Skybox as Info Billboard

Jana Dadova [*]
Faculty of Mathematics, Physics and Informatics
Comenius University
Bratislava

Abstract

In this paper we propose a new way of information mapping to the virtual skybox. This mapping brings improvement to the existing virtual world's projects with providing relevant information instead of a static or animated skybox. Information are in a visible chronological correspondence. We explain background terms as well as previous similar projects. Application of the Ajax3D technology is described as a powerful method to provide the connection between database and X3D scene. Moreover, except the visible order also solution for the user interaction is provided. Everything is shown in the SkyBoard application, that can be found in the page http://www.dzana.sk/projects/bakalarka/index.html. This approach fills unused or free space in the virtual world. Moreover it is possible to use this method for educational or informative purposes, where additional information are provided.

Keywords: AJAX3D, Virtual World, Skybox, Education

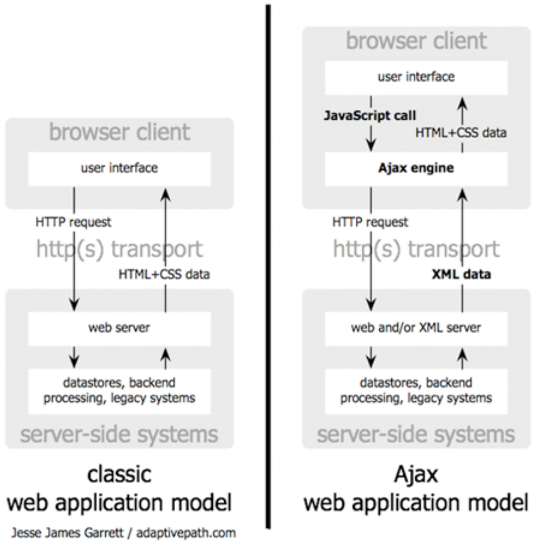
1 Introduction

Information are everywhere nowadays. On the billboards, advertisements, TV's, walls, etc. On the other hand, virtual worlds either as social networks like Second Life [Lin 2003], or historical reconstructions like Virtual Hearth of Central Europe project [VHC 2000] become more famous every day. Nevertheless these worlds still have free space that can be filled with the information. In the real world there is tendency to effectively use everything. Why not combining all of this together? We propose effective combining these methods. Our approach fills relevant information in the skybox space of a virtual world. These information are ordered in a way that the chronological order is visible.

There are approaches that deals with the information processing in the virtual world project, or scenes. In this paper only those that process Virtual Reality Modeling Language (VRML) or X3D formats are interesting. Reason for this assumption is that there have been already created worlds in VRML format. Some of already existing approached use Java programming language that allows use and cooperation with database, but are complex and hardly reusable. Therefore Ajax3D solution is more proper for this approach.

Moreover also some problems arise. Either the problem with the cooperation between the X3D scene and database of entities or with the display of the entities in the desired way. Another problem is user interaction. Successfully all of these problems are solved, but there still left part that could be better investigated.

This paper is structured as follows. At first, brief description of the basic terms is provided. Afterward previous and relevant work is discussed. Finally contributions and the future work possibilities are proposed.



Jesse James Garrett / adaptivepath.com

Figure 1: The traditional model for web applications (left) compared to the Ajax model (right)

2 Background

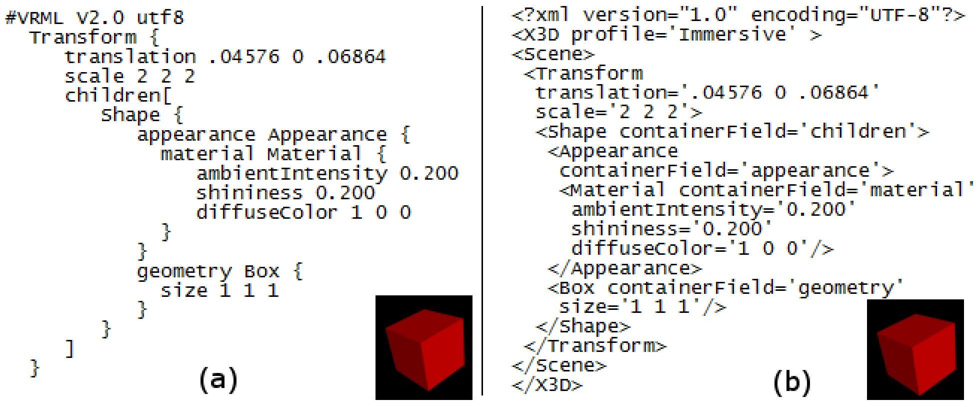
Several terms need to be explained before further method description. First of these terms is Ajax as a method of creating dynamic web pages and another is X3D that is related to the scene description.

