New tools for 3D HMI development in Java

Ondřej Severa, Martin Čech, Pavel Balda
Department of Cybernetics
University of West Bohemia in Pilsen
Pilsen, Czech Republic
e-mail: osevera@kky.zcu.cz, mcech@kky.zcu.cz, pbalda@kky.zcu.cz

Abstract — **In this paper a new approach to 3D HMI suited to Java platform is presented. The technique described is based on Java3D package (rendering package) and a VRML (Virtual Reality Modeling Language) Java loader which creates a bridge between the CAD system and the rendering engine. The new approach is step-by-step demonstrated on a real Furuta pendulum HMI. Moreover, it is shown that after adding a simulation core containing the pendulum model one gets an interactive virtual reality laboratory which can be simply embedded into a web page as the Java applet. Such applet can be even stored at some embedded control device with built in web server. The authors believe that the presented technique will be applicable in both academic and industrial sphere.**

Keywords – Java applet, 3D HMI, CAD, VRML, Web based education, furuta pendulum, virtual reality, Java 3D

I. Introduction

Over the years, HMI became an integral part of any control application. Traditionally, it is composed of several screens displaying current values of important variables from a machine or process. Initially, the screens were quite simple and based on 2-dimensional (2D) graphics. Fifteen years ago, a boom of 3-dimensional (3D) scenes started thanks to the growing power and capabilities of computers (in particular graphics cards and accelerators). Such visualization provides more realistic view and interface to the existing object. The number of elements in the scene is basically not limited, thus one can create not only a view of some particular manipulator but even a full 3D model of a factory including its production lines or plants. If the visualization platform together with data acquisition subsystems satisfies common real-time requirements then one gets in each moment an up to date live view of the reality. Let us call it further *RT-3D-HMI* (real-time).

On the other hand, the 3D-HMI may be supplied by the simulation core containing the physical model of both the machine and its control system. Then one gets a fully virtual reality model of the machine or plant. Such model is usable f. e. for distance learning [1, 4, 5] or training new human operators [3]. Let us denote such tools as *VR-3D-HMI* (virtual-reality). The terminology is clearly explained in Fig. 1.

Nowadays, the 3D-modules are available in the majority of leading visualization SWs (e.g. Indusoft, LabView). Unfortunately, the utilization of mentioned large software packages has several limits. They usually do not allow exporting 3D screens into self-contained Java applets which can be further embedded into a platform-independent web page. Consequently, such HMIs are often runnable only on classical desktop computer equipped with MS Windows. Further, the HMI integration with the machine and control system model is quite complicated. Finally, such SW is too expensive for simple or embedded applications.

One has always to take two basic parts into account when creating the 3D model. The first part is the 3D rendering engine which takes care on drawing model on the screen. The second part is some SW tool for 3D model creation – usually CAD system (SolidWorks, Autodesk Inventor, CATIA, etc). Note that the CAD system is always used when designing and manufacturing new machines. Hence, using it as a source of model for 3D visualization is quite natural. However, there must exist a ‘bridge’ between these two parts. In other words, the CAD system must provide export of the 3D model into some common format acceptable by the rendering engine.

Java platform was already proved suitable for developing interactive web tools [9]. It completely obviates the above mentioned problems. In this paper a new approach to 3D HMI suited to Java platform is presented. The technique described is based on *Java3D* [11] package (rendering package) which is often used to display 3D scenes in Java [2], [6]. Next, the Java loader for *Virtual Reality Modeling Language* (VRML) files creates a bridge between the CAD system and the Java rendering engine. The free NetBeans IDE [12] is used as the development platform. The connection to the real time device controlling the pendulum model is ensured by JavaREX communication protocol [7].

The paper is organized as follows: In Section 2, the common features of the most used CAD systems are summarized. Section 3 discusses the pros and cons of well known file formats for exchanging 3D data. The key features of Java3D package are highlighted in Section 4. Especially the cooperation with Java VMRL loader is focused here. As an example, the development of Furuta pendulum HMI is step by step demonstrated in Section 5. Both variants (*RT-3D-HMI* and *VR-3D-HMI*) are briefly described here. Conclusions and ideas for further work are given in Section 6.

