THE CONTROL SYSTEM OF THE ELETTRA BOOSTER INJECTOR

L. Battistello, D. Bulfone, V. Forchi', F. Giacuzzo, G. Gaio, M. Lonza, R. Marizza, R. Passuello, L. Pivetta, C. Scafuri, L. Zambon
Sincrotrone Trieste, Trieste, Italy

ABSTRACT

A new injector made of a 100 MeV Linac and a 2.5 GeV booster ring is being built at ELETTRA to provide full energy injection into the storage ring in top-up mode. This project has required a novel design of the control system with the adoption of a new architecture and the use of up to date technologies, which will also be the basis of the control systems for future projects and accelerators upgrades. In this context, ELETTRA has adopted Tango as the control system software framework and has joined the Tango collaboration as a developer partner.

The main aspects regarding the control system structure, the development of graphical user interfaces and the technical solutions adopted for the equipment interfaces are discussed. Use of industrial controllers for the interlock and personnel safety systems and their integration into the control system are also presented in this article.

INTRODUCTION

ELETTRA is the third generation synchrotron light source based in Trieste, Italy. Its present injector is an electron linear accelerator (Linac) working at a maximum energy of 1.2 GeV. During normal users operation electrons are injected at 0.9 GeV into the storage ring and ramped to 2 or 2.4 GeV depending on the users requirements. A refill is performed every 36 hours and typically lasts 40 minutes. The new injector will significantly increase machine stability and reliability and, above all, will allow for rapid refills that top-up the stored electron beam while keeping insertion devices and beamlines in operation.

The full energy booster injector [1] is composed of a 100 MeV Linac pre-injector feeding a 3 Hz cycled booster synchrotron that matches the maximum energy of the storage ring, namely 2.5 GeV (Figure 1). It will be placed in the open space inside the present storage ring building. After the completion of the new injector, the existing Linac will become available for the FERMI@ELETTRA Free Electron Laser (FEL) facility [2], which is presently in the design phase.

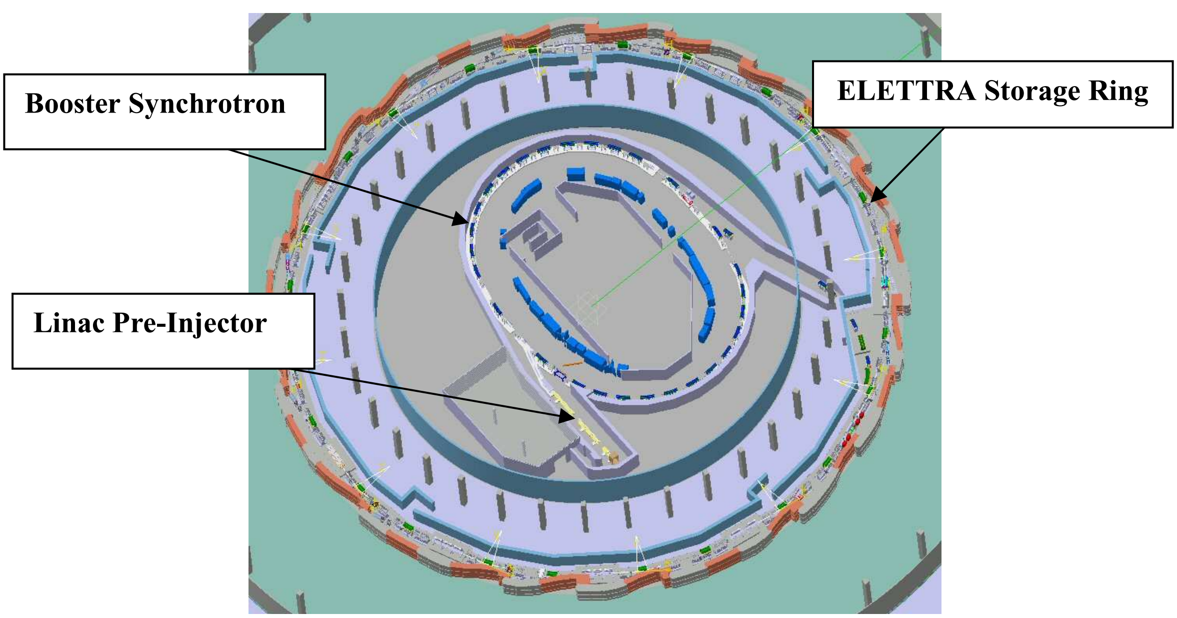

Figure 1: The ELETTRA facility with the new booster injector.

