Experience and Trends in Control System for Particle Accelerators

G. Cuttone, A. Rovelli
INFN-LNS, V. S. Sofia, 44 - 95123 Catania - Italy
D. Giove
INFN-LASA, V. F.lli Cervi, 201 - 20090 Segrate (Mi) - Italy

Abstract

The architecture of a control system for particle accelerators has to be tailored according to a lot of requirements arising from the particular environment in which the machines work. In this paper we will discuss actual and future trends considering the experience gained designing and commissioning a control system for the heavy ions facility at LNS. Particular care will be devoted to investigate the architectures which mainly require limited investment concerning budget and man power. The growing role of software and graphical programming will be discussed in detail.

1. INTRODUCTION

The control system has a central role operating a particle accelerator and represents one of the major items in manpower and budget investment. This implies that the basic architecture has to guarantee a flexibility and modularity to allow any modification during its lifetime and maintainance has to be carried out without any interference with accelerator operations. A distributed control system represents an ideal answer to these demands and may be easily implemented using a lot of different commercial standards products nowadays available. The lifetime of a technical solution is one of the more important problems designing and maintaining a control system. Taking into account both evolution and economic budget it is hard to define an architecture which will guarantee more then 3 years of interest. The availability of commercial products in the high-tech market (high resolution analog boards, large multiplexer, etc) and in the consumer market (computers, high quality mass storage and peripherals) forces the designer to implement the rules defined in the general project according to this point of view.

We followed these general considerations designing and realizing the control system for the heavy ions facility at LNS, based on a K=800 Superconducting Cyclotron as booster of a 15 MV Tandem. The cyclotron and all the related beam transport lines are totally computer controlled. The first beam, delivered by the Tandem and radially injected in the booster, has been recently accelerated. The experience gained commissioning the acccelerator control system will be presented and discussed mainly looking to hardware and software choices and the results obtained. New trends and possible improovements will be reported.

2. CONTROL SYSTEM OVERVIEW

The heavy ions facility computer control system has been designed as a distributed one. It follows a three level architecture and the intelligence has been distributed following a functional scheme (i.e. similar equipment or complete subsystem is under the control of a single computing unit) both for the compact size of the cyclotron and for the limited dimensions of our laboratory. At the field level simple intelligent units directlyi nterface each device allowing the implementation of command and data acquisition prrocedures. These units are interconnected to the higher level by means of a low cost serial bus. The stations, representing the process level, coordinate the functioning of each subsystem and guarantee the communication with the console. Process stations and operator console are interconnected by means of a local area network (LAN). The presentation level is implemented as a set of graphic workstations for man-machine interactions.

The main features of our control system are: the capability of driving accelerator and beam lines toward a predefined state maintaining it stable, a central console with enhanced graphic capabilities to allow an easy and flexible man-machine interaction and the local station for each accelerator subsystem for servicing and troubleshooting.

The control system is now experiencing its second stage of revision. The first one, which allowed the first cool-down of the superconducting coils and the first magnetic measurements (Milan , 1988), was based on an extensive home-made work. All the levels so far described were developed in the laboratory, and the control group worked as a research OEM in different fields of interest (optical interfaces, high resolution boards, industrial computers structure, multiprocessor operations, etc) . The experience gained in the first period of operation and, particularly, the evaluation of the efforts and the costs necessary for the completion of the whole control system suggested us a major consideration about the subject. The project was reconsidered according to the man-power availability and budget constraints: the process level and the main console were completely redesigned for the commisioning of the cyclotron in Catania.

We are now evaluating a less dramatic revision after the first year of beam operations. It will be aimed to reduce the number of channels managed by the computer control and to introduce specialized consoles based on LABVIEW software.

3. CONTROL SYSTEM HARDWARE

The field level of a control system may be realized using different commercial solutions. In our control system it has to be not only a physical interface to the devices but it has to have the capability to implement control algorithms too. Moreover the analog and digital signals treatment has to be as close as possible to the devices and a high speed bus has to permit a fast data transfer to the local controllers. We found a good compromise for our specifications using intelligent boards based on an Intel 8044 microcontroller interconected by means of BitBus. It is a high speed (375 Kbit/sec) asynchronous serial bus able to connect up to 28 different boards (nodes) in a hierarchical protocol. A real-time, multitasking operating system (iRMX51) is part of the onchip firmware in the 8044. The standard trasmission media is a 2 wire twisted cable at a maximum distance of 300 m. but an optical interface module allows to overcome distance limitations and to increase noise immunity has been developed by us.