2.1 Ajax

Asynchronous JAvascript and XML (AJAX) is a method of the creating of the dynamic web pages [Garret 2005]. The method is new in the philosophy, but the technologies used are well know for the web designers. This method allows to build interactive internet applications that are real-time (user do not need to wait for the response). Actually, the difference between traditional way is only in the one command:

```
XMLHttpRequest()
```

[*] e-mail: dadova.jana@gmail.com



```
#VRML V2.0 utf8
  Transform {
     translation .04576 0 .06864
     scale 2 2 2
     children[
        Shape {
           appearance Appearance {
             material Material {
                ambientIntensity 0.200
                shininess 0.200
                diffuseColor 1 0 0
             }
           }
           geometry Box {
             size 1 1 1
           }
        }
     ]
  }
```

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive' >
<Scene>
 <Transform
  translation='.04576 0 .06864'
  scale='2 2 2'>
  <Shape containerField='children'>
   <Appearance
    containerField='appearance'>
    <Material containerField='material'
     ambientIntensity='0.200'
     shininess='0.200'
     diffuseColor='1 0 0'/>
   </Appearance>
   <Box containerField='geometry'
    size='1 1 1'/>
  </Shape>
 </Transform>
</Scene>
</X3D>
```

Figure 2: (a) VRML source (b) X3D source - scene in the right bottom corner

This command is used for the communication between Ajax engine and server script. As the Ajax engine is usually used JavaScript, and we also used this programming language, because of it's simple cooperation with the X3D scene. In this method there is not needed to download whole new page with the server script response. In the older, or more traditional way the whole browser is reseted and therefore also X3D browser will be deleted. This reset will cause the bringing back default values. In that case, in our virtual world, view point will be bring back to the first.

X3D scene is embedded to the web browser as Document Object Model (DOM). There are well known approaches of working with this model. The difference is shown in (Fig. 1). Following commands are used for the cooperation between X3D browser and the Ajax engine:

- `browser.createX3DFromString()` - creates scene from the input string
- `browser.loadUrlS ()` - creates scene from the input urls, or file
- `context.getRootNodes()` - as a result gives an array of the rootnodes
- `context.removeRootNode()` - removes the node from the scene and the scene graph
- `context.addRootNode()` - adds new rootnode
- `addFieldEventListener()` - add listeners to the objects, so the browser observes attempt to interaction

These few commands are sufficient for the cooperation between JavaScript and X3D browser. This cooperation is essential, because this is the bottleneck of all other communications.

2.2 X3D

EXtensible 3D Graphics (X3D) is ISO standard for 3D scene description, whose main purpose is to be displayed in the web browser. This allows to build robust, interactive virtual worlds applications. This standard originates in the VRML standard but brings XML syntax that is popular and widely used in the creation of the web pages. Instead of the fact, that the scenes used in our final application SkyBoard are saved in the VRML format, X3D can be used. Because of the back compatibility of these two formats, that means that in the X3D browser can be VRML file uploaded. The difference between these two formats can be seen in (Fig. 2) Moreover, to be able to display the X3D scene, the plugin needs to be installed and the whole scene needs to be embedded in the web browser window with the following line:

