

THE EVOLUTION OF THE DAFNE CONTROL SYSTEM: A HISTORY OF LIBERATION FROM HARDWARE

G. Di Pirro, G. Mazzitelli, I. Sfiligoi, A. Stecchi, INFN-LNF, Frascati (RM), P.O. Box 13, 00044 ITALY

Abstract

During the DAFNE [1] commissioning and run operations the Control System [2] has been continuously evolving in order to fulfill the user requirements and the needs of a complete accelerator management. The original structure of distributed CPUs relaying to a central shared memory proved to be scalable and suitable for adding functionality 'on the fly'.

After 5 years of operation, the system had reached some intrinsic limits so that we focused on an upgrade plan mainly regarding the user level and the capability of the system to connect to external resources.

The re-engineering was done thinking to the maximum software reuse and the original choice of a commercial software environment for all the control applications demonstrated to be valid allowing to redesign the user level with no worries for the porting.

1 DESIGN STATEMENTS

When we started the Control System development we had a large amount of devices (Tab. 1) and a few people involved in the job.

Table 1. Device List

Device	Quantity	Interface Type
Magnet Power Supplies	425	Serial
Vacuum Pumps	157	Serial
Beam Position Monitors	123	MUX,DVM
Vacuometers	43	Serial
Fluorescent Flags	23	I/O
Ion Clearing Electrodes	39	ADC,DAC,I/O
Beam Loss Monitors	31	Scaler
Kickers	10	ADC,DAC,I/O, GPIB
Video Multiplexers	10	Serial
Scrapers	12	Stepper Motor Control
Programmable Delays	7	GPIB
RF Cavities	3	ADC,DAC,I/O
Beam Current Monitors	3	DVM

In order to optimize the development time we decided to use commercial technologies as much as possible. A commercial product is characterized by a

broad distribution, which means a lot of feedback from the users and, consequently, deep debugging. Furthermore the wider the distribution of a product, the more reliable is its support from the producer.

Another criterion was to provide "easy development and maintenance".

We decided to use:

- LabVIEW [3] as the development environment for all the software;
- industrial VME bus to house the front-end hardware.

2 SYSTEM GENERAL DESCRIPTION

We distributed VME crates all around the machine in order to have the front-end hardware close to the devices to be controlled. At that time, LabVIEW was available only for Macintoshes so we decided to develop a customized processor based on the Macintosh LCIII mother board as a VME controller. The result was a fully operative Macintosh 68030 computer, joint to an interface, performing the VME System Controller functions.

The distributed CPUs make up the system 3rd level where the applications dedicated to device handling and control reside.

All the CPUs run asynchronously and write into their own VME memory the result of the control tasks for all the devices of which they are in charge. The data refresh time ranges from a few Hz to 50 Hz depending on the number of devices, the interface type and the complexity of the process.

From a data point of view, the system 3rd level consists of several local memory pages where all the machine objects are represented with descriptive records continuously updated at their own rate. All 3rd level VME crates are connected to a central cluster of VMEs through point-to-point optical links to constitute a central common addressing space called 2nd level.

The end result is a virtually central memory where the machine RTDB (Real Time Database) resides.

The transaction from the 2nd up to the 3rd level is transparent to the user, that can fetch any descriptive

record from a remote memory page with a simple VME read cycle. These read actions are performed from the system 1st level where the consoles and the user applications reside.

3 SYSTEM IMPLEMENTATION

In the first system version we used Macintoshes 68040 as consoles. The connection between the operator's consoles and the VME was done by dedicated VMV [4] interfaces.

The system (see Fig. 1) main peculiarities were:

- "true" memory mapping of the central virtual memory into the consoles internal address space;
- uniform hardware and OS (Macintosh at any level).

The DAFNE commissioning started and continued until December 1999 with this setup.

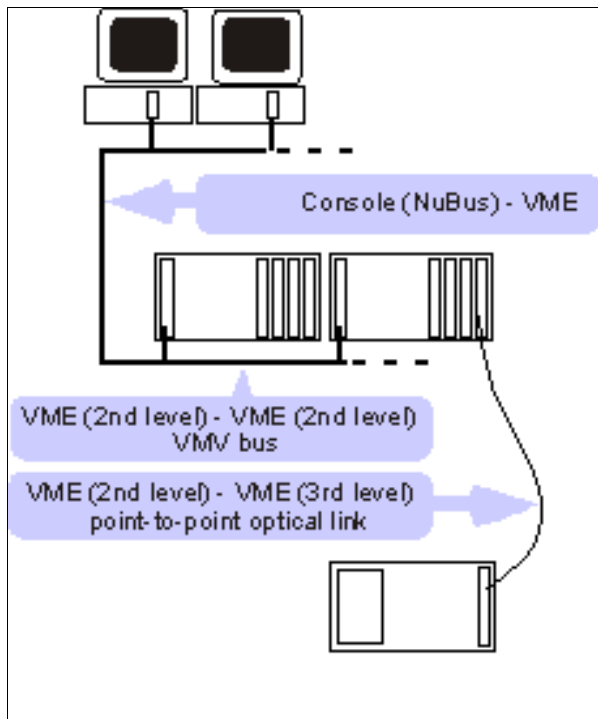


Fig. 1 Original System Implementation

4 SYSTEM UPGRADE

After 5 years of operation the system general structure based on distributed CPUs and a central shared memory demonstrated to be valid with no limitations from the hardware. It allowed easy and fast data gathering and correlation.

Also, the distributed processors were shown to be suitable for the front-end tasks, hence we focused on the 1st level for the upgrade project.

