

# Simusol: Simulating thermal systems using Sceptre and Dia

Dolores Alía de Saravia  
loli@unsa.edu.ar  
Facultad de Ciencias Exactas  
Universidad Nacional de Salta  
Buenos Aires 177, 4400 Salta, Argentina

Luis R. Saravia  
saravia@unsa.edu.ar  
INENCO (Instituto U.N.Sa – Conicet)  
Buenos Aires 177, 4400 Salta, Argentina

Diego Saravia  
dsa@unsa.edu.ar  
Departamento de Física  
Facultad de Ciencias Exactas  
INENCO

**Abstract**—This paper shows how two specific programs Sceptre and Dia have been used to develop Simusol, a tool for simulating thermal systems.

The thermal system must be outlined with Dia which has to be extended with a specially prepared thermal circuit sheet. The resulting XML Dia-file is translated, by Simusol, into several text files, one of them with input data for Sceptre to simulate the system. Using output from Sceptre, Simusol prepares text files with input data for Gnuplot to represent results. So, in some sense, Simusol converts graphics to code to simulate the system.

All the software used, Simusol included, is free; most of it distributed under the GPL license: GNU/Linux, Dia (which works with XML and SVG files), Sceptre, Perl, gfortran, gcc and Gnuplot.

## I. INTRODUCTION

Inenco Institute, *Renewable Energy Research Institute*, is very interested in developing tools to simulate the behavior of thermal systems.

When a system, formally described by some equations, has to be simulated with a computer program, it is represented by a specific data scheme.

Sceptre makes it possible to numerically simulate electric circuits; and thermal systems and many other systems, can be seen as electric circuits.

A diagram describing the system is often a very good visual aid to the person who has to prepare the alphanumeric data.

Nowadays, there is software such as Dia, which gives more than a visual aid: Dia makes it possible to completely automate the translation of the diagram.

## II. SCEPTRE

Sceptre was designed as “a unified system of digital computer programs by which the electrical engineer can communicate with the computer to determine the DC, transient or AC response of electronic circuits”. [3].

One could also say that, when used to calculate transient responses, it is a tool to solve some systems of linear differential equations. Coefficients of the equations do not need to be constant.

Electric circuits diagrams are quite frequently used and universally understood, so it is almost certain that any person, before using Sceptre, will draw a diagram for his or her circuit.

In order to use Sceptre, the person will identify each element with a name and also will identify the points of connection between elements. There are Sceptre rules, which are quite simple, to follow in order to accomplish this identification. Then the input file for Sceptre will have lines describing each element and also the required outputs. For example “R1, 1-2=0.1” means that “the resistor 1 which is connected between nodes 1 and 2 has a value of 0.1 ohms” (see Figure 1). And if the current through that resistor were required there will be, under “OUTPUTS”, a line containing “IR1”; which will mean the current through resistor 1, from node 1 to node 2.

There are several ways of making mistakes when preparing the input file for Sceptre:

- 1) syntax errors,
- 2) one or more elements forgotten,
- 3) wrong connection points,
- 4) wrong values.

Some of these errors may be not detected by Sceptre: it will simply simulate a different circuit.

Sceptre[7] is written in FORTRAN.

Inenco developed a special version of GNU/Linux, Ututo, to help people run Sceptre from a CDROM [9].

## III. SCEPTRE FOR THERMAL SYSTEMS

It is also well known that thermal systems can be thought of as electrical circuits with energy flux as current and temperature differences as voltage differences. In this way, the components of thermal circuits can be interpreted as resistors, capacitors, and current or voltage sources. [6], [5].

Therefore, Sceptre is also useful to calculate the transient response of thermal circuits.

## IV. DIA

“Dia is a vector-based drawing tool similar to Win32 OS Visio ... It is ideal when the diagram has shapes that recur and are connected by lines... Dia is extensible through the use of *sheets* which contain collection of shapes for use in diagrams”, [10], [1].

Dia home page: <http://projects.gnome.org/dia/>. Today, version 0.97 can be downloaded. But version 0.88 and most later versions are also good for Simusol.

## V. DIA AND A SHEET FOR THERMAL CIRCUITS

One of the sheets Dia comes with is the “circuit sheet” which contains many shapes to represent different electronic elements. So, one can draw beautiful electronic diagrams using Dia. But those shapes are not designed to receive an identification and they are no good for Simusol. It is not difficult to redefine the shapes adding an identification for them, and so we did it to create a *Simusol-Electrico sheet*.

But, instead of using electric element shapes for thermal simulations, we decided to use new icons, so the person who has to draw the diagram —using Dia— can forget about the electrical analogy and concentrate on the thermal aspects.

So we designed a *thermal circuit sheet* with several shapes described in SVG format; and one more shape — a C-programmed one— borrowed from the *UML sheet*.



