

AUTOMATED GENERATION OF ECOSIMPRO CODE FOR PROCESS PLANTS USING SIMPD

Luis Felipe Acebes (felipe@cta.uva.es)
Department of System Engineering and Automatic Control.
University of Valladolid. Spain.

KEYWORDS

Industrial processes. Model design. Dynamic modelling. Program generators. Computer Aided Engineering.

ABSTRACT

In this paper a new approach to obtain dynamic models (differential equations) of a complex system will be shown. This approach tries to emulate the way of reasoning of an expert modeller and, starting from AI and graphic modelling techniques, faces up to the model reusing task. Besides, from the user point of view, this approach makes easier the modelling task, avoiding many of the problems that the symbolic manipulation of equations is taking with it.

The paper is organised in this way: First, in the introduction we present our background and the motivations of this work. Second, the core of the ideas of this modelling methodology will be discussed. Third, we will show the implementation of these ideas. Finally, some conclusions are given.

1. INTRODUCTION

Modelling and Simulation of systems are basic tasks in several activities related with the system engineering, such as: the design of control systems, training simulators, process optimisation,...

In our case, we are developing a training simulator for operators of beet sugar factories. This simulator is formed by three nodes: (a) a dynamic simulation of the process; (b) some operator consoles and (c) a trainer console. Obviously, each factory is different, then, the dynamic model must be different, and, due to the human and material cost of a mathematical model of a plant, we needed some tool to automate this task.

Although, the simulation techniques are well established, it is not true for the modelling ones. However, there are a set of techniques and tools that deal with model reusing, some of them are:

1. The block oriented languages (simulation environments with graphical user interfaces: SIMULINK, GRAPHIC MODEL, MMS,...).
2. The bond-graphs techniques.

3. The object oriented modelling languages that use the symbolic manipulation of equations (DYMOLA, OMOLA).
4. The graphic modelling languages, where the models depend on the context (PMT).
5. The use of expert systems.

As far as we are concerned, each modelling tool, that we know, has some drawbacks for the final user that obstruct its setting-up in the industry. Then, we planned to develop a modelling tool of easy use for people that, without knowledge about modelling and simulation, know the process that they want simulate. This tool is based on a new modelling approach that will be presented subsequently.

2. THE PROPOSED MODELLING APPROACH

The key idea: emulating the way of reasoning of an expert

The main idea is that the methods of the nowadays-modelling tools are different to the ones of an expert modeller. When a modeller develops a simulation model, he does not think about the system as a set of equations that must be manipulated in order to solve some variables. Modelling is not only a task of juxtaposition of equations and its subsequent symbolic manipulation. Some works of AI experts have shown this difference.

The modelling methodology can be resumed in three steps:

1. System conceptualisation: to describe physically the system and the aims of the model. This description is triple: topological (subsystem interactions and boundary conditions), structural and phenomenological. The analysis of the aims of the model drives us to the variables of interest.
2. Model formalisation: to determine the applicable laws to the relevant phenomenon that are joined to the variables of interest, to obtain the corresponding equations and to solve their causality.
3. To give values to the model parameters.

Modelling principles

The majority of the production process are formed by hundreds of constituent units (tanks, heat exchangers, mixers, evaporators,...). Then, it looks like logical that a modelling tool for people that are familiar with the production process must be based on the process units as the minimum modelling units. This idea allows to develop a graphical user interface in which the relationship between the model scheme and the process scheme is total.

In many cases the dynamics of the processes in the basic units are not independent of the whole system. In the field of the process industry, and in other domains too, it is possible to model the dynamics of each unit in an independent way if the interactions are modelled in a global way.

Finally, the physical laws are general (the same law is applied to make a mass balance in a tank or in an evaporator). Then, we use a general description of the physical laws, whose results depends on the physical descriptions and the interactions with the environment of the units in which are applied.

To resume, the modelling principles of this approach are:

1. The basic modelling units do not have a mathematical model that is associated in a single-minded way, but they have a physical description.
2. The interactions between process units are modelled in a global way.
3. The description of the physical laws is unique, general and not included in any particular system.

Modelling units

We define modelling unit to each functional unit that is described physically. During the model generation, to each unit will be assigned a set of equations and parameters in function of its physical description and the aims of the global model.

