USING SCIENTIFIC VISUALIZATION IN SIMULATIONS
OF REAL-TIME PROCESS-CONTROL SYSTEMS

Tereza G. Kirner (*)
Alan S. Felinto
Federal University of São Carlos, SP, Brazil

Abstract

This paper focuses on the use of scientific visualization as a form of improving simulations of real-time process-control systems. First, basic concepts are discussed. Then, a software tool for graphic and animated simulation of the considered systems is described. The tool incorporates facilities from scientific visualization to make the validation of systems being developed. The main goal of the paper is to contribute to better validation of real-time process-control systems and, consequently, to higher-quality applications.

Keywords: Scientific Visualization; Real-Time Systems; Process-Control Systems; Graphic Simulation.

1. Introduction

Process-control systems comprise a class of real-time systems, usually composed of interdependent and interactive tasks, which must execute under stringent timing-constraints in order to the system produce its intended mission. Classic examples are manufacturing control systems, flight mission control, medical monitoring systems, robotics applications, control of laboratory experiments, etc. ([JAP91], [KYO88], [LEV90]).

The development of better and higher-quality control-process systems depends strongly on the use of effective validation techniques. In this sense, simulation is pointed-out as a powerful technique for the validation of dynamic behavior and results of complex, real-time process-control systems.
Traditionally, simulation of process-control systems generates many tables, graphics, and statistics, which although useful are not easy to understand and manipulate. System users normally need to spend much time with these tables and graphics. Besides, these aggregated data sometimes do not express a reliable view of the system behavior, primarily of critical situations that are possible to occur along the system execution.

This paper focuses on the use of scientific visualization as a form of improving simulations of real-time process-control systems. Scientific visualization allows a correct, complete and understandable reproduction of system behavior and results, by graphic and of animation facilities.

The paper begins in section 2 discussing basic concepts that oriented the work. In section 3, a software tool that was developed for supporting simulations is described. The tool incorporates facilities from scientific visualization to make the validation of systems being created. Finally, in section 4 the conclusions of the work are presented. A simplified System of Mixture of Liquids, which is used in the paper as an example of real-time process-control system, is summarized in Appendix A.

2. Conceptual Foundations

This work, which resulted in the tool described in the next section, was based on concepts of real-time process-control system, scientific visualization, and simulation, derived from the literature. These concepts are discussed below.

2. 1. Real-Time Process-Control Systems

Process-control systems (in short, PCS) are a class of real-time system normally composed of interdependent and interactive tasks, which must execute under stringent timing-constraints in order to the system produce its intended mission. Classic examples are manufacturing control systems, flight mission control, medical monitoring systems, robotics applications, control of laboratory experiments, etc. ([JAF91], [KOV88], [LEV90]).

There are two main components in PCS: environment and computational system. Environment includes the user application and physical activities being supported, plus the interfaces (sensors, actuators, and channels of data traffic). Computational system is composed of hardware and software, which are responsible by monitoring and controlling the system tasks execution. In summary, environment refers to the controlled system and computational system represents the control system.

The relationship between environment and computational system is made by stimulus and responses arriving to and leaving the system, which are manipulated and controlled by events and state variables.

Figure 1 gives a schematic representation of process-control systems.
Process-control is a time-based concept and therefore timing is a fundamental aspect of the development of any process-control system [JAF91]. Thus, basic parameters, related to timing-constraints, were considered in this work, according to discussion provided by Dasarathy [DA85]. These parameters involve values of minimum, maximum, and duration, which define the stimuli, responses, and related events that arrive and leave the system.

2.2. Scientific Visualization

Scientific visualization refers to specification of transformations, theory, or models, aiming to produce visual representations compatible with the situation of the real-world being focused. These visual representations comprise graphic and animated objects that are organized in a manner that permits to show important information on the target problem ([CAR92], [MCM87]).

The computational solution of a problem, by using scientific visualization, is normally composed of: graphic representation of numeric values; and graphic representation of dynamic behavior and different states assumed during the problem functioning [KIM90]. Scientific visualization can also express: results of aggregated data; logical processing of a solution; and dynamic processing of physical activities, monitored by programs or software tools.

The monitoring of the behavior of physical activities is crucial for the development of real-time process-control systems, in which scientific visualization may be very helpful [HEG89]. Let's consider, for example, the representation of a System of Mixture of Liquids. This system may allow to the observer (or user) to view, in a graphic form, the dynamic behavior of flow of liquids during the mixture preparation. An appropriate graphic representation
leads the observer to perceive possible failures in the system execution and therefore make the necessary decisions to eliminate or recover these failures (this example of System of Mixture of Liquids, summarized in Appendix A, is used to illustrate some aspects discussed in the paper).

Scientific visualization is an inter-disciplinary area, comprising themes of computer graphics, image processing, computer vision, and human-computer interaction ([CAR92], [HEG89], [McM87]).

Computer graphics involves the synthesis of pictures and images, obtained through specifications given by the user. It includes creation, storing, and manipulation of models and images of objects, by means of geometry and mathematics resources ([BUR89], [CAR92]).