```
<embed TYPE="model/x3d">
```

After successfully fullfiling these two conditions no other applications are needed.

3 Previous Work

Virtual reality transfers from books and computer games to virtual worlds over a past years. These worlds extends social networking to the new level of understanding. Except social networking, virtual worlds could be also useful and powerful tools for informational and educational purposes.

Essential requirements for interactive virtual world include:

- objects in a scene look like three dimensional
- user interacts with the scene by virtual hand of avatar
- response is real-time and framerate is at least 25 frames per second

as is stated in [Zara 2000]. The most important requirement for real-time applications is the last one, but in our case this can be affected by the speed of internet connection. Also other aspects depend highly on the browser used.

Interesting projects related to this work and their relevance are described in the next subsections.

3.1 Related Projects

In the last years the expansion in the area of virtual worlds has increased. These improvements bring interest to a wide public.

3.1.1 Active Worlds

Active Worlds is one of the first projects in the field of virtual worlds. Interaction is made by sensitive objects that reacts on the avatar's hand. Moreover transportation here is also possible, through these sensitive objects. After clicking on the object user is transported on the designed place. This is applicable when whole

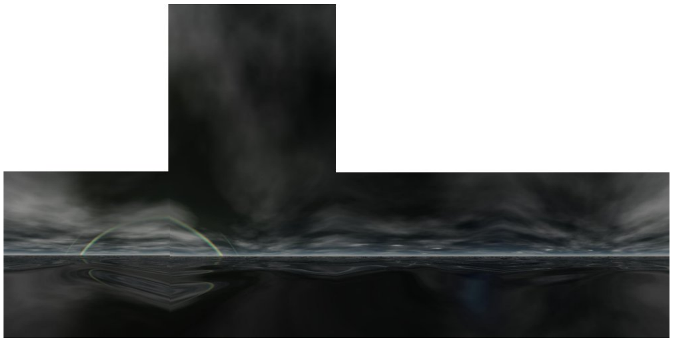


Figure 3: Texture for skybox used in Alpha World

scene need to be changed. On the contrary in our world, only view is rotated, because there is not necessity to change whole environment. User therefore could be confused when moving somewhere else.

Basically, this world consists of many, that are created by users. Users could contribute in the building creation and the whole visual representation. This is first attempt of virtual world, therefore also skybox is conventional. It is usually statical image that depends on the light change (Fig. 3) [Qvortrup et al. 2002]. From education point of view there exists also Active Worlds Educational Universe. It is possible to develop new way of distant education with cooperation of teachers and students [Act 2008]. The educational process is created in virtual buildings, but the way is similar to one in the real world. Therefore our method is in this way different that can not be used by placing similar approach in real world. There is imposed potential given by virtual world.

3.1.2 Second Life

One of the most popular approaches is Second Life (SL) project that was published in 2003 and was created by the Linden Lab [Lin 2003]. It is also user-created and apply idea of distant education [Ritzema and Harris 2008].

As most of the similar projects, also SL tries to be as realistic as possible. Unlike our application, where we try to extend virtual reality with additional information provided. Basically, in the conventional approach of virtual world stands skybox as a statical image or animation. On the other hand, in our application, enhancement is provided by filling the skybox space with relevant information.

3.1.3 Twinity

Twinity world is the youngest from the mentioned. The main goal of this world is to bring models of famous building from real world [Met 2008]. Abstract and real meshes up here in virtual space. Moreover our approach is adapted exactly for purposes of this world, where information add interest to the real buildings.

3.2 Related implementation approaches

Firstly, there are limitations because of given requirements:

- available access through web browser window
- use of existing VRML models

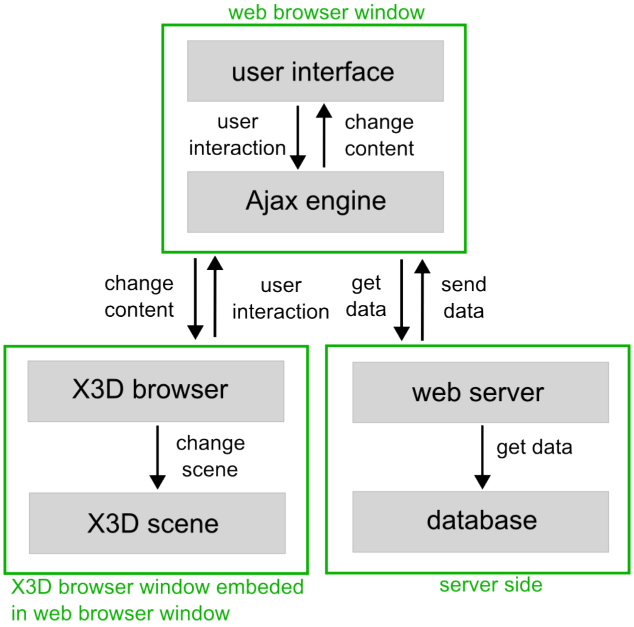


Figure 4: Communication between web browser, X3D browser and the server

- interaction
- database of entities
- visible chronological order

