MAN MACHINE INTERACTION TOOLS BASED ON X-WINDOWS

D. Giove, G. Cuttone

INFN Milano and INFN-LNS Catania

Abstract

The development of tools which makes comfortable both the work of a software engineer designing man machine interfaces both the work of an operator in a control room is one of the most important items in a control system of a modern accelerator.

The availability of powerful workstations and of X-Windows has made possible to modify the way to discuss the problem of man-machine interaction.

Introduction

The role of operators of accelerators has changed greatly in recent years. A major reason is the increasing prevalent trend toward automation in the control of large complex systems. The introduction of automation changes the role of the operator responsible for the system from a controller to a monitor and, when circumstances warrant, to a troubleshooter responsible for fault detection, diagnosis and compensation. The computer is responsible for making low-level control decisions; for the most part the human passively monitors the evolving states of the computer controlled system.

The effectiveness of supervisory control of complex systems depends on the way which has been choosen for coping with the overall system. The concept of complexity is not an objective feature but depends rather on the methodology which has been adopted for the operator interface. Traditionally, control rooms have been defined at a very complex level of representation; this has been due to the hardware-oriented and one-sensor/one-indicator philosophy followed in the design in order to reproduce electromechanical displays by means of computer generated pages. Such an approach may be useful during the development phase of a control system, when it may be considered a tool for debugging single equipments, but it is unfeasible during operation of the whole accelerator.

The alternative to this approach is to consider an operator, rather than hardware, driven design. This choice requires to define a model for the processes of decisionmaking and problem-solving involved in supervisory control functions and to take advantage of the available technology and tools for its implementation.

Control systems layout

Modern accelerator control systems are based on quite a conventional scheme which has to fulfill several general requirements.

The control system plays a central role in operation of an accelerator and it represents one of the major items in the budget. This implies that the basic architecture has to guarantee such a flexibility and modularity to allow any sort of modifications during its lifetime and that maintenance have to be carried out without any interference with accelerator operations. Distributed systems represent an ideal answer to these demands and may be easily implemented using commercial standard products. The availability of components which follow international or de-facto market standards represents a great improvement in the way to think a control system. The attention may be moved from the home made development of basic components, which is a really expensive and time consuming activity, to the definition of general rules for the integration of commercial products in a multivendor, suitable environment. Proprietary solutions and closed systems must be avoided if possible to limit the risk of lack in the technical support.

Regardless of the real complexity of the accelerator, a modern control architecture develops on three levels[1,2] : the plant level, where we have the single equipment control; the process level, which is responsible for the operation of a set of functional or geographic related devices; the supervisor level with the operator interface. Three different networks provide the connection between the different levels and between units in the same level. A hierarchical, serial, high speed network is dedicated to the plant controller boards; processor stations are connected usually on Ethernet or Token Ring networks, sometimes using fibre optics as a transmission media to improve reliability. Control rooms have their own network, Ethernet or Token ring based, connected directly or by means of gateways or bridge to the other networks.

The design of the software architecture costs considerably in manpower and suffer from the lack of products to be taken as a general reference. The choice of the operating system, of the language , of the support of multiprocessing are all arguments of discussions between people involved in control design.

Control room layout

The establishment of accepted general rules for the lower levels and the already discussed importance of the operator interface in the management of a control system have stimulated a lot of work in the design of new approaches to man machine interaction tools.

A dramatic change in the design of main and local consoles has been produced fewyears ago by the availability of the first workstations at low cost and with a lot of basic software tools. These computers, which were already well appreciated for their use as technical tools in electronic and mechanical design but which presents very high costs, proposed themself as the answer to a lot of questions and doubts about the architecture so far followed for console nodes. According to the structure of process stations, consoles were designed as a set of specialized boards fitted inside a cardcage and with a particularly developed software suited for those boards and that particular configuration. Expandibility was guaranteed, from the point of view of hardware, as a board would be easily inserted in the cardcage, and also for the software, but the time involved was too high and the environment was too much bound to the particular boards used. Availability of new, more powerful hardware forced to rewrite most of the software and this limits the capability of the whole system to be able to meet future needs and requirements. On the other hand, there were no alternative choices up to the availability of 80286 based Personal Computers and at least of cheap workstations.

The availability of standard graphic software and the capability to create networks are the two features which make a workstation a practical and cost effective way to provide a universal environment for the development of operator interfaces. The availability of powerful hardware and software standards makes straightforward the setting up of a network where hardware and software resources can be easily shared in a really cost-effective way. Moreover, the development of more and more powerful Personal Computers IBM-like or Apple and the establishment of wide network strategies which include them, claims for the integration of these computers as dedicated units for less demanding tasks.

The Milan computer control

The first tests on the magnetic field of the Milan Superconducting Cyclotron were carried out since November 1988 till March 1989. During this period the control structures of the equipments involved in the operations were tested too and some experience was gained. Main hardware and software choices have been proved satisfactory but the flexibility of the system was not so good as we expected, particularly for the operator console. Taking into account all the considerations we have made so far , we decided to develop a general purpose structure of console with enhanced performances and flexibility. It was decided also that the project and the development of most of the software would take not more than 2 man-years of work .

