2011 International Nuclear Atlantic Conference - INAC 2011
Belo Horizonte,MG, Brazil, October 24-28, 2011
ASSOCIAÇÃO BRASILEIRA DE ENERGIA NUCLEAR - ABEN
ISBN: 978-85-99141-04-5

CONSTRUCTION OF NEW OPERATION INTERFACE FOR THE LABIHS SIMULATOR USING THE ELIPSE E3 STUDIO SOFTWARE

Silas C. Augusto and Mauro V. Oliveira

Instituto de Engenharia Nuclear, CNEN/RJ
Universidade do Estado do Rio de Janeiro
Rua Hélio de Almeida, 75
21941-906 Rio de Janeiro, RJ
silas@ien.gov.br; mvitor@ien.gov.br

ABSTRACT

The Human-System Interface Laboratory (LABIHS), located at the Instituto de Engenharia Nuclear (IEN), has a compact simulator that simulate the processes of a Pressurized Water Reactor (PWR) Nuclear Power Plant (NPP) of 930 MWe of power. This simulator is composed by a HP-UX workstation computer, where the simulation software runs, and a set of computer stations, that represent an advanced control room, where the simulator is operated by software control panels that represent several systems of the simulated nuclear power plant. The current HSIs for the LABIHS simulator was built using iLog software tool. The development of new human-system interfaces (HSIs) for the simulator is one of the research fields of LABIHS. This paper presents the screen components development process for a new HSI for the LABIHS simulator, using the software Elipse™ E3 Studio. These new components developed using the E3 Studio are similar to the ones used in the current simulator interface. The article shows some comparisons between the component and screen development with Elipse™ E3 Studio processes and using iLog Studio.

1. INTRODUCTION

The human-system interfaces (HSI's) comprises an important part of an industrial plant, with which operators interact to perform their functions and tasks. The HSI's include alarms, information displays and controls. All HSI is made of hardware and software and is characterized in terms of its physical significance and functional characteristics. The NUREG-0700 [1] reviews the physical and functional characteristics of HSI's. The design aspects of human factors engineering (HFE) of HSI's are covered by NUREG-0800 [2] and NUREG-0711 [3].

Based on literature, interviews and visits to industrial plants, O `Hara *et al*. [4] identified challenges in the technology of human-system interfaces and its potential effects on the performance of people. The topics were evaluated taking into account their impact on plant safety [5]. With respect to human-system interfaces of plants it was found that with the rapid development of computer technology, more and more human-computer interface systems has been introduced in the conventional man-machine systems. Currently, for example, some computers may represent a large control room with numerous gauges, buttons and actuators of a nuclear power plant or a system of aid to the pilot of an aircraft cabin. Gradually, man-machine interface systems with conventional meters and actuators connected by numerous wires and panels are being replaced by human-computer interface systems with a few screens. This means that currently the man-machine interfaces can be considered as human-computer interfaces and digital interfaces.

In addition, the control systems of industrial plants are becoming more complex and, consequently, the operator interfaces of these systems are also becoming more complex, which makes the design of these interfaces a complex task. When a system of a plant becomes more complex, its dynamics becomes more complex. The increasing complexity of the dynamics of the plant can lead to difficulties in monitoring/control of this system, they introduce extra needs in the design of the interface.

As in any complex system design, its highly desirable to use a theory/methodology for the design of interfaces. The methodology to be used in the design of an interface must take into account the following issues:
a) What should be presented?;
b) How should it be presented?;
c) When should it be presented?.

To help answer these questions, it has been developed in recent years some methodologies/philosophies of building interfaces, among which we mention: the task-oriented interface [6]; ecological interfaces [7], and function-oriented interfaces [8].

The main objective of this article is to document and briefly present the development of a new operator interface to the LABIHS simulator through the development of new screens with all components similar to the previous ones, using Elipse™ Software E3 Studio [9-11].

The remainder of this paper is organized as follows. Section 2 presents an overview of the LABIHS simulator. Section 3 presents the development process on the E3 Studio software tool and some of its features. In section 4 is made a comparison between iLog Studio and E3 Studio. Finally, section 5 brings some conclusions and comments on the current stage of the human-system interface development for the LABIHS simulator.

2. OVERVIEW OF THE HUMAN-SYSTEM INTERFACE LABORATORY

2.1. LABIHS Objectives

The LABIHS laboratory has the following objectives:
- providing technological expertise in the design of graphical interfaces;
- design systems to aid the operator;
- carry out modernization of control rooms;
- carry out evaluation of control rooms and interfaces considering aspects of ergonomics and human factors;
- analyze the interaction between operators and the various systems operated by them;
- assess the reliability of human operators in simulated accident scenarios and normal operation.

2.2. LABIHS Advanced Control Room

A control room contains the systems and operation manuals needed to control the operating conditions of an industrial plant, so to ensure that its operation and shutdown processes are reliable and safe either under normal circumstances and during accidents [12].

