A Configurable Interface Generator for Simulated Modules in Sugar Industry

Miguel A. García [*], Diego R. Llanos[**], César de Prada[*]
[*]: Department of Systems Engineering and Automatic Control &
Sugar Technology Center (CTA)
[**]: Department of Computer Science
University of Valladolid. Spain
e−mail: miguel@autom.uva.es

Introduction

Simulation languages have known a continuous improvement and development since the arrival of the CSSL'67 standard and the identification of their field of application as different from that for the general purpose languages. Amongst the characteristics introduced can be outlined a model description closer to the mathematical format and the detachment between model and experiments, that is, the answer to how to describe the model and how to perform the experiments on the simulation [1]. Related to this second question, a command language becomes an integral part of the simulation language, allowing changes in the model variables, plotting, etc, without the need of recompiling the simulation program and preserving the structure of the system.

Hence and as a metric, one of the aspects that defines the quality of a simulation language is the key point of user interaction. Ideally, the separation between model and experiments should be total, in such a way that once the simulation is finished, its use in different experiments made by different users could be carried out as easy as possible, taking into account the differences between users because of their aims, level of expertise, etc. Going deeper in this separation and taking into account the inherent condition to the experiments as a user interface with the models, it makes sense to try to improve the use of preconfigured sets of experiments for a sort of end users foreign to simulation details.

ACSL [2] is a well established simulation language and have thousands of users all over the world. The way in which ACSL handles the separation between models and experiments is by using different files and syntax for each one. The models are implemented in a Fortran−based language and compiled separately, while the experiments are performed typing commands that involve the names of the variables and parameters of the model, as well as procedures composed of groups of predefined commands with a given syntax. This kind of performing experiments using procedures and commands requires a certain degree of expertise and knowledge of the implementation of the simulation model, which can be a real barrier for certain classes of users.

So, a tool for generating graphical interfaces for general purpose ACSL simulation models can be very useful. The tool is aimed to construct simulators for education and training in fields such as process control. Final users can manipulate the simulation from an upper level, avoiding the details of simulation, names of variables and experiments, and focusing the attention in the work with each particular simulated system. The tool is based on graphical operation and transparent communication with the supporting simulation. The steps that must be followed by the programmer to prepare the graphical interface for users of the simulation are outlined. Some application examples interesting for the personnel of sugar factories as end users are shown in order to suggest the power of this tool for the comprehension of the principles of the most common elements in this area.

Users of simulation modules

Sugar industry operators are a good example of the needs and requirements for the users of simulations. In the context of a sugar factory, there are many components and elements working together, usually with interactions, constraints and multiple objectives. Many times, the foundations and physical principles of single elements are not available in an isolated way and remain hide for the operators. This fact drives to a misunderstanding of the deep aspects of the process and consequently, to a bad interaction with it. So, a simulation tool being able to present every element appart from the rest of them, allowing an easy interaction with it and providing a set of selected experiments that make evident the causes and the effects of the actions over the element can be a useful complementary tool. Of course such a tool must hide the syntax and the technical aspects of the simulation, and present an end−user oriented interface, in such a way that the interaction, change of the value of variables and the execution of experiments can be made in a high level and intuitive way. Besides, the tool must be easily configured for different simulated elements with preconfigured experiments, ready to be used in standard training sessions.

The requirements of an interface suppose the adaptation to the needs of specific users and aims, and this is particularly truth when the aim of the simulation is to train people, or when the users are non−experts in simulation details. For this kind of users, a tool for developing graphical interfaces of general purpose simulation models and a way to perform experiments can be considered the best choices.

The current implementation is made in C language for Windows environments and communicates with ACSL for Windows by using DDE [3], but with the same policy and different communication methods, versions for another operating systems could be developed. The work with the experiments can be carried out graphically, using controls such as scrollbars, buttons, bitmaps, etc, belonging to an interface designed with a complementary application of the one that communicates with the ACSL command tool in order to load the commands remotely.

The capability of dealing with the models transparently with regard to the background simulation enhances its power and accessibility by users without the need of knowing the simulation details. The advantage of this approach is that the attention can be focused in the experiments and the evolution and operation of the simulated system, which is their proper object of interest, reducing the simulation to its condition of supporting tool.