Therefore solution has to satisfy all of them and that leads to problems. The main problems from the view of implementation are the connection to database and the display of entities. The connection between the scene representation and the database of entities. Following approaches discuss this topic.

3.2.1 Virtual Reality Searching Machine

Virtual Reality Searching Machine is name of the application that was introduced in 1999 on the Central European Seminar on Computer Graphics (CESCG). The main goal is to combine data stored in the database and 3D model stored in VRML file. This approach is interesting in a way of combining technologies for database and VRML manipulation [Hanuska 1999]. Implementation is realized in a Java programming language and is displayed in the web browser window through Java Applet. The disadvantage of this method is dependence on the implementation. Java Applets are usually used for the specific purpose.

3.2.2 Ajax3D

Ajax3D is an approach that was presented on the SIGGRAPH 2006 as a White Paper [Parisi 2006]. It introduces a way of building interactive and dynamic virtual worlds and combines strength of AJAX (see Section 2.1) and X3D (see Section 2.2). Basically, X3D scene is embedded in a web browser creating a Document Object Model (DOM) that allows interaction of model with a web document. Therefore scene is controlled by Scene Access Interface (SAI). This communication is show in the (Fig. 4). Both these models are independent of the programming language, but in the Ajax3D approach Javascript is used for the control. In the paper mentioned above there is not described any particular way of cooperation with database. However, data retrieval used by Ajax afford way of co-operating with server-based programming language, for instance PHP.

3.2.3 Embodying and Extracting Data in Web3D Models of Proposed Building Developments

There are many ways of providing extra information in the X3D scenes [R. Hetherington 2007]. This approach is one of them. Ajax3D model is used here for adding extra information to the existing models of buildings. Data are saved in the database. This approach is similar to ours in a way of connecting between database and X3D scene, not in a way of displaying the entities.

4 SkyBoard Application

SkyBoard is a name for the application that we developed to show our approach in practice. We have used two churches that have been artificially added to the scene, which means that are not in real world near each other. One of this models was created by Peter Kan [Kan 2007].

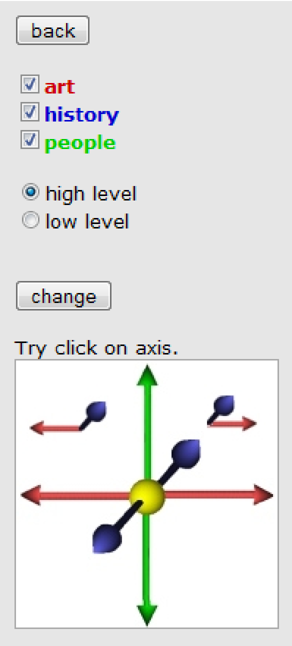
The largest difference between our proposed solution and the one used in all previously described virtual worlds (see Section 3.1) is in access. All three are accessible through internet, but downloading the special browser is needed. They are not displayed in the web browser window. Moreover, AJAX3D approach was chosen, because our application should be easily combined with existing projects instead of specific implementation provided by the Java solution.

Facts that are included in the entities are exported from the book [Gombrich 2006]. Therefore these entities in our example application are not really relevant for the place they are associate to. In the next section is detailed description of the implementation we have used specific for the target groups.

4.1 Target Groups

Different groups of people have different requirements. It is necessary to adapt our application for the special needs of each user. Difference is in the amount of entities displayed. Therefore each user can customize the environment depend on the needs. The groups mentioned above are these:

- **students** - want to easily learn relevant facts usually because of school purposes. *Option:* display all relevant entities.
- **historical enthusiast** - want to have entities in a transparent order and want them as many as possible. *Option:* use high detail option and coloring of the categories.
- **sightseers** - are more interesting in visual part of the city and less about the fact. *Option:* choose option less detail.
- **others** - not interested in the facts. *Option:* for these is there an option of not clicking on the sensitive object and no entities will be displayed.