Image processing focuses on the analysis of data obtained by the digitalization of images from the real world. The transformation of an image to another one that is easier to understand by a certain user is the primary goal of image processing ([BUR89], [CAR92]).

Computer vision intends to emulate the human interpretation through an interpretation produced by computer. This involves automatic analysis, rebuilding, and recognizing of objects (or specific characteristics of objects) that take part of a scenario, by means of one or more images ([CAR92], [HEG89]).

Human-computer interaction allows an adequate implementation of the capture, presentation and use of created representations. These are essential aspects for reaching the best results of scientific visualization (see Shneiderman[SHN93], for a consistent discussion of the subject).

Furthermore, scientific visualization involves extensive use of animation techniques ([PUL93], [WIL90]). According to Pulli[PUL93], animation is an interactive, simulated, and graphically visualized execution of a system model for increasing human understanding of system behavior, function, and/or structure. Typically, animation is carried out in a simulated time-base to ensure that the execution speed is convenient for the human perception of events and their causal relations.

2.3. Simulation of Process-Control Systems

Simulation consists of investigating properties and behavior of a real-world system, through a second system that is constructed as a model of the original, real-world one. Thus, the model is used to make experiments and the results of these experiments are applied in the original system [SCH90].

Simulation environments are extensively used as a very effective resource to validate complex systems (Tanir[TAN94] gives a review of concepts and applicability of simulation environments). Simulation represents a powerful technique for monitoring and validating the behavior and the partial and final expected results of PCS. As these systems involve solution of highly dynamic and complex problems, they often produce large amount of data combinations, during their operation. Traditionally, simulation of PCS
generates many tables, graphics, and statistics, which although useful are not easy to understand and manipulate. System users normally need to spend much time with these tables and graphics. Besides, these aggregated data sometimes do not express a reliable view of the system behavior, primarily of critical situations that are possible to occur along the system execution.

Scientific visualization allows correct, complete and understandable reproductions of system results, by graphic and of animation techniques. With these techniques, simulation of PCS can show all system components, interactions among them, execution of physical activities, occurrence of events and their impact on the system behavior, etc. For example, the simulation of System of Mixture of Liquids can provide information, such as: moving of products that will compose the mixture; characteristics of liquids outlet; measures of volume, heating, pressure, density, etc.; inadequate values and non-predicted states; etc. By scientific visualization, it is possible to provide the necessary information in a quick and easy-to-understand manner. In consequence, the system validation is improved, since it is possible to make previsions of dangerous situations and implement the required decisions.


The goal of the tool next described is to support users and software designers who are involved in the development and validation of PCS. The tool makes graphic and animated simulations, and its construction was driven by concepts discussed in the previous section.

The relevant characteristics of the tool are summarized in the following subsections.

3.1. Architecture of the Tool

The primary components of the tool are: manager module; creator module; graphic and system libraries; and interface with the application. These components are shown in Figure 2.

A) Manager Module

PCS systems have two basic components, that is, environment and computational system. According to this concept, the user application is responsible by the simulation of the computational system, and the tool here described is responsible by the simulation and graphic demonstration of the system environment. In this context, the manager module makes the control of the environment of the target PCS.
In the performed implementation, the control of animation of graphic objects is based on the concept of stimulus and response. The communication between the user application and the environment is implemented by parameters of tasks calls. Thus, the manager module receives calls from the user application and these calls "stimulate" the objects. The objects, after be stimulated, will process the changes in their functioning and will send the adequate response to the application, by means of the manager module. The application will process this response, and after will continue its execution. As calls as are necessary can be made.

The control of time was implemented by formulae, which calculate the real-time the objects spend to complete their tasks. In the environment simulation, the time spent for executing the tasks of each object is proportional to the real-time that is calculated by the formulae, also considering the computer internal clock as a basis.

B)-Creator Module

This module is responsible by graphically construct the environment of the target system. So, it is accessed a library of resources, which contains all graphic components that are required by the system being simulated. Each graphic component is represented as an object, with specific characteristics of functioning and representation.

The graphic system produced may be stored in the system library, to be subsequently used by the manager module.

During the graphic construction of system environment, the creator module makes the connection between the objects that were specified by the user. This connection requires the identification of possible communications between system objects and pre-defined initial state of the system, which, obviously, will be changed along the execution.
C)-Libraries

The tool has two libraries, that is, graphic library and system library.

Graphic library contains pre-defined graphic objects, which are required in the construction of graphic systems. Procedures for representing the environment behavior, such as manipulation of colors, and resources for animation, also take part of this library.

System library contains the objects and their interconnections, related to the system being simulated. So, from a start call, the manager module accesses the system library and gets the requested system.

D)-Interface with the Application

The interface between application and tool is made by parameters indicated in tasks calls. So, the call of a certain application contains the specification of graphic objects to be used, each one including the following parameters: identification of the object; type of information to be manipulated; desired action; and a variable for actualizing or reading the resulting information from the action. It is followed a pattern which is similar to that one used by the X-View to describe primitives [HEL90].