Of course, this advantage requires an extra−work for the programmer of the simulation. Simulation programming will consist not only of implementing the algorithms of the model and preparing the battery of experiments for using this model, but also of designing a graphical user interface that corresponds both with the variables in the model and with the commands in the experiments. This last task must be achieved by the programmer of the simulation himself, who is the person with the necessary deep knowledge to prepare a user interface for users not familiarised with the simulation aspects.

Application structure

Hence, the interface generator consists mainly of two applications: PConfig, that allows the programmer to design an user interface, and SimuModu, that allows the user to run the simulation through the graphical interface, avoiding to know the details of the simulation language. The link between them is the configuration file obtained as result of the use of PConfig, which is recovered by SimuModu in order to get the configuration information and to execute ACSL with it. Fig.1 shows the group of different applications, files and communications between them and with the simulation project.

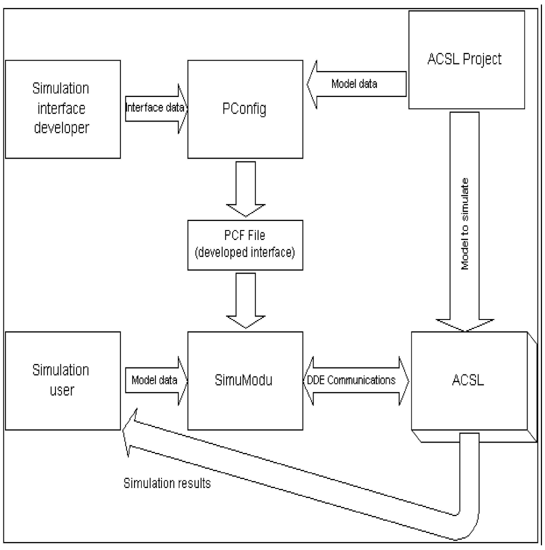


Fig.1. Block diagram of the set of applications and files

Fig. 2 and Fig. 3 show the application PConfig used for the configuration of a new interface, and a worksession with SimuModu executing the simulation and using the configured simulation. SimuModu uses not only the configuration information of the file genereted by Pconfig, but also the information about variables and experiments in the simulation, in order to offer them to the user. With this information, the user can select the experiments, modify the value of the variables on the simulation from the dedicated interface and even use the graphical tools of ACSL representations, in the same way all this operations could be made from command line of ACSL.

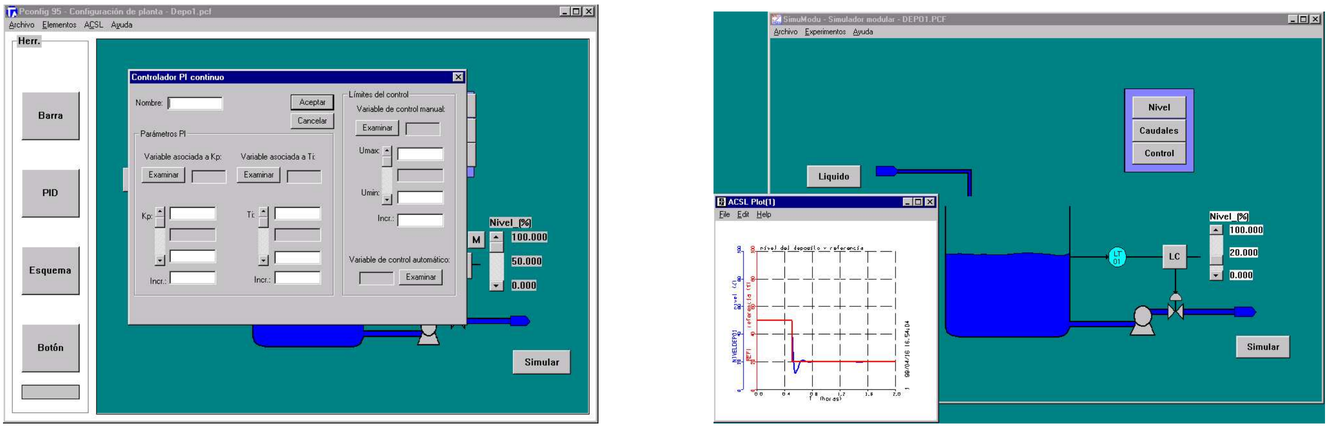


Fig.2 y Fig.3. Configuring an interface with PConfig and executing a simulation with the generated interface using SimuModu

Although Pconfig y SimuModu are introduced through the description of the successive steps of the heat exchanger example, many other modules of another typical elements in the sugar factories, such as diffusors or crystallizators, are already available.

