USING THE SYSTEM DYNAMICS PARADIGM IN TEACHING AND LEARNING TECHNOLOGICAL UNIVERSITY SUBJETS

José María González de Durana[1], Oscar Barambones[1],

Enrique Kremers[2] , Pablo Viejo[2]

[1] University College of Engineering, University of the Basque Country
Vitoria-Gasteiz (SPAIN)
[2]European Institute for Energy Research, Karlsruhe Institute of Technology and EDF
Karlsruhe (GERMANY)
E-mail: josemaria.gonzalezdedurana@ehu.es, oscar.barambones@ehu.es,
kremers@eifer.org, pviejo@eifer.org

Knowledge of Differential Equations is applied to various scientific fields such as physics, chemistry, biology and engineering and therefore often an important part in the basic subjects of mathematics in the first college courses related to those areas. The logic and common sense seems to indicate teachers use these basic skills acquired by students and employ them to curricula development in the following intensification courses, but unfortunately it is not usually the case. According to the authors, that is because instead of using a generic software to set up and solve the problems of Differential Equations that arise at different areas, what we have is a proliferation of software applied to solve special case problems. Some of these programs offer sophisticated graphical user interfaces to create complex system models, usually by putting together some library components, as if it were a puzzle, but without the need to set up the differential equations. According to the authors, this method, although valuable to solve some specific problems very quickly, is aberrant from the educational point of view, because it allows students to solve problems without knowing what they are doing or how they are doing. Worse, if a complication arises in the problem statement, for which there are no pieces in the puzzle, or execution errors occurs due to an incorrect construction, then they are not able solve the problem.

Because of this, software that does not hide the equations and with the user can know at any moment what he/she is doing, from the mathematical point of view, is missing. According to the authors, any simulation program including the System Dynamics paradigm meets this condition because its GUI is very close to differential equations and the Initial Value Problem. The modeling of a system using this paradigm is simply to raise because an initial value problem associated with the system is quickly represented by the graphical user interface of the simulation program.

This article presents some learning experiences focused on "problem based learning" using AnyLogic, which provides the System Dynamics paradigm to perform simulations of physical systems. The program provides a graphical environment that allows to perform animations very easily.

The first on is to simulate the filling of a tank of water where the model is a first order non-linear differential equation. This case is instructive as it is very easy to raise the initial value problem and may be valid to review some concepts already forgotten by the students such as for example the derivative, integral, differential equation and initial value problem.

Other simulation exercises posed to students are the control of a cart by a force, a parabolic shooting, and other mechanical, electrical and thermal examples.

Keywords: Problem Based Learning, Differential Equations, System Dynamics, Technology-Enhanced Learning, Computer Software on Education, ECTS Experiences.

1 INTRODUCTION

The knowledge of our students about Mathematics, Physics and Informatics, essential to initiate the study of Automatica, has been decreasing lately. Teachers we are compelled somehow overcome this deficiency, through activities and tasks that allow them to regain some of that knowledge forgotten. No easy task because assimilate some of these concepts requires a considerable amount of time.

It is said that our students have lost the habit of reading but are experts in console games and computer. And also that if the teacher writes many mathematical formulas on the board, more than one of them gets scared and does not return to class. How then we're going to teach the basics of control systems, full of derivatives, integrals and differential equations?

The simulation of a water tank is an excellent example to introduce some of these concepts. If we were able to explain without formulas, maybe things would change.

In the following, the authors give a personal experience in this regard, based on the realization of simulation exercises with the program AnyLogic

1.1 Solving methods for differential equations

The differential equations are an important conceptual part not only of automation or control systems but also in many other areas of science and engineering, each of which has developed its own methods to find solutions quickly. The wheel-and-disc integrator invented by James Thomson [1], brother of Lord Kelvin, was the first device that allowed for (mechanically) the operations of analog computation. Using the integrator as basic element, the two brothers built a device to calculate the integral of the product of two given functions. Kelvin designed other machines capable of integrating differential equations of any order, but they were never built.

To find the solution of a given explicit ordinary differential equation,

$$\frac{d^n y}{dt^n} = f\left(\frac{d^{n-1}}{dt^{n-1}}, \ldots, \frac{dy}{dt}, y, u, t\right),$$

together with the initial values, the idea of Lord Kelvin was to integrate with his device n times $y^{(n)}$, thus obtaining the values $y^{(n-1)}$, . . . , y' , and carry out with them the necessary (mechanic) arithmetic operations to obtain $f(\cdot)$, and then close the loop [2].

