Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-6 (2016) 133–137

A new Model for a Remote Connection with Hardware Devices using Javascript [⋆]

J. Sáenz [*] F. Esquembre. [**] F.J. García [***] L. de la Torre [*] S. Dormido [*]

[*] *Departament of Informatics and Automatics, Computer Science School UNED, Spain, (e-mail: jacobo.saenz@bec.uned.es,ldelatorre@dia.uned.es, sdormido@dia.uned.es).*
[**] *Mathmatics Faculty, Universidad de Murcia, Campus Espinardo,Murcia, Spain (e-mail: fem@um.es)*
[***] *Departament of Computer Engineering and Technology, Informatics Faculty, Campus Espinardo, Murcia, Spain. (e-mail: fgarcia@um.es)*

Abstract: Nowadays online resources play an important role in teaching and learning thanks to new advances in technology. This importance is enhanced in scientific areas and even more for distance education universities, where the classic hands-on laboratories are not always possible. For that reason, these face-to-face laboratory practices have been replaced or even complemented with online virtual and remote laboratories (VRL). Normally, these applications are developed with high level programming tools, and these, to greater or lesser extent, use Java. Unfortunately, the newly discovered Java security issues and the impossibility to run Java in smart devices are restrictions to the dissemination of this kind of applications. This work is the first step towards get a new structure to Easy Java/Javascript Simulations (EjsS) that allows a remote connection with hardware devices using javascript. In this regard, the first objective is to solve the problems with Java applications when using EjsS. The proposed solution provides the user a structure to reuse their VRLs using a Java model that runs in a server and a Javascript graphical user interface in the client device.

Keywords: Virtual and Remote Laboratories; Distance Education; Control Engineering