Advanced control rooms of industrial plants consist of an array of systems and equipment, where operators can monitor, control and intervene in the process through various graphical

user interfaces and monitoring stations. These interfaces have significant implications for the safety of the plant, since they affect the activity of operators, affecting how operators receive information regarding the status of key systems and determine the requirements necessary for operators to understand and supervise the system. The main task of operators is to keep the plant operating under acceptable conditions of safety and efficiency. The actions taken by the operators are supported through starting and stopping procedures, emergency procedures, alarm systems, communication systems, control systems, security systems and fault diagnosis. Operators also interact among themselves and with their support teams, i.e., the maintenance, testing and planning teams. In control rooms, the operator is seen as an agent that monitors the automated process and should act in case of failure of any automated system. However, operators often do not receive the proper training, resulting in several problems. Also, systems often do not provide enough information that could help operators to act properly when required.

2.3. LABIHS Simulator

The simulator of a nuclear reactor is a tool that simulates the process of generating electricity from nuclear fission, and the main features of the systems that make up the nuclear plant, using mathematical models to analyze the dynamic behavior of these systems. Simulators are used as support for regulatory agencies, the training and qualification of operators, security analysis and validation of operating procedures. The simulator shall transmit to the operators a realistic view of the nuclear plant, providing the illusion that the operators are operating a real plant, showing the operation of key systems and enabling operators to experience actions they should perform in a real control room.

The LABIHS simulator consists of a set of equipment and computer programs that simulate the processes of a PWR (Pressurized Water Reactor) nuclear power plant that generates 930 MWe of power, forming a compact nuclear power plant simulator and offering a control room with several advanced graphical interfaces representing the various systems that make up the nuclear reactor. Figure 1 shows the basic components of LABIHS.

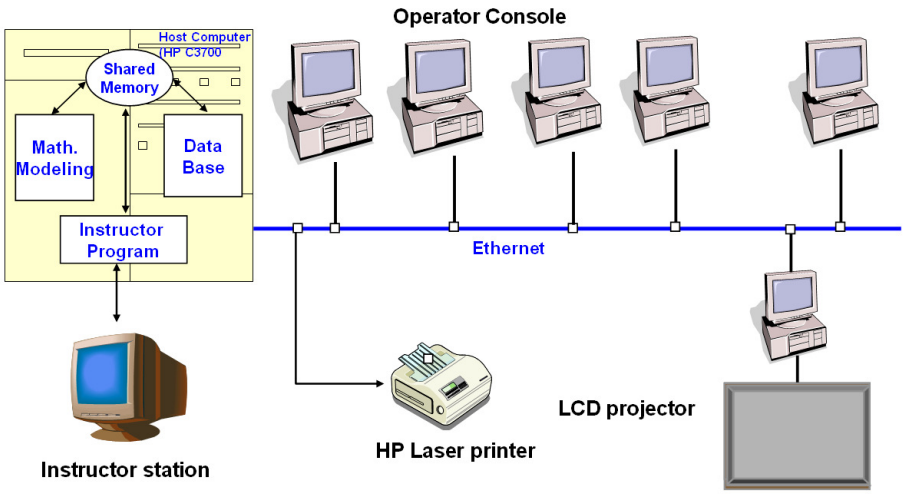


Figure 1. Basic components of LABIHS.

The simulator's operation team consists of three operators: the operator that supervises the reactor and primary circuit systems, the operator that supervises the turbine and secondary circuit systems, and the supervisor. Each operator controls and monitors the systems under its responsibility through three LCD computer screens, a keyboard and a mouse. On the room's front wall is a screen that shows the overall operation of the nuclear plant and systems. This screen is intended to provide the operator with an integrated view of the reactor operation. Figure 2 presents an overview of the LABIHS control room.

Figure 2. LABIHS advanced control room.

In addition, there is a small room within the control room where, through a workstation, the instructor controls the simulation software. Figure 3 shows the main screen used by the instructor.

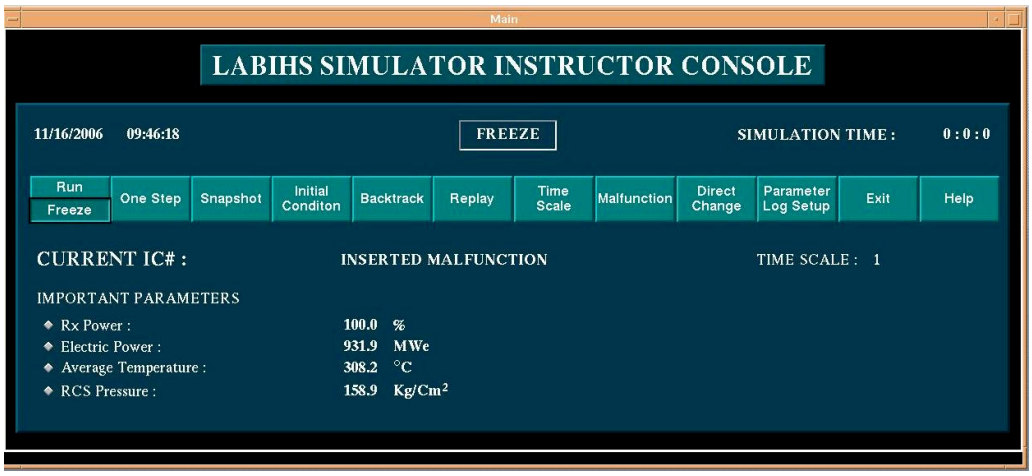


Figure 3. Instructor Console main screen.