back

☑ art
☑ history
☑ people

● high level
○ low level

change

Try click on axis.

Figure 5: Skybox navigation

4.2 Problem Definition

As is written before, some problems have arisen in our work (see Section 3.2). These problems include:

1. connection between entities in database and virtual world
2. interaction
3. displaying of entities

Solutions for the first problem are given in the previous section. Moreover in our application the Ajax3D approach is applied for it's flexibility and compatibility in re-using. Although this problem is well discussed in the previous section and solutions are found, there had to be build specific solution.

The second problem also can be solved with Ajax3D approach that allows cooperation between the browser (web and also X3D) and the server side program. Interaction needed in our application is defined as objects sensitive for the user response in the X3D scene. These objects needs to be defined also in the X3D, because Ajax3D handle only response of the actions. Moreover these need to be observed by the X3D browser. To be observed by the browser the entities has to be find in the node tree. The problem is that the nodes from this tree are accessible by the names only when they are rootnodes and are not dynamically created. Therefore the entities that are dynamically created has to be accessed by the position in the tree.

Entities that are going to be displayed in the skybox space need to have visible chronological order. This is not a trivial problem. Although, there are solutions provided, they have not been tested in this specific implementation. Usually chronological facts are displayed in 2D applications, but here we can use the whole potential of the third dimension.

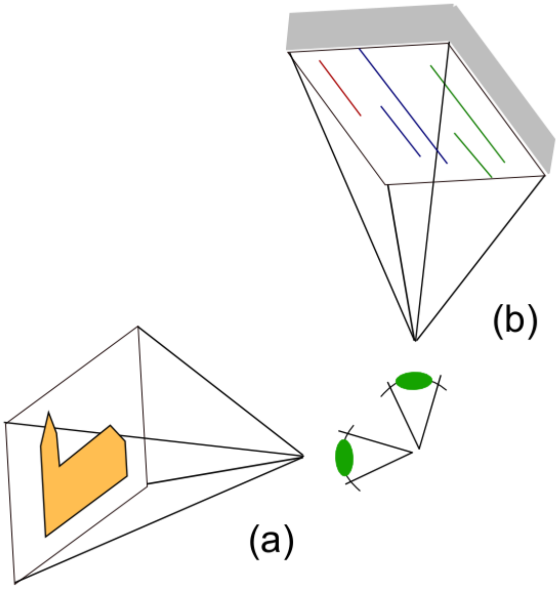


Figure 6: (a) standard view (b) rotated view to the sky

4.3 Technology

In the following section the technologies used for the SkyBoard application are described.

4.3.1 Web Page

Web browser window is the first part of the application seen by the user. The whole virtual world is embedded in the scene, but there are still some controllers left out of the world. These is status bar for the better orientation of the user, navigation window (Fig. 5), and text window. In the navigation window user can navigate whole skybox with the entities and create events that are processed with the Ajax Engine. On the other hand the text window is displayed only when specific entity is interacted and detailed information are displayed here.

Additionally, the Ajax Engine is also part of the browser window, because it is implemented in the JavaScript programming language which is Client Oriented. It is responsible for handling of the events either from X3D browser, or objects from the page.

4.3.2 Scene

Firstly the scene is initially uploaded from the VRML file to the X3D browser. As is written before (see Section 2.2) this is possible, because X3D format is back compatible with VRML format. There is not used VRML browser, because Ajax3D approach uses X3D and also this is a newer format. Therefore it is better in the perspective view. By this substitution it is easy to exchange our world to the one defined in another VRML/X3D file.

After initial upload of the scene, user can move and navigate in the world by the controllers provided by each browser as in normal X3D scene. Unlike the common X3D scene in ours are yellow balls (SkyPorts) near interesting places. These SkyPorts start to pulse, when virtual hand is on them. This indicates response for user interaction. After clicking to the SkyPort, user is smoothly teleported to the new ViewPoint, that has the same coordinates but rotation directs to the sky. This teleportation is provided by the X3D browser. Therefore it looks like the user only rotates virtual head of avatar and this is not confusing. It is also possible to rotate back, but this option is not supported by the X3D browser (Fig. 6). The view point is changed with rotation back to the scene and again with same x, y coordinates.