The physical description of each unit is made in two levels:

1. Physical and topological level: (a) interactions with the environment, using the real physical connectors; (b) internal topology: (subelements, internal paths,...); and (c) physical characteristics.
2. Phenomenological level. In this approach this description is essential, because, when the

user selects the level of detail of the model, he is selecting, indirectly, the phenomenon that are happening. For instance: in an closed tank containing a water solution it is possible to model the vapour accumulation or not, then the model will be different.

In our modelling approach, when the user selects each elemental units, he must respond to a set of questions to describe it physically.

For instance, for a tank the user responds to:

- The number of inputs and outputs (pipes), and their location.
- Is the tank closed?
- Do you consider the density as a constant?
- ...

Interactions

From the definition of modelling unit, these share physical properties through their connectors. In the case of the process industry the main interaction are the material flows and these flows transport properties such as temperatures, concentrations.... These flows are result of a cause-effect relations, for instance, when two pressure supplies are connected a material flow is produced. It is applicable in other domains, for instance: in the electric one the main interaction is the current.

We consider line of flow to the entity formed by the elements that connect two process units in which accumulation or generation of material is produced. A similar concept is used in a modelling tool called PMT, *Process Modelling Tool*.

Physical laws

The main idea of this approach is that there are not mathematical models (equations) linked to the elemental units that form the whole system. In place of them, there exists a set of rules to build the global model of the plant, which depends on the context in which each elemental unit is working.

This rules and procedures are based on a set of physical laws: first principles, equilibrium and transport equations,

To model the behaviour of a magnitude of a process unit exists a unique rule to generate the equation that models the dynamic of it. Each rule is linked to one magnitude and the result of its application depends on the physical characteristics and the interactions of the process unit in which is applied. The use of this rules produces a causal equations, a set of new variables of interest (this variables are included in the

generated equation) and a set of parameters and initial conditions that are needed to solve this equation when the model is simulated.

For instance, the rule to model the accumulation of mass of liquid has this structure:

1. To detect the mass flows of liquid and vapour that input and output (W_i , W_o) in the receipt and they have an influence on the mass of liquid.

$$ec = SW_i - SW_o$$

2. If phenomenon of evaporation and condensation exist, then the equation must contain the evaporation and condensation flows (W_e , W_c).

$$ec = ec + W_c - W_e$$

3. If the receipt is open, the equation must consider the running over flow (W_r).

$$ec = ec - W_r$$

4. The equation of the mass is the integral of the sum of flowrates, with $mass_i$ the initial value of the mass of liquid. Also, the mass of liquid is pointed out as solved variable.

$$mass' = ec$$

5. Each of the flows is marked as variables of interest. Besides, the expressions of the flows has a set of activation conditions, that imply a set of parameters and variables of interest, for instance, the expression of the overflowing flow contains a parameter (the overflowing high) and a variable of interest (the high of liquid).

These rules are general and they are not integral part of any elemental unit. This description has several advantages:

1. Each rule is applicable in different units.
2. As each magnitude is modelled in an independent way. This set of rules avoids to have generic models of elemental units, containing a finite number of equations. This models would be closed and the inclusion or exclusion of assumptions would drive us to a multiply of models.
3. The symbolic manipulation of equations is not required; the equations are generated having solved the causality. Besides, the set of required parameters and i.c. is the minimum one.
4. A library of rules can be developed. Then, this library avoids that the number of the

description of each unit will be multiply by the sum of each possible description of each magnitude of this unit. With this methodology, it is possible to model a number of units using the physical principles that they share, and to avoid an exponential explosion of the number of models of elemental units.

The generation of the simulation code

The global scheme (Figure 1) of this modelling approach is divided into two parts, one is made by the user and the other is made by the modelling tool. This scheme will be detailed as follows.

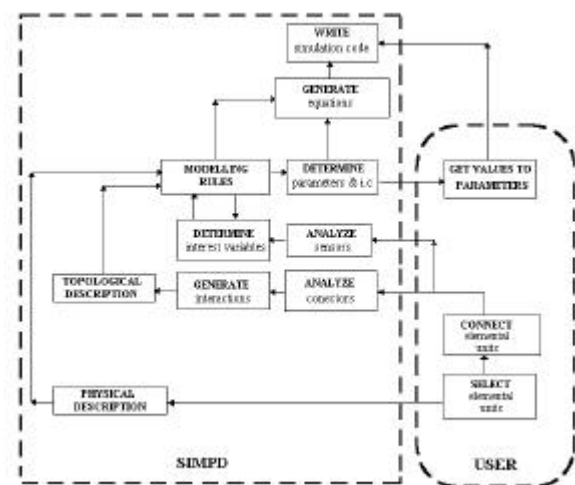


Figure 1

The user builds a schematic representation of the system

The user selects a set of elemental units specifying, to each unit, its physical characteristics and some assumptions about its performance.