3. SCREEN AND COMPONENTS CREATION USING E3 STUDIO

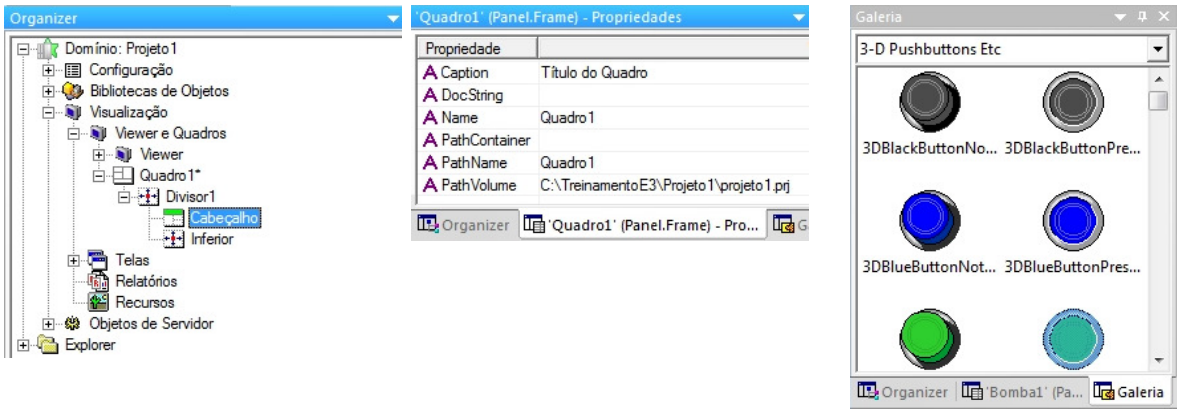
The E3 Studio is a software developed by Elipse™ Software that enables the construction of computer user interfaces for control and supervision of automated systems. On the scope of this article, the focus will be on the abilities of E3 Studio to create screens and components for a new user interface for the LABIHS simulator software. The following sub-sections describe the details on how to develop screens and components in E3 Studio.

3.1. Screens Creation In E3 Studio

On the E3 Studio, during the creation of a new project, a database file and a communication driver can be specified. A database file stores the data acquired or generated by the application. A communication driver enables the application to communicate with a specific device to acquire/send data, simplifying the development of applications. If a driver for a certain application do not exist, it can be developed by a E3 specific package software.

3.1.1. Organizer, frames and screens in the E3 Studio

After running the E3 Studio software and opening/creating a new project, on the left side of the software window is an area where the user can view the Organizer, the Properties and the components Gallery, by clicking of the corresponding tab at the lower left corner of the window. The Organizer tab shows a tree-like structure that manages the various elements that compose an application developed on the E3 Studio. The Properties tab shows properties of the elements currently selected on the edit window. The Gallery tab shows a list of drawings, organized by category, selectable on the combobox at the top, that come pre-installed with the E3 Studio and are useful for creating new components. Figure 4 shows the Organizer, Properties and Gallery tabs.



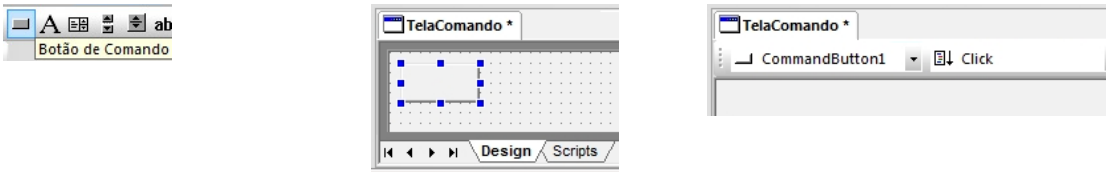
(a) (b) (c)

Figure 4. Opening/creating a new project 0n E3: (a) Organizer tab; (b) Properties tab related to Frame1; (c) Gallery tab.

Among the elements managed by the Organizer are screens and frames. Screens enable the user to aggregate the graphical components chosen for a future screen by placing them on a window, and defining their position and attributes. Frames enable the user to define the placement and size of screens on the application windows, by dividing a frame into two or more sections or sub-sections and assigning a screen to each section or sub-section.

3.2. Creating Screen Navigation Buttons

In an application made in E3 Studio, navigation between screens is accomplished by buttons that, when clicked with the mouse, change the screen displayed in a given sub-section of the frame on the application window. Navigation buttons have scripts associated to its Click event that define which screen will be displayed on which frame once the button is pressed. Figure 5 shows the icon to create a new command button, a newly-created Navigation button and its Click event.



(a) (b) (c)

Figure 5. Screen navigation buttons: (a) Command Button highlighted; (b) new Navigation Button; (c) Click event of a Navigation Button.

3.2. Screen Components Creation In The E3 Studio

Before creating components in the E3 Studio, an object library must be create to store them, as shown of figure 6.

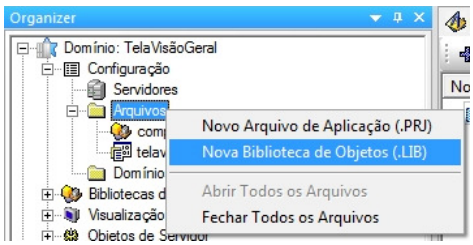


Figure 6. Creating a component library.