978-1-61284-361-2/11/$26.00

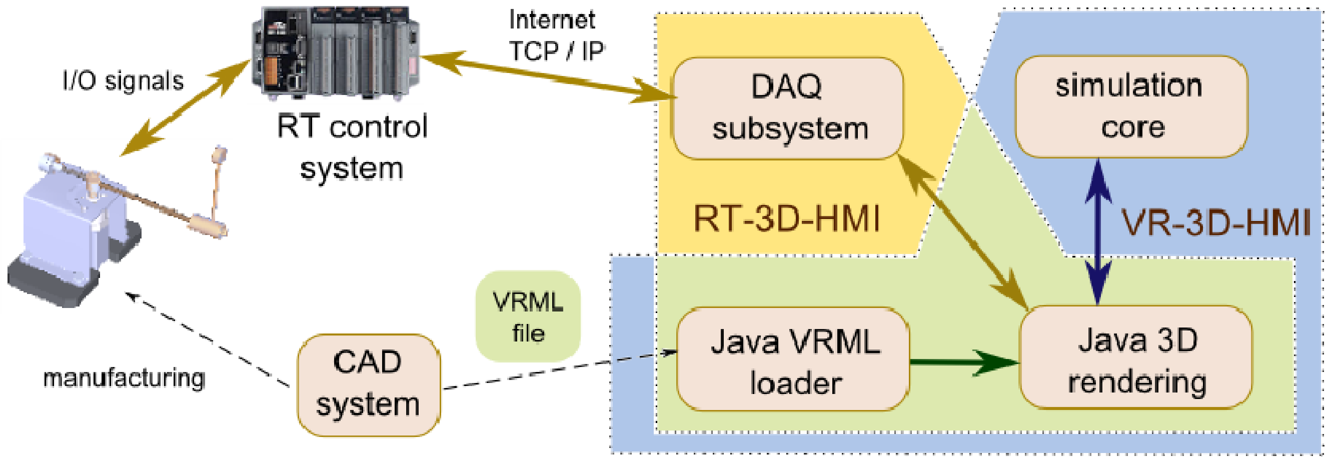

Figure 1. RT-3D HMI vs. VR-3D HMI schema.

II. CAD SYSTEM COMMON FEATURES

Computer-aided design (CAD) is the use of computer technology for the process of design and design-documentation. CAD may be used to design curves and figures in two-dimensional (2D) space; or curves, surfaces, and solids in three-dimensional (3D) objects.

Outputs from the 3D CAD software have a similar form. There is a final product which should contain all necessary features for production. This product is assembled from basic parts like a building kit. These parts should contain both their final shape and type of material which they are made from.

A. Tree structure and object groups

It is evident, that there is a need to organize parts and elements to define their mutual mechanical connection and dependencies. A commonly used natural structure is a tree. Nodes of the assembly tree may contain basic parts or, more often, whole components represented by the group of basic parts. For example, the robotic arm (tree node) has children - the efector 'fingers'. Let us point out, that the machine structure is a key input for product (model) animation process. When some transformation (rotation, movement) is applied on a particular tree node it affects directly also all of its children.

It is a pity that the tree structure is quite often used only inside the 3D CAD system and is exported just into its native files. After exporting the model into one of the exchange formats described in Section 3 one gets the model in a simplified form of unorganized list of elements. In such case it is necessary to recreate the model tree structure outside the CAD system.

B. Materials, colors, textures

Like it was said before, each basic part of assembly should have assigned some type of material. There are different concepts of material representation. Materials can be understood only in visual interpretation, thus they are represented as a set of colors (ambient, diffusive, emissive, specular), value of shininess and transparency or they can be covered by a texture. Color attributes or textures define final look of the part and are used in visualization.

It is very common that 3D CAD software includes also physical representation of the material (rigidity, mass density, inertia matrix), so it is possible to analyze parts in mechanical point of view (e.g. stress analyses, deformations). Such information is not necessary for creating 3D HMI. The real-time HMI gains the dynamics in fact from the real object while the virtual reality HMI contains the dynamics inside the simulation core. For 3D HMI, it is usually sufficient to suppose that the machine is assembled by rigid bodies (often called as MBS – multi body system). More comments to the machine dynamic model are given in Section 5.