Moreover it is possible to interact also with the displayed entities by clicking on them. This is than followed to the Ajax engine and detailed information is displayed. The request to the server side script has to know which entity was interacted. As was said before this is not trivially done (see Section 4.2), because these entities are not rootnodes. Therefore it is not possible to access them by the node name. We solved this problem, by accessing the skybox that is the last node, because it was the last one dynamically created. Afterward when entities are created, in the touch sensor is information, about the consecutive number of the entity in the display pass. With these information, entity can be easily found in the database and specific information can be returned to the Ajax engine.

4.3.3 Server and Database

Finally, when the scene is uploaded and the interaction is provided by the user. The request is sent to the Ajax engine. This request is then send to the server script as is described before (see Section 2.1). In our application the server side script is implemented in the PHP programming language that has intuitive cooperation with the database. Here we have used MySQL database for it's robustness.

PHP script has as input the position of the actual viewpoint. Afterward with this position, the specific SkyPort can be identified from the table of SkyPorts. Moreover with this identification relevant facts are extracted from the table of facts. Entities are created as nodes of the X3D scene filled with these facts. This whole part of the world is then positioned above the avatar and centered. Processed output from the PHP file is sent back to the Ajax engine.

4.4 Requirements

As every web-based application, our also have some client and server requirements. Nevertheless almost every requirement is common in the standard web applications. Therefore, instead of X3D played, nothing more has to be added to the server, or client side.

4.4.1 Client

Firstly our application is displayed in the window of the web browser, therefore user requirements starts with the web browser. Firefox 2.0 is preferred, because the application was tested in this environment. Ajax engine as well as HTML page are another client-oriented parts, but these are provided by the browser and user does not care about it.

On the contrary the X3D browser is more interesting part to be obtained by the user. This browser has to be downloaded and installed as the plugin to the browser. Theoretically our application should be runnable with any X3D browser, but is recommended Vivaty player, because the application was tested with it. Vivaty player is free and easily to use with intuitive navigation in the scene [Viv 2008].

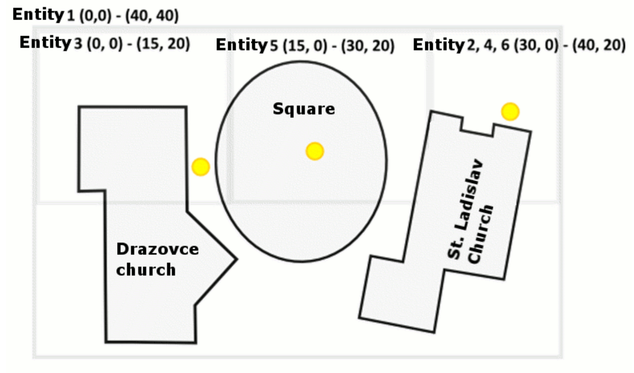

Entity 1 (0,0) - (40, 40)
Entity 3 (0, 0) - (15, 20)
Entity 5 (15, 0) - (30, 20)
Entity 2, 4, 6 (30, 0) - (40, 20)
Square
St. Ladislav Church
Drazovce church

Figure 7: Ground-plan of the scene. Yellow balls - SkyPorts. Rectangles - relevancy of the entities. Only the first entity is defined for the whole scene, because all SkyPorts are inside the rectangle.

4.4.2 Server

At first, server requires common web-based client-server communication. This includes database, in our case we tested the application with MySQL database. To run server based script, in our case PHP, the PHP server is needed. Recommended is the Apache server. Also the world that is initialized when the page is uploaded has to be stored on the server side.

4.4.3 Pre-processing

Pre-processing is necessary, because the whole scene and database has to be created. Assumption is that this application is only extension to the existing virtual world, therefore the scene is already created. Afterward SkyPort objects have to be added into the file. SkyPorts are defined as yellow balls, but for the customization is available any object with defined Sensors in a special form. These sensors are TouchSensor for interactivity with the user and TimeSensor to create pulsing effect. Defining is as follows:

