possibilities and the acceptance of such information systems.

REFERENCES

[1] Karkola, C.: Untersuchungen der Anwendbarkeit des STEP "Applikationsprotokolls" 225 auf das 2-1/2D Geometriemodell des CAFM-Systems KOPERNIKUS in Hinblick auf die Erweiterung zur Verwaltung von Gebäudeelementen, Diplomarbeit TU Dresden, Fakultät Bauingenieurwesen.

[2] Benitz, S.: Entwurf und Realisierung einer internetfähigen Anwendung zur Abfrage von Geometriedaten aus dem CAFM-System KOPERNIKUS zur Präsentation von Raum- und Bauwerksdaten für ein Leitsystem der TU Dresden, Diplomarbeit TU Dresden, Fakultät Bauingenieurwesen.

[3] Web 3D Campus Information System CISKA of University of Karslruhe, website: http://www.uni-karlsruhe.de/~Campus3D,

[4] Facility and Infastructure Information FIIS, System of the University of Kansas, wegsite: http://www.ku.edu/~fmkuhtml/fiis/projsch.htm,

[5] VR University 3D Map U3D of University of Tasmania, website: http://www.comp.utas.edu.au/kca300/2000/uni3d.html,

[6] On-line UCLA Virtual Campus ucla center for digital art, website: http://www.cda.ucla.edu/caad/virtual/vucla.htm,

[7] The 3D Virtual Campus,Tour of the University of Queensland, website: http://www.geosp.uq.edu.au/projects/Urban_Landscape_Lab/cover.htm

[8] Realax AG Karlsruhe http://www.realax.de