DATA REPORTING ON INTERNET FROM SENSORS THAT MEASURE ENVIRONMENTAL PARAMETERS


Abstract: This paper describes a development stage of a 3D application developed using WebGL for reporting temperature, humidity, pressure and dew point from a sensor installed in a location where constant monitoring is required. The data from the sensor is stored in an Oracle database. The sensor measures the environmental parameters once per minute every day. The web pages that make the data available to Internet users are developed using Oracle APEX. The 3D model of the monitored location and the sensor were designed using Blender. The 3D models are loaded into the Oracle database and rendered, using WebGL inside web pages.

Keywords: WebGL, Java, Internet server, 3D web application, open source

1. INTRODUCTION

Currently, a software platform is under development that will gather information from several sensors (thermometers, hygrometers, anemometers, manometers, lightmeters) and record the data in an Oracle XE database. The data (sensor measured values like temperature, humidity, pressure and configuration) can be accessed by users from the Internet, outside the monitored locations where the sensors are installed. The data is displayed in graphs and reports in web pages developed using Oracle APEX. The graphs and reports are designed in a manner to facilitate data processing for statistical purposes to Internet users. The software platform has a security scheme to allow each Internet user to see the sensor data only from the monitored locations that he owns [5][8]. In figure 1 the blue user has access to information from the blue location, while the yellow user has access only to information from the yellow locations.

Present research was started as a development stage for the development of a fully automatic system for a mushroom cultivation greenhouse, first sub-stage being set for gathering and monitoring data (temperature, humidity, pressure, air flow, light intensity) as well as founding background for the second sub-stage of development including targeting the automatic (remote) adjusting of environmental parameters from mushroom incubation and fructification rooms [6].

2. PROBLEM FORMULATION

The research group is focused on developing a 3D interface for Internet users to be able to view the exact positions and data of the sensors installed in the monitored locations in order to facilitate an enhanced collaboration between physically remote people and places (between people inside the monitored location where the production process takes place and engineers from remote locations) [2][3]. Also, it is preferable to use open source code for developing the 3D interface.

3. PROBLEM SOLUTION

The first attempt was made using Java 3D. This requires installing Java and Java 3D plugin on the computer Internet browser[4]. The java applet is packed in a *.jar file and uploaded into the Oracle database in a BLOB type column and can be viewed from web pages designed with Oracle APEX.

The second approach for the development of the 3D interface was using WebGL. WebGL consists in Javascript code written inside the web page that is able to render 3D scenes. In the development the “three.js” WebGL engine was used from the following web link “http://www.chromeexperiments.com/”. All the necessary “three.js” libraries, needed for the construction of the 3D scene for the 3D model of the monitored location, were uploaded into the Oracle database. The 3D model (mesh) of the sensor and the location has been modeled in Blender, an open source 3D modeling software available at “http://www.blender.org/”. The sensor and location models files were exported from Blender as wavefront (*.obj) and converted in *.js files using “convert_obj_three.py” from “three.js”. The mesh files of the sensor and of the location have been loaded into the Oracle database in BLOB type columns.

Of several computer configurations, on which the 3D interface has been tested, two are displayed in table 1. The first is an average desktop computer system, the computer that a client will most likely have. Unlike Java 3D, WebGL does not require any additional plugins to be installed on the computer Internet browser and loads and renders the 3D scene faster then the Java applet.

<table>
<thead>
<tr>
<th>PC No</th>
<th>CPU</th>
<th>RAM</th>
<th>GPU</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intel Core 2 Duo E7500 2.97 GHz</td>
<td>2 GB</td>
<td>ATi Radeon HD 4300</td>
<td>Windows 7 32 bits</td>
</tr>
<tr>
<td>2</td>
<td>Intel Core i7-920 2.66 GHz</td>
<td>12 GB</td>
<td>Nvidia 470 GTX</td>
<td>Fedora Core 16 64 bits</td>
</tr>
</tbody>
</table>

Tab. 1. Technical specification of the computers used to test the 3D application

4. CURRENT LEVEL OF THE RESEARCH

In present, the software platform has gathered information from one sensor (Fig. 1) from October 2011 until present May 2012. The next figures present screenshots from the current stage of the development of
A software application that stores and reports over the Internet, data from sensors that measure temperature, humidity, pressure and other measurable digitized elements.

The sensor used in this research is installed in a designated location and is displayed in figure 2, the technical specifications can be seen in table 2. All graphs and reports in the next figures contain values actually measured by the sensor. The sensor is connected to a local area network and can be accessed for configuration and data gathering using its IP address. The sensor measures the temperature, humidity, dew point and pressure once per minute every day. Figure 3 displays the 3D model of the sensor rendered exactly as is rendered inside a web page using “WebGL Three.js”.

