# THE EVOLUTION OF THE ELETTRA CONTROL SYSTEM

C. Scafuri, L. Pivetta on behalf of U.O.Controlli, Sincrotrone Trieste S.C.p.A., Trieste, Italy

*Abstract*

The evolution of the Elettra control system is presented by focusing on the major technical upgrades. The Elettra control system has been operating since 1993. The orginal control system architecture was based on a three layer design. A field bus connected the low level computers used to interface the accelerator devices whilst a ten Mbit/s shared Ethernet network linked the middle layer computers to the servers and operator workstations. A first control system upgrade started in 1998 in order to dismiss the field bus and to provide more computing power. A couple of years later a major rework of the network infrastructure was carried out with the introduction of a switched Ethernet architecture. Starting from 2003, in view of the construction of a new booster injector for the storage ring and of the FERMI@elettra free electron laser, new control system hardware and software platforms have been selected. Driven by the additional necessity of cutting development and maintenance costs, the Tango control system has been adopted. The tools developed in order to effectively manage the integration and coexistence of the legacy and new control system are described.

## THE ORIGINAL CONTROL SYSTEM

The original contol system of Elettra is based on a three level architecture, respectively named *presentation*, *process* and *equipment interface* layer [1].

A Local Area Network (LAN) based on a 10 Mbit/s shared Ethernet interconnects the *presentation* and *process* layers. A multidrop field bus, based on the MIL-1553B originally developed for military aircrafts, is used for the Low Level Network.

Modular Single Board Computers (SBC) based on the Motorola 68K family microprocessor, hosted into VME-bus [2] based systems, have been selected for the *process* and *equipment interface* layers. In order to meet the required performances the OS-9 real-time operating system has been chosen to run on these platforms.

The Local Process Computers (LPC) installed at the *process* layer execute the main control tasks and provide all the services to the application programs running on the control room workstations and servers. A multi-master architecture has been adopted for the LPC. The commercial SBC used are equipped with a Motorola 68030 microprocessor running at 25 MHz and 16 Mbyte of RAM.

A number of Equipment Interface Units (EIU), typically hosted inside the controlled equipment, compose the *equipment interface* layer and perform the input/output connections with the field. Digital I/O, ADC and DAC VME boards are widely adopted, as well as serial RS232 and RS422 adapters.

One LPC and the required number of EIUs, sharing a dedicated field bus branch, are assigned to each homogeneous group of controlled equipment: one for the vacuum system, one for the radio frequency plants and one for the corrector power supplies, etc. Remote Procedure Call (RPC) is the core technology used to build the distributed application infrastructure allowing for transparent communication between the workstations and low level computers. It is really interesting to note that some of these systems are still in use at Elettra; the three LPCs and the fortyone EIUs dedicated to the control of the magnet power supplies are still the original installations.

UNIX based workstations and servers have been chosen as the control room computers. The HP9000/720 workstations, used as operator console, are equipped with 48 Mbyte of RAM. An HP9000/827S server computer has been used to provide the NFS centralized storage, both for the HP workstations and the LPCs. Moreover, all the centralized services, such as Remote Procedure Call (RPC) name resolution, clock synchronization, gateway applications toward the beamlines and the accelerator database run on the server, fitted with 96 Mbyte of RAM and two 1.3 Gbyte mirrored external disks. The workstations and the server run the HP-UX 9 operating system.

## REMOVING THE FIELD BUS

The experience gained during the first years of operation togheter with the technology advances lead to a review of the control system architecture. Starting from 1998 a new VME SBC, based on Motorola 68040 microprocessor with MMU and on-board etherent interface has been selected and a completely new low level framework has been adopted [3]. This novel low level computer, that unifies the characteristics of both the EIU and LPC and integrates the *equipment interface* and the *process* layers, has been named Equipment Controller (EC).

Being able to run both the low level control loops and the high level communication tasks, the EC allowed the dismissal of the MIL-1553B field bus.

The central element of the new design is a distributed database resident in the ECs memory where all the input data of the machine are stored. A client request, instead of directly accessing the field, is served with a simple query on the Local Data Base (LDB), with an immediate performance advantage. Some *poller processes* continuosly update the LDB by reading the data from the *device servers* dedicated to the equipment. The distributed application software is still based on RPC and completely compatible

with the existing middleware, though improved in performance and flexibility,

Starting from these new framework, in the following years a number of existing accelerator sub-systems have been upgraded to the new control system architecture. By the end of 2003 the whole vacuum system, the four radiofrequency plants and all the insertion devices took advantage of the new hardware and software platform. During the same period also the newly installed systems such as the super-conducting wiggler, the $3^{rd}$ harmonic cavity and the multibunch feedback systems have been interfaced with this new technology.

