GRAFCET simulation using *Easy Java Simulations*

José Luis Gayo Vélez  
Escuela de Ingenierías Industrial, Informática y Aeronaútica  
Universidad de León  
jgayov00@estudiantes.unileon.es

**Abstract**—This paper deals with the simulation of a simple GRAFCET diagram that controls a small training industrial plant using Easy Java Simulations, a powerful free software tool that allows a great variety of systems to be modelled and simulated with very little knowledge of the Java programming language, in a short developing time and fully portable.

**Index Terms**—PLC simulation, GRAFCET, IEC 1311 standard, Easy Java Simulations, EJS.

I. INTRODUCTION

A great research has been made in the field of industrial, and not only industrial, simulation. The product of this research is the development of a vast number of simulating tools for a wide variety of environments. These simulating tools offer a number of benefits in terms of training, time-saving and so on. In the field of PLC simulation there is a lot of research that has been made too. As a result a vast quantity of free and commercial simulators are available to the public.

There is, nevertheless, a field in which no great attention has been paid until now. This is the integration of both PLC and industrial simulation into a single frame and that is the goal of the present paper.

The software chosen for that purpose will be Easy Java Simulations (EJS), a potent programming graphical tool that will allow us to perform our system in a fast, easy, intuitive and portable way.

This paper is divided as follows:

The first section will deal with the IEC 1311 standard, whose goal is to normalize the syntax and data of languages used in the programming of PLCs. Then we will continue with a section on the language GRAFCET, which is the language that we will simulate in our system. In the next section a short list of both commercial and free PLC simulators will be provided.

In the second part of this paper we will introduce EJS. A brief overlook will allow us to come to the last section, the simulation of a simple GRAFCET program that control a simple training industrial plant simulated with EJS too.

The paper will end with a section on the conclusions about this work and future developments.

II. THE IEC 1311 STANDARD

In 1979 the International Electrotechnical Commision (IEC) established a standard API for PLC programming. The goal was to respond to the growing complexity of control systems and the diversity of incompatible PLCs.

The result was the IEC 1311 or IEC 61311 standard, a standard divided in 5 parts of which we are only interested in the third one, IEC 1311-3, which deals with the standardization of programming languages. The standard IEC 1311-3 specifies the data types and the syntax of the languages used for that purpose. The defined languages are:

1. Instruction List (IL), a simple assembler-based language with a reduced instruction set close to machine language. It is specially used for short highly optimized programs.

2. Structured Text (ST) is a high level language with syntax similar to Pascal mainly used for programs with long and/or complex calculations.

3. Ladder Diagram (LD) is a graphic electrical-contact language used mainly by people with no
knowledge on programming language. It is the most widely used.

4. Function Block Diagram (FDB), a graphic language based on blocks similar to those used in electronic circuits.

5. Sequential Function Chart (SFC) is a graphical language based on blocks used for sequential, concurrent systems. It is the most widely used for complex systems. GRAFCET is a SFC language.

III. THE GRAFCET LANGUAGE

GRAFCET is a SFC language with very interesting capabilities to perform industrial systems. Its wide use combined with the clarity to overview processes, that make it ideal to detect errors and problems, is the reason why we have chosen to simulate this language. Its main features, programming structures and evolution rules are commented.

GRAFCET may be successfully implemented for a vast number of industrial processes, for a wide range of purposes and with a great number of advantages compared to the rest of the IEC 1311 defined languages.

A program in GRAFCET is a number of its constituting elements (States, Transitions and Arrows) combined in a tree structure representing the succession of all the phases and the stable states the process must go through to perform the goal it was designed for.

There are a number of rules to be respected when implementing our systems and a number of ways of combining elements to achieve different kinds of automatic systems.

IV. LIST OF SIMULATORS

In this section a short list of both commercial and free IEC 1311 compliant PLC simulators are enumerated and briefly described.

V. THE EJS ENVIRONMENT

Easy Java Simulations (EJS) is a graphic tool to create and simulate a wide variety of systems (physical, industrial, etc.) in a easy, fast, portable and user friendly way with a minimum knowledge of the Java programming language. The version used for this project is EJS 4.3.7.

It is divided in three mainly sections: Description, Model and View.

In the Description section we can make a description of our system. This description may be distributed in as many pages as we wish. Each page is indicated by a tab on the upper part of the page.

