PROSCAN CONTROL SYSTEM STATUS REPORT

D.Anicic, G.Dave, M.Gasche, T.Korhonen, H.Lutz, A.C.Mezger
Paul Scherrer Institute, Villigen, Switzerland

ABSTRACT

The 250MeV compact cyclotron delivered by ACCEL (Germany) has been installed at PSI and is now in its commissioning phase. The cyclotron, COMET, will provide beam for the PSI proton therapy project PROSCAN. The Control system for this facility comprises three control systems: the COMET control system as provided by the supplier ACCEL, the Machine Control system (MCS) and the user control system (UCS) for each user area. The MCS is basically a copy of the control system used at the High Intensity Proton Facility at PSI, with some additional components required for a proper operation of the medical facility. In particular we will describe the mechanisms used for the interfacing of the UCS through a Beam Allocator System (BALL). We will also report on the progress in development and installation of the hardware and software as well as some first results of the operation of the facility.

INTRODUCTION

In the final setup, the PROSCAN will comprise four areas: Gantry1, Gantry2, Optis and Experimental. At the moment temporary TEST area is being built (and is almost complete), in front of the later GANTRY2 area. All the main hardware and software components (as well as the beam line itself) can be used and tested.

Presently installed components are:

Kicker magnet	1
Bending magnets	2
Quads (focus/defocus magnets)	15
Steering bending magnets	10
Beam stoppers (with Faraday-Cup)	4 (3)
Beam stopper (as personal safety element)	1
Collimators (selectable 6 sizes)	2
Movable collimator slits	2
Energy Degrader	1
Selectable scattering foil (5 foils)	1
Movable Energy Degrader shielding	1
Multi-Leaf Faraday-Cups (16/32/64 leafs)	5
Beam intensity monitors	5
Ionisation chambers (intensity)	3
Halo monitors	7
Harps (Multi-stripe Profile Monitors)	32 (12 16-stripe, 20 32-stripe)
Pneumatic drives	14
High Voltage source (for Harps)	8 (dual channel)
COMET Accelerator connected PLC devices	254

THE MCS HARDWARE

The hardware for the MCS consists mainly of standard control systems components that are already in use in other PSI proton accelerators. The hardware is based on the VME64x standard and the extensive use of mezzanine cards like Industry Packs. CPU cards from Motorola and the Industry Pack (IP) carrier boards from Hytec Electronics are used. Control of devices that need an intelligent low-level controller (for example for monitoring the device status and generating interlock signals [1]) has been implemented using the intelligent IP carrier board VICB8003 from Hytec. The beam diagnostics group uses mainly in PSI developed VPC VME board (contains two Virtex2Pro FPGAs with two PowerPCs each).

The control system infrastructure is in place as far as the construction allows and the components have been connected to the devices and configured with the corresponding firmware.

The components to be monitored or controlled up to the TEST area are the magnets, the drives for the beam blockers, the multi-stripe profile monitors (Harps), Halo monitors and ionization chambers, Faraday-Cups (including Multi-leaf), the beam intensity and beam position monitors, the energy degrader and scatter foil, collimator motor drives and pneumatic drives.

The degrader has two axes that have to be driven synchronously. The drive is implemented with stepper motors and the amplifier developed for the Swiss Light Source (SLS) was selected for use with a small modification for the use with brakes. The degrader controls have been extensively tested and the requirements for speed and accuracy have been fulfilled. The control electronics is based on the Hytec VICB8003 carrier board with a Hytec ADC and Digital I/O IP modules and an OMS stepper IP module.

The kicker magnet is used for fast beam on/off switching. For the kicker, a switch, that allows the safety systems (RPS and PaSS) to be connected, was developed and installed. It is controlled, like other magnets, also with VICB8003 carrier board containing one RPM IP module and up to 3 dual opto-link IP modules for communication with the corresponding power supplies.

The stepper motor drives (for example for collimators) and a selectable scattering foil are also implemented with the VICB8003 carrier board with Hytec and OMS IP modules.

The Run Permit System (RPS, interlock) [1] has been implemented in VME hardware using IP modules and corresponding transition modules to connect to the signals. The firmware logic has been implemented up to the TEST area and is in operation/testing.

The signals for the Patient Safety System (PaSS) have been defined and the wiring prepared for the existing components. The PaSS implementation is underway; the structure has been defined and is being refined and undergoing analysis for safety.

Diagnostics devices, like Harps, Halo monitors, Ionization chambers, Faraday-Cups, beam position and intensity monitors, have been implemented in VPC boards. Due to the late delivery of VPC boards some are still controlled by their CAMAC equivalent.

Figure 1: PROSCAN control room console

So-called safety switch boxes for interfacing the safety systems (RPS, PaSS and the Personnel Safety System) to the components that are used to switch off the beam have been developed by ACCEL according to our specifications and have been installed, tested and connected.