Components created on E3 Studio are broken into parts placed at different points of the Organizer tree-like structure. They are:

- XObject, on Domain -> Object Library -> XObject;

- XControl, on Domain -> Object Library -> XControl;
- Command screens for some components, on Domain -> Visualization -> Screens;
- Scripts, accessible by clicking on the Script tab while viewing a XObject or Command screen;
- Data Object, on Domaín -> Server Objects -> Data Object.

Since components' parts are placed at different points on the Organizer, a naming convention was adopted to better relate these parts to its related component:

- XObjects have a Data (*Dados*) prefix;
- Common XControls don't have prefix or suffix;
- XControls that compose a Command screen have a Command (*Comando*) prefix;
- Command screens have a CommandScreen (*TelaComando*) prefix;
- Data objects have their names on plural, i.e. a *-s* suffix.

As an example, the component *Bomba1* (Pump1) is composed by the XObject *DadosBomba1*, XControls *Bomba1* and *ComandoBomba1*, Command screen *TelaComandoBomba1* and Data Object *Bombas1*.

3.2.1. XObject

A XObject define the properties of a component. Properties have the following fields: name, type, public, initial value and description. Figure 7 shows the creation/editing of XOject properties.

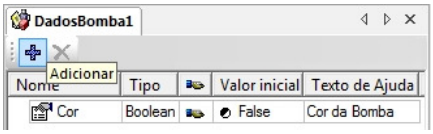
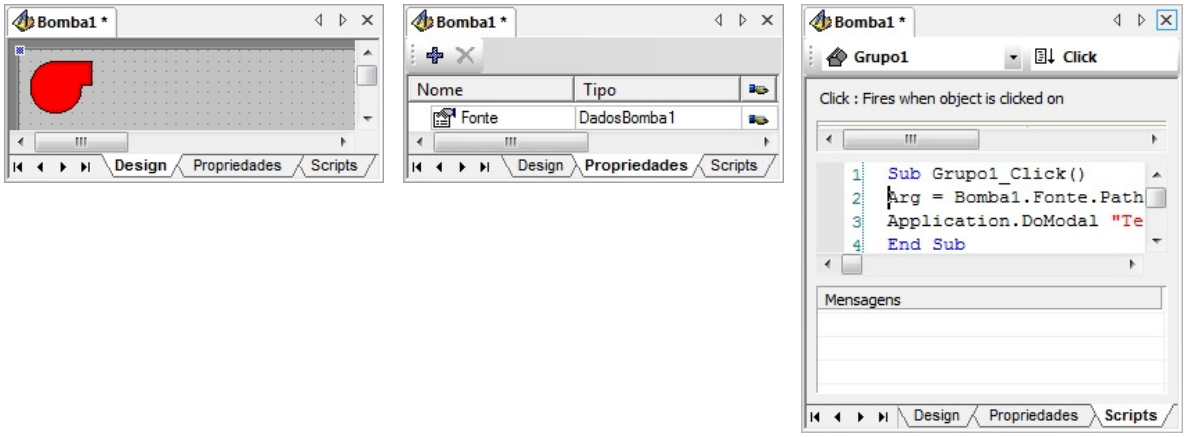


Figure 7. XObject properties.

3.2.2. XControl

A XControl define the component's appearance and behavior. Its development is divided into three tabs, Design, Properties and Scripts, located on the lower corner of the XControl editing area, as shown on Figure 8.



(a) (b) (c)

Figure 8. Design a XControl component: (a) Design tab; (b) Properties tab; (c) Scripts tab.

In the Design tab, the component is designed by inserting and modifying objects such as rectangles, text boxes and command buttons on the editing area. To ease the design of components, the drawings on the E3 Studio Gallery can be dragged to the editing area and utilized as a starting point.

In the Properties tab, the data source for the component is defined, which is the respective XObject to the XControl. As illustrated in Figure 9, its similar to a XObject property, but there's only one property, the Name field is always equal to Source (*Fonte*), and the Type field is the respective XObject to XControl (e.g., the XObject *DadosBomba1* for the XControl *Bomba1*).

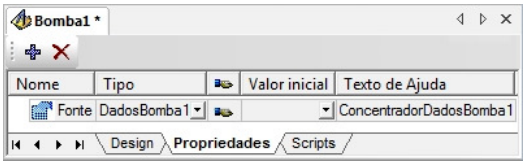


Figure 9. Properties tab of XControl Bomba1.

The Scripts tab is where are programmed the scripts associated with specific events of certain objects placed on the Design tab, such as the Click event of a Command Button, as shown in Figure 10. Command screens also have scripts.

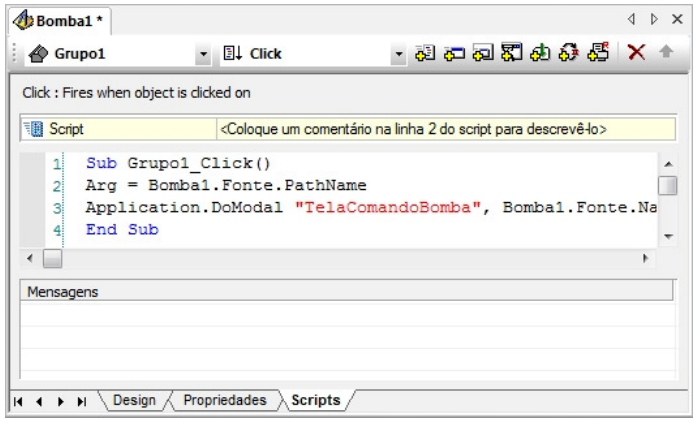


Figure 10. Scripts associated with specific events of certain objects.