When the elemental units has been selected, they must be connected using real connectors (pipes, wires,...), then a plant scheme with process, control and measurement units is built.

The modelling tool analyses the plant scheme

The modelling tool reasons over the plant scheme and it determines the interactions and the variables of interest (the user selects the variables of interest using sensors in the plant scheme).

The information about the interactions plus the physical information, that the elemental units contain inside, constitute the physical, phenomenological and topological description of the system.

The modelling tool generates the simulation model

Starting from the initial set of variables of interest and using the set of rules to build the mathematical models and the physical description of the system, the modelling tool generates the simulation model of the system.

If the set of rules of generation of models complies with a set of conditions, the algorithm of generation of mathematical models works well (these conditions escapes from the aim of this paper, but they can be found in Acebes 1996).

The user gives values to the model parameters

The generated simulation model contains a set of equations, parameters and i.c. To simulate this model some values must be given. Due to the physical diversity of the process units this task can not be automated and it requires the action of the user.

3. IMPLEMENTATION

Given that, this work is being developed in the C.T.A (Institute of Sugar Technology in Spain), to be used by process engineers, the proposed approach has been implemented in a specific purpose-modelling tool called SIMPD.

The aims of SIMPD are: (1) to generate code for a training simulator of plant operators; (2) to design control structures and study changes in the strategy of the production process.

SIMPD has been implemented using the shell to develop expert systems G2®. This shell is suitable to program this application, because, apart from to store the knowledge of an expert, it supplies an easy-use environment to make a GUI and it allows the communication with external applications.

SIMPD is organised in a modular way:

1. A library of elemental units: a set of classes of process and control units, whose instances allow the physical description of the equipment that they symbolise.
2. A model scheme: a graphical description of the whole system. This scheme (flow-sheet) is like a process scheme and it uses the schematic representations of each elemental unit and the physical connections between them.
3. A scheme analyser that extracts the relevant information that the model scheme contains (variables of interest, interactions,...).

4. A generator of the mathematical model of the whole system, it is formed by the set of rules that contains the expert knowledge.
5. A generator of simulation code, for a specific simulation language: ACSL® .
6. A communication module with ACSL®, that allows to use the generated simulation code using the same scheme that has been used to generate the model and give values to the model parameters.

To model a system with SIMPD the user must complete the next steps:

- From the graphical user interface, the user selects the elemental units (and their characteristics) those compose the whole system. The graphical representation of these units is located in the plant scheme (Figure 2). Later, the user connects them, in a real way, and he builds the process and the control structure (Figure 3).

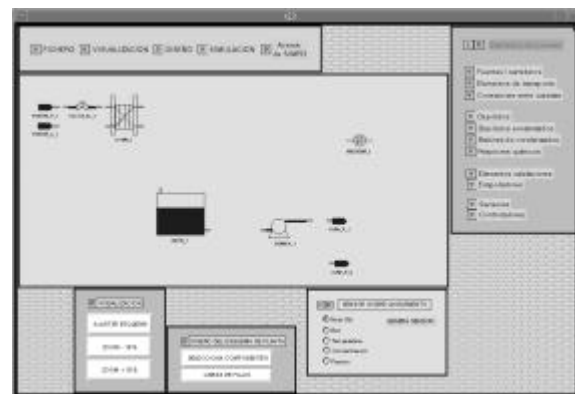


Figure 2. Selecting elemental units.

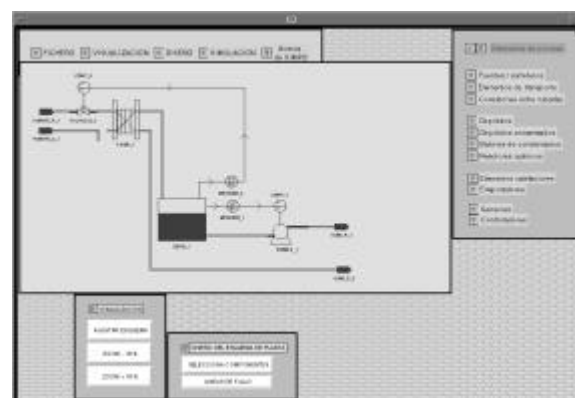


Figure 3. Connecting elemental units.