ELETTRA is operating since 1992 with the original control system. VME crates with 68k CPU boards running the OS-9 operating system are used as field computers, while the operator consoles are

Unix workstations with Motif graphical user interface. A MIL-1553 fieldbus and an Ethernet network connect the above computers in a three-layer architecture. During the past years several partial upgrades have been carried out, mainly driven by new required features and obsolescence of either hardware or software components. The incoming booster project demands for a complete review of the technologies and the architecture employed.

REQUIREMENTS AND GUIDELINES

The booster control system will be an extension of the ELETTRA control system, since it will share the same network, control room workstations and servers. All the readings and commands necessary to the operation of accelerator equipment must be handled and made available remotely by the control system, which provides control room operators and technical groups with the graphical tools to interact with the equipment in a comprehensive and ergonomic way. Automatic procedures executing complex operations will be implemented to ease the operator tasks and to avoid loss of time or mishandlings. Support to machine physicists to carry out experiments in characterising the accelerator beams will be provided.

In top-up mode, brief injections into the storage ring will be continuously repeated to keep the stored current constant with evident benefit for the machine and the beamline experiments. In this operational mode, all of the booster components, including the control system, must assure high reliability and stability.

The booster accelerators and the associated technical equipment, including power electronics, instrumentation and control system field computers, will be placed in the same hall (technical gallery), which will be inaccessible by the personnel when the booster is operating. Diagnosis and possibly fixing capabilities must be implemented remotely to quickly recover from malfunctions of both the machine equipment and of the control system itself with minimum impact on machine operations.

The booster cyclic operation made of injection, energy ramping and extraction, will be repeated at 3 Hz. Real-time capabilities are required to the control system to accordingly change the equipment settings and to measure beam parameters with adequate time resolution. Moreover, a 3 Hz correction loop is needed to stabilize the electrons trajectory in the booster to storage ring transfer line in order to keep the injection efficiency optimal. Transfer line beam position monitors and corrector magnets will be used respectively as sensors and actuators by the feedback loop.

Thank to the extraordinary progress in the field of electronics and information technology, control systems with superior performance, novel useful features and tools can be built nowadays deploying off-the-shelf components, open standards and free open-source software. Moreover, in the past years a number of control system frameworks (EPICS, ACS, Tango, etc.) developed by international collaborations among laboratories have become available free of charge to the controls community. This allows collaboration partners to share efforts and reduces considerably the development time, with the additional advantage of adopting, in most of the cases, stable and well tested software.

Following the mentioned guidelines, the development time required to build a control system can be noticeably reduced and the overall cost (including hardware, software, man power, maintenance, etc.) lowered.

CONTROL SYSTEM HARDWARE

The layout of the booster control system is depicted in Figure 2. The data network is the backbone of the control system. Several types of computers distributed around the machine will be attached to the network and communicate with each other using the TCP/IP protocol. An extension of the existing ELETTRA control network will serve the booster control system. It consists of a switched 100 Mbit/s Ethernet network with 1 Gbit/s fibre-optic uplinks and is separated from the rest of the company network by a firewall to protect the control system from external intrusions or harmful traffic. The booster technical gallery will be provided with a wireless network for the connection of notebook PCs used in the debugging and commissioning phase.

Most of the machine equipment like RF amplifiers, power supplies, instrumentation, etc., will be interfaced to the control system via direct connection to the Equipment Controllers (EC) [3], which are diskless VME systems equipped with Motorola MVME5100 PowerPC boards running Linux and the Tango control system software (see below). The chosen VME crate features a 16-slot VME64x back-plane, two hot-swap redundant power supplies and a local controller with Ethernet interface for remote monitoring and reset of the crate.