3.2.3. Command Screens

Command screens are small pop-up windows that appear when clicking on certain components and contains controls that allow to change the component's state, such as ON and OFF buttons that enable to turn on/off a pump. The development of a Command screen is divided into:

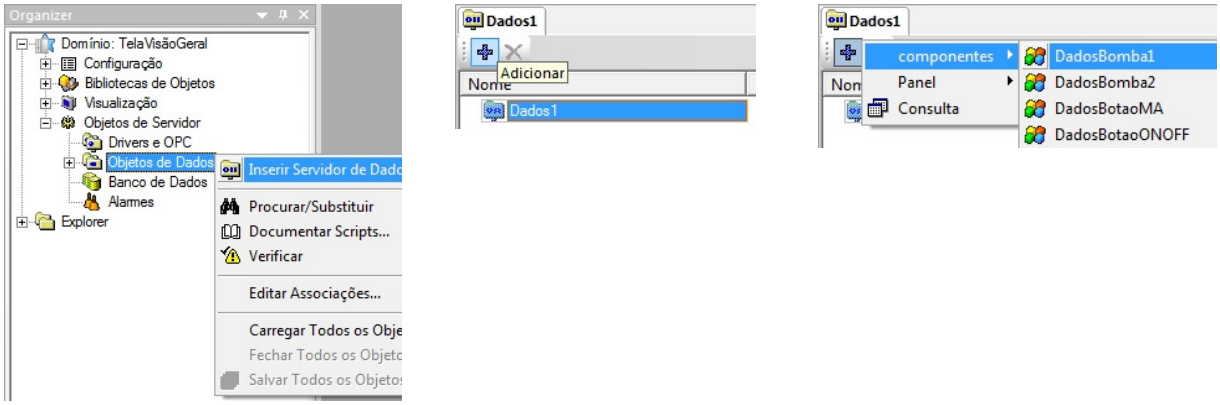
- A Script of a XControl's Click event;
- A XControl where are placed the controls and/or indicators and its scripts (ex: the XControl *ComandoBomba* with the ON and OFF buttons, each one with a script for its Click event);
- A screen where the aforementioned XControl is inserted; this screen also has a script (ex: *TelaComandoBomba*).

A Command Screen is exemplified on Figure 11.

Figure 11. A Command Screen.

3.2.4. Data Objects

In the Organizer, under Domain -> Server Objects are placed the Data Server nodes, that aggregate Data Objects. Put simply, Data Objects function as 'instances' of components. For example, for the Pump1 (*Bomba1*) component, there are Data Objects named *RHR*, *ChargingPump1* through *3*, *RHRPump* and *SprayPump*, placed under the *Bombas1* Data Server. Figure 12 illustrates the creation of a Data Server and the creation of a Data Object on a Data Server.

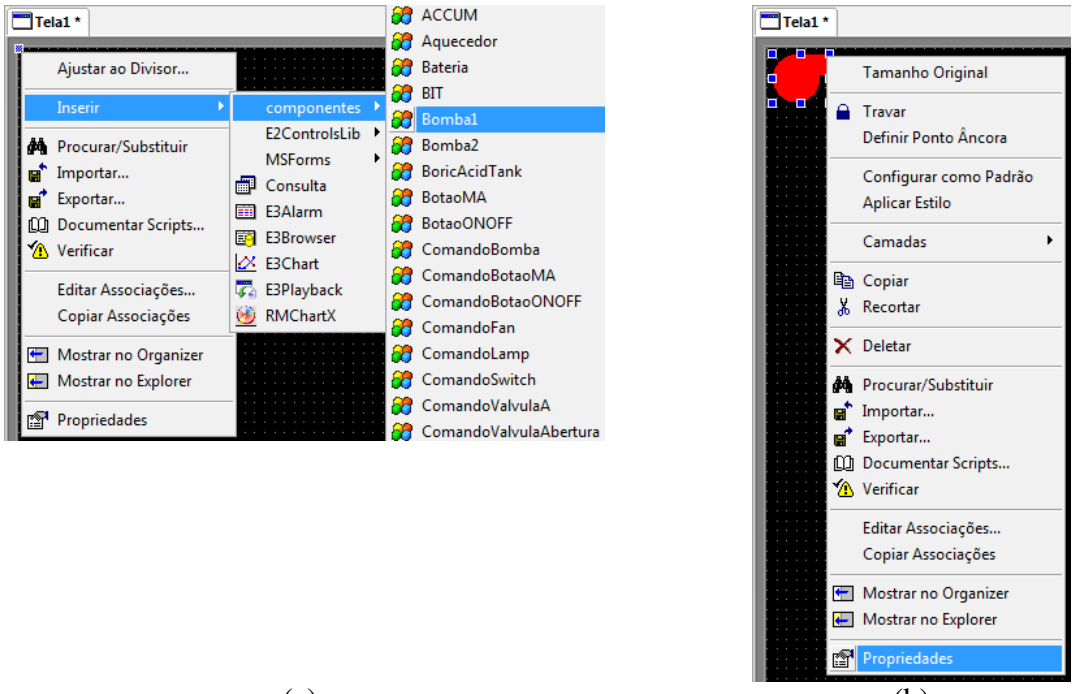


(a) (b) (c)

Figure 12. Example of: (a) Creating a Data Server;. (b) Selecting a Data Server; (c) Choosing a XObject.