In 1950s came the analog computer, equipped with electronic integrators made by electron valves and based on the same ideas of Kelvin. This device allowed for obtaining solutions of differential equations in the form of electrical signals. Despite its high efficiency (especially compared to the mechanical integration methods), the analog computers lost importance with the advent of computers and now digital methods are predominant. Anyway Kelvin's method (or Forrester's) is still applied in the numerical algorithms.

Also on the the 1950s, Jay W. Forrester [3] had the need to make a simulation (of inventory control type) for the company General Electric. This first simulation that Forrester did with pencil and paper can be considered as the beginning of System Dynamics [4]. Later he asked Richard Bennett for help to solve the equations using the computer and created a compiler, called SIMPLE, for this purpose. Interestingly, the method of Bennet was the same (adapted to the computer) that Kelvin used to solve differential equations by mechanical methods.

Since then, successive generations of SD people have spoken in the *Forrester language*.

1.2 Simulation and modeling tools used in this paper

Matlab is probably the most widely used simulation program for control systems at university, although there are many other programs like Maple, Mathematica, Octave, Scilab, etc. that are also very common. With Matlab, simulations are possible in a mathematical sense, i.e. to apply numerical methods for solving some differential equations representing the system. However, being a program designed with the technologies of the 1950-60s, it lacks the advantages of more modern object-oriented based software. These advantages are evident if the system to model is of discrete event type and even more if it is a hybrid or an agent-based system.

AnyLogic is a recognized program in the community of multi-paradigm simulations, but little known in the areas of automation and control engineering, which is based on the latest advances in object-oriented modeling applied to complex systems [5]. It currently support three approaches or modeling paradigms:

- System Dynamics (SD)
- Discrete Events (DE)
- Agent Based (AB)

These three paradigms are mutually compatible, so that, for example, to model a hybrid system we will use the SD method to model the continuous part of the differential equations and the DE method for modeling the events. AnyLogic models are portable Java applications that can run on their own. They are also multiplatform and can run anywhere a Java Runtime Environment (JRE) is installed. So the models can be also run in a web browser in form of a Java Applet, which allows for an easy way of publishing the models. Moreover, it is very easy to develop animations of active objects: the assembly of the image is done automatically. In this way the animations are highly reusable and can be displayed on the applets.

2 SYSTEM DYNAMICS EXAMPLES

The examples presented below have been chosen to expose some typical problems well known by DS people, especially electrical engineers, accustomed to use sophisticated circuit analysis tools, or mechatronics analysis tools, and to encourage them to use the System Dynamics paradigm. These examples prove that (with a little effort) SD methods are also perfectly valid for analysis of mechanical and electrical systems and that they are easily integrable into multi-approach modeling environments, allowing for integration of DS people within multidisciplinary modeling projects.

2.1 Numerical ODE solution using an SD model

An initial value problem (IVP) is an ordinary differential equation (ODE) with a given initial condition (in form of a specified value) of the unknown function at a given point in the solution domain. In physics and engineering resolving these kind of problems is common, as the differential equation describes a system which evolves with time according to the specified initial conditions. Using SD we can obtain the numerical solution of the first order control differential equation (initial value problem),

$$x(t)=f(t,x(t),u(t))$$
$$x(0)=x_0$$

where $t\in\Re$, $x(t)\in\Re$, $u(t)\in\Re$ and $f:\Re\times\Re\times\Re\rightarrow\Re$, the (control) function *u(t)* is given.

Note that the above equation can also represent a multidimensional system if for some given integer *n* there are $t\in\Re$, $x(t)\in\Re^n$, $u(t)\in\Re^m$ and $f:\Re\times\Re^n\times\Re^m\rightarrow\Re$

AnyLogic hyperarrays allows us to model these multidimensional differential equations, using a very single Forrester diagram, as can be seen in Figure 1.