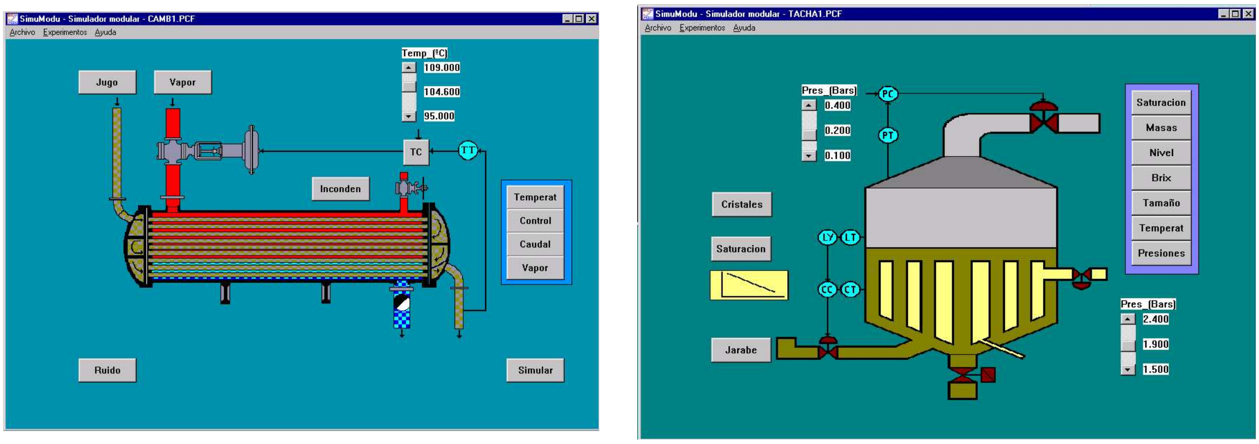


Fig. 4a y Fig. 4b. SimuModu configured for different elements: A heat exchanger and a vacuum crystallizator

One of the most important characteristics of the proposed tool is the easy production and interchange of modules. The simulations programmers can design as an appendix of their work the interface for every simulation by using PConfig, and the personnel of the sugar factories can have a set of interfaces and their corresponding simulations in order to load and save them in the same or different sessions of training with SimuModu.

Abstract

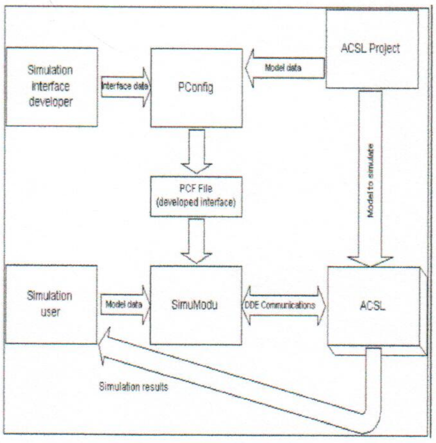
This paper present an integrated software tool for designing and executing modular simulations of common elements in sugar industry environments. The developed system allows to avoid the knowledge of the details of the simulation and the use of specific commands. The implementation has been made with regard to ACSL (Continuous Advanced Simulation Language), although the ideas behind are general and the tool could be adapted to other environments.

References

[1] F.E. Cellier, Continuous Systems Modelling. Berlin, Springer Verlag, 1991
[2] ACSL, Reference Manual v.11, MGA Software, 1995
[3] C. Petzold, Programación en Windows, Madrid, Ed. Anaya, 1992.