3.2.5. Example of How to Insert a Component on a Screen

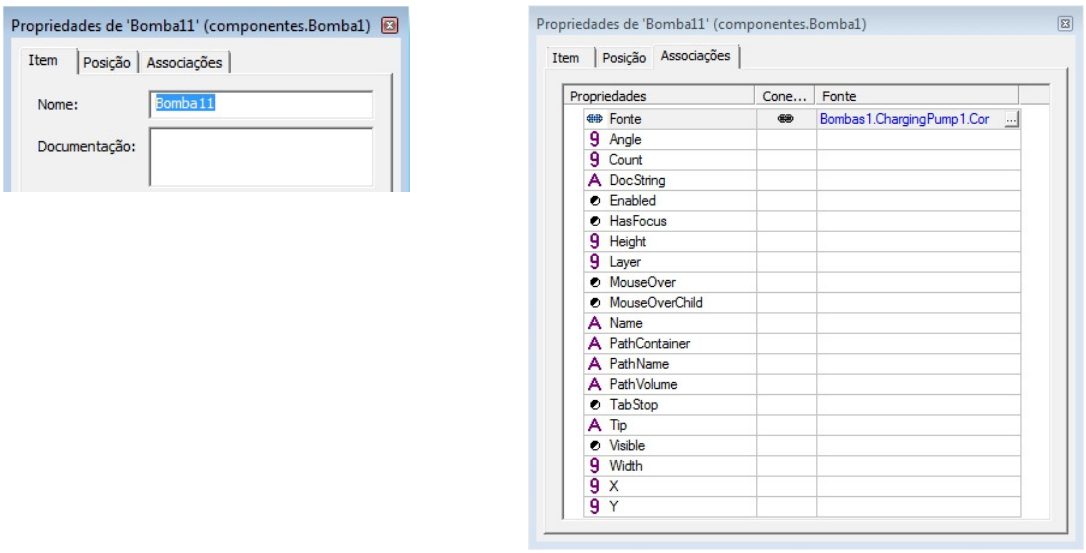
As shown on Figure 13, to insert a component on a screen, open a screen and, in the editing area, open its context menu and choose 'Insert' -> 'components' -> name of XControl desired, such as *Bomba1*. After inserting the component, open its context menu and choose Properties.



(a) (b)

Figure 13. Example of inserting of: (a) Component on a screen; (b) Component properties.

As shown on Figure 14, on the Item tab, change the name of this 'instance' of the component to better identify it. On the Associations tab, on the *Fonte* property, set its value as the property of the Data Object that must be associated with this XControl 'instance', e.g. *Bombas1.ChargingPump1.Color*.



(a) (b)

Figure 14. XControl 'instance': (a) Properties' Item tab; (b) Properties' Associations tab.

4. SCREEN AND COMPONENTS CREATION COMPARISON BETWEEN E3 STUDIO AND ILOG STUDIO

The development of screens and their components in E3 Studio is quite different from how it is done on iLog Studio [14-15]. The software has a very different user interface, in which its structure work is quite different. The following sections presents a comparison of the components development process on iLog Studio and on E3 Studio, and the screens development process on HSI Builder and on E3 Studio as well. Finally, are pointing out the positives and negatives aspects of both software.

4.1. Development of Screens

For creating a screen using the components created in the iLog Studio was developed a graphical user interface software called HSI Builder. The development of screens on the HSI Builder is simple. After running the HSI Builder, simply follow 4 steps:
1) open/create a screen;
2) select a tab for a group of components;
3) drag and drop a component to the screen;
4) double-click on the component to edit its properties.