Most of the Input/Output (I/O) interfaces are Industry Pack (IP) modules mounted on VME carrier boards. PMC (PCI Mezzanine Card) modules and VME boards are also employed for applications where high performance or special features are required. Several types of I/Os will be used: analogue and digital signals, RS-232/422/485 serial lines, GPIB interfaces, etc. Voltage-to-frequency conversion is used to avoid electromagnetic noise in the transmission of analogue signals in the Linac klystron room. Transition modules can be mounted on the rear side of the crates providing a clean connection of the cables and, if necessary, signal conditioning for the I/O modules.

Machine equipment featuring its own embedded controller with Ethernet interface will be directly connected to the controls network, as well as Programmable Logic Controllers (PLC) adopted for the interlock and personnel safety systems.

The existing ELETTRA control room workstations and servers are Unix computers. In view of the construction of the booster, an ongoing renovation program aims at replacing them all with Linux PCs by the beginning of 2006. Most of the existing control room programs based on the old control system software (RPC and Motif) have been already ported to Linux by recompiling the source codes. The booster control room software will run on the same computers.

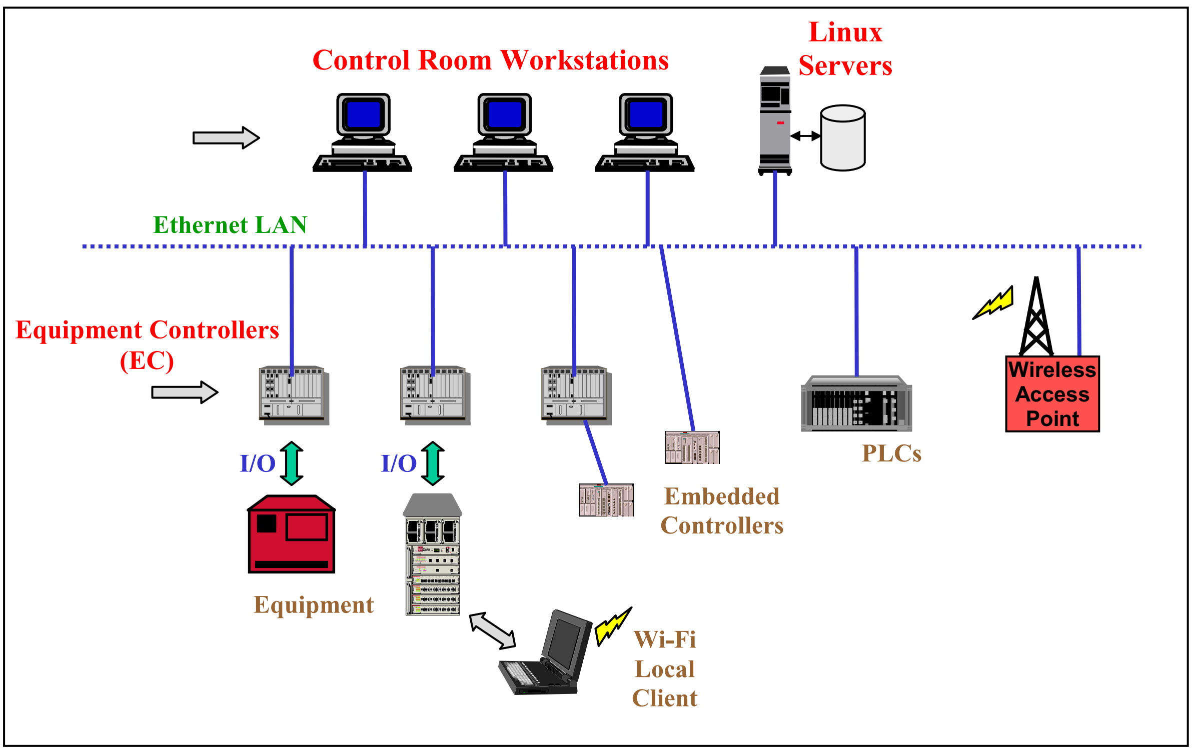

Figure 2: The booster control system layout.