In the same time the old 10 Mbit/s shared network has been reworked. The new control network, based on switched technology, has a star topology. A central switch/router is conncted to the peripheral layer-two switches with 1 Gbit/s fibre optic uplinks. The equipment controllers could take advantage of a full 100 Mbit/s dedicated network connection, altough almost all the SBCs were still using a 10 Mbit/s ethernet interfaces. Nevertheless, the bandwidth gain was noticeable and brought a sensitive increase of the broadcast/multicast traffic coming from the company network. This caused malfunctioning of some of the older OS-9 based systems. For this reason a firewall has been installed to protect the control system network from this harmful traffic and from unhautorized accesses. Subsequently all the control system computers connected to ethernet needed to be re-configured to a new private network.

The presentation layer control room workstations were also upgraded; the chosen operator consolles are HP9000/C200 technical workstations with 512 Mbyte of RAM, while the server is a HP9000/D730 with 1 GB or RAM and a RAID 5 external disk cabinet whit redundant controller equipped with 10 GByte of disk space. A second D730, with the same configuration of the main server, is kept as a cold backup. HP-UX 10.20 is the initial revision of the operating system.

## GOING FORWARD

During the year 2001, in view of the construction of a booster injector for the storage ring, and taking into account that the hardware in use reached the middle of its guaranteed industrial life span, a market survey has started to choose the future hardware platform for the control system.

The VME industrial modular system turned out to be a good choice in terms of flexibility and boards availability. The natural substitute of VME was supposed to be Compact PCI (cPCI), offering better performances and hot swap capabilities. In 2001 however, cPCI still suffered some major limitations such as few available slots, single mastering capabilities and scarce availability of commercial-off-the-shelf (COTS) interface boards on the market. More than five years later, the VME bus is still strong and widely adopted, thanks to the military and legacy markets, and the active development still done by all the consolidated VME manufacturers. Our final choice was in favour of the renewed VME64x that provides good performances and allows to reuse legacy hardware.

Taking into account the overall performance, quality of manufacturing and market life span, the selected CPU board is the Motorola MVME51xx, a sigle slot SBC based on the MPC74x0 PowerPC microprocessor, running at up to 500 MHz and equipped with 512 Mbyte of RAM. The SBC accomodates up to two on-board PCI Mezzanine Card (PMC) modules for high performance PCI peripherals. Up to six modules can be used with a stackable carrier adapter. Running at a relatively low clock rate, resulting in low power consumption and easy board level integration, the MVME51xx is well suited for embedded applications. Moreover, the in-core availabile AltiVec technology allows to effectively run compute intensive tasks such as DSP applications or matrix calculations.

Concerning the I/O hardware, besides the legacy VME boards, the mezzanine technology became availabile and attractive for the offered flexibility. PMC and Industry Pack (IP) are already widespread. IP is an adequate low cost solution for high density low-to-middle performance I/O, in fact one-slot VME IP carrier board can host up to four IP modules. On the other side PMC is used where high performance and/or intelligent I/O is needed.

The new hardware platform lead to the necessity to renew the operating system. The opportunity of using the GNU/Linux operating system appeared quite attractive because it is open source, free of charge, very stable, reliable and does not come from a single supplier.

The present installation is based on a size-reduced Debian 3.1 distribution, featuring a 2.4.25 Linux kernel supporting the Motorola MVME51xx SBC, that have proven excellent reliability, stability and good overall performance. The diskless production systems boot from a server using the DHCP/TFTP protocols. This permits a clean and smooth upgrade of the installations to different Linux versions, even, possibly, on a per-machine basis.

Traditionally, accelerator control system designs feature low level computers running a real-time operating system. Altough in most cases current CPU performance provides really fast response even running a general purpose OS kernel, sometimes tighter application requirements need a guaranteed real-time behaviour. Being designed to privilege throughput over responsiveness and determinism Linux is not real-time. As a solution, a real-time micro-kernel can be inserted beneath the Linux OS taking control of the interrupt management and running all the latency demanding applications, whilst Linux runs as a low-priority idle task. Real Time Application Interface (RTAI), a free implementation developed at the Milan Polytechnic and adapted to the PPC architecture by DENX Software Engineering is adopted at Elettra.