A Configurable nterface Generator for Simulated Modules in Sugar Industry

Miguel A. García [*], Diego R. Llanos [**], César de Prada [*]
[*]: Department of Systems Engineering and Automatic Control &
Sugar Technology Center (CTA)
[**]: Department of Computer Science
University of Valladolid. Spain



Objectives:

- To train with isolated modules in order to understand their principles
- To hide the syntax of the simulation language
- To present an end-user oriented interface

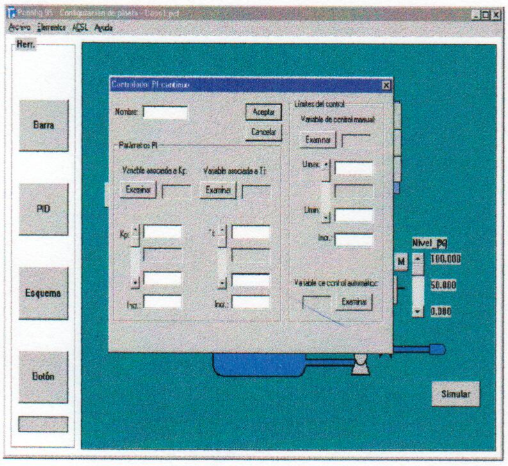
Features:

- Windows environments (NT, 98)
- Simulation language: ACSL (Advanced Continuous Simulation Language)
- Portability of modules and interfaces

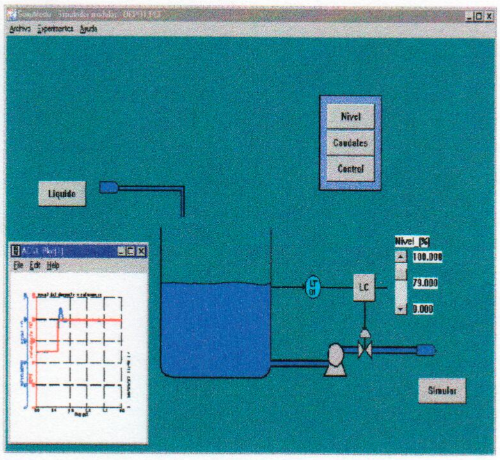
Tools:

- PConfig:Design and development of interfaces, access to variables and experiments
- SimuModu:Execution of simulation, communication with ACSL
- Configuration file: Generated by PConfig, read by SimuModu

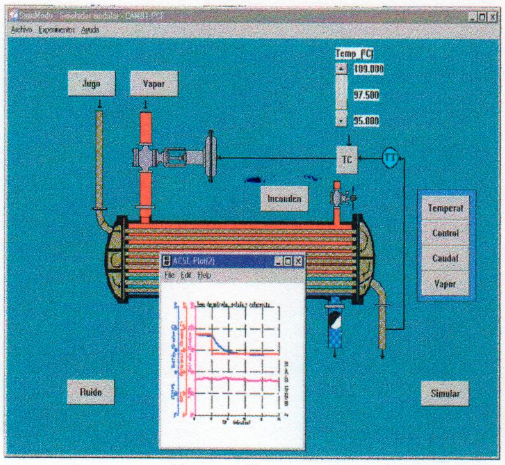
Programmers of simulations design with PConfig as a final step of their work the interface to use them.
Users of simulations interact with them directly from the interface provided by SimuModu and avoiding the details of the simulation language.



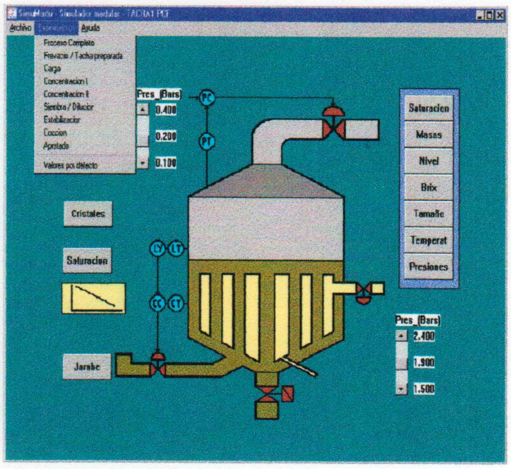
Designing the interface with PConfig for a tank with level control



Executing the module with SimuModu: ACSL plot of the result after experiment



Module execution with SimuModu: Heat Exchanger



Module execution with SimuModu: Vacuum Crystalizator with set of experiments