The operator level had reached some intrinsic limits:

- the 68040 μ P was no longer able to stand the load of always-heavier requirements ;
- the Macintosh NuBus platform was dismissed.

We set the following targets:

- to improve 1st level reliability and performance;
- to get rid of the consoles connection bus and therefore the limitation on the number of consoles;
- gain remote access on the consoles;
- to have Internet media and services fully available.

An obvious issue was to reuse, as much as possible, the software already developed and this imposed the requirement to adopt computers able to run LabVIEW.

We chose Sparc VME embedded computers by FORCE [5] with Solaris operating system instead of Macintosh consoles. The first benefit of using VME embedded processors was to get rid of all bus-to-bus interfaces and cables needed to access the RTDB.

First, we re-wrote the VME read/write basic routines and then we encapsulated them into conventional LabVIEW graphic nodes. After this, porting all the user applications in LabVIEW for Solaris went on smoothly and required just a little cosmetic make-up and a few minor adjustments.

We estimated to load up to 5 user sessions on each Sparc CPU. We installed four diskless FORCE computers and a Sun Enterprise 250 as server over a switched 100 Mbps Ethernet network.

The interaction with the user applications running on the VME processors is done by mean of *SunRay* [6] lightweight terminals. These terminals are centrally managed by, and draw their computing resources from the *SunRay* server software that runs on the Enterprise 250.

The system performance, concerning the graphic presentations and the window management, greatly improved and also console hangs due to low memory disappeared.

This architecture (Fig. 2) is fully scalable: it is possible to increase the number of VME embedded processors and hence the power dedicated to user applications and to add "on the fly" *SunRay* terminals in order to have more working points.

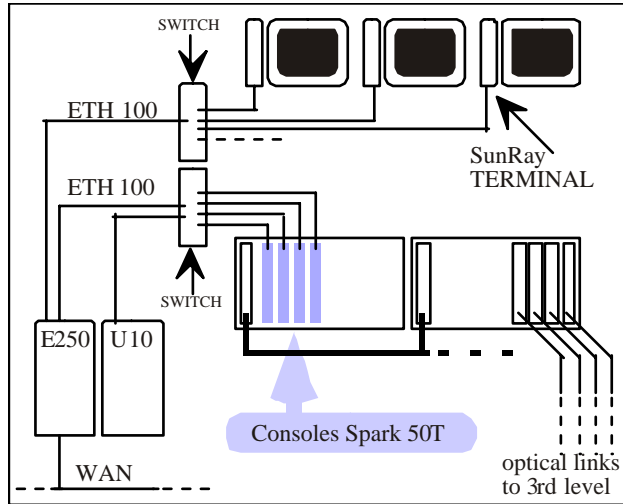


Figure 2: New implementation of the Control System.

After 6 months of development, tests and debugging and 3 months for the installation, the new system met all the upgrade targets, becoming operational in March 2000 (Table 2 summarizes the number and type of processors employed in the system in its present status).

Table 2: System main components

3rd level CPUs	Custom Mac LCIII	43
2nd level CPUs	FORCE 50T	4
1st level term.	SunRay	15
System server	Sun Enterprise 250	1
WWW server	Sun Ultra 10	1

5 NEW DEVELOPMENTS

Presently, we are working on the upgrade of the 3rd level layer of CPUs. This became essential for more than one reason:

- we are running out of spare CPUs;
- the new LabVIEW version does not run on 68K processors;
- dealing with MacOS 7, the remote development and debugging is hard to do.

On the basis of our present experience we looked for a VME controller that would allow us to reuse all the existing LabVIEW software even on a different platform. This led to the choice of a Pentium diskless board by VMIC [7] with Linux OS, which matches all the new requirements and still has a reasonable cost. Now, we have written the basic VME access routines

and we plan to test the new 3rd level CPU at the beginning of 2002.

6 CONCLUSIONS

The system has evolved through 3 different operating systems and 5 major LabVIEW releases, smoothly withstanding these shocks. This has been possible thanks to the use of software written for personal computers with the characteristic of portability.

We are confident that we will always be able to use more suitable hardware, reusing the software work done. This is the meaning of "liberation from hardware".

7 ACKNOWLEDGEMENTS

We want to thank G. Baldini and M. Masciarelli for their commitment and essential contribution in the system implementation, O. Coiro and D. Pellegrini for their support in the hardware installation and setup.

We are also grateful to all the Accelerator Division staff for their continuous suggestions and encouragement for making a good and useful job.

A dedicated acknowledgement is due to C. Milardi for her tight collaboration to the high-level software development.

REFERENCES

- [1] G. Vignola and DAFNE Project Team, DAFNE: The First Φ -Factory, EPAC'96, Sitges, June 1996, p. 22.
- [2] G. Di Pirro et al. "DANTE: Control System for DAFNE based on Macintosh and LabView", Nuclear Instrument and Methods in Physics Research A 352 (1994) 455-475.
- [3] LabVIEW, National Instruments Corporation, 11500 N Mopac Expwy, Austin, TX 78759-3504 USA (<http://www.ni.com>)
- [4] Creative Electronic System S.A., Route du Pont-Butin 70 CH-1213 Petit-Lancy 1 Geneva Switzerland (<http://www.ces.ch>)
- [5] Force Computers GmbH, Prof.-Messerschmitt-Str. 1, D-85579 Neubiberg/München (<http://www.forcecomputers.com>)
- [6] Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, CA 94303 USA (<http://www.sun.com>)
- [7] VMIC, 12090 S. Memorial Pkwy, Huntsville, AL 35803 USA (<http://www.vmic.com>)