III. 3D GRAPHICS FORMATS

Generally, both RT-3D HMI and VR-3D HMI use the same 3D model created by CAD software. For Java platform, it was decided that the final input format for Java 3D renderer will be VRML (Virtual Reality Modeling Language). It is well supported and very widely used 3D format. Nowadays it has a successor called X3D which is backward compatible.

Although VRML or X3D formats are broadly used on the World Wide Web, only a minority of the 3D CAD softwares support these formats in default (e.g. SolidWorks). The standard data exchange format supported by almost all CAD systems is STEP. Basically, all of mentioned formats have a capability to preserve the objects tree structure. Let us describe their key features in more details.

A. VRML

Virtual Reality Modeling (Markup) Language is a standard text file format for representing 3-dimensional (3D) interactive vector graphics, designed particularly with the World Wide Web in mind. It is standardized by ISO (ISO/IEC 14772-1:1997) and maintained by Web3D Consortium. Common extensions are *.wrl or *.wrz for gzip compress vrml files.

Example 1: Simple sphere made from shiny blue material:

```
#VRML V2.0 utf8
DEF dad_Sphere Transform {translation 0 .5 0
 children [ DEF Sphere Shape {
   appearance Appearance {
    material DEF Shiny_Blue Material {
     ambientIntensity 0.200
     shininess 0.400
     transparency 0.100
     diffuseColor 0 .5 .8
     emissiveColor 0 0 .2
     specularColor 0 .5 .8
    }
   }
   geometry DEF GeoSphere Sphere {radius 1.000}
  }
 ]
}
```

B. X3D

X3D is a successor of VRML. It is standardized by ISO (ISO/IEC 19775/19776/19777) and also maintained by Web3D Consortium. X3D scene can be stored in a text XML file (*.x3d), Open Inventor-like syntax of VRML97 (*.x3dv) or in binary file (*.x3db). X3D specification support several profiles. These profiles define various levels of capabilities. The big advantage of XML structure is that there exist data parsers (SAX, DOM) in almost all leading programming languages.

Example 2: Sphere in X3D (equivalent to Example 1):

