Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-6 (2016) 144–149

Comparison of Various Technologies Used in a Virtual Laboratory

Pavel Trnka* . Stanislav Vrána**
Bohumil Šulc***

Czech Technical University in Prague, Faculty of Mechanical Engineering
Department of Instrumentation & Control Engineering, Prague, Czech Republic
(Tel: +420 2 2435 2769; e-mail: pavel.trnka@fs.cvut.cz*,*
*stanislav.vrana@fs.cvut.cz**, bohumil.sulc@fs.cvut.cz***)}*

Abstract: The usage of remotely accessible sources increases in education. On one hand, there are passive sources as electronic libraries, scanned books and webservers allowing cooperative production of education materials. On the other hand, there are virtual laboratories allowing participating of users on the conducted experiments. The server side and client side parts of virtual laboratories can be based on various technologies; the aim of this paper is to give a kind of comparison of several of them. The paper also introduces an approach to the laboratory models with extended use that can be used in both ways, as remotely accessed models allowing only a limited control and as locally accessed models allowing full control. Because of the limited length of the paper, the aim is not to provide a complete survey through all possible technologies usable to the virtual laboratories development, but to provide the practical experience with an implementation of some of them.

Keywords: education, educational aids, human-machine interface, interactive, process models, remote control, computer simulations, speed control

1. INTRODUCTION

In technical education, there is a growing trend to allow the students to access to some study materials and supporting tools remotely. Common study texts are often placed on web pages, in the form of downloadable PDF files, wiki texts (e.g. In Czech Republic, there are WikiSkripta established by medical faculties at Czech and Slovak universities, *wikiskripta.eu*), and sources combining both approaches; i.e. the text is put at the wiki in the editable TEX format and the final document is downloadable as PDF file (e.g. WikiSkripta of Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, *wikiskripta.fjfi.cvut.cz*).

In the field of automatic control education, universities often offer the remote access to the control tasks via virtual laboratories. These virtual laboratories can contain (as the name can suggest) not only virtual control tasks but they can also mediate access to the real tasks. For both, pure virtual tasks and remotely accessed real tasks, various technologies can be used.

2. TASKS IN THE VIRTUAL LABORATORY

The virtual laboratory of Department of Instrumentation and Control Engineering at Faculty of Mechanical Engineering, Czech Technical University in Prague, accessible through *vlab.fs.cvut.cz*, contains four types of tasks: pure virtual tasks in Java, real tasks controlled via Java, real tasks controlled via Matlab, and pure HTML/JavaScript controlled real tasks. In the past, the Matlab simulations accessible via MatWeb technology were a part of the virtual laboratory, but as the MatWeb technology is no longer supported, the tasks were abandoned.

2.1 Pure virtual tasks

The pure virtual tasks are based on Java technology. They were created according to their real versions. There are virtual tasks for the logic control (Hofreiter and Urbanek, 2010) – “Entrance gate”, “Rotary table with the conveyor belt”, “Water level control”, “Railroad control”, “Pneumatic motor and vacuum pad”, “Parcel lifting platform” and for the continuous control – “Bathyscaphe”, “Ball and elliptic rail”, “Ball and beam”, “Water reservoir”, “Water levitation”, “Vehicle position control”, “Air levitation”. All the virtual tasks are created in Java with no support from HTML. An example of pure virtual task is shown in Fig. 1.

2.2 Real tasks controlled via Java

In principle, those tasks are similar to the pure virtual tasks; only the real device is used instead of its mathematical model. And also, the web camera is used instead of the graphical visualisation.

The virtual laboratory of Department of Instrumentation and Control Engineering contains a “Watt’s governor” as a task of this type (see Fig. 2). To ensure that more than one student can access the task simultaneously, four instances of this task are present and user is directed to the one that is currently free. If no free task is available, the visitor has to wait till any other visitor finishes his experiments (Trnka et all, 2013).