A couple of pilot installations have been carried out during 2003 to put online some high performance Beam Position Monitors (BPM) prototypes. A digital detector

VME board provides the beam position data at 8 KHz rate through tme VME interface. Three RTAI application modules, the fast local orbit feedback system, a fast calibration system for the Electromagnetic Elliptical Wiggler and a general purpose orbit acquisition/monitoring system provide the data to user space application servers [4].

Software development practices, tools and platforms were also reconsidered, with the aim of reducing development times, improving code quality and eliminating as much as possible dependencies from proprietary or closed source products. The GNU/Linux operating system reached a sufficient maturity and relability to be used as the main server operating system. By switching to GNU/Linux we where also able to switch from HP proprietary hardware platform to PC workstations and servers, which now have enough computing, storage and graphical performancese for our purposes. This low cost solution give us the benefit of a drastical reduction in acquistion and maintenance costs. Most of the legacy control software has been ported without major problems to Linux. Another major decision was to adopt, as much as possible, object oriented design and programming thecniques for our new developments. Following this decision we had to choose all the new development tools and libraries. Keeping in mind our constraints, for example the possibility to resume most of the existing C software libraries to handle the transition period, we selected C++ as our main developing laguage and the Qt toolkit as our graphic library [6].

## TANGO

The other major decision was to change our communications middleware library from RPC to the Tango Control System [7]. This choice is the result of a long selection procedure, focussed on CORBA based solutions[6], that lead to a list of requirements. The fundamental is to adopt a Distributed Object Oriented (DOO) model based on the Device abstraction. The Tango DOO model is based on many years of experience in the field of accelerator controls programming. During the design phase of new Tango Devices, the object model compels the programmer to have a very clear picture of what the Device must accomplish. On the other hand, a generic Device model support is fundamental to allow clients programs to interact with new Devices always using a standard API. Tango meets also other requirements such as the support for different programming languages and for multithreaded programming on both client and server side. Asynchronous event notification, based on the publish/subscribe model, is availabile. Tango also adds the capability of runtime discovery of Devices and of their characteristics (introspection).

Tango meets all our requirements in terms of capabilities, technologies and performance. The adoption of Tango by a number of different institues guarantees the support and development efforts on Tango core and related tools for the future. We joined the Tango collaboration at the beginning of 2004.

## STITCHING TOGETHER OLD AND NEW

Since the beginning of 2004, Tango has been in use at Elettra for developments related to the new projects. It was also used for the renovation of some plants of existing storage ring (e.g. RF distribution). We had to find a way to guarantee that legacy control programs based on RPC work with the new Tango based plants. We exploited the fact that we can map the semantics of all the RPC calls of our old control system to equivalent Tango calls. A configurable, generic Tango to RPC bridge server has been developed. The bridge acts on one side as an RPC server, on the other as a Tango client. The bridge is configured to respond on the same host and port number of the RPC server of the legacy plant which is converted to Tango. Incoming RPC calls are analyzed and mapped to the correspondig Tango calls, the results are extracted from the response and formatted according to RPC conventions and finally the reponse is returned to the original RPC client. Tango errors and exceptions are also translated or mapped to equivalent or similar RPC error codes.

Since Tango offers a wider choice of data types and behaviour than the old RPC, the design of a new Tango device server must be done with care if we need to make it compatible with the old RPC library. Needles to say, the possibility of mapping the old calls to Tango is one of our most important requirements for the new control system middleware.

Presently we have several subsytems which are critical for machine operations (RF distribution, injection kickers and septa, orbit reading, gloabal orbit feedback) based on the new Tango architecture and the new booster control system is completely based on Tango. Generic and specilized bridges allow us to continue to use all the old control programs without any change in the original source code or configuration. The deployment of the new tools and bridges is also very smooth, with a neglibile impact on machine operations.

## REFERENCES

[1] P. Michelini, The Elettra control and data acquisition system, EPAC'90

[2] The VMEbus specification, ANSI/IEEE STD1014-1987, IEC 821 and 197, VITA, 1987

[3] M. Lonza et al., Design and development ot a new control system for the Elettra linac, EPAC'96

[4] D. Bulfone et al., New front-end computers basedon Linux-RTAI and PPC, ICALEPCS'03

[5] L. Battistello et al. , The Control System Of The Elettra Booster Injector, ICALEPCS 2005, Geneva, October 2005.

[6] D. Bulfone et al. , Toward The Elettra New Injector Control System, ICALEPCS 2003, Gyeongju, October 2003.

[7] A. Goetz et al., "TANGO a CORBA Based Control System", ICALEPCS 2003, Gyeongju, October 2003.