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive' >
   <Scene>
      <Transform DEF='dad_Sphere' translation='0
   .5 0'>
    <ShapeDEF='Sphere'containerField='children'>
        <Appearance containerField='appearance'>
         <Material DEF='Shiny_Blue'
            containerField='material'
            ambientIntensity='0.200'
            shininess='0.400'
            transparency='0.100'
            diffuseColor='0 .5 .8'
            emissiveColor='0 0 .2'
            specularColor='0 .5 .8'/>
        </Appearance>
        <Sphere DEF='GeoSphere1'
         containerField='geometry'
         radius='1.000'/>
       </Shape>
      </Transform>
   </Scene>
</X3D>
```

C. Step

Step-file standardized under ISO 10303-21 is the most widely used data exchange form of STEP (Standard for the Exchange of Product model data). This ASCII text file represents data according to a STEP Application Protocols.

Remind that Step-files are well supported by various 3D CAD softwares. If there is no direct export into VRML available, such format may serve as a bridge between some 3D CAD and VRML. However, one must find a way to convert STEP files into VRML outside the CAD system. One possibility is to use CADExchanger [15] as a convertor from STEP-File into X3D and then use Flux Studio [16] for further modifications and final export to VRML.

IV. Java 3D Package and Java Vrml Loader

Java3D is a set of APIs which is the Java expansion in the three-dimensional field based on OpenGL and D3D. It implements the characteristics of real-time and interactivity in the three-dimensional environment.

A. Java3D Scene structure

Three-dimensional scenes in Java3D use the tree structure. In a simple Java3D virtual universe (tree root) there are two main branches – the *viewer* and *3D scene*.

The first branch defines how the viewer looks to the 3D canvas. There is a possibility to add multiple behavior nodes to the viewer. For example, orbital behavior ensures spherical movement around 3D model, key navigator behavior provides pitch and yaw of the camera, etc. Furthermore, some of the children in the viewer branch handle also 3D rendering settings like antialiasing, minimal frame rate, etc.

The second branch represents 3D scene with imported 3D model. This branch contains typically numerous `Shape3D` instances (parts of an assembly) placed as leafs into the tree. Every `Shape3D` node is a child of `TransformGroup` (TG) node which connects shapes with standard 3D transformation matrixes. The movement or animation of objects is then invoked by changing those matrixes. It is possible to synchronize transformations according to data from real sensors (RT-3D HMI) or from mathematical machine model (VR-3D HMI). Consequently, when the information from TG is used during the repetitive rendering process of the scene, the displayed model reflects reality (RT-3D HMI) or virtual reality (VR-3D HMI). Detailed description of Java3D function is well described in [14].

Note that Java3D contains several optimization tools. For instance the user has to define the transformation nodes that will be changed during animation. Once this task is done, the whole model should be compiled to optimize the computation of transformation sequence in the tree.

B. Java VRML Loader

It flows out from previous sections, that the CAD software provides the 3D model and Java3D can render any 3D scene. Anyway, there is need for a bridge between these two technologies. Java3D VRML Loader is a part of Java3D project. By a few lines of Java code one can parse VRML files and create all necessary instances of Java3D scene graph. The created graph is then simply connected to the 3D scene branch. Finally, the whole tree is passed to the Java3D renderer which takes care on the periodical repainting of the scene.

V. Example: Furuta Pendulum 3D HMI

Let us show the ideas on the development of a Furuta pendulum 3D HMI. A common background for both RT-3D-HMI and VR-3D HMI is three-dimensional model created in SolidWorks. It was primary intended for manufacturing.

However, the direct VRML export available in SolidWorks allows to use the CAD model almost without any change as an input for the Java VRML loader and Java 3D.

A. RT-3D-HMI