Fig. 1. “Thermal circuit” sheet (“Circuito Termico”)

The three first shapes under the `Circuito termico` title

represent nodes; the nine next shapes represent simple thermal elements; the next one stands for any defined `model`: some combination of connected elements; we use the last one, from the *UML sheet*, to put textual data in.

A diagram representing a thermal circuit can have errors; for example:

- 1) elements not properly connected.
- 2) elements or nodes not properly identified.
- 3) elements without a value or values without an element
- 4) values or functions or tables not properly defined

## VI. SIMUSOL

Simusol consists of several Perl scripts. It translates Dia diagrams into Sceptre input data using the Perl module `XML::Parser`. The translation is possible because of the well designed Dia XML format.

Because the values we have to feed Sceptre with, are not the values of thermal or geometric properties, but the values of the equivalent electric properties, one has to be specially careful to avoid the mentioned fourth type of error (page 1).

For example, the value to assign to a resistor representing the heat conduction through a wall, depends on more than one wall property. More generally, we can distinguish three types of thermal elements which through the electrical analogy became resistors: conducting resistors, convecting resistors, radiative resistors. Even for each one of them, we use different formulas to get their values in ohms, depending, for example, on the involved geometry.

Simusol is started from a terminal by, for example, `simusol tanque` if `tanque.dia` or `tanque` is the diagram file. If everything is OK, it shows the graphs representing the results of simulation. It also produces some other interesting files.

It is easier for people to visually verify correct connection lines between elements than to tell Sceptre about them. So, it is good to have Simusol to interpret the diagram.

Simusol detects some errors in diagrams (page 2) and informs the user about them. It also interprets message errors given by Sceptre related to simulation.

Simusol is being used by several investigators, among them [8], [4]; and some students who have helped to detect programming errors and to improve its characteristics. Our thanks to all of them.

At the moment, we continue working to improve the facilities to communicate with Simusol and to take advantage of some features of Sceptre. For Dia, we would like to be able to program “buttons” in Dia’s canvas in order to be able to start Simusol without using a terminal.

## REFERENCES

- [1] Dia Project.  
<http://live.gnome.org/Dia>.
- [2] Subversion repository of this article sources, 2011.  
<https://www.inenco.net/repos/loli/jfsfk>.
- [3] David Becker and Wolf-Rainer Novender. *Extended SCEPTRE, User Manual*.  
<http://www.iem.thm.de/iem/84.html>, 1999.

```

CIRCUIT DESCRIPTION
Archivo producido por S I M U S O L:
simusol1.pl
para S C E P T R E
tanque.d, a partir del archivo-diagrama tanque
ELEMENTS
C1, 1-0 = 41860.
E1, 0-2 = 5
J1, 0-1 = T1 (TIME)
R1, 1-2 = 0.1
DEFINED PARAMETERS
PA1 = T1 (TIME)
OUTPUTS
VC1, IR1, J1
INITIAL CONDITIONS
VC1 = 18
FUNCTIONS
TABLE 1
0, 500
14400, 500
14400, 0
21600, 0
RUN CONTROLS
X PLOT DIMENSION = 0
START TIME = 0
STOP TIME = 21600
INTEGRATION ROUTINE = TRAP
END

```

Fig. 3. Part of the input file for Sceptre; it corresponds to fig. 2

- [4] C. Cadena and L. Saravia et al. *Análisis del comportamientoermico de absorbedores de aluminio empleados en cocinas solares con concentración*. AVERNA, ISSN 0329-5184, No. 5, AR107-08, 2001.
- [5] John A. Duffie and William A. Beckman. *Solar Energy Thermal Processes*. John Wiley and Sons, 1974.
- [6] Jan F. Kreider and Frank Kreith. *Solar Energy Handbook*. McGraw-Hill, 1981.
- [7] W. Novender. Sceptre: Simulation of Nonlinear Electrical Circuits. *Linux Journal*, (63), July 1999. <http://www.linuxjournal.com/article/3008>.
- [8] M. Quiroga, L. Saravia, and C. Cadena. *Aplicación del programa Sceptre para la simulación de una cocina solar tipo caja*. AVERNA No 5, AR107-08, 2001.
- [9] D. Saravia and L. Saravia. *Ututo GNU/Linux, SCEPTRE*. Inenco, UNSa, ISBN 987-9381-06-8, 2000. <http://www.ututo.org>.
- [10] Kayon Toga. *The GNOME Desktop*. 2001.