In the Model section we can describe all the variables that constitute the different elements in our system. It is divided into six different subsections, each of them with a specific purpose and each of which being divided at will into pages indicated by tabs. The four subsections that we will describe are:

- In Variables we define all the variables used in the system. Their name, type and initial values are given with a brief description of its performance.

- Initialization allows us to give an initial value to the variables defined, normally when the value of the variable is determined by an expression.

- In Evolution we define the rules that will describe the performance of our system, normally controlled by differential equations. E can also describe it with our own Java code.

- In Fixed relations we describe the relations among the different variables defined before.

The View section will allow us to perform the user interface of the system. To this purpose a wide variety of graphical elements is provided. This section is composed of two different parts:

The left panel contains all the elements needed to describe the system. It is divided into three menus, each of them subdivided into different submenus each containing its own elements grouped by categories. These menus are:

- Interface, with several submenus containing elements to create user friendly interfaces. It contains windows, graphic panels, menus, buttons, etc.
– 2D drawables, that provides many 2D elements geometrical figures, arrows, groups of elements, etc. and other interesting elements like pumps, valves and so on.

– 3D drawables provides the same elements of the preceding but in 3D.

The left panel contains a window called Tree of elements in which we will place all the elements selected in the previous menus. These elements will be placed forming a hierarchical structure of windows, panels, etc.

VI. SYSTEM SIMULATION

In this section we will describe the major part of this work. The system to be simulated will consist of a small training industrial plant containing four tanks (T1, T2, T3 and T4) with an orifice on the lower side, a pump, a three-way valve, four valves below each tank to empty them at will and throw water to a general deposit. T4 is opened to T2. T3 is opened to T1. T1 and T2 and opened to the deposit. Figure 1 shows the layout of our plant.

The system works as follows:

The system starts by pushing the Start button. The pump is then activated at 50% of its flowing capacity and pumps water to the three-way valve, which distributes only to T2. When the level of T2 reaches 30% of its capacity the pump is opened to 90% and the three-way valve opens its third way to T3. When the level of T2 reaches 90% the pump is deactivated until T2 is at 5% and then the cycle begins again. We can open the valves below the tanks to empty the tanks at will.

In order to achieve our goal we have to build a user friendly interface containing all the necessary controlling elements to perform a correct supervision of our system as well as to retrieve all the information provided by it. This will be done with the help of the View section mentioned earlier. The interface is shown in fig.2.

We also need an interface to show the evolution of our GRAFCET diagram. This will be achieved in a similar way as we have obtained the previous one.

This part is not directly controlled by us, because we can only act on the control of the panel. The GRAFCET diagrams only shows which state and which transition are activated at every moment.

We must say that this interface is not as elegant as we would like it to be. This is because we have used the elements provided in EJS, which are not specific to GRAFCET. This will be discussed in the Conclusions section at the end of this paper.
VII. CONCLUSIONS AND FUTURE RESEARCH

EJS has allowed us to perform our system in a fast, easy, intuitive and portable way with a very user-friendly interface as we had planned, even if the graphic interface was not as elegant as we would have wished.

The next developments will then deal with the elaboration of a graphic library of GRAFCET symbols with the goal of improving the capabilities of EJS to develop GRAFCET diagrams. This will facilitate the development of more complex systems.

The elements included in this new library will fully comply with the IEC 1311 standard and will include a number of methods in Java code internal to the elements themselves that will make programming much easier and simpler for the user. Figure 4 shows some of the new elements to be added to EJS in order to achieve this goal.

There is also the intention that the above-mentioned library will provide some capacities for remote control. To reach this goal it will be necessary to combine this library with the client OPC element which is to be developed. In this way the GRAFCET programmed for a real industrial process will be executed by establishing a link between the actions and the transitions programmed in the GRAFCET and in the OPC server connected to the PLC that will handle the inputs and outputs of the industrial process.

ACKNOWLEDGMENTS

The author of this paper would firmly like to express his gratitude towards the Grupo de Investigación SUPRESS (SUPRESS Research Group), of the Universidad de León, and specially to Sergio García de Prado, author of most of the simulation in section VI, for the support provided in this project as well as for the interest shown for all suggestions, comments and proposals of improvement made by the author.

REFERENCES


