

UN-VirtualLab : A web simulation environment of OpenModelica models for educational purposes

Oscar Duarte

Universidad Nacional de Colombia, Department of Electrical and Electronics Engineering
ogduartev@unal.edu.co
Carrera 30, Calle 45, Ed. 453, Of. 202. Bogotá, Colombia

Abstract

In this paper a free web simulation environment is presented: UN-VirtualLab . It is a virtual laboratory in which users can perform experiments on precompiled software models. UN-VirtualLab uses OpenModelica in order to compile software models written in Modelica language. The main features and internal architecture of the system is presented. Some potential applications are discussed.

Keywords: simulation environments, web applications, OpenModelica, education

1 Introduction

UN-VirtualLab is a web simulation environment distributed under GPL license. It is a virtual laboratory in which users can perform experiments. Using a web browser, any regular user can pick up an experiment, modify its parameters and simulate its response. The plant in which the experiment is done does not physically exist, it is a pre-compiled software model. UN-VirtualLab uses OpenModelica in order to compile software models written in Modelica language.

The term ‘virtual laboratory’ refers, in a broad sense, to an electronic and software workspace for experimentation. Many different approaches satisfy this definition. In order to clarify what kind of them UN-VirtualLab is, we may use some classification criteria:

- *Web availability:* Some virtual labs are installed in a PC as a local software package where as others are installed in a web server. Usually, in the second approach the user connects with the server through a conventional internet navigator. In this paper we use the term ‘virtual’ for web based tools, such as UN-VirtualLab .
- *Physical vs. software models:* In some virtual labs the plants in which the experiments are con-

ducted physically exist ([1], [2]) where as in others they are software models([3],[4]). In the first approach there are sensors and actuators connected with the virtual lab, and the user can manipulate them. In UN-VirtualLab the plants are software models.

- *Interactivity:* In [5] a distinction is made between two types of software model based Virtual labs: those which allow the user to perform actions during the simulation and those which not. The first approach is referred as *runtime interactivity* and the second one as *batch interactivity*. UN-VirtualLab has batch interactivity.
- *Subject orientation:* Some virtual labs are designed to satisfy an specific need ([4], [6]) where as others are of general purpose ([7]). In the first approach, the solution is not easy to use in a different context; for example, a virtual lab for biological plants can not be converted to a virtual lab for mechanical plants. UN-VirtualLab is of general purpose.
- *Course orientation:* Some virtual labs are course oriented ([3]) where as others are not. In the first approach, there are tools as student and grade management, individual and group progress reports, etc. UN-VirtualLab is not course oriented.

Some efforts have been made to combine Modelica and virtual labs: In [8] a web version of the well known DrModelica software is presented. In [9] and [5] virtual labs are developed by using a combination of Dymola and Sysquake. In [10] a Modelica-based algorithm is developed to implement the interactive mechanism. In [11] a web service is developed to compile and simulate remotely Modelica based models.

In this paper we show a simple and different approach, based on OpenModelica. OpenModelica is

an open source modeling and simulation environment ([12], [13]). Using OpenModelica it is possible to compile Modelica models. Once the compilation is done, an *executable file* is available. When running, the executable reads an *input file* and writes a *results file*, as shown in figure 1.

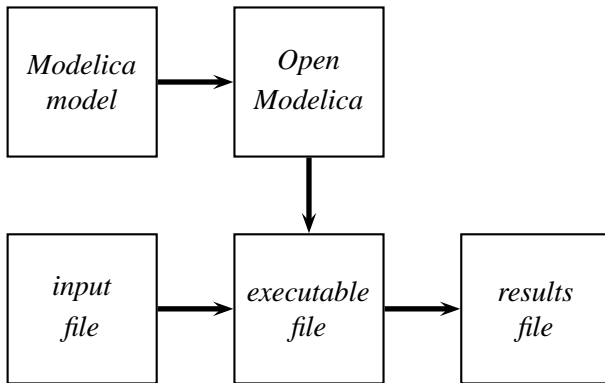


Figure 1: Files in an OpenModelica simulation.

UN-VirtualLab interacts in a web environment with these files in the following way:

- Brings a graphical interface to modify selected parameters of the *input* files.
- Runs the *executable* file.
- Displays the data of the *results* file using plots, tables as well as 2D and 3D animations.