```
<TouchSensor DEF='TouchSensor1' />
```

```
<TimeSensor cycleInterval='5.000' />
```

Where TouchSensor1 is the name of the sensor, therefore it can be easily identified. Exactly these names are stored in the database. The name of the TimeSensor is not important, because it is not dynamically manipulated. Better adding algorithm is left to the future work.

Afterward, when sensitive objects are added, they has to be processed and therefore SkyPort database need to be created. In the database are the unique names and position of the skyports stored to be easily identified after interaction.

Another database has to be created to store facts, dates and relevancy to the facts. Relevancy is stored as x, y coordinate of the rectangle's corners where the entity is displayed. Therefore after the interaction with the SkyPorts located inside this rectangle the specified entity is displayed (Fig. 7).

4.5 Display

As was said before the displaying of the entities in the skyspace is not a trivial problem. There is possibility to fill information with using the all dimensions offered in the 3D space. Our approach is inspired by the book [Gombrich 2006]. Comparison of these results can be seen in the (Fig. 8). In this approach, information is displayed only in two dimensions. Although we successfully improved this method by moving the box with entities in all 3 dimension. Therefore user can move whole sky with the entities forward and backward to have better look at the display.

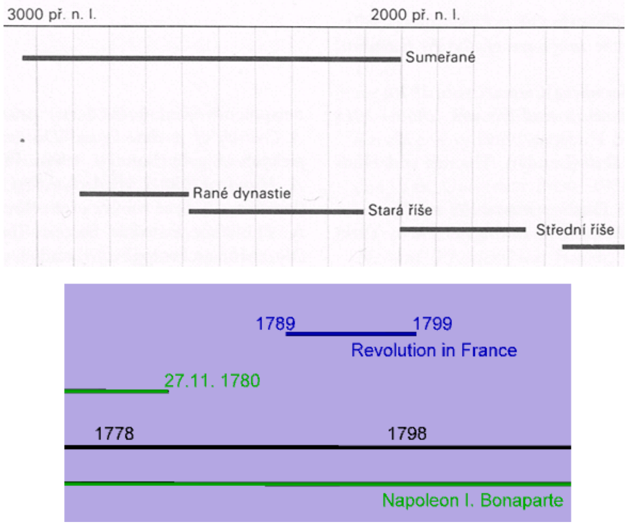


Figure 8: (top) scan from [Gombrich 2006] (bottom) our method

Firstly the whole skyspace is displayed as a group, with an unique name that includes all entities. This group is then the last added rootnode and can be therefore easily found in the scene graph. The entities are displayed as the thin cylinders, where each represents time distribution of the entity. The length of the cylinders is counted from the difference between begin and end time of the fact. If only year is provided, the distribution is the whole year. It means counting begin date as first of January and end time as end of December. Similarly with only year and month dates, without the day.

Entities are categorized so that the groups are displayed together. Therefore these groups are colored by a specific color.

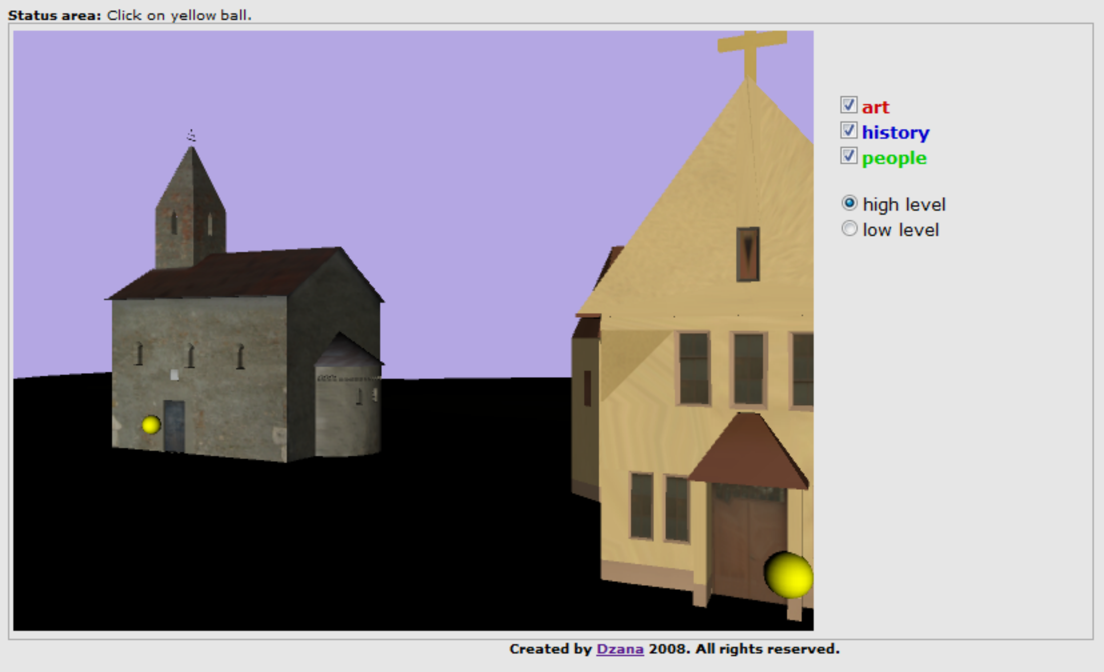
Moreover for the better orientation the scale is displayed in the middle of the skybox. Therefore, when moving whole skybox also the scale is moved and entities stay at their right places relative to the scale. To compare, in the Gombrich's approach scale is at the top (Fig. 8 (top)) and in our method scale is in the middle. This improvement allows scale to be seen also when whole box is moved in the top or bottom direction.

Because the facts are not distributed linearly in the scale, the problem in displaying of them arises. Therefore holes can be seen, when there are short entities far away. Further investigation of this problem is left to the future work.

5 Results

In order to show our solution for the fitting the skyspace with the relevant information, there was created application with name SkyBoard. For now, only sample information and scene are used.

