A centralized Control System for the DAΦNE Φ Factory

G. Di Pirro, C. Milardi, M. Serio, A. Stecchi, L. Trasatti.
INFN, Laboratori Nazionali di Frascati
P. O. Box 13, 00044 Frascati (Rm) ITALY

Abstract

We describe the control system for the new DAΦNE Φ-factory under construction at the Frascati National Laboratories. The system is based on a centralized architecture for simplicity and reliability. A central processor unit coordinates all communications between the consoles and the lower level distributed processing power, and continuously updates a central memory which contains the whole machine status. This memory constitutes the machine database prototype. A simple message passing scheme built on a system of mailboxes takes advantage of high speed busses and of Fiber Optic interfaces. Macintosh II personal computers are used as consoles. The lower levels are all built using the VME standard.

I. DAΦNE

The DAΦNE accelerator complex [1] of the INFN Frascati National Laboratories consists of a two ring colliding beam Φ-Factory and of a 510 MeV e^+/e^- injector for topping-up. (See Fig. 1).

The project has been approved by the INFN Board of Directors in June 1990 and the engineering design has started in January 1991. Construction and commissioning is scheduled for the end of 1995.

The luminosity target is $\sim 10^{33}$ cm^{-2} sec^{-1}.

The main features of the Φ-factory are :

- electrons and positrons circulate in two separate storage rings and collide at an horizontal half-angle $\theta_x = 10$ mrad (in one or two interaction points) in order to achieve high collision frequency without parasitic crossings;
- the novel design of the magnetic lattice is a 4-period modified Chasman-Green type, with a 1.9 Tesla normal conducting wiggler magnet inside the achromat.
- a crab-crossing option is contemplated (if needed).
- injection is obtained through a LINAC and an accumulator damping ring.

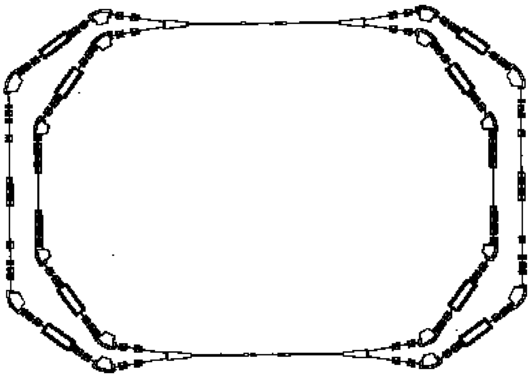


Fig. 1 : The DAΦNE Φ-Factory layout

II. SYSTEM STRUCTURE

Fig. 2 shows the general architecture of the control system. Three levels are defined:

PARADISE (PARAllel DISplay Environment) is the top level, implementing the human interface. Several consoles, built on Macintosh II personal computers, communicate with the rest of the system through high speed DMA busses and fiber optic links.

PURGATORY (Primary Unit for Readout and GATing Of Real time Yonder) is the second and central level of the system. It essentially contains only a CPU and a Memory in a VME crate. The CPU acts as a general concentrator and coordinator of messages throughout the system. This does not constitute a bottleneck, since we use very high speed busses (MacVee[2]) and fast fiber optic links. A measurement of the throughput of these busses [3] shows performances more than one order of magnitude better than using a conventional LAN. Using a polling mechanism the CPU also checks the lower level units, which relay significant changes in the machine situation. The updated information is stored in the central memory, where it is accessible from the consoles and it represents the prototype of the machine database.

HELL (Hardware Environment at the Low Level) is the third level of the system and it is constituted by many (about 60) VME crates distributed around the machines. Each crate is equipped with at least one CPU which performs control and readout of the related elements in the machine. Only significant changes in the parameters are transferred to the Purgatory, thus hiding useless information from the central processor. A first estimate of the system gives about 7000 channels to be controlled.

Centralized Control

In a system with a large amount of distributed processing power there is a choice on the control configuration: distributed, where every peripheral CPU is autonomously responsible for accessing the central database, or centralized, where all communications run through a central control unit.

We chose centralized control for two orders of reasons: reliability and simplicity.

A central control system is more reliable than a distributed one, at least in an environment such as ours, where the whole apparatus to be controlled is contained in a single building or cluster of buildings. This is a very different situation from that of a network which has to keep functioning, at least partially, under adverse breakdown conditions. Our central CPU will have a second unit acting as warm backup, capable of intervening very rapidly in case of failure.