In this paper we summarize the main features of UN-VirtualLab (section 2) and describe its internal architecture (section 3). We also discuss possible applications in section 4. Conclusions and future work are presented in section 5.

2 Features

In UN-VirtualLab experiments are organized by nested subjects. It is possible to define a tree of subjects and include any number of experiments in every subject.

The same Modelica model can be used in several experiments. As an example, using the same model of an electrical vehicle it is possible to design an experiment to analyze de controller performance, another to perform sensitivity analysis of the vehicle mass, and another one to study power consumption.

Once a user selects an specific experiment, he/she can:

- Modify the parameters of the experiment. The parameters are those defined in the *input file*. For convinience, they are classified in three types:

Simulation parameters: such as *start time*, *stop time*, *step value*, *tolerance* and *method of integration*.

Initial conditions: start value of simulated variables.

Model parameters: any other parameter in the *input file*. These parameters can be arranged in groups.

- Simulate the model and visualize the results. There are up to four options of visualization:
 - Plots.
 - 2D animations.
 - 3D animations.
 - Data tables.
- Read or download the experiment documentation, including:
 - Model description.
 - Modelica source code.
 - Author information.

Other features are:

- Multilanguage support, selected by the administrator.
- Easy appearance customisation, using css themes.

3 Architecture

UN-VirtualLab is written in php language. Modelica models, *input*, *executable* and *results* files of every experiment are stored in individual directories. Every single experiment is defined by an *xml file*. System administrator can modify these xml files using a graphical interface (see figure 2).

When a user picks up an experiment, the system reads the corresponding *xml* file and creates a graphical interface that shows the outputs of the simulation with the default values of the parameters. It also shows a form so the user can change these values and launch a new simulation (see figure 3).

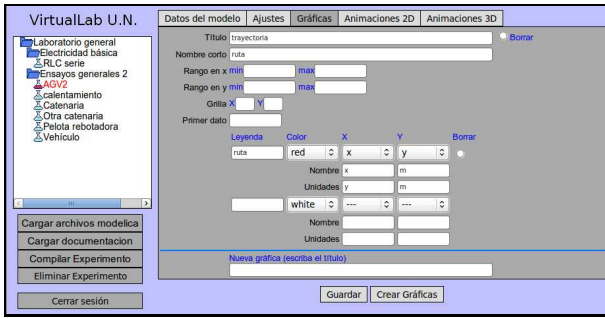


Figure 2: Administrator interface

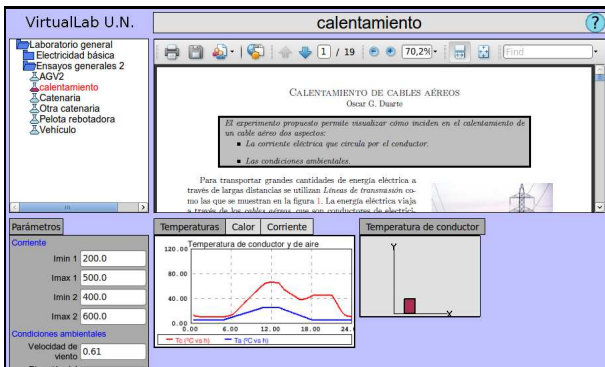


Figure 3: Experiment interface

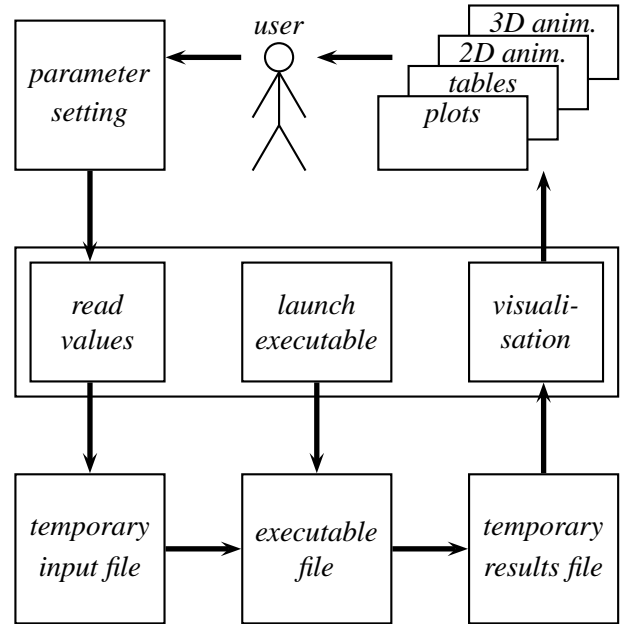


Figure 4: Files in a UN-VirtualLab simulation.