Figure 2 shows the overall view on the RT-3D-HMI. The application is divided to four major parts. There is a control panel on the left side followed by the main 3D canvas and the user control panel on the right side. Last noticeable component is a trend panel on the bottom of the application. This trend shows actual values of arm and pendulum angles.

The user can control orientation, zoom, view and setting of the 3D canvas through 3D control panel. The user control panel suits remote visualization needs. It is possible to send commands to the control algorithm, read values and error states. More specifically, the user can control a set point of the arm angle, position of the pendulum (up or down) and the state of demonstration mode.

DAQ subsystem

RT-3D-HMI is entirely depended on data acquired from RT control systems, i.e. the visualization only reflects the state of reality. Hence, the live connection to the process is necessary (DAQ subsystem in Fig. 1). Here, the communication uses a TCP/IP protocol which client part is described in [7]. The developed HMI is able to run as an application or applet. It implies that user is able to connect to the control system even remotely through any web browser with basic JRE (Java Runtime Environment).

B. VR-3D-HMI

As oppose to RT-3D-HMI, VR-3D-HMI is totally independent and self-contained application. It is suitable for training, demonstration and education purposes.

One can see on the Fig. 3, that the look of the application is quite similar to the previous one. Changes reflect different usage of the application. For example, the communication error indicators were removed. On the contrary, a special user control panel was added. It is located between the canvas and trend (see Fig. 3). The panel controls environment influence at the model. One can push, pull and add noise to the pendulum angle. Moreover, by the special slider the user may try to control the pendulum in upper position manually.

The Furuta pendulum VR-3D HMI (FPM Demo) is free accessible on the department virtual laboratory http://lab.fav.zcu.cz/virtuallab. The real pendulum with control system and RT-3D-HMI is depicted in Fig. 4.

Simulation core

The creation of the simulation core (see Fig. 1) is quite an advanced task which description exceeds the page limit of this paper. Let us give only few notes to it.

It is impossible to write the code for Java simulation core manually. It must be generated fully or at least partly automatically. Here a special technique for automatic generating of simulation core directly from Matlab/Simulink model was used [8].

Next, the problem of obtaining the own mechanical differential equations must be considered. The development of real machine is an iterative process. The first sketch of the machine may be designed for example in Matlab/Simmechanics. Here one can optimize the initial dimensions and mass of individual mechanical parts. Consequently the whole system may be simulated even with the control system. Simmechanics provides also export of linearized differential equations in the machine steady state. The Simmechanics model serves as a basic idea for detailed model in CAD system. Although CAD generates automatically the mechanical equations including the proper position of the centre of gravity and inertia matrixes, there is usually no way to gain the model from CAD for further usage in Java simulation core. Hence the parameters of the real model are often refined by some more or less heuristic identification experiments on the real machine. The mentioned problems will be researched in the future.

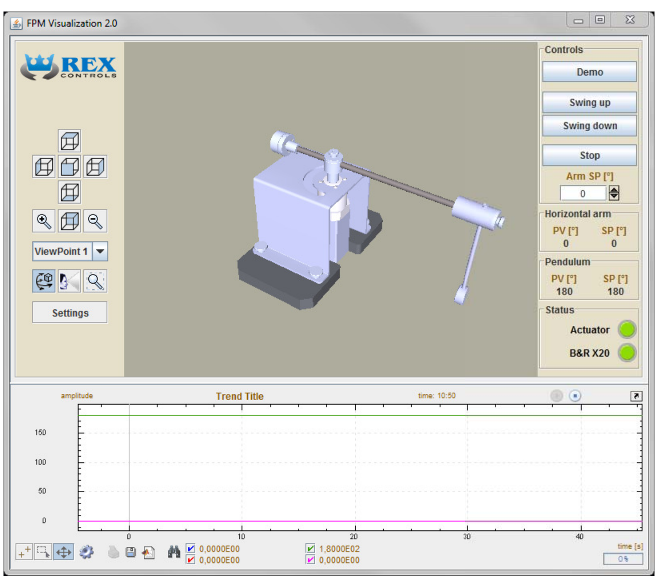

Figure 2. Furuta pendulum RT-3D HMI.

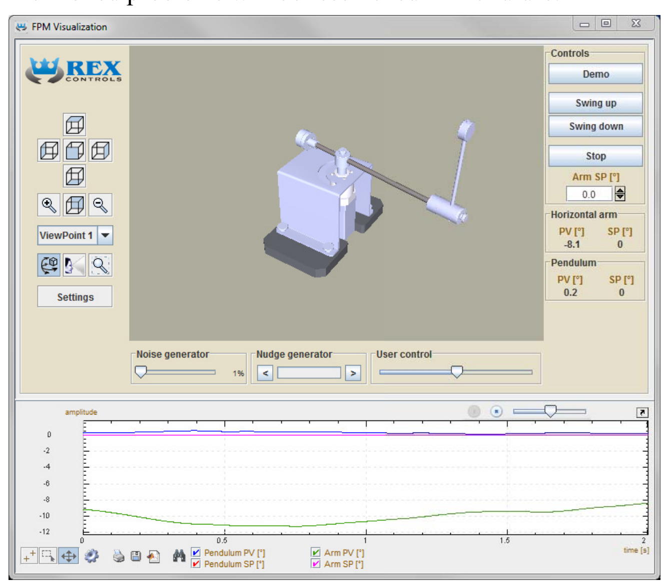

Figure 3. Furuta pendulum VR-3D HMI.


Figure 4. Real Furuta pendulum with its control system and 3D HMI.

VI. CONCLUSIONS