![Fig. 1. Research description](image1.png)

![Fig. 2. The sensor used in this research, Comet T7510](image2.png)

| Intelligent barometric pressure, humidity and temperature probe with Ethernet interface |
|---------------------------------|-----------------|
| **Characteristic**         | **Value**       |
| Manufacturer               | Comet           |
| Model                      | T7510           |
| Temperature                | -30 to 80°C     |
| Humidity                   | 0 to 100%RH     |
| Pressure                   | 600 to 1100hPa   |
| Ethernet                   | RJ45            |

![Tab. 2. Sensor specifications](image3.png)

![Fig. 3. The 3D model of the sensor designed in Blender and rendered with WebGL Three.js](image4.png)
In figure 4, the temperature graph is displayed and figure 5 displays the temperature report. The temperature values can be seen by placing the mouse cursor over each dot. The measured values can be downloaded as a *.csv file that can be opened with Microsoft Excel, LibreOffice Calc or OpenOffice Calc.

In figure 6, the average, minimum and maximum values of temperature can be seen per day in a calendar format. For example, on the 26th of May 2012 the maximum value of temperature was 27.8°C, the average value of temperature was 26.4°C and the minimum value of temperature was 24.4°C.

In figure 7, the 3D model of the location and the sensor positioned inside the location are displayed. The user can rotate and zoom the location. When the user clicks the sensor, the information (name, measured values and position) about the device will be displayed in the right side of the window.

In figure 8, the average, minimum and maximum temperature values per hour in the last 24 hours can be seen. The 1st line (green) represents the maximum values per hour, the 2nd line (blue) represents the average values per hour and the 3rd line (red) represents the minimum values per hour.

In table 3 the applications used for developing the software platform are presented. The sensor communicates with the computer where the data is stored using SNMP protocol. The data from the sensor is stored in an Oracle XE database. The software that loads the measured values of temperature, humidity, pressure and dew point into the database was developed using Java Developer. The operating system of the computer that stores data is Linux Fedora Core. Backup configuration files using Bacula, backup database by making “cold backup” and exporting the SQL tables scheme.
Figure 9 represents a graph for displaying the average, minimum and maximum temperature values per day in the last 31 days. Unlike the calendar in figure 6, the graph format can better display the maximum, average and minimum temperature values.

Figure 10 presents the graph for the average, minimum and maximum temperature values per month.

Figure 11 highlights the main modules with the main characteristics for inserting, updating and deleting data in the application SQL scheme from the database [7]. Each module contains more than one PL/SQL table. PL/SQL (Procedural Language / Structured Query Language) is Oracle Corporation’s procedural extension language for SQL and it is used to manipulate data (select, insert, update, delete) from tables and write procedures and functions in Oracle relational database [8].

The “Security Scheme” module manages the username and password for each user that logs to the Application Server database. The password is stored in an encrypted format, not plain text. The privileges for each user are also managed by the security scheme. A privilege means granting access to a certain user to
The maximum temperature was 32.8°C on April 5th, 2012.

The lowest temperature was on December 2011 is 13.9°C.

manage specific data. For example a user will have privileges to add new locations to the database, while another user will have only the privilege to view the locations, but not to make modifications.

The “Location Definition” module stores the locations where the monitoring takes place. It stores the location name, address, blueprint, 3D model file of the location and the owner of the location. The location depends on the users from the Security Scheme because the owner of the location is a user.

The “Object Definition” stores the sensor name, type, manufacturer, image of the sensor, 3D model file of the sensor and technical data of the sensor. The Object Definition module depends on the Security Scheme because of the user privileges. Some users are allowed to modify sensors data, others only to view it.

The “Warehouse” module manages the number of devices in the database and the locations where the devices are placed. It stores how many sensors of a type there are and where the sensors are located. It merges the 3D model of the location with the 3D model of the sensors, resulting in a 3D Scene that displays the position of the sensors in the monitored location. Also it holds a history of the device when the location of the device changes. The “Warehouse” module depends on the - 0079 -
“Object Definition” and the “Location Definition” because it merges devices data with locations data.

The “Data management modules” represent any type of data processing that can be done with values measured by a certain device defined in the “Object Definition”. For the current research it represents the “Monitoring” module that contains tables that store the measured values from the sensors and generates reports and graphs and issues warnings if parameters limits exceed. Another application is sending email messages before the calibration period of the sensor expires. The calibration period of a sensor is usually 1 year, after which it must be recalibrated by a metrological laboratory.

All the modules highlighted in figure 11 can be extended to fit application specificity.

4. CONCLUSION

This paper has presented the development of the 3D interface for a software platform that allows data reporting on the Internet and data recording from sensors that measure environmental parameters from locations that require constant monitoring.

For this research, WebGL was the solution for building and rendering complex 3D scenes of the monitored locations to facilitate a mixed reality/virtual infrastructure.

The original contribution for this stage of the research is the development of modules for monitored locations, sensor installation, data processing and data representation and SQL database structure (graphs, diagrams, tables, reports, warnings, and 3D models built in Blender).

The commercial solutions of this type of software application that can report on the Internet data from sensors are too expensive, or do not process certain range of sensors, and the compiled code and license do not allow to adapt the code to the production process specificity [7][9].

5. REFERENCES


Fig. 11. The database relational tables scheme