Figure 4 shows how UN-VirtualLab processes a simulation order. When the simulation is launched, the actual values defined by the user are read and a *temporary input file* is created, the system runs the *executable file* and creates a *temporary results file* which is used to generate on line plots, animations and tables. Then, temporary files are removed.

Notice that the same structure shown in figure 4 can be used with any executable file that uses input and results files, not just by OpenModelica compiled files. In that sense, UN-VirtualLab has a very general structure and can be used with a broad spectrum of simulation packages. However, UN-VirtualLab actually recognizes just the input and results file formats used by OpenModelica.

3.1 Plots and tables

Curves to be plotted are defined by selecting a pair of simulated variables. Usually, the first one is *time* but not necessarily, so it is possible to plot two-dimensional phase portraits. As an example, consider the bouncing ball plant, in which h represents the height of the ball, v its velocity and t the time. We can plot $(h$ vs. t), $(v$ vs. t) and $(h$ vs. v).

Plots are made on line taking the values written in the *results file*. A php class has been written to produce the images that are displayed using the png format. Using the same plot more than one curve can be drawn, as figure 5 shows.

The data plotted is also available in data tables. User can download the data of his own simulation as plain text. Columns are separated by <TAB> character, so it is possible to import directly the downloaded data into an spreadsheet as OpenOffice Calc.

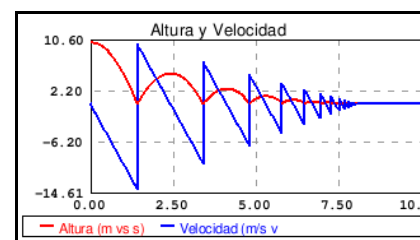


Figure 5: Plot example. Height and velocity of a bouncing ball.

3.2 2D animations

2D animations are based on some primitives whose properties are changed by the values stored in the *results file*. The primitives available are: axis, rectangles, ellipses, rings and polylines. The properties that can be driven by the simulation results are: size (x and

y), position (x and y) and rotation (around z axis, orthogonal to the animation plane).

Following with the bouncing ball example, a 2D animation can be made with two primitives: a rectangle for the floor, and a circle for the ball (see figure 6). The y coordinate of the circle position can be driven by the h variable in the *results file*.

2D animations of UN-VirtualLab are animated png/gif files. In order to build every frame, several php classes have been written. The combination of the individual frames in a single png or gif animated file is done with the *apng-creator* and *dgifanimator* libraries respectively [14]. The number of frames and the time between frames of every animation can be adjusted. It is also possible to configure the ($x - y$) position of the camera and an scale factor.

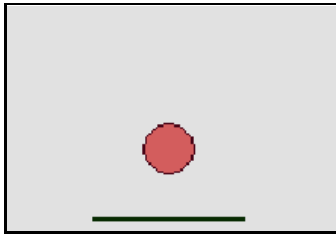


Figure 6: 2D animation snapshot. Height of a bouncing ball.

3.3 3D animations

3D animations are made in a similar way to 2D animations. Primitives available are: axis, cubes, spheres, pipes and cylinders. The properties that can be driven by the simulation results are: size (x , y and z), position (x , y and z) and rotation (around x , y and z axis).

Figure 7 shows an snapshot of a 3D animation of the bouncing ball. It has been made with two primitives: a cube for the floor, and an sphere for the ball (see figure 6). The z coordinate of the sphere position is driven by the h variable in the *results file*.

As in 2D animations, the number of frames and the time between frames of every animation can be specified by the system administrator. It is also possible to configure the (x, y, z) position of the camera, (x, y, z) focus point and an scale factor.