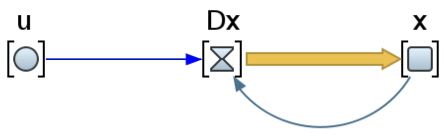

Figure 1: Forrester diagram of a multidimensional ODE

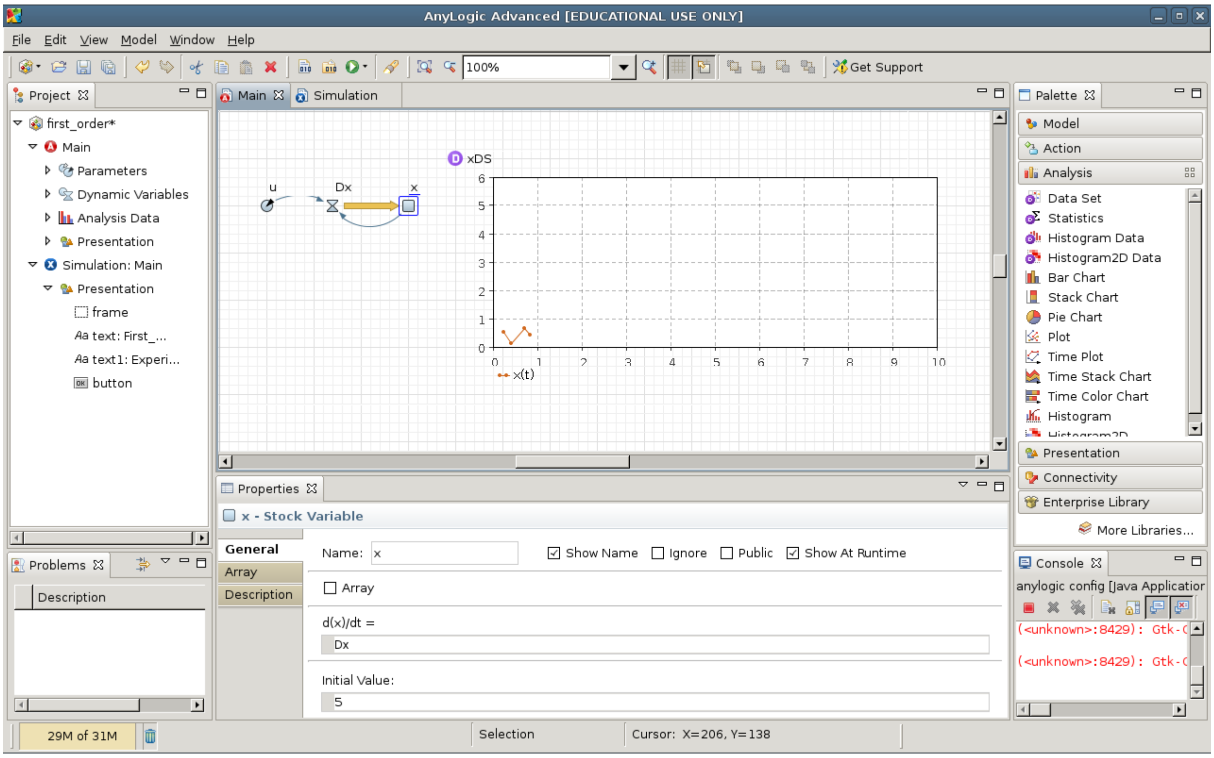

Figure 2: Anylogic interface enables differential equation modeling by drag & drop

This diagram can represent a first order system when $n = 1$ or, in general, an orden n system for $n > 1$ and then u(t), Dx(t) and x(t) are hyperarrays. The (given) function u(t) represents the control input. Dx is a flow variable representing the derivative of the unknown function x(t), represented as a stock.

The thick arrow is nothing more than the integrator DS object whereas the thin arrows indicate dependencies of the function

$$f\left(t, x\left(t\right), u\left(t\right)\right).$$

that is, f depends on x and u (also it depends implicitly on t).

2.2 First order system

To explain a concrete example, a first order IVP given by

$$\frac{dx}{dt} = ax\left(t\right) + bu\left(t\right)$$
$$x\left(0\right) = x_0$$

where parameters a, b and control input are given. This system will be modeled as follows.