Friendliness of our application and the whole new approach was validated by the questionnaire. It was included in the page with the whole world. The question was, if this new method is better than



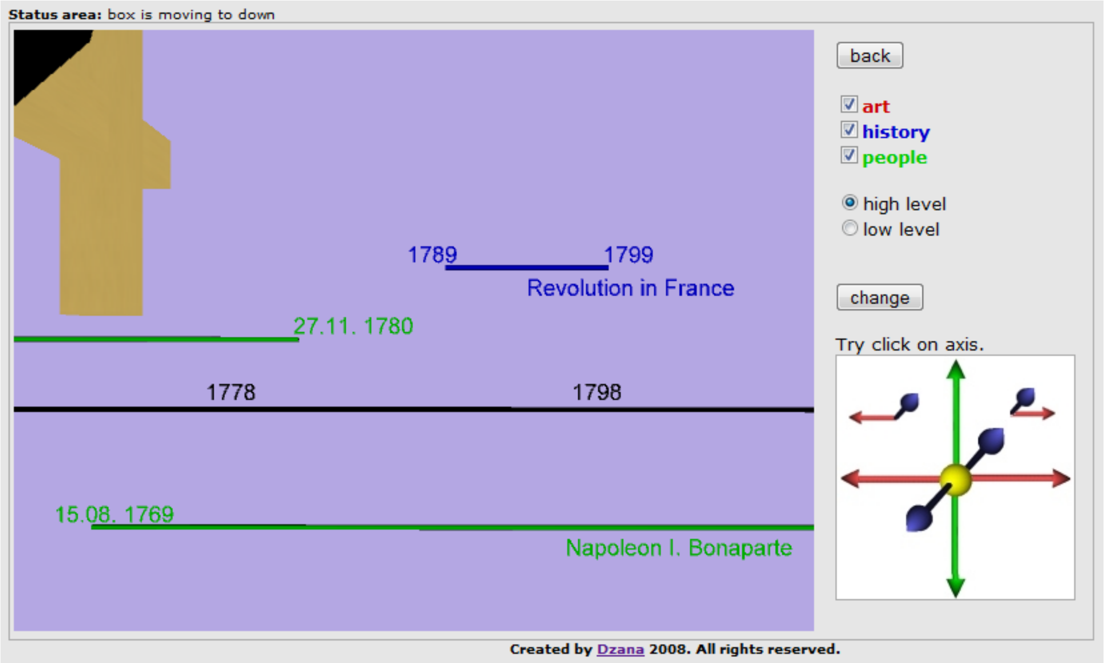
Status area: Click on yellow ball.

☑ art
☑ history
☑ people

● high level
○ low level

Created by Dzana 2008. All rights reserved.

Figure 9: Front View



Status area: box is moving to down

☑ art
☑ history
☑ people

● high level
○ low level

Try click on axis.

Created by Dzana 2008. All rights reserved.

Figure 10: Sky View

the traditional was of displaying of information. Amazingly more than 60% users claim that like our display more.

Our results are shown in the (Fig. 9) where is the front look at the sample scene without any additional information. Second figure (Fig. 10) show view from another view port, which is rotated to the sky and entities are added.

6 Conclusion and Future Work

In conclusion, this approach is one of the first which uploads entities in the skybox in the virtual world. There are still parts of the algorithm that needs to be further investigated. There is display of the entitled and user interaction provided, but the display still has holes and better solution could be find. In our method there are some pre-processing steps, that are time consuming therefore the more automated approaches has to be more investigated.

To sum it up, this approach could be used in the virtual worlds to provide more information or in the virtual museums projects for educational and informational purposes.

References

ACTIVE WORLDS CORPORATION. 2008. *Active Worlds Educational Universe*, [online 23.1.2009]. http://www.activeworlds.com/edu/awedu.asp.

GARRET, J. J. 2005. Ajax: A new approach to web applications. *[online 23.1.2009]. http://www.adaptivepath.com/ideas/essays/archives/000385.php*.

GOMBRICH, E. H. 2006. *The Story of Art*. Phaidon Inc Ltd.

HANUSKA, D. 1999. Information systems and vrml worlds. In *Proceedings of the Central European Seminar on Computer Graphics*.

KAN, P. 2007. *3D model of Drazovsky kostolik*, [online 23.1.2009]. http://www.peterkan.com/kostolik.php.

LINDEN LAB. 2003. *Second Life*, [online 23.1.2009]. http://secondlife.com/.

METAVERSUM GMBH. 2008. *Twinity Powered by Real Life*, [online 23.1.2009]. http://www.twinity.com/en.

PARISI, T., 2006. Ajax3d: The open platform for rich 3d web applications. SIGGRAPH 2006, Whitepaper.

QVORTRUP, L., JENSEN, J., KJEMS, E., LEHMANN, N., AND MADSEN, C. 2002. *Virtual Space Spatiality in Virtual Inhabited 3D Worlds*. Springer-Verlag.

R. HETHERINGTON, B. FARRIMOND, P. C. 2007. Embodying and extracting data in web3d models of proposed building developments. In *Proceedings of the Computer Graphics, Imaging and Visualisation*, 528–534.

RITZEMA, T., AND HARRIS, B. 2008. The use of second life for distance education. *Journal of Computing Sciences in Colleges 23*, 6, 110–116.

VHCE TEAM, BRATISLAVA. 2000. *Virtual Hearth of Central Europe*, [online 23.1.2009]. http://www.vhce.info.

VIVATY, INC. 2008. *Vivaty Player*, [online 23.1.2009]. http://www.vivaty.com/downloads/player/index.php.

ZARA, J. 2000. *Jazyky pro popis virtualni reality*. CVUT, Praha.