10.1016/j.ifacol.2016.07.168

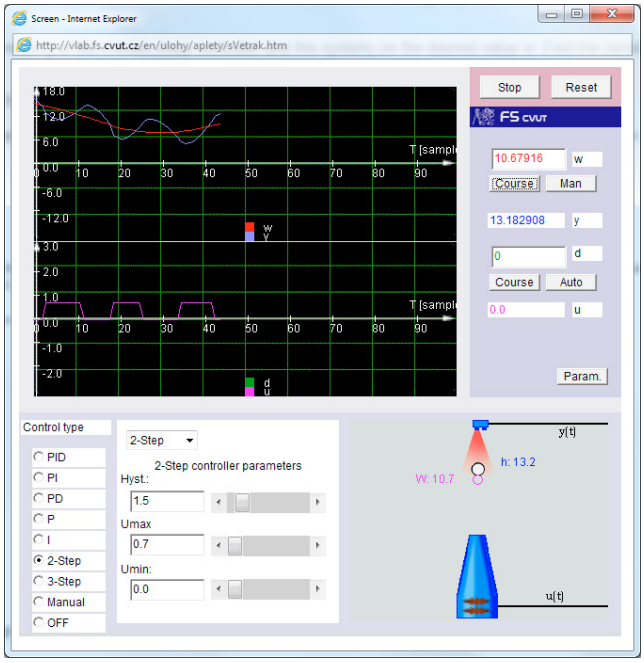

Fig. 1. The example of pure virtual task coded in Java – "Air levitation"

2.3 Real tasks controlled via Matlab

This type of task contains a network interface that expects commands from outside and that provides the current data about the state of a laboratory model. The control part is downloadable in the form of *.m* file, *.mdl* file, or newly *.slx* file.

2.4 Pure HTML/JavaScript controlled real tasks

Each of these tasks is equipped with its own mini-computer used to the control and to the remote access via WebSocket technology. Due this fact, the task can be controlled from any web browser supporting the WebSockets. The main control algorithm is written in non-web language, but all its parameters and variables are available via the WebSocket. Whole interface to control those tasks is created in HTML and JavaScript.

It is possible to see the courses of algorithm variables continuously, and also it is possible to see the task itself if a web camera is available.

2.5 Matlab simulation accessible via MatWeb

The MatWeb was an interface between Matlab and webserver, in principle working similarly like e.g. PHP. Before execution of simulation, the simulation parameters needed to be set and then transferred via MatWeb to a simulation model. After the simulation was finished, the results were generated in the form of figures and a HTML file. It was not possible to see the progress continuously, only the final results were shown. The only possibility to see the experiment online would be to control the real process using Matlab and to observe it via webcam. The MatWeb technology is no longer supported by MathWorks, so some replacements have been performed; on the other hand, they have similar advantages and disadvantages as MatWeb had.

Instead of Matlab and MatWeb, it is possible to use own algorithm in C++ (Costa et al., 2010) or Scilab (Magyar and Zakova, 2012).

2.6 Comparison of available approaches

The approaches to virtual laboratory task can be compared from many viewpoints. The most important is the accessibility. From this point of view, it is better when the task does not require downloading of any other supporting tool. According this criterion, the suitable tasks are those controlled purely from HTML/JavaScript and those accessed through MatWeb; in both cases, the user needs only the web browser.

Less suitable are tasks requiring a download of the Java Virtual Machine. Although it is declared that the programs written in Java are executable in any device, the reality is different and even incompatibility between two different versions occurs. The least suitable is the requirement to have a Matlab program in the computer. This limits the range of people able to access the task only to people using Matlab already; it is not expectable that somebody will buy the Matlab only to connect to one laboratory task.

Another important property of the virtual laboratory task is the possibility to show the current process variables. From this point of view, the tasks controlled via Matlab accessed through MatWeb are not suitable, because this approach requires to set the parameters at the beginning, then to conduct the simulation, and only then it is possible to show the result in a static page. It is not possible to see the variable courses during the simulation. It is also impossible to leave the task running 24 hours a day and just see the recent values after reconnecting to it.

It is also necessary to solve the simultaneous access of more

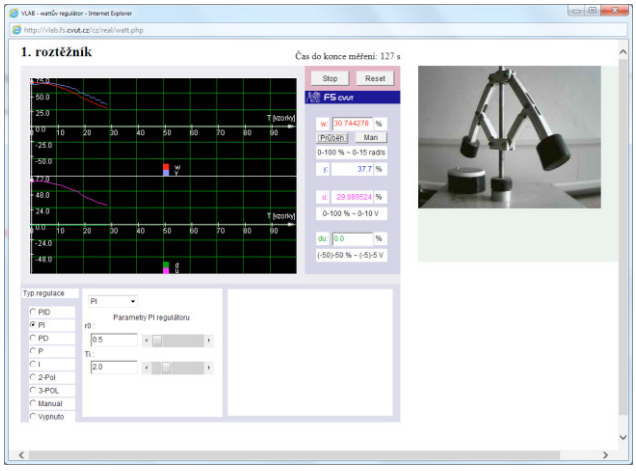

Fig. 2. The Java controlled real task "Watt's governor"