The way to model this system is very easy: after placing the selected objects from Palette into Main window, some Properties should be assigned to them, by clicking them. After dragging a Stock variable and two Flow variables from Palette to the Main window, by clicking at each one of them we will rename them with appropriate names and assign them some pertinent properties. So in Stock x variable properties we write

$$\frac{dx}{dt} = Dx$$

as well as its initial value *x*0. In Flow D*x* variable we should write the function description that in this case is

$$ax+bu$$

where *a, b* and *x0* are Java variables of type double, declared within the Main window properties.

2.2.1 Tank model

To better understand the first order differential equation, we analize one of simplest phisical systems: a liquid holding tank. The tan example is very appropiate to start because everybody has the intuitive idea of a container that fills with water, and may be a good example to encourage the student to remember some important concepts. It lies in a natural way the idea of integral as well as the notions of differential equation and initial conditions problem. This simulation will help you to remember all these important concepts.

From their basic notions of physics students know the law of conservation of mass so the difference between input and output flows must be equal to the liquid volume variation, so

$$q-q_s=A\frac{dy}{dt}$$

where *A* is the constant tank area and *y*(*t*) is the liquid level. If the liquid friction is negligible then the output velocity is $v_s=\sqrt{(2gy)}$ so the output flow is $q_s=bv_s=b\sqrt{2gy}$ and we obtain the Initial Value Problem:

$$\frac{dy}{dt}=\frac{-b}{A}\sqrt{2gy}+\frac{cp}{A}a$$
$$y(0)=y_0,$$

where *g* is the gravity constant, *p* is the input pressure and *c* is a constant.

Simulation of this model in AnyLogic has the advantage that easily allows an animation simply by drawing a rectangle to which we assign the dynamic property height (height) equal to y. Thus we see how to fill the tank in the simulation. A picture (animated!) Is worth a thousand words.

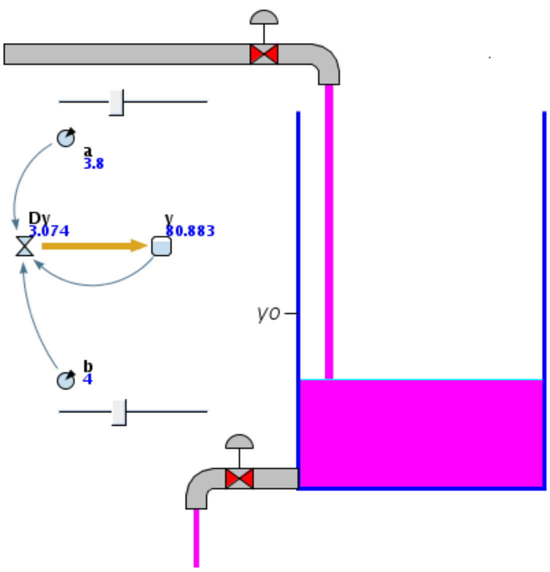

Figure 3: Tank Model

Figure 3 shows the diagram for the IVP System Dynamics and design of the animation of the tank with two valves, each of which is associated with a slider that can vary their section. Thus the tank can fill and empty at will during the simulation user.

Our students, playing the simulation (it must be remembered that if some have diminished the appeal of reading then it is almost certain that they are experts in digital games), will soon realize that the liquid level can exceed the maximum height the reservoir, which in reality lead to a spill of liquid. And so arises the inevitable question: professor, why water is flowing through the valve once the reservoir is filled completely? Then, the teacher, looking perplexed and mind on the theory of differential equations, may give them an unconvincing answer. We think that, instead, it would be more effective to reuse the resources of the student, the ideas he has, unknowingly, on discrete event systems and hybrid systems. He knows wow to fill a jug of water and also that must observe its level and close the valve when the level reaches its maximum value. With the teacher help and a little effort on his part, the student will be able to learn to use Statecharts in order to model the Discrete Event parts (sensors, actuators, controller) and this way complete the model as he pleases.

Students will not only managed to increase their motivation, but also will acquire some basic notions of the area of automation, which will later be helpful for learning other subjects related to this issue.

2.2.2 RL circuit

As first example of an electrical system, the RL circuit will be discussed. It consists of a resistor, represented by the letter R and an inductor, represented by the letter L. Resistor and inductor are connected in series in this example, as shown in the Figure 3.