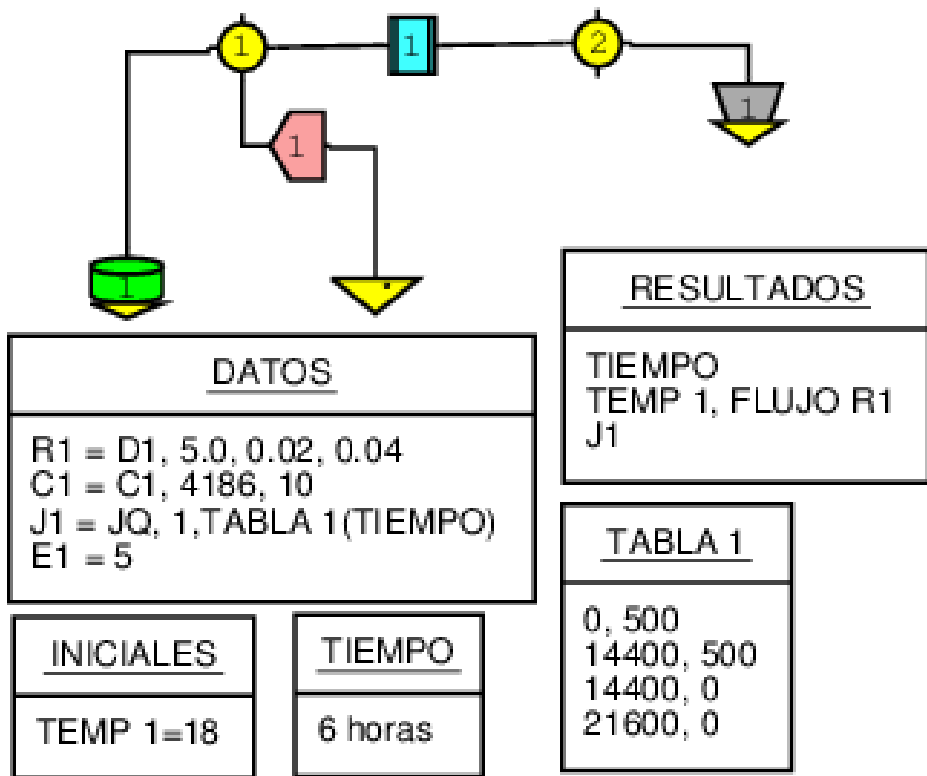


Fig. 2. A thermal circuit, drawn with Dia

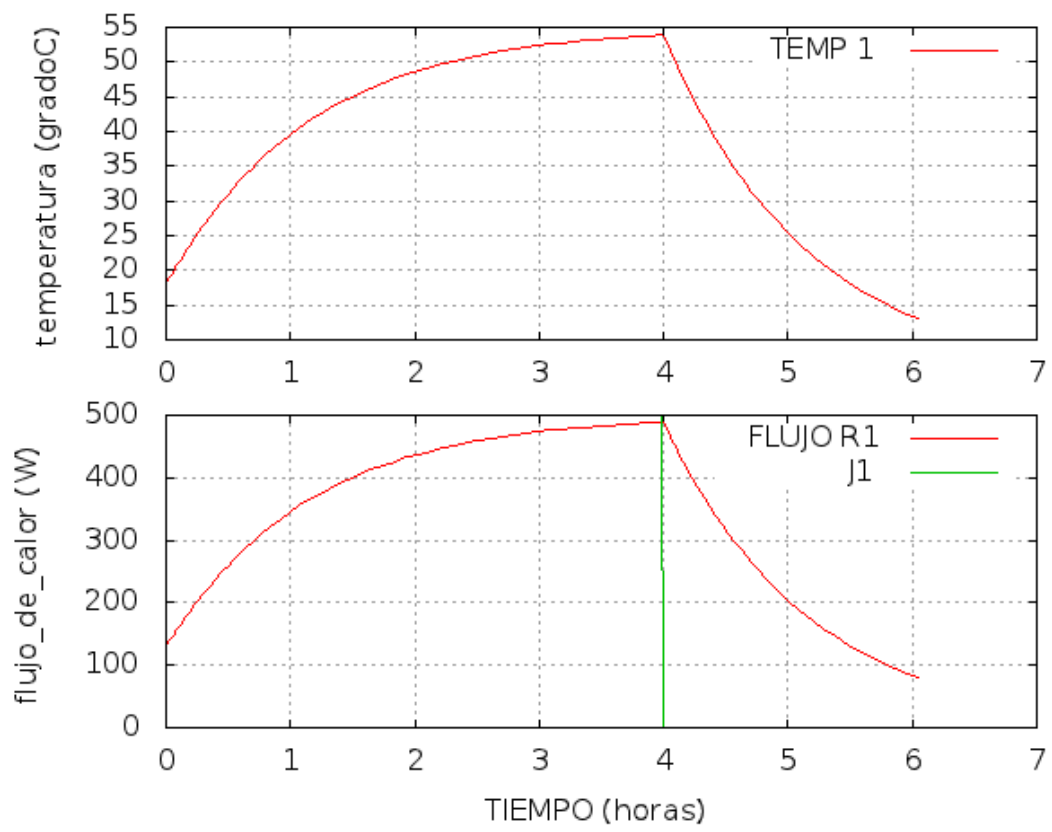


Fig. 4. Graph with results for diagram of fig. 2