At the beginning process level stations have been designed as a Multibus I multiprocessor system[1]. According to the industrial philosophy all the software was loaded on EPROM and there were no supports for on-line debugging at source level and for the operator interface. 80286 based boards were choiced as general reference. The major drawback in this design was the lack of a confortable debugging environment. This was more and more unfeasible because we had to test both the home-made software and the electronics of the different equipment interfaces. During the first revision we changed completely the basic design and we choiced to use Personal Computers. The availability (less then half price!) of more powerful processors compensate the lack of real-time operating systems.

The control stations and the console are interconnected by means of a LAN. For the requirements of our accelerator facility an Ethernet LAN is satisfactory considering its bandwith (10 Mbit/sec) and the limited number of stations we have on the network. Moreover a severe synchronization is not a request for us both for the structure of the accelerator and for the choice to solve the time critical problems at the field and control level. The non-deterministic time access to the network due to the CSMA/CD technique may be considered a not significant drawback for us.

The console realized in the first revision [2] based on the same multiprocessor structure used for the control stations with high cost specialized graphic boards, showed us a flexilbility not so good as we expected. Considering our requirements in terms of complexity and operator interface units (at least three), our experience, the availability of standard graphic and network communication software we decided that workstations are a pratical cost effective way providing an universal environment for the development of the presentation level. For its design we fixed some general rules. Its architecture has to be independent of the number of workstations in use, the worksites and the structure of lower level. Full operations of each facility subsystem have to be posssible by each workstations. An easy configuration of the parameters has to be guaranteed as well as high level alarms handling tools has to be available for an easy troubleshooting.

We found a good compromise for all these requirements designing the console as an Ethernet segment with distributed software running on the resources connected to it [3]. We implemented this hardware architecture using a Local Area VaxCluster (LAVC) of Vaxstation 4000/60 with a 4000/90 as boot member. The intrinsic problem to have a central machine which will paralyze the whole system in case of failure has been considered, but our experience gained in similar applications used for computer rooms reports a very low failure rate.

The local control stations are physically on the same Ethernet backbone. The Vaxstations are the hardware platforms for the operator interaction tools; they share the same applications and are able to manage each control stations accesible via network. Particular elements of the console are two PCs 486, connected via Ethernet to the DEC machines, managing the off-line accelerator parameters database, alarm report and troubleshooting operations.

3. CONTROL SYSTEM SOFTWARE

The first software architecture followed the hardware choices. The operating system at process and console levels was an industrial real time one (INTEL iRMX86) and the programming language was PL/M 86. Process level network was designed and written by us using an INTEL library (iNA 960). Programs for input and output devices (including graphical synoptics) were written in "quasi" machine language.

The first revision involved nearly 90% of the developed and used software. The leading idea was to follow as much as possible DEC specifications for network and workstation software. At the process level MS-DOS has been adopted guaranteing the use of available programming standard tools. Performance and architecture limits of MS-DOS have been taken into account, but the presence of the field level allows to overcome these problems.

The process level network has been designed as a set of task-to-task connections between PCs and the console according to a client-server model. Decnet libraries have been used and new functionalities have been added by us to fulfil the particular requirements of a completely automatizated system (connection open, repair in case of failure, avoiding network dead lock). 'C' language has been adopted as standard for the two higher levels. At the presentation level we adopted VMS as operating system and Dec Window Motif as graphical programming environment. The choice of VMS has been quite a conservative one, and it was mainly due to our experience and to the available network products at that time.

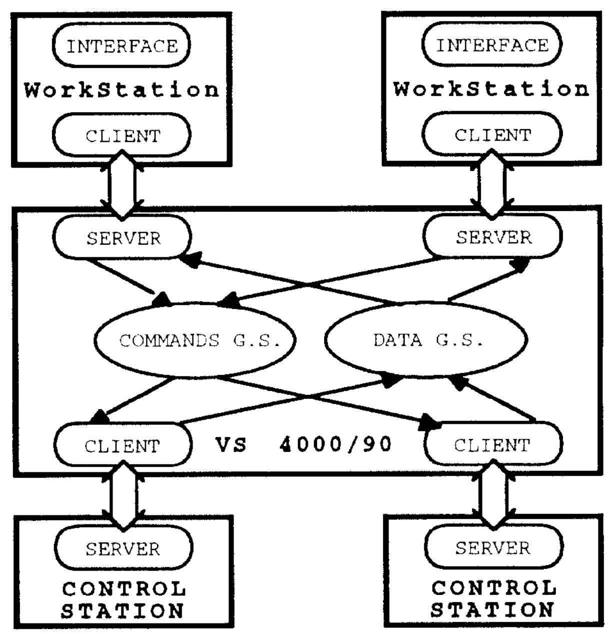

Figure 1. Control system software architetcture.