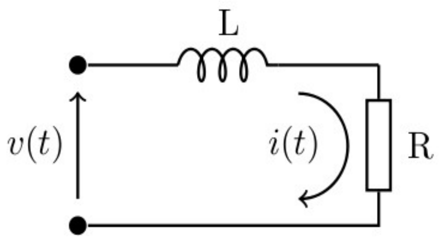

Figure 4: RL circuit

The problem that arises is: for given *v*(t), *i*(0) = *i*0 , calculate *i*(t).

In order to obtain the ODE initial value problem, we use the 2th Kirchhoff law. Then we get

$$L\frac{di}{dt}+Ri=v \quad ,$$

so the IVP is

$$\frac{di}{dt}=-\frac{R}{L}+\frac{1}{L}v$$
$$i(0)=i_0$$

and the solution can be computed with AnyLogic in the same way as the previous first order system, taking into account that in this case the unknown function is i(t) and we have

$$a=-\frac{R}{L}, \quad b=\frac{1}{L}, \quad x(0)=i(0)=i_0$$

Of course the Forrester diagram is also the same as in the previous example.

2.2.3 RC circuit

The next example discussed is the RC circuit, which is quite similar to the previous one but instead of using a inductor, a capacitor, represented by the letter C, is considered here.

Like the RL circuit, the RC circuit can be used as filter for signals by letting pass only certain frequencies. Together with the RL circuit, the RC circuit exhibits a large number of important types of behaviour that are fundamental in analog electronics.

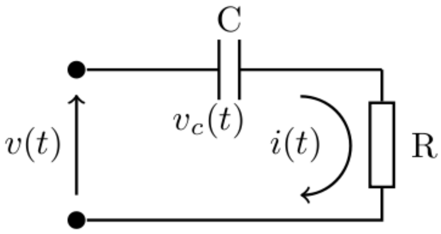

Figure 5: RC circuit

The problem that arises is: for given v(t), vc (0) = v0 , calculate vc (t).

The voltage across the resistor and capacitor are as follows:

$$v_r = Ri$$
$$v_c = \frac{1}{C}\int i(t)\,dt \quad .$$

From the 2th Kirchhoff law, we get

$$Ri = RC\frac{dv_c}{dt} = v - v_c$$

so the IVP is

$$\frac{dv_c}{dt} = -\frac{1}{RC}v_c + \frac{1}{RC}v$$
$$v_c(0) = v_0$$

and the solution can be computed again with AnyLogic using SD, in the same way as the previous example but in this case the unknown function is *v*c (t) and

$$a = -\frac{1}{RC}, \quad b = \frac{1}{RC}, \quad x(0) = v_c(0)$$

The Forrester diagram is also the same as in the previous examples.

2.3 Second order system

A second order differential equation is an equation involving the unknown function *x*(t), its first and second derivatives *x'*(t) and *x''*(t), and, for control differential equations, the given control function *u*(t). We will consider the Initial Value Problem