Some general rules were fixed for the project:

- the architecture must be independent from the number of workstations in use : adding or removing a workstation must be invisible to the system. This is the safest way to provide an "easy expandible system".
- the architecture would allow to have the same workstation connected both in the main control room both in a remote place close to the accelerator.
- the architecture of the consoles must be fully independent from the lower levels : the choices made in process and plant levels must not influence the supervisor structure.
- the presentation level of the software must not require any practice in computer science and must be icon driven. The operator must be able to define its own working environment with a few choices and to access every information he want to deal with.
- the software configuration must be guaranteed by dedicated computers and must be allowed to change particular parameters without have to stop the system.
- the allarm and malfunctioning logging task must be performed by a dedicated unit able to provide particular tools to help the operator in his troubleshooting job.
- maintenance of the whole structure must be easy and centralized as much as possible on a single machine.

The choice of the final solution was not easy and a lot of different considerations, as estimated technical support available from Italian vendors, experience with similar hardware platforms and operating system software, availability of man-power, were taken into account.

At last we decided to base our hardware architecture on a Local Area Vaxcluster of 3100 Vaxstations with a microVAX 3100 as boot member. The gateway toward the other control networks was provided using a microVAX II where a real time picture of the "virtual accelerator" is present. Two 80386 PCs were dedicated to allarm logging and to configure the data necessary for the application tasks.

Hardware architecture

The main console has been designed as an Ethernet segment with a distributed software running on the resources which are connected to it. The Ethernet segment is physically the same which interconnets the proces control stations but we have superimposed on it the Local Area Vaxcluster(LAVC) Digital protocol. This particular choice reflects the experience made by ourselves using LAVC scheme in computer rooms. This software is quite reliable and is well tested in a lot of installations all around the world. The structure resulting is easy to maintain as it recalls centralized architectures but it is really flexible and able to suit growing requirements. The intrinsic problem to have a central machine which will paralize the whole system in case of a failure, has been considered but the experience with similar architectures reports a very low failure rate.

The boot member is dedicated to the management of Cluster operations and of print and tape resources. Historical logging of all the variables acquired or calculated by the control system is provided by the microVAX II, which is a satellite in the LAVC. The requirement of linking the control room structure with the process level network is not a trivial one, as the two data link protocols are quite different and there is no reported experience on such a job. We decided to meet this need and to provide the functional decoupling of the two structure at the same time implementing a gateway. A hardware connession between a Multibus I based cardcage, connected to the process Ethernet, and the microVAX II makes a segment of memory on Q-bus available to the Multibus processors as local memory . The channel is really performant and we have reported an error rate of less than $1x10^{-9}$. Similar connessions, based for example on DMA transfer mechanisms, were satisfactory too but require some more computing power and an higher error rate was measured during operations.

The previously mentioned Multibus I cardcage provides another capability : a local task, running on a 80286 CPU, updates a network of TV black/white monitors which distribute accelerator informations all around the plant at a very low cost.

Powerful 3100 Vaxstations with 16" Trinitron color monitors have been chosen as the hardware platform for operator interaction tools. All the workstations, whose number is completely unrelated to the architecture of the console, share the same programs and are able to work on every access point to the coaxial Ethernet cable. This is the true sense of the statement that "the console is a network".

Two particular nodes on the control room Ethernet are two 80386 based PCs. The first one is dedicated to run a database application used for the storage of all the relevant parameters of each sensor and actuator of the cyclotron. Data are organized according to different query schemes to provide an interactive tool for everyone needs a particular information. A particular user of this program is the console itself : the data section of the applicative programs refers to this database and every change in the database is immediately available to the operator workstations taking advantage of the possibility to share a disk between DOS and VAX processes. The second PC provides a lot of tools to help the operator during allarm handling and troubleshooting operations. This computer was conceived as an hypermedia machine, with enhanced graphic capabilities for the display of pictures or drawings related to a particular event and with the possibility to play back defined speeches as an auxiliary tool. Both the PCs are dedicated machines, mouse driven, and they have been chosen for their high performances, low cost, easy programming and integration in the Digital network architecture.

Software architecture

The most relevant challenge in the development of a new, flexible and "portable" console is in the software design. The whole hardware architecture described in the previous section would be meaningless if it would not be provided an adequate software support.

The main choice in the software architecture has been that to use X-Window as the basic element. X-Window is a network oriented windowing system for bitmapped graphic displays developed at MIT. It arises from the need to allow not only data but also application sharing between different computers on the same network . The mechanism which has been developed to fulfill this requirement provides for a complete logical separation between the application (called the client process) and the operator interface (called the server process). The server may be local if the two processes works on the same machine, or it may be a network resource. X-Window is based on 3 software levels : the client uses the Xlib library to produce graphical outputs which are converted according to the rules of a particular protocol, called X-Protocol, before of being sent to the server, where they are decoded by the X-Server process. X-Server translates the received packets in a format suitable for the particular device in use and therefore is the only device dependent part of X-Windows.