Moreover, this architecture generates a system which is very simple to maintain and to control, where the failure of a peripheral unit can be diagnosed and isolated very efficiently.

Polling Mechanism

We have chosen a polling mechanism instead of the usual interrupt structure again for reasons of reliability and efficiency. Contrary to widespread belief, interrupts are not very fast (typical 50 µsec + software for a MicroVax), and they increase the system indeterminacy, thus making debugging and failure detection much more complicated. Since we will have a rather large system (recent estimates run up to 60 VME crates), system and software complexity should be kept to a minimum.

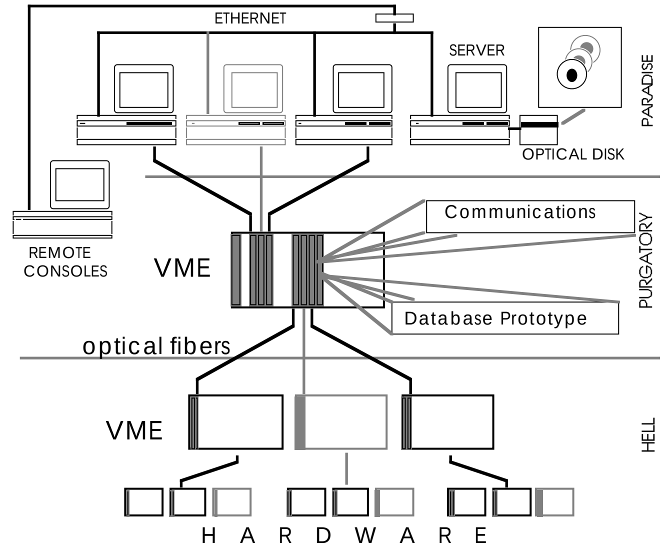


Fig. 2: Control System Schematic Diagram

III. THE HARDWARE

VME has been chosen for both Purgatory and Hell. The ease of implementation of multiprocessing, the large amount of modules commercially available, the low cost make it the obvious choice in this moment.

Communications between levels are implemented through high speed parallel busses where distance and noise considerations allow it. On the other hand, since we must control a complex cluster of machines (LINAC, Booster Ring, Storage rings and transport lines between them), distances are greater than the 70 meters typically allowed by parallel busses. Therefore we are developing a VME module, VINETA, capable of 125 Mbit/s serial transmission on optical fibers. VINETA is a simple passive point to point optical link acting as a VME slave. It is capable of supporting the maximum throughput of a standard VME CPU, offering output in a 2048 word FIFO.

IV. CONSOLES

Macintosh II family personal computers have been chosen for the system consoles. In the last few years we have seen an impressive effort by personal computer firms and third parties to supply large quantities of very high quality software at very low prices. The situation is now, as far as software is concerned, definitely in favor of the use of large diffusion machines as opposed to high cost, "high" power, low diffusion workstations. Hardware prices keep getting lower, while the cost of software development has reached about 80% of the total cost of an installation, with all the reliability risks of in-house software development. The Macintosh family of computers is at the moment the best candidate for a human interface development, since the effort expanded on software development on this machine has been the most striking on the market. On the other hand, hardware performance keep increasing; a 68040 machine has already been announced, and work is apparently in progress on a 88000 Macintosh. One can be confident that the hardware power will keep increasing and that our present Macintosh IIfx will be a very slow machine by the time DAFNE will be commissioned.

Previous experience with Hypercard [4,5], on the other hand, has shown that high level software packages can decrease software development times by strong factors. Faster and more powerful human interface packages are coming out every day. Prototyper 3 is an example. You design the windows, buttons and menus interface graphically, and the program generates the corresponding C (or PASCAL) code, with plug-in modules for the specific requirements of your application. After all, writing over and over again the same 100 lines of code to open a window is a definite waste of time and effort.

V. VME OPERATING SYSTEM

In our previous experience with a similarly structured control system we used no operating system for the lower level CPUs. Simple FORTRAN or C programs took care of the relatively easy tasks of a small and dedicated CPU which only has to perform a few simple tasks. The general idea is still:"A CPU for each task". While this is a rather extreme statement, we think that the software environment for the lower level CPUs must be kept as simple as possible, possibly at the expense of increasing their number. On the other hand, the advantages of using a standard environment are obvious as far as bookkeeping and standardization are concerned. At the moment we are evaluating several Operating Systems and we plan to reach a decision at the end of this year.

VI. MARCO

One of the main problems that are facing us is that of human interface. This does not only mean elegant windows, nice buttons and full menus; it also involves presenting the machine operators and physicists with a coherent interface toward both the real operation world and the simulation and modeling programs.