- The user invokes the scheme analyser, and if it is necessary, it requests information about the system (flow directions,...). If the scheme is well built, SIMPD generates the simulation model.

Then, SIMPD requests the values of the parameters and i.c that are necessary to simulate the system (Figure 4). SIMPD writes the simulation model according to the ACSL® or Ecosim Language syntax in an ASCII file.

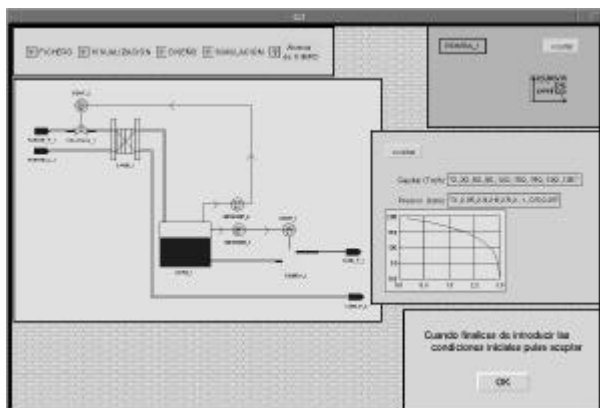


Figure 4. Giving values to the model parameters.

- Finally, if the user selects ACSL as simulation language and SIMPD is running over a Sun workstation he has two possible options. First, if he is an expert user of ACSL®, he can compile the model and use the command language of ACSL®. On the contrary, if the user doesn't know ACSL®, SIMPD allows him, intuitively, to simulate the system from the plant scheme (using the module that communicates ACSL® and SIMPD). In this case, SIMPD allows to the user to compile the model, start and stop the simulation, change parameters and plot variables.

However, if the user selects EL as simulation language, he must compile the generated model using the GUI of EcosimPro®.

5. GENERATION OF EL MODELS

Initially SIMPD was developed to generate ACSL models and in a subsequent step it was modified to generate EL models. As ACSL doesn't allow the object oriented modelling we don't use this modelling capability of EcosimPro®. So, the EL models generated by SIMPD are monolithic COMPONENTS.

Adding to SIMPD the capability of generate EL models implied to include a new module that using the parameters, initial condition and equations of each instance of the P&I scheme generates EL code.

Having into account the differences between the statements of ACSL® and EL (for instance: *integ* and *'*) and comparing the ACSL® and EL structures (Figure 5), we can deduced that this task is not conceptually complicated but it is required time and

programmatic skills of G2, ACSL and EcosimPro (Redondo & Sanz, 2002).

program name	COMPONENT name
...	
initial	PORTS
...	We don't use in this case
end	DATA
dynamic	Some constants of ACSL
derivative	DECLS
...	It is not necessary in ACSL
end	TOPOLOGY
discrete	We don't use in this case
...	INIT
end	Initial conditions of ACSL
...	DISCRETE
end	ACSL discrete statements
terminal	CONTINUOUS
...	ACSL derivative
end	statements
end	END COMPONENT

Figure 5

4. CONCLUSIONS

From the user's point of view, this modelling approach has several advantages: (a) it makes easier the conceptualisation of the system; (b) it formalises automatically the mathematical model and avoids the problems that are linked to the symbolic manipulation of equations; (c) the user gives values to the model parameters intuitively, through the GUI.

However the weak spot of this approach concerns to the developer of the modelling rules. To make a set of these rules together with the library of elemental units is a similar task to develop a model library in an equation oriented general purpose modelling language. Then a possible further research is to develop a tool that makes easier to write the modelling rules.

Nowadays, we haven't compared the accurate and speed of execution of ACSL and EcosimPro models but we have in mind to make it in the near future.

REFERENCES

Acebes, L.F. 1996. *SIMPD: Sistema Inteligente de Modelado de Procesos Dinámicos*. Universidad de Valladolid, (Ph. D.).

Redondo, I., Sanz J. 2002. *Generación de código EcosimPro para SIMPD*. Universidad de Valladolid (Project Study).