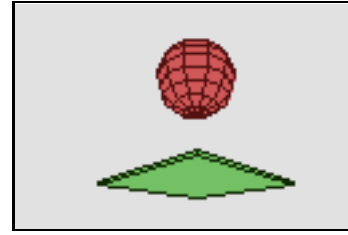


Figure 7: 3D animation snapshot. Height of a bouncing ball.

3.4 Documentation

UN-VirtualLab displays information about the experiment. The experiment author must prepare this information as pdf files. UN-VirtualLab uses *pdftohtml* to produce the html files from the pdf.

Authors can use \LaTeX and a suggested \LaTeX style to produce the pdf files. The suggested style has a customisation of the *listings* package ([15]) for publishing Modelica source code. It recognizes a subset of the Modelica language specification, often enough to produce fancy documentation files (see figure 8).

File 1: Asinh.mo

```
within Catenary ;
function asinh
  input Real x;
  output Real y;
  external "C" y=asinh(x);
end asinh;
```

Figure 8: Example of \LaTeX output of Modelica source code using the suggested style.

3.5 Layout and appearance

The user interface has five blocks, as shown in figures 3 and 9. They are:

1. Experiment selection block: a tree menu to pick up the subject and experiment.
2. Parameter settings block: a dialog form to change de default values of the choosen experiment parameters.
3. Documentation block: a frame display the pdf/html experiment documentation, and the links to downloadable files (Modelica source code and documentation files).

4. Plots and tables block: two frames to display the simulation results as plots and data tables, respectively.
5. Animations block: some frames to display the simulation results as 2D and 3D animations.

The appearance of the interface can be changed using different css themes. Not only colors and fonts can be changed, but also the size and position of the blocks.

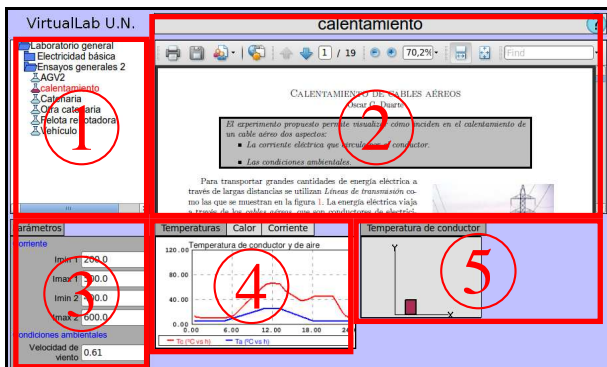


Figure 9: Layout

3.6 Multilingual support

UN-VirtualLab has multilingual support. Actually there are two languages available: English and Spanish. Two different aspects has been addressed to implement multilingual support:

- Common interface: all the strings that are common to all experiments. They are defined in a single file.
- Experiment data: every single experiment has specific strings as experiment name, parameters names, plots titles, etc. These strings are defined in the *xml file*. It is possible to define different *xml files* for the same experiment, each one for a different language.

4 Possible applications

UN-VirtualLab is not intended to replace simulation tools as Dymola, SimulationX or OpenModelica. The main purpose of UN-VirtualLab is to publish simulation experiments on the web. Some potential applications are the following:

1. Publishing research results: suppose you have finished a research project and as a result you have a novel dynamic model of something. You have published good papers in recognized journals, but you also want to explain the results to a wider public. You may use UN-VirtualLab to let the visitors of your web site to explore your model.
2. Novice students laboratories: suppose you are in charge of a first year course in an engineering program. You want your students to know something about more advanced topics, perhaps just to help them to understand some basic concepts. You want them to experiment with some plant, but they do not have yet enough skills and knowledge neither to do it in real life nor to write a simulation software. You may use UN-VirtualLab to bring them a convenient simulation environment.
3. Complement traditional teaching: the benefits of simulation environments in traditional teaching have been widely reported (see, for example [16]). Using UN-VirtualLab it is possible to design some experiments that help students to explore more aspects of a concept than those explored in the classroom.

5 Conclusions and future work

UN-VirtualLab provides a light web simulation environment for pre-compiled software models. Using web simulation environments, many people can access the same simulation engine and the licenses costs is reduced. Using UN-VirtualLab and OpenModelica, it is possible to implement a totally free software solution.