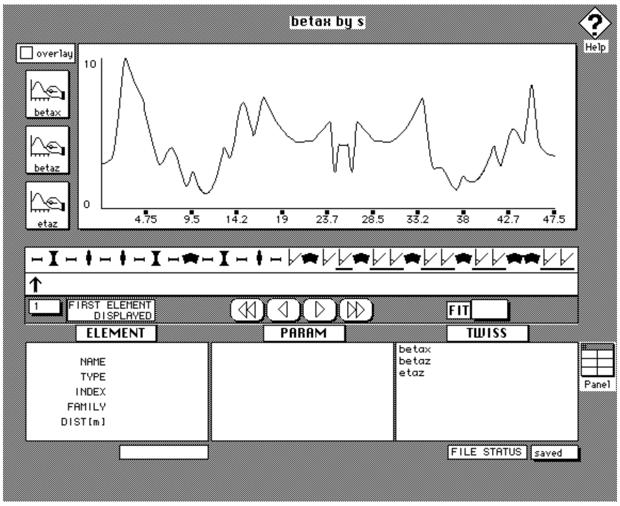
In this framework we have started to develop a common human interface toward the standard simulation and modeling programs that machine physicist use on standard computers (VAX, IBM, etc.). The idea is that it should be possible to predict the effects of a change of parameters of the machine before actually trying them out, and to run an optimization program, without leaving the human interface environment of the control system. It has been proven several times that a better human interface is not only helpful in carrying out a task, but generally allows a broader view of the problem at hand, allowing better control of the situation.

MARCO [4] is a first attempt to build a common interface and a graphical machine representation format. It has been connected for the time being to two lattice design codes: LEDA and COMFORT. Hypercard has been used as programming environment. Fig. 3 shows the machine element display and histogram window of MARCO.

VII. AN OPEN SYSTEM

The development of MARCO has shown another important requirement of a modern control system: openness. While some programs, like LEDA, can be translated to run on a Macintosh, the problem is much worse for big and old programs like COMFORT. Therefore we developed a set of translators and of communication utilities to run the human interface on the Macintosh, while the actual simulation code was run on a VAX machine, without any operator intervention and without changing the human interface.

We believe this to be a very important capability for a control system. We live in an environment where several computing elements will have to coexist: Personal Computers of different flavors, Mainframes with their large baggage of written and debugged code, very powerful and cheap RISC machines in the VME standard, and so on. It should be possible to the user to get the results from all possible worlds without having to live with several user interfaces.



FIRST ELEMENT DISPLAYED: 1

FILE STATUS: saved

Fig. 3. Machine element display and histogram window for MARCO.

All this will require a really open system. We are planning to install Ethernet cards on all the consoles, linked to the storage medium and, through a bridge, to the rest of the world.

In this respect, we will use standard databases to collect data and to standardize the machine operational and structural aspects. We shall use both ORACLE for its high diffusion in the controls world and 4th Dimension on the Macintosh for its ease of operation and affordability.

VIII. SUMMARY

The control system we are building is based on highly distributed hardware and software capabilities, with a strong accent on openness to other environments. Previous experience with smaller systems tells us that these are the most stringent requirements we will have to face, and that the human interface will be the most arduous problem to solve. The use of modern software techniques is a big help in this direction.

IX. ACKNOWLEDGEMENTS

We would like to thank the Accelerator Group of the LNF for continuing discussions and encouragement. The work with the LISA group helped us develop a set of techniques that will be very useful in the future.

VI. REFERENCES

[1] G. Vignola et al., talk presented at the San Francisco Particle Accelerator Conference, May 1991.

[2] B.G. Taylor, "The MacVEE Hardware User Manual"; Rev.4.1, CERN EP Division, 1987.

[3] L. Catani, G. Di Pirro, C. Milardi, L. Trasatti, F. Giacco, F. Serlini "The High Speed Bus System for LISA" Presented at the International Conference on "Accelerator and Large Experimental Physics Control System" Vancouver, Canada, Nov 3, (1989) *In Press to Nuclear Instruments and Methods.*

[4] L. Catani, G. Di Pirro, C. Milardi, A. Stecchi, L. Trasatti: MARCO - Models of Accelerators and Rings to Commission and Operate, presented at the San Francisco Particle Accelerator Conference, May 1991.

[5] HyperCard v2.1, Apple Computer Inc,©1987-90.