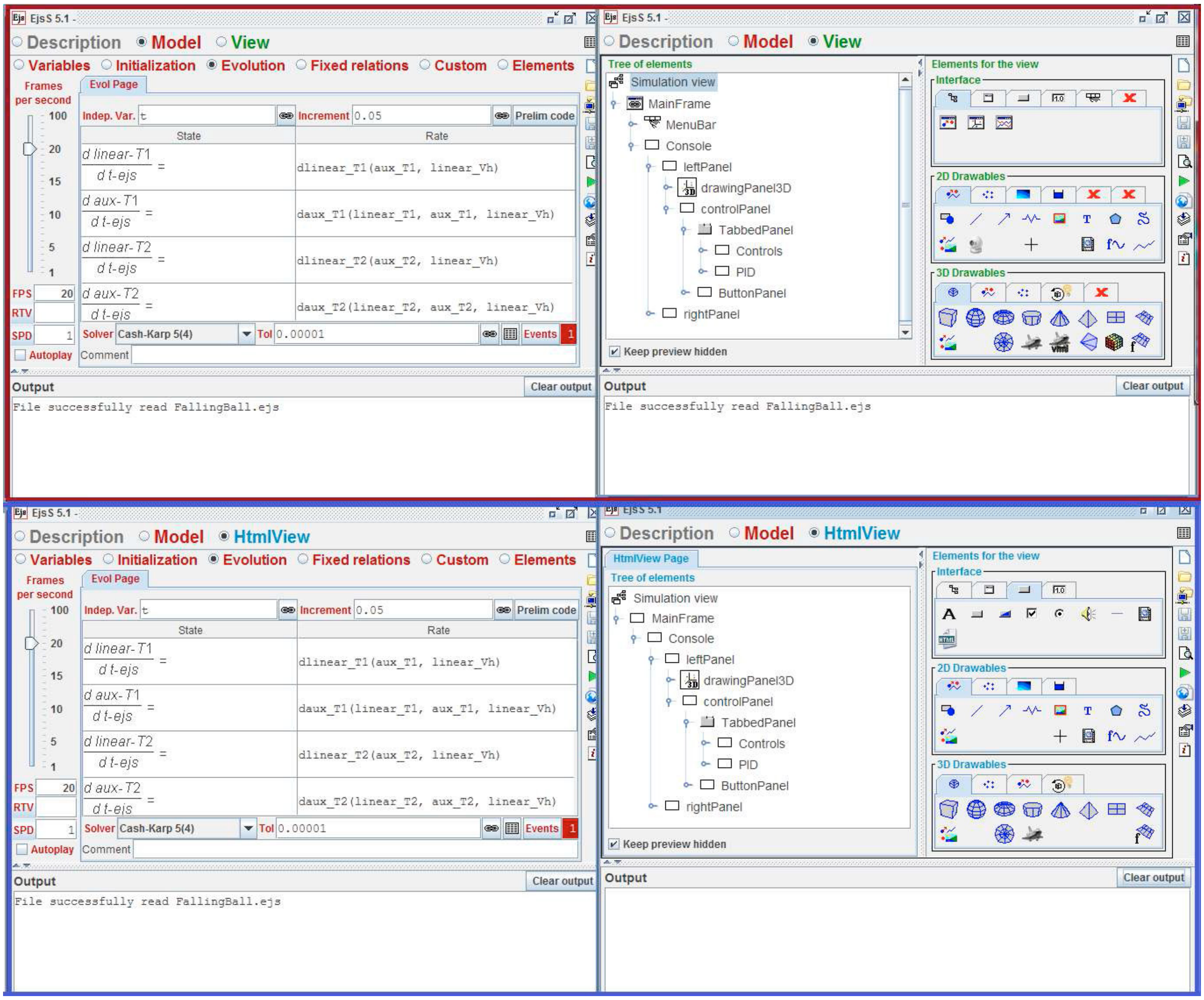
1. INTRODUCTION

The use of online resources as a complement to the theory have become widespread over the last years. Thanks to videos, applications, simulations and VRL the student can enhances their traditional lessons to obtain a better understanding of the theory. This kind of tools and applications have been added by many universities to their courses: (Andreja Rojko, 2010; Farias et al., 2010; Bose, 2013; Guimaraes et al., 2011; Hassan et al., 2013; Santana et al., 2013; Vavougios and Karakasidis, 2008; Tawfik et al., 2013; Yuliang Qiao and Hu, 2010; Rojko et al., 2010; Yazidi et al., 2011)).Normally these extra materials are created by using integrated development environments or tools that use Java. Over the last years many Java vulnerabilities have been appearing, in consequence, some browsers like Google Chrome do not support Java. In addition, the restrictions on running non digitally signed Java applets makes even worse the common dissemination of VRL based in this technology. Furthermore, new mobile devices do not support Java either, adding another impediment for the publication of VRL. To deal with the Java security problems some developers have written their applications using Javascript, works like (Frank and Kapila, 2014) are examples of this kind of solution. In this regard, the use of Javascript solves the two problems but involves some new issues due to:

- The limited computational resources of a smartphone or tablet pc. A complex VRL can overload the device while running.
- Multiple applications developed in Java will become useless with this change, and rebuild all existing VRL from scratch can be a huge task.

This work presents a solution applied to solve these problems when using EjsS for creating VRL, also is the first step towards obtaining a functional communication over internet between a GUI in Javascript and hardware devices. The paper is organized in five sections. Section II describes the two different versions of the latest update of EjsS. Section III contains the new architecture proposed to expand the EjsS capabilities in order to answer the problems presented before. Section IV gives an overview of the advantages using this architecture. Finally, Section V adds some conclusions and describes further work in order to improve this investigation.

[⋆] This work was supported in part by the Spanish Ministry of Economy and Competitiveness under Project DPI2012-31303.



Indep. Var. t — Increment 0.05 — Prelim code

State	Rate
d linear-T1 / d t-ejs =	dlinear_T1(aux_T1, linear_Vh)
d aux-T1 / d t-ejs =	daux_T1(linear_T1, aux_T1, linear_Vh)
d linear-T2 / d t-ejs =	dlinear_T2(aux_T2, linear_Vh)
d aux-T2 / d t-ejs =	daux_T2(linear_T2, aux_T2, linear_Vh)

FPS 20 — RTV — SPD 1 — Solver Cash-Karp 5(4) — Tol 0.00001 — Events 1

```
File successfully read FallingBall.ejs
```

Fig. 1. Both, main views of the EjsS editor. Red: Java, model, with some differential equations, and view tabs (green text). Bottom: Javascript, model, with the same differential equations, and view tabs (cyan text)

2. EASY JAVA/JAVASCRIPT SIMULATIONS

2.1 The EjsS Tool

EjsS is an open source authoring tool designed for students and teachers. It offers an easy way to create a GUI for simulations according to the user needs of interactivity and visualization. Due to the problems presented in the previous section, the last versions of EjsS allows the user to create applications in both Java or Javascript. The figure 1 shows both editors together. EjsS also allows the user to run a finished application directly from the editor. If the aim is to publish the VRL, the developer can package it in order to run it later in a standalone mode (in the case of Java) or inside a web page to be run in a web browser (Java and Javascript). This tools have been presented in other works (Farias et al., 2010; Chacon et al., 2015), where EjsS have been defined as a tool that facilitates the development of applications by researcher, teachers and students who want to focus in the simulation theory and not in the technical programming aspects.

2.2 The EjsS Java mode

The figure 1 shows, into a red square, the main view of the Java enabled version editor. The top part in the editor contains three tabs: description, model and view, which are shown in the figure. EjsS editor allows to develop an application structured in two main parts, the graphical user interface, or view, and the model:

- The *model* can be seen as Java programming code, differential equations (left side of images in Figure 1) and/or connections to other software or hardware. The simplicity or complexity of the model depends only on the developers requirements and their knowledge about the simulated system.
- The *view* provides to the users a GUI that determines the interaction and visualization capabilities of the application. This view can be built using the editor