There are some aspects that must be reinforced in the short term:

- As stated in section 3, even that UN-VirtualLab internal structure is very general, actually it recognizes just the input and results files produced by OpenModelica. More formats should be recognized.
- The animations can be more complex, by driving more primitive properties such as colors and line widths. The use OpenGL and other graphical libraries must be explored.
- It is important to research how to implement interactive simulations. The use of the interactivity option of OpenModelica through the web is not trivial, but must be explored.

The first public application of UN-VirtualLab is available at the Virtual Academic Services of the National University of Colombia ¹. According with the actual schedule, in June of 2011 it will include at least a hundred of experiments, most of them from engineering subjects.

References

- [1] R. Pastor, C. Martín, J. Sánchez and S. Dormido Development of an XML-based lab for remote control experiments on a servo motor International Journal of Electrical Engineering Education. Vol 42 No.2 2005 pg 173- 184
- [2] Paul I-Hai Lin and Melissa Lin Design and Implementation of an Internet-Based Virtual Lab System for eLearning Support Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05) 05-08 July 2005, Kaohsiung, Taiwan.
- [3] Dongil Shin , En Sup Yoon, Kyung Yong Lee, Euy Soo Lee A web-based, interactive virtual laboratory system for unit operations and process systems engineering education: issues, design and implementation Computers and Chemical Engineering 26 (2002) 319– 330
- [4] Diwakar, Achuthan and Nedungadi Biotechnology virtual labs- Integrating wet-lab techniques and theoretical learning for enhanced learning at universities. IEEE. 2010 International Conference on Data Storage and Data Engineering Bangalore, (International), (February 2010)
- [5] Carla Martín Villalba Object-Oriented Modeling of Virtual Laboratories for Control Education Doctoral Dissertation Universidad Nacional de Educación a Distancia Escuela Técnica Superior de Ingeniería Informática Madrid, 2007
- [6] Hu Shaobin, Yin Daiyin, Cao Guangsheng, Guo Lingling Construction and Application of Autonomous Learning Based Remote Downhole Tool Virtual Lab 2010 2nd International Conference on Education Technology and Computer (ICETC) Shanghai, June 2010
- [7] Alfonso Urquía, Carla Martín-Villalba Laboratorios Virtuales Interactivos <http://www.euclides.dia.uned.es/simulab-pfp/index.htm>
- [8] Eva-lena Lengquist S , Susanna Monemar , Peter Fritzson , Peter Bunus DrModelica – A Web-Based Teaching Environment for Modelica In Proceedings of the 44th Scandinavian Conference on Simulation and Modeling (SIMS'2003)
- [9] Carla Martín-Villalba, Alfonso Urquía, Sebastian Dormido Object-oriented modelling of virtual-labs for education in chemical process control Computers and Chemical Engineering 32 (2008) 3176–3186
- [10] Wenbin Jiang, Zhen Tang, Hai Jin, Chao Liu MISM: Modelica-based Interactive Scheduling Mechanism for Virtual Educational Experiments The Fifth Annual ChinaGrid Conference Guangzhou, China, July 16-18, 2010
- [11] Sven Meyer zu Eissen, Benno Stein Realization of Web-based simulation services Computers in Industry 57 (2006) 261–271
- [12] OpenModelica project web site. <http://www.openmodelica.org>
- [13] Peter Fritzson, Peter Aronsson, Adrian Pop, Håkan Lundvall, Kaj Nyström, Levon Saldamli, David Broman, Anders Sandholm: OpenModelica - A Free Open-Source Environment for System Modeling, Simulation, and Teaching, IEEE International Symposium on Computer-Aided Control Systems Design, October 4-6, 2006, Munich, Germany
- [14] PHP Classes Repository. <http://www.phpclasses.org>
- [15] CTAN, the comprehensive T_EXarchive network. Listings package. <http://www.ctan.org/tex-archive/macros/latex/contrib/listings/>
- [16] Zacharias C. Zacharia, Georgios Olympiou Physical versus virtual manipulative experimentation in physics learning Learning and Instruction (2010) article in press
- [17] Dirección Nacional de Servicios Académicos Virtuales. <http://www.virtual.unal.edu.co>

¹<http://www.lab.virtual.unal.edu.co>