Taking as reference the software model proposed by Dec Window a modular object oriented code has been developed: GIULIA . Its features has been extensively reported elsewhere [3] Giulia is composed of two different part running at least on two different machines: the first is dedicated to the management of the accelerator data while the second provides the man-machine interaction tools. In fig. 1 the software architecture is shown.

The data management part consists of a server task running on a VaxStation 4000/90 created by the client task running on the operator workstation; they realize a transparent task-to-task communication between these two machines. Two different VMS Global Sections created on the VaxStation 4000/90 contain the accelerator data received from the control stations (Data Global Section) and the commands to be sent (Command Global Section). Tasks running on the operator workstation look and actuate the accelerator facility in a indirect way through these sections, making the operator software really device independent. They are based on three different classes: representation, interaction and beam diagnostics. Representation has three subclasses (table, bar chart and time diagram) identifying the different way to display each parameter. Interaction has three subclasses (button, knob and input box) identifying the the different ways to operate each parameter. Beam diagnostic is a particular class realized to display the beam current distribution acquired by the Beam Profile Monitors and calculate the main physical beam parameters.

Table 1
LAN Performances

CLIENT		SERVER		
PathWorks 4.1			tran/s	Mbit/s
VS 4000/60	↔	PC 386SX/20	19	1.2
VS 4000/90	↔	PC 386SX/20	22	1.3
VS 4000/60	↔	PC 486DX/33	57	3.3
VS 4000/90	↔	PC 486DX/33	61	3.5
VS 4000/60	↔	PC 486DX/66	61	3.5
VS 4000/90	↔	PC 486DX/66	69	4.0
VS 4000/60	↔	VS 4000/60	50	2.9
VS 4000/60	↔	VS 4000/90	50	2.9

The performance of our control system are strongly dependent by the LAN communication performance. Table 1 presents our measured performance tests. Each transition is composed by 2 Kbyte of Commands Packet sent to the server task on a control station replying 5 Kbyte Data Packet to the client task. In this configuration we can send up to 50 commands per second from the console and we have an overall update rate of the accelerator facility of about 3 Hz.

4. FUTURE TRENDS

Before the end of 1994 the first extracted beam tests has been scheduled. According to this time table we plan to test all the different accelerator subsystems. The parameters number nowdays controlled and the quality of control algorithms are strongly influenced bv the need to gain experience and useful information by them. After the first functioning of the whole facility we are going to revise this situation in the aim to semplify control schemes. At the same time using the knowledge so far gained we plan to close in an automatic way control loops. From the point of view of the control architecture we are also going to move the actual philosophy from DEC to a PC based structure. The network at the process level will be a Novell one with a support for TCP/IP and task-to-task communications. Process level stations will be modified to disk-less PCs, client of the Novell server. Remote boot capability and hot reboot of the control software will be provided helping maintenance and adding flexibility to the system. The data structure image of the facility situation, will be organized in a true database in the Novell server, replacing DEC machine so far used. A lot of attention will be devoted to the development of small data acquisition and analysis units, dedicated to troubleshooting management. These units, which may be conceived also as specialized consoles for particular equipments, will work in a parallel way with respect to the control system architecture. LABVIEW has been chosen as the basic environment for these units. The experience which we will gain using LABVIEW will be valuable in the evaluation of this environment for a redesign of the main console software

3. REFERENCES

[1] F. Aghion et al., "The Distributed Control System with decentralized acces to an optical bus for the Milan Superconducting Cyclotron", in Proceedings of Int. Conf. on Cyclotron and their Applications, Tokio, Japan, 1987, pp. 428-431

[2] F. Aghion et al., "The Operator Interface for the Control System of the Milan Superconducting Cyclotron", in Proceedings of Int. Conf. on Cyclotron and their Applications, Tokio, Japan, 1987, pp. 432-435

[3] D. Giove et al., "The Operator View of the Superconducting Cyclotron at LNS", in Proceedings of Int. Conf. on Accelerator and Large Experimental Physics Control System, Tsukuba, Japan, 1991, pp. 131-133

[4] G. Cuttone et al., "Operational Experience with the Control System of the LNS Superconducting Cyclotron", IEEE Trans. on Nuclear Science, vol. 41, pp. 188-193, Feb. 1994