A specification example of call primitive, taken from the System of Mixture of Liquids, is:

Pipe (i, state, reading, value)

Where:

- Pipe = Contains the name and type of the object. In this case, it was specified the pipe object, which allows the passage of liquids, depending on its state.

- i = Is the identification of the object, among other objects of the same type. This means that more than one pipe can exist in the system.

- State = In the case of pipe, state can assume two values, open or closed.

- Reading = This allows to obtain the value of the state of the considered object.

- Value = This is a variable that is associated to the current state of the object, in the instant of the call.
3. 2. Implementation of the Tool

Most current tools for graphic support were implemented with UNIX Operating System, by means of programming language C and X-Window System. UNIX is adopted because it is becoming a pattern; C because of its potentiality and portability; and X-Windows because it makes easier the construction of good human-computer interfaces.

In the implementation of the tool here described, besides the above resources, it was used the X-View tool kit [HEL90], which contributed to the creation and control of windows and graphic objects.

The version currently implemented runs in Sun Spark-2 Workstations. The prototype incorporates facilities of graphic manipulation, interactive functioning based on menus and windows, options of controlling the presentation area of screens, etc.

The visualization of the system simulation is triggered by a call of start, which generates a graphic representation showing the system initial state and the system execution environment.

When the creator module is accessed, the tool shows the objects of the graphic library, so that the designer can choose the objects of interest for the system being developed. In the System of Mixture of Liquids, some important objects are: pipe, reservoir, tap, and devices for measurement and control of volume, temperature, density, outlet of liquids, etc.

Next, the initial state and attributes of each object may be defined. For this, the tool opens specific windows that ask the necessary information.

After be generated the graphic representation of the system environment, the animated visualization of the system simulation can be obtained. For that, the designers have to prepare the PCS computational control, by primitives provided by the tool and, sometimes, by simple coding pieces made in C language. The primitives work as if they would be sensors and actuators of PCS.

The system behavioral representation must allow not only the visualization of the system in its current state, but also the identification of critical and of exception situations, which can endanger the system results. Thus, for the System of Mixture of Liquids, the designer may observe problems, such as surpassing the accepted liquid volume in the reservoirs, inappropriate liquid outlet, and other harmful situations occurring during the system operation.

Figure 3 examplifies a screen of the tool.

The complete description of the tool, including its architecture, functionality, sample of screens, and simulation of the System of Mixture of Liquids, is given in Pelinto [FEL93].
4. Conclusions

The development of high-quality process-control systems depends heavily on the use of effective validation techniques. In this sense, simulation is pointed-out as a very powerful technique for validation of dynamic behavior and partial and final results of complex, real-time process-control systems.

This paper exploits the use of scientific visualization as a form of improving simulations of real-time process-control systems. Relevant concepts on the subject are discussed, and a tool which was implemented with basis on these concepts is therefore described. A simple example of System of Mixture of Liquids is used to explain some concepts and functioning of the tool.

In summary, the paper intends to contribute to the creation of correct and safe process-control applications, through software tools for graphic and animated simulations using scientific visualization.

References


[FEL92]-Felinto, A., Support to the Visualization of Scientific Data from Real-Time Systems Simulations, UFSCar, São Carlos, Brazil, 1994 (Master's Dissertation - in Portuguese).


Appendix A - System of Mixture of Liquids

The goal of the system is to produce a chemical mixture, by the composition of two different liquids. For this, some values of characteristics related to the initial liquids and final mixture (such as: volume, temperature, density, outlet, draining, etc.) must be reached and maintained.

The system behavior needs to be monitored to the system accomplish its goals. The following components take part of the controlled application: (a)-four reservoirs to store liquids and mixture; (b)-input and output pipes, in where the liquids flow; (c)-taps connected to the pipes, which allow the appropriate outlet and draining of liquids; (d)-devices for measuring and controlling volume, temperature, density, outlet, and other physical properties.

In relation to the control of volume, the system has the following rules:

A)-Respective input pipes will be immediately closed if the volume of liquid in the reservoir-1 completes 95% of its maximum capacity. Otherwise, they will be open if the volume of liquid in the reservoir-1 completes 55%, being also assumed the rule B.

B)-Respective pipes will be immediately closed if the volume of liquid in the reservoir-1 is sufficient to fill 95% of the maximum capacity of the other reservoirs 2, 3, and 4.

C)-An output pipe will be immediately closed if the respective reservoir where the pipe releases liquid completes 95% of the total volume.

D)-For values below of 40%, the output of a reservoir will be turned off, being turned on again when the volume of liquid in the reservoir reaches 60%.

About the Authors:

Tereza G. Kirner is Adjunct Professor at Department of Computer Science, Federal University of São Carlos, SP, Brazil, currently developing Post-Doctorate Program at Center for Software and Systems Engineering, University of Colorado at Colorado Springs, with grant provided by FAPESP (Proc. 93/0341-7). E-Mail: tkirner@lalo.uccs.edu.

Alan S. Felinto received his M.Sc. Degree in Computer Science from the Federal University of São Carlos, SP, Brazil. E-Mail: alan@fuel.br.bitnet.