Taking as a reference the software model proposed by X-Windows a modular, object oriented code has been developed : **GIULIA** (Graphical Interactive Unit Level for Improved Automation)[3].

GIULIA has been designed to take full advantage from Digital architecture; it is based on DEC-Windows, a Digital extension of X-Windows mainly concerning the availability of a powerful toolkit and of a User Interface Language (UIL) able to define the structure of the graphical interface. Object description is stored in a separate file where the methods to which the objects will respond are outlined. The necessary code for the implementation of every method described in the UIL file is contained in a second file. Such a structure enhances the separation between form and content of an application. DEC-Windows is based on X-Windows so it is possible to use high level DEC provided calls and low level Xlib functions in the same code.

Another important choise has been that of the operating system to use for GIULIA. The natural one, which seems to fulfill portability requirements, was to use UNIX. Nevertheless we had no experience on it while a lot of work has been done using VMS. Moreover we extimated that the use of operating system services would be limited to the handling of the shared memory, while all the window functions are source compatible using both UNIX and VMS. These considerations induced us to start working with VMS and, eventually,to change to UNIX later. There were no doubts on the choice of the language being "C" the most suitable for such applications.

GIULIA starts on power up of an operator workstation and is composed by two main parts running on at least two different computers : the first part provides man-machine interaction tools while the second one is dedicated to the managenement of accelerator related data. When a workstation is connected to the control Ethernet GIULIA proposes to establish a connection to the whole control system. A positive answer by the operator starts a remote task on the microVAX II which complete a transparent task to task scheme for exchanging data between the remote workstation and the rest of the networks. Data exchange has been optimized to reduce unnecessary traffic on the network and to simplify the handling of the shared memory. This segment, physically housed on a board inside the microVAX II, has been structured as a global section shared between the GIULIA activated process and a process which updates the section records according to the packets received from the process stations. Tasks running on remote workstations look at the control hardware in an indirect way by means of this section. This is the reason of the portability and of the device independence of GIULIA. The bandwith measured on the network channel is of nearly 1.1 Mbit/sec for 2.5 Kbytes packets and it quite agrees with the performances of the other control networks.

The interaction tools in GIULIA are based on the definition of *a class*, called *rappresentation*. Main attributes of this class are "true" resizing and iconic interface. Related to these graphical attributes some other functional attributes have been defined, as the capabilities to modify on-line different characteristics of the parameters being in use. Rappresentation has 3 *subclasses : table, bar chart and graphic.* These subclasses identify the different ways into which one may display every parameter of the cyclotron. The environment provided to the operator for his job is like that one of a writing-desk : foils,particular rappresentations, are distributed on the screen and the displayed parameters have been chosen during a short "navigation" inside the program by the operator itself according to his preferences. General schemes have been defined and are available as a preliminary view into the system. Foils may be opened, iconified, deleted, reconfigured. Graphical attributes of the rappresentation subclasses may be changed easily thanks to interactive setup utilities or modifing the UIL description file.

The data section of GIULIA is based on files obtained by the database program running on a PC. This reduces the complexity of the software and guarantees a longer lifetime to GIULIA. A process in GIULIA checks all variables for allarm or safety limits, and when a threshold is exceeded all the related informations are transferred to the second PC for further dedicated analysis. The operator may go into deep details taking advantage of immediately available drawings or of speech training on some specialized items. The experience gained may be used to improve the quality of the services from this computer.

GIULIA provides an extensive on-line context sensitive help explaining the way it works and all the details concerning the object classes it implements. Moreover, all the operations allowed by the menu in GIULIA are software protected to prevent malfunctioning and warning messages are issued to the operator if some sort of mistake has been done during navigation. Some choices, which may be dangerous as the modification of an allarm limit by the operator, are always remembered in the display of the workstation and are effective only for that is related to the display.

Conclusions

The goal to develop both the structure and the software so far discussed in one year has been reached. Preliminary tests have been carried out in order to verify the overall architecture and all seems to work according to the design requirements.

The development environment has proven valuable and the experience gained on DEC-Windows has encouraged us to start a further enhancement of GIULIA to include a new class, which we call *interaction*, and new tools for interactively building synoptic views of accelerator components. The time estimated for these developments will be of nearly 8 man-months.

First tests on the Cyclotron in Catania will be carried out using GIULIA as an advanced interactive tool.

References

1) P.G. Innocenti - "The LEP Control system" - International Conference on Accelerator and Large Experimental Physics Control Systems - Vancouver, Canada, November 1989

2) M. Mignacco - "The ELETTRA Control System" - International Conference on Accelerator and Large Experimental Physics Control Systems - Vancouver, Canada, November 1989

3) D. Giove - "GIULIA : a User-Friendly Tool for Accelerator and Experiment Control" - INFN Internal Report