SOFTWARE

GNU/Linux

We have decided to adopt a uniform and homogeneous software environment for all of the new developments, including the booster control system. The chosen platform is GNU/Linux. This choice has been already described in [4]. After three years of actual development, deployment and daily use at our institute we can confirm the goodness and the validity of our choice. The GNU/Linux system has proven to be sturdy and reliable. The rich and extensively documented development tools have shortened the coding and debugging cycle and greatly improved the quality of our software.

GNU/Linux is deployed for both PC-x86 and VME-PowerPC platforms. The standard Linux kernel does not offer, however, real-time characteristics that are necessary whenever a deterministic behaviour is required. Several patches or extensions to the kernel are available enabling hard real-time performance. At ELETTRA we are using RTAI (Real Time Application Interface), which well fulfils the requirements of all the booster real-time applications. The choice and performance of RTAI are described in [3].

The adoption of the GNU/Linux platform for the booster control system opened up the possibility of developing a single code base for all the hardware platforms. We use CVS [5] as our centralised source repository and for managing releases and revisions of our sources.

CORBA & Tango

CORBA has been selected as the basic middleware for distributed applications [6]. After evaluating the characteristics and performance of several CORBA solutions [4] and after a thorough evaluation of our needs and requirements we decided to adopt Tango, a fully object oriented CORBA based control system software [7]. The ELETTRA machine controls group is now a member of the international collaboration which is developing Tango.

The adoption of Tango is mainly motivated by the fact that it meets our requirements in terms of capabilities, technologies and performance. By joining the collaboration we can spare a lot of development time and benefit from a larger developer and user community.

Tango offers a set of useful tools to ease and speed up the development of both device servers and clients, but the most useful "tool" for the developers is a well defined object model to which all Tango devices must comply. The Tango object 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. This is a great benefit: a good, clear and well thought design will result in clear, efficient and maintainable code, fewer bugs and a reduced programming effort.

In order to maintain a good degree of compatibility with the existing high level control room programs based on Remote Procedure Call (RPC), we have developed and deployed a configurable bridge which translates RPC calls into equivalent Tango calls.

Some new tools for the Tango framework are being developed at ELETTRA and are made available to the whole Tango community: an alarm system [8], a Web interface for Tango and a Web based browser of historical data (see below).

GUI & Qt

The graphical control panels and user interfaces for the booster control system are based on the Qt graphical toolkit [9] and C++ and Python Tango APIs. We developed a set of tools to better and more easily write graphical user interfaces with Qt and Tango. The new tools take the form of three packages: *Ttk* (Tango toolkit), *QtControls* and *QTango*.

Ttk is a non graphical library that provides some classes to handle the most common client read and write actions. It transparently manages device proxy creation and caching, event subscription, polling threads and client side error logs.

QtControls is a small set of custom Qt based widgets used to input and display data in a format suitable for controls and not yet well supported by existing Qt widgets. The set of widgets will grow as needed.

QTango is built on top of *Ttk* and *QtControls*. It automates the most common usage patterns employed for writing graphical control applications, i.e. to associate a Tango device attribute or command with a viewer or a controller.

Scripting Languages

The old control system is accessible mainly from C language API and libraries; the other supported language is Matlab. In our experience, the availability of a scripting language has proved to be extremely useful for non routine operations like machine physics experiments or commissioning of new equipment. The control system of the booster will be accessible from Matlab and Python, which are both very well supported by Tango. Python [10] is an object oriented interpreted language. Many different libraries are already available for Python, covering almost every aspect of scientific and technical computing.

Database Applications and Web Interfaces

The new booster control system will use Mysql for all its database operations: Tango device configuration handling, machine settings save and restore, archiving of historical data, etc. The existing Oracle applications are also being migrated to Mysql.

A Web based data browser for the Tango historical archiver [11] is under development, which will give the possibility of searching and displaying machine historical data. Another Web tool will provide remote access to the new control system by browsing the operating Tango devices and displaying their attributes and properties. Both tools are being developed as part of the Tango collaboration.