In this paper a novel approach to creating 3D HMI suited to Java platform was presented. It is based on Java3D rendering engine and a Java VRML loader, which is able to interpret files obtained directly from some CAD system. The whole technique was step by step demonstrated on the development of Furuta pendulum 3D HMI. The *RT-3D-HMI* version communicates with the pendulum control systems and updates the 3D scene according to the reality. The *VR-3D-HMI* version is a self-contained Java applet equipped by simulation core containing the model of both the pendulum and its control system. Compared to *RT-3D-HMI*, the GUI is extended by the ability to control the pendulum manually and by the slider defining the amplitude of the noise affecting the pendulum model. Such virtual reality tools are close to the computer games and are suitable f. e. for teaching students or engineers before they come in touch with the real machine. In the future, the big effort will be put to adapting all GUI components for NetBeans GUI designer. Also the possibilities of automatic generation of mechanical equations directly form CAD will be investigated. Solving these tasks will speed up the development of 3D HMIs. The authors believe that the presented technique will be applicable in both academic and industrial sphere.

ACKNOWLEDGMENT

This work was supported Czech Science Foundation – project no. P103/10/P208 and by Czech Ministry of Industry and Trade – project No. FI-TI1/077. This paper was also supported from the funds of the European Union (ERDF/INTERREG IVC) through the Innovation 4 Welfare project (subproject I4W-22-MRH).

REFERENCES

[1] Douar A, Harous S, Djoudi M, et al., “An Interactive 3D Distributed Environment For Distance Practical Works,” Proc. International Conference on Innovations in Information Technology Al Ain, Arab Emirates, Dec 2009, IEEE Press, pp. 76-80, ISBN: 978-1-4244-5698-7.

[2] Vogt N, Popov E, Rudert R, et al., “3D visualization of molecular structures in the MOGADOC database,” Journal of Molecular structure vol. 978 iss. 1-3 pp: 201-204, Aug. 2010, ISSN: 0022-2860.

[3] Pei Y, Wu H, Zong R, “Simulated Joint Robot-hand System Based on Java3D,” Proceedings of the 2009 2nd international congress on image and signal processing, vols. 1-9, pp. 2805-2808, IEEE 2009, ISBN: 978-1-4244-4130-3.

[4] Le Blanc, A., Bunt, J., Petch, J. and Kwok, Y. 2005, "The virtual learning space - An interactive 3D environment", Proceedings of 10th International Conference on 3D Web Technology; Bangor, Wales, March 2005, pp. 93-102, ISSN: 15529886.

[5] Vani, V. and Mohan, S. 2010, "Interactive 3D class room: A framework for Web3D using J3D and JMF", Proceedings of the 1st Amrita ACM-W Celebration of Women in Computing in India, Sept. 2010, ISBN: 978-145030194-7.

[6] Wu, Y. and Miao, Z. 2008, "Construction of the virtual environment based on java3D", Proceedings of IET 2nd International Conference on Wireless, Mobile and Multimedia Networks, ICWMMN 2008; Beijing, Oct 2008, pp. 512-515, ISBN: 978-086341925-6.

[7] Balda, Pavel; Čech, Martin. “Java interface to REX control system.” In Process control 2006. Pardubice : Technical University , 2006. s. 1-8. ISBN: 80-7194-860-8.

[8] Čech, Martin; Balda, Pavel. “A new technique for automatic generation Java applets for web-based control education”. In Proceedings of the 17th international conference on Process control '09. Bratislava : Slovak University of Technology, 2009, s. 491-497. ISBN: 978-80-227-3081-5.

[9] Čech, Martin; Balda, Pavel; Schlegel, Miloš; Severa, Ondřej. “New virtual internet laboratories presenting advanced control schemes”. In 11th International Carpathian Control Conference. Eger, Hungary : REKATEL Bt., Miskolc, Hungary, 2010. s. 1-4. ISBN: 978-963-06-9289-2.

[10] Presser, C. G. “A Java web application for allowing multiuser collaboration and exploration of existing VRML worlds”, Proceedings of the Tenth International Conference on 3D Web Technology, Bangor, United Kingdom, March 2005, pp. 85-92, ACM, New York, NY.

[11] Oracle. *Project Java3D.* 28 Feb. 2008. . https://java3d.dev.java.net/ 14 Jan. 2011

[12] Oracle. *Netbeans* . 23 Aug. 2010. http://netbeans.org/ 14 Jan. 2011

[13] Web3D Consortium. *Web3D Consortium.* http://www.web3d.org/ realtime-3d/ 14 Jan. 2011

[14] Selman, Daniel. “Java 3D Programming”, Manning Publications; 1 edition, 2002, ISBN-13: 978-1930110359

[15] Roman Lygin. *CAD Exchanger.* 28 Dec. 2010 http://www .cadexchanger.com/ 14 Jan. 2011

[16] Mediamachines. *Mediamachines web blog.* 21 Feb. 2007 http://mediamachines.wordpress.com/flux-player-and-flux-studio/ 14 Jan 2011