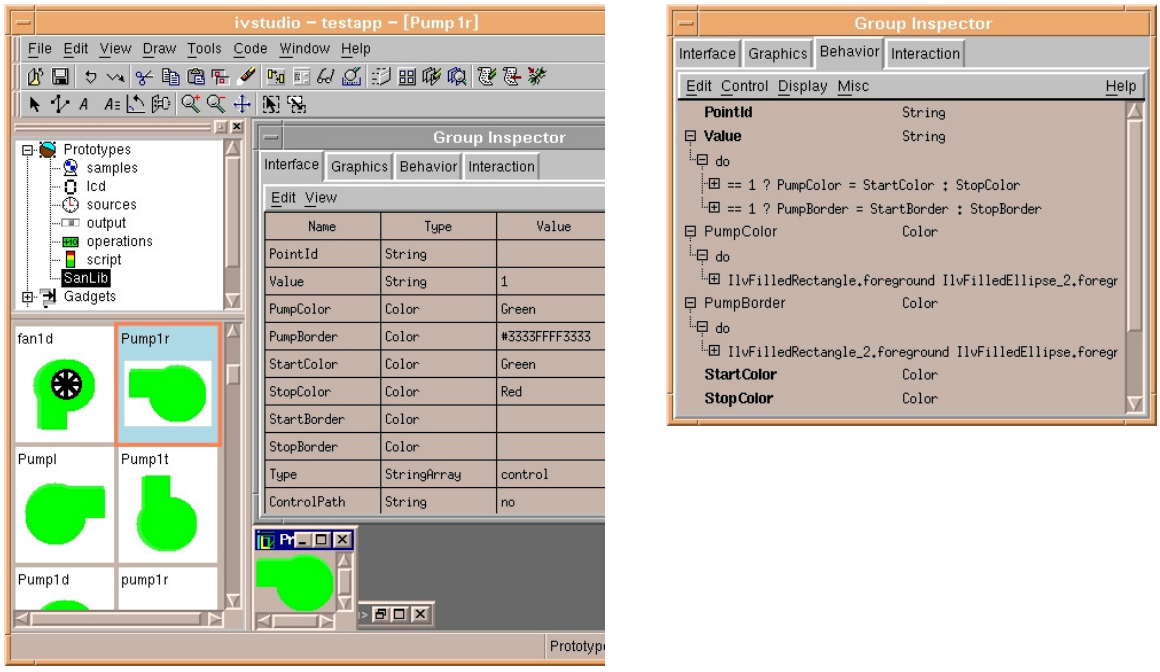
However, for each component placed on a screen, the user must set the value of its attribute PointID (tag component name), which needs to be checked on another software.

The development of screens on the E3 Studio is also easy. After running the E3 Studio and open the project file, simply follow 4 steps:
1) open/create a screen;
2) with the mouse pointer over the screen, open the context menu with the mouse;
3) on the context menu, choose Insert -> component's library -> component's name to insert the desired component on the screen;
4) edit its properties.

The development of both components and screens integrated into the same software ease the development process. On the E3 Studio, opening several screens at the same time for editing works better than on HSI Builder, as each opened screen has its own tab to quickly access and view it. Also, since the data source is defined during a Data Object's creation, once all Data Objects required for a project are created, its easier to add new 'instances' of existing components to screens, e.g. to add a component that represents a specific pump on a screen. On HSI Builder, it's always necessary to check a PointID's list to find the desired PointID and copy its name to the component's PointID property.

4.2. Component Development

On the iLog Studio, after the component is drawn, the rest of the development is carried out in the Group Inspector window, which has the Interface, Graphics, Interaction and Behavior tabs. The most common procedure is to create attributes in the Interface tab, define behaviors in the Behavior tab, and then save the component in a component library. Figure 16 illustrates the iLog Studio and Group Inspector windows.



(a) (b)

Figure 16. (a) iLog Studio window and Group Inspector window Interface tab, (b) Group Inspector window Behavior tab.

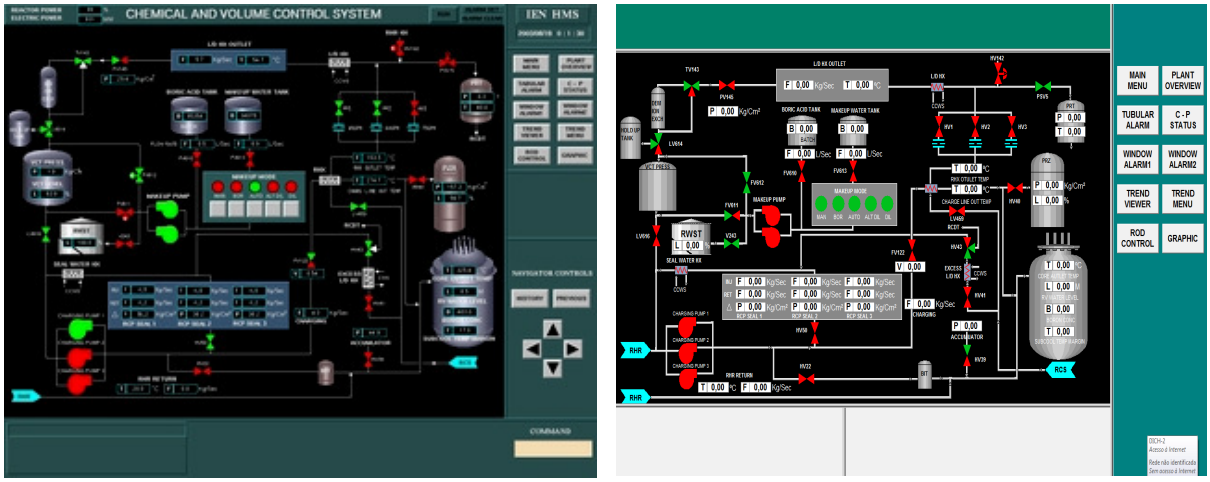
Due to the development process being concentrated in the Group Inspector window, the development process is better organized. Create and define behaviors for the component is made easier by the menus present on the Behavior tab, which allow you to choose certain behaviors and set them up without the need to program scripts. On the other hand, the technical support for the iLog Studio software in Brazil is less accessible when compared to the technical support for the E3 Studio software.

On the E3 Studio software, as components are 'broken' into XObjects, XControls, screens, Data Objects and Scripts, and each kind of these is placed at a different node on the Organizer tree-like structure, the component development process is somewhat 'scattered' in the software's internal structure. Also, the development of a component should follow the aforementioned order due to cross-references between them. These factors make the development process confusing and harder for novice users of the E3 Studio. On the other hand, Elipse™ Software is a Brazilian company, which facilitates the purchase of new software licenses and access to its technical support, which was more accessible than the technical support of iLog Studio. Also, the use of scripts allows more flexibility and customization in the development of components.