INTERLOCK AND PERSONNEL SAFETY SYSTEMS

Machine interlock and personnel safety systems are autonomous systems. Each of them adopts a centralized architecture with a Programmable Logic Controller (PLC) in charge of the execution of the control programs and a number of distributed I/O peripherals connected via Profibus. The adoption of the Tango toolkit and of a Supervisory Control And Data Acquisition (SCADA) system for the operator console software has been evaluated. Taking into account also our past experience with both systems, we have chosen Tango, which gives the advantage of avoiding additional, proprietary and expensive software. PLCs will be connected to the controls network and will communicate with the control system using the TCP/IP protocol.

Interlock System

The interlock system will execute the required actions to protect vacuum system components, magnets and other part of the accelerator from damaging. The PLC is a Siemens S7 series 300. A dedicated Tango device server will communicate with the PLC by means of a TCP/IP socket, enabling any client application to access the interlock system through its Tango interface. In the control room, operator panels will collect, display and archive alarms coming from the PLC. Additionally, a complete map of input/output values will be displayed for debugging purposes.

Personnel Safety System

The personnel safety complex is in charge of protecting people from radiation hazards by denying access to potentially dangerous areas. Moreover, it takes all the actions necessary to switch off the accelerators in case of anomalous events that could imply danger for the personnel. Given the extremely high degree of safety required, fail-safe versions of PLC (Siemens S7 315F), Profibus (Profisafe) and I/O peripherals are being adopted. In case fail-safe sensors and actuators are not available, standard devices will be deployed in a redundant configuration. In the control room, a supervisory system will interact with the personnel safety system through a dedicated Tango device server and TCP/IP socket.

Special care will be taken to protect beamline personnel during top-up injection, which is normally performed with beamline stoppers open. Cross checks using charge detectors and current monitors will be done to detect whether electrons accelerated by the booster and injected into the ring are stored or, instead, are lost, with a consequent production of high energy photons and neutrons.

CONCLUSIONS

Construction and commissioning of the booster injector will be carried out in parallel to the ELETTRA operations, which will be eventually suspended for some months to connect the new injector to the storage ring. The completion of the project is planned by the end of 2007.

The control system structure is defined; software and hardware components have been chosen privileging the adoption of standards, commercial solutions and existing systems. In this respect we have adopted the Tango control system framework that definitely assures performance, reliability and optimization of the development cycle. Additional work has been spent to complete this framework with the necessary tools and to integrate it into the existing control system environment.

In-house hardware developments for special modules not available on the market are almost completed.

By relying on this well established arrangement, the software developers can concentrate on applications specific to the equipment, with the goal of ensuring a high level of functionality available from the beginning of the commissioning phase.

The control system framework resulting from the work described above will also be the basis for the new projects, including FERMI@ELETTRA and the upgrade of the existing ELETTRA control system.

REFERENCES

[1] C.J. Bocchetta et al., “A Full Energy Injector for ELETTRA”, EPAC 2000, Vienna, June 2005.

[2] C.J. Bocchetta et al., “FERMI@ELETTRA: A Seeded Harmonic Cascade FEL for EUV and Soft X-Rays”, FEL 2005, Stanford, August 2005.

[3] M. Lonza et al., “New Front-End Computers Based on Linux-RTAI and PPC”, ICALEPCS 2003, Gyeongju, October 2003.

[4] C. Scafuri et al., "Toward the ELETTRA New Injector Control System", ICALEPCS 2003, Gyeongju, October 2003.

[5] P. Cederqvist et al., "Version Management with CVS", http://ximbiot.com/cvs/manual/; http://www.nongnu.org/cvs/

[6] Object Management Group, "The Common Object Request Broker: Architecture and Specifications", 1998, ftp://ftp.omg.org/pub/docs/formal/98-12-01.pdf

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

[8] L. Pivetta, "Development of the TANGO Alarm System", these proceedings.

[9] http://www.trolltech.com

[10] http://www.python.org

[11] E. Taurel et al., "The TANGO Collaboration Status and Some of The Latest Developments", these proceedings.