$$x''(t) = a_1 x(t) + a_0 x'(t) + bu(t)$$
$$x(0) = x_0, x(0) = v_0$$

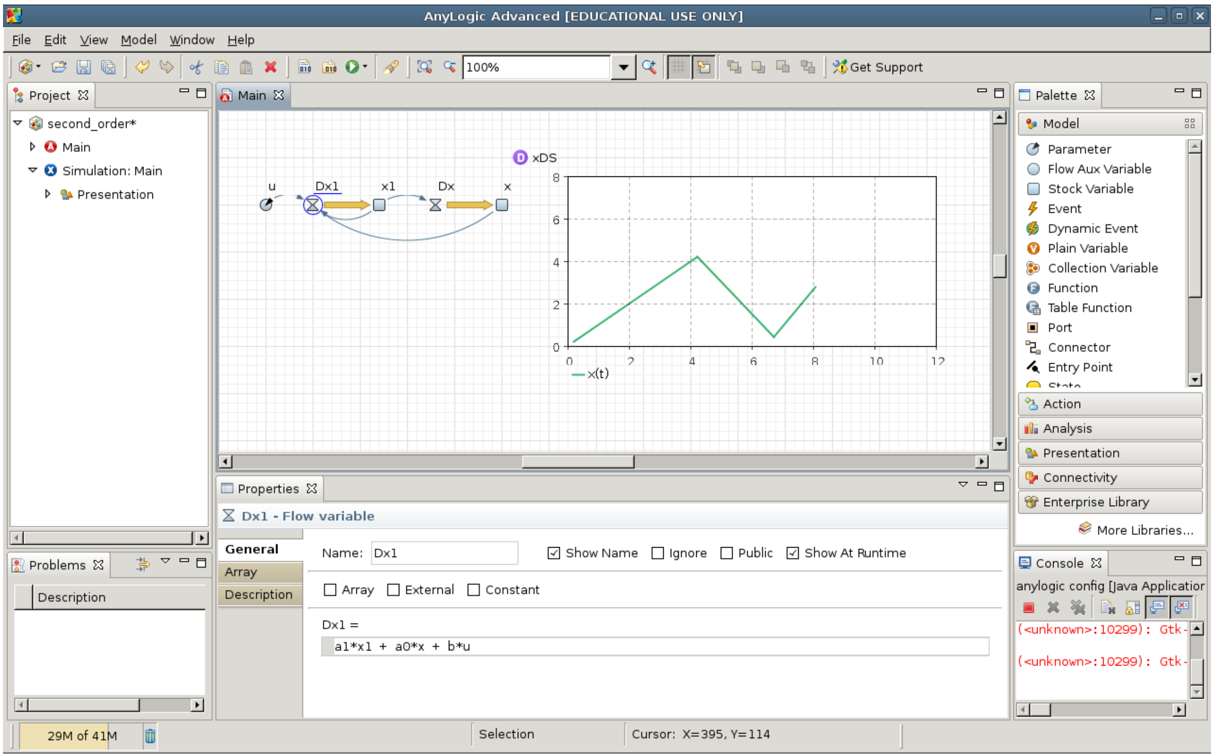

Figure 6: Second order system modelled in Anylogic

As it is know, any *n* order explicit differential equation, with some single variable changes, can be translated to a system of *n* differential equations of first order. So in this second order system, making the changes

$$x := x_1, x := x_2$$

we will obtain the following IVP with two first order differential equations:

$$x'_1 = x_2$$
$$x'_2 = a_1 x_2(t) + a_0 x_1(t) + bu(t)$$
$$x(0) = x_0, \quad x(0) = v_0$$

It very easy to model this IVP in AnyLogic, in similar way as the previous first order examples, as it is shown in Figure 6.

2.3.1 Mechanical system

Second order ODEs appears in many electromechanical systems. As an example, the movement of an object through a viscose fluid tied to a spring will be discussed. The object of mass m is moving through a fluid of viscose dumping *b* and tied to a spring which is fixed on one side and has an elasticity *k*. The mass is pushed with the force *f*(*t*).

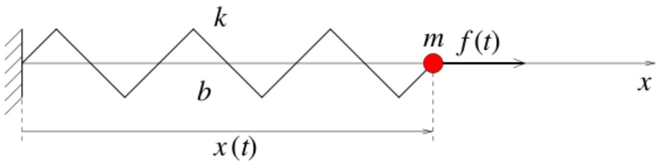

Figure 7: Spring-mass system in viscose fluid

Applying Newton's second law gives

$$m\frac{d^2x}{dt}=-b\frac{dx}{dt}-kx(t)+f(t)$$

Now, with the changes $x := x$ (no change), $x' := v$, $f = u$, we obtain:

$$x'=v$$
$$v'=-\frac{k}{m}x(t)-\frac{b}{m}v(t)+\frac{1}{m}u(t)$$
$$x(0)=x_{0,}\quad v(0)=v_0$$

And again it results very easy to model it in AnyLogic, as it is shown in figure 8. It is even possible to reuse some parts of the previous second order example to build this model.