Table 1 highlights relevant differences between development on iLog Studio and on E3 Studio and Figure 18 shows the chemical and Control Volume System of the PWR simulator developed in both systems.

Table 1. Relevant different aspects between iLog Studio and E3 Studio.

iLog Studio	**E3 Studio**
Component development more cohesive on Group Inspector window	Component development 'scattered' on Organizer tree-like structure, harder to novice users, caution needed to sub-component cross reference.
For each component instance that will represent the same simulated object (eg. a specific pump), the PointID of such object must be entered. Another software called Instructor Panel is necessary to search for PointID value.	Once a Data Object representing a simulated object is created, new component instances can be added to screens with no need to research simulated object data source.
Needs HSI Builder to create screens and Instructor Console to search PointID.	Does not need external tools.
Technical support less accessible on Brazil	Technical support more accessible on Brazil.
Harder to integrate to simulation software	Easier to integrate to simulation software, provide creation of custom data source driver
Using terminal emulators, several developers work simultaneously using same license, as version acquired for LABIHS is for HP-UX (Unix-like)	Just one developer works at a time, as version acquire for LABIHS is for Microsoft Windows



(a) (b)

Figure 17. Chemical and Volumetric Control System Display: (a) iLog Software; (b) Elipse™ E3 Software.

5. CONCLUSIONS AND COMMENTS

As information technology constantly evolves, the obsolescence of hardware and software must be taken into account. The future migration of the LABIHS simulator to a new hardware and software platform will allow a better maintenance of the laboratory and ease the development of graphical interfaces.

One of the issues to be addressed in this process is to complete the development of a new HSI for use in this new environment. Presently, the LABIHS simulator is running using the HSI developed with iLog Studio. In this articles was presented the use of E3 Elipse™ software to build the new HSI for the LABIHS simulator. In this way, most of the components used to compose the screens of the LABIHS simulator were recreated on E3 Studio and some of the screens of the simulator were also recreated on the E3 Studio as well. The development process of new screens on E3 Studio was easier when compared with the process made thorough the iLog Studio.

The next steps in the project is to establish data communication between the new simulator interface developed on E3 Studio and the current simulator, so that components can be functionally tested and evaluated in a working simulator. However, it depends on the development of: 1) a communication driver for the E3 Studio that will enable the communication with the simulator, and 2) a software running on the simulator workstation that will make accessible the contents in the shared memory for the communication driver. After, the remaining undeveloped screens will be recreated on E3 Studio.

REFERENCES

1. NUREG-0700, “Human-System Interface Design Review Guidelines”, **U.S. Nuclear Regulatory Commission Research**, Washington & (2002).
2. NUREG-0800, “Standard Review Plan, Chapter 18 Human Factors Engineering”, **U.S. Nuclear Regulatory Commission Research,** Washington & United States (2004).

3. NUREG-0711, “Human Factors Engineering Program Review Model”, **U.S. Nuclear Regulatory Commission Research**, Washington & United States (2004).
4. O´Hara, J. Stubler, W., e Nasta, K., “Human-system interfaces management: Effects on operator performance and issue identification”, **BNL Report W6546-1-1-7/97**, Upton, Brookhaven National Laboratory, Nova York & United States (1997).
5. Stübler W., Higgins J., and O´Hara J., “Evaluation of the potential safety-significance of hybrid human-system interface topics”, **BNL Report J6012-T2-6/96**, Upton, Brookhaven National Laboratory, Nova York & United States (1996).
6. Balbo S., Draheim, D., Lutteroth, C., “Appropriateness of User Interfaces to Tasks”, TAMODIA, Gdansk, Polônia pp. 26–27, (2005).
7. Vicente K.J., Rasmussen J., “Ecological interface design: theoretical foundations”, ***IEEE Transactions on Systems, Man and Cybernetics***, **Vol**. 22 (4), pp. 589-606 (1992).
8. Lin, Y., “Experimental study based on eye gaze measurement for computer interface”, **Technical Report (AEDL-2000-L7Z01)**, Advanced Engineering Design Laboratory, Department of Mechanical Engineering, University of Saskatchewan & Canada (2000).
9. Elipse™ Software Ltda., “Tutorial do E3”, Versão 3.2 (2009).
10. Elipse™ Software Ltda., “Manual do usuário do E3”, Versão 3.2 (2009).
11. Elipse™ Software Ltda., “Manual de referência de scripts do E3”, Versão 3.2 (2009).
12. CNEN-NE-1.01, “Licenciamento de operadores de reatores nucleares”, **Comissão Nacional de Energia Nuclear**, CNEN, Brasil (1979).
13. HSIL Simulator - Human System Interface Laboratory Simulator, **HIS Builder User’s Manual**, Doc. ID: IEN-HSIL-DOC-06-APPENDIX 2, Instituto de Engenharia Nuclear, Brasil (2002).
14. iLog, “iLog Views Studio 4.0 - User’s Manual” (2000).
15. iLog, “iLog Views Controls 4.0 - User’s Manual”. (2000).