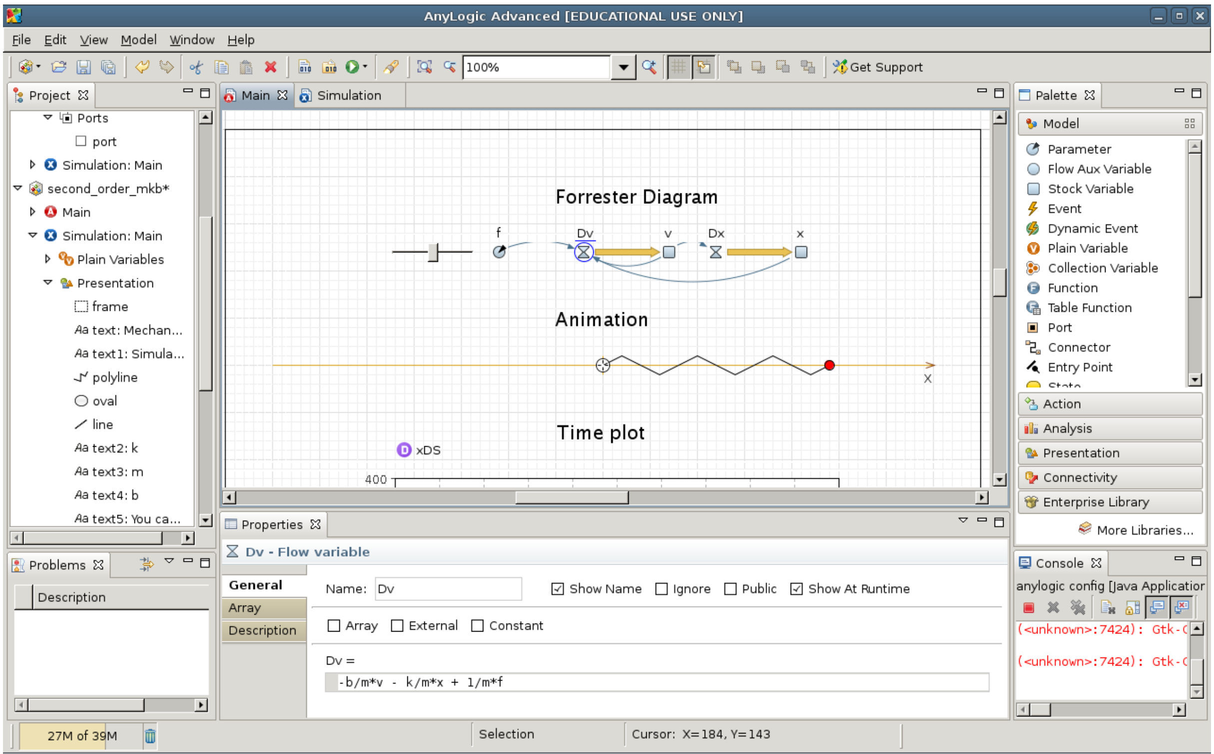

Figure 8: Spring-mass system modeled in Anylogic

3 CONCLUSIONS

According to the authors, simulation can be used to review some basic skills about Mathematics, Physics and Informatics, in subjects that require a high degree of interdisciplinarity.

It is important to choose appropriate exercises in which important concepts such as derivative, integral and differential equation be obvious so that the student, by running simulation, can visualize them.

Also, the results obtained in applying these methods during the last years have passed courses been really good, getting to the acquisition of basic knowledge of the area of Automatics for the students has been satisfactory and quite fast, and noting that the degree of learning of content is much higher than using traditional methods.

Moreover, the degree of students satisfaction has been high, as claimed by its comments made in the forums created in the Moodle platform of our university.

In conclusion, we believe that it has achieved its major goal: to obtain a more effective learning of these basic content. We emphasize the didactic importance of the learning simulation in continuous systems and event systems discrete.

ACKNOWLEDGMENT

This work was made possible through cooperation between the University College of Engineering in Vitoria-Gasteiz and EIFER in Karlsruhe. The authors are very grateful to the Basque Government for the support of this work through the project S-PE09UN12 and to the UPV/EHU for its support through the project GUI07/08.

REFERENCES

[1] Thomson, J. On an Integrating Machine Having a New Kinematic Principle. Proc. of the Royal Society of London, 24 : 262-5, 1876

[2] Kailath, T, Linear Systems, Prentice Hall, Inc., 1980

[3] Forrester, J.W., The Beginning of System Dynamics, Banquet Talk at the international meeting of the System Dynamics Society, Stuttgart, Germany, 1989

[4] Gordillo, F., Aracil, J., Dinámica de Sistemas, Alianza Editorial, 1998.

[5] Borshchev, A., Filippov, A. From System Dynamics and Discrete Event to Practical AgentBased Modeling: Reasons, Techniques, Tools. The 22nd Int. Conf. of System Dynamics Society, Oxford, England, 2004.