THE MCS SOFTWARE

The MCS software is the copy of the existing high intensity proton cyclotron control system. For its implementation, we have installed the server and the operator workstation computers as well as the data acquisition and control computers (Input/Output Computer, IOC).

The server and the operator workstation computers are running a Linux (presently Fedora Core 1, now migrating to Scientific Linux 4.1) operating system. There are actually four servers: the IOC boot server, the file and the development server, the database server and the archival server. The special dedicated workstation/server for the BALL functionality has also been installed.

We have currently two operator consoles, one in the control room (Figure 1), consisting of two workstations with four screens each, accompanied by one “knobbox” [2] (the four-knobs operator convenience tool). Another console is a workstation with two screens installed near the PROSCAN hardware equipment for test and commissioning purposes. There are eight IOCs in operation at the moment. One is dedicated for connection to the COMET's PLC based control system. There is one for each beam section (reflecting RPS topology), and four for the diagnostics equipment.

All servers are fully functional. The boot server provides the boot images for the IOCs and contains their data acquisition and control (DAQC) configuration files. It also acts as a Dynamic Host Configuration Protocol (DHCP) server in the PROSCAN MCS sub-network. The file server contains the control system client software (for operators, physicists and HW experts). The Database [3] server runs the ORACLE database and the corresponding data entry and data retrieval applications. It also distributes the necessary DAQC configuration files for the IOCs and the workstations. The archival server runs the data-archiving, and also the centralized message-logging and the set-value-logging applications [4].

The IOC software supports all hardware and functionalities being used at this moment. The additional drivers will be implemented as soon as needed and as additional hardware becomes available. Access control has also been implemented, but will not be used until it becomes necessary. The BALL has still to be extended for this functionality.

The IOCs already have a complete list of the data acquisition and control parameters for the COMET accelerator PLC and also almost complete list of parameters up to the TEST area. The COMET control system will be commissioned with its own SCADA based system. Afterwards, MCS will take over.

The initial beam setup up to TEST area and corresponding measurements are already in daily use.

THE BEAM ALLOCATOR SYSTEM

The PROSCAN system has the beam line from the COMET cyclotron directed into four different areas. Therefore a control system and mechanisms and procedures are required to be able to allocate and control and manage the beam and the related resources in these areas. The Beam Allocator System (BALL) has been developed to perform and fulfill these requirements. The primary task of the BeamAllocator is to handle the requests and commands of the beam users from the Gantry1, Gantry2, Optis, and the Experimental area, and to control the beam parameters and provide beam setting and adjustments functionality. The access to the system is controlled through login and password to prevent unauthorized operation and provide graded command ability. The user capabilities have been assessed as being at THERAPY, DIAGNOSTICS and MACHINE operation level, each level giving correspondingly higher levels of command ability. Beam ownership in an area is controlled through the application of MASTERSHIP allocation on a first come first serve basis. This allows ownership rights and full operation for the relevant area, depending on the user type. If there is an already existing MASTER, then a SUBMASTERSHIP is allocated to other Clients in their own area, provided there is no conflict of ownership. Ability to set Beam Tunes, and read and write to selected devices is provided. Machine status information is also provided.

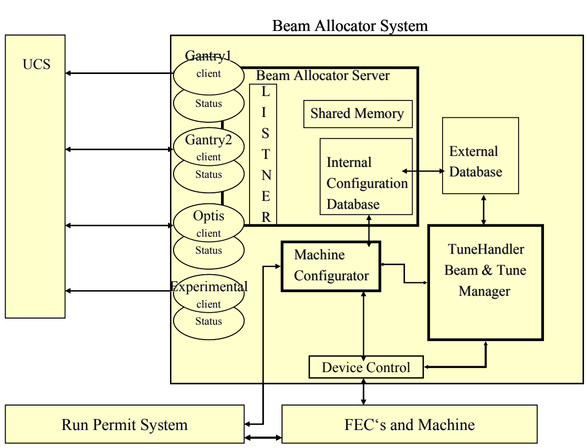

Figure 2: The BeamAllocator environment

To implement this functionality, various software components were developed, to form and build up the Beam Allocator System, as shown in Figure 2. In particular, the components: the BeamAllocator, the MachineConfigurator and the TuneManager, which communicate with the underlying MCS. Status-clients are also provided for information broadcast.

The BALL system is a Client/Server design, using multi-processing and multi-threaded techniques to decouple the components and handle them separately. This gives the flexibility of being able to run any of the components on separate computers if needed. The components communicate with each other using an interface protocol of request and response sequences. The communications between the components are established using TCP/IP and UDP protocols.

For early development, fast prototyping was used initially to confirm and test the principles of the design and communication, and the message sets and the architecture before proceeding to detail development for a testable system. Ease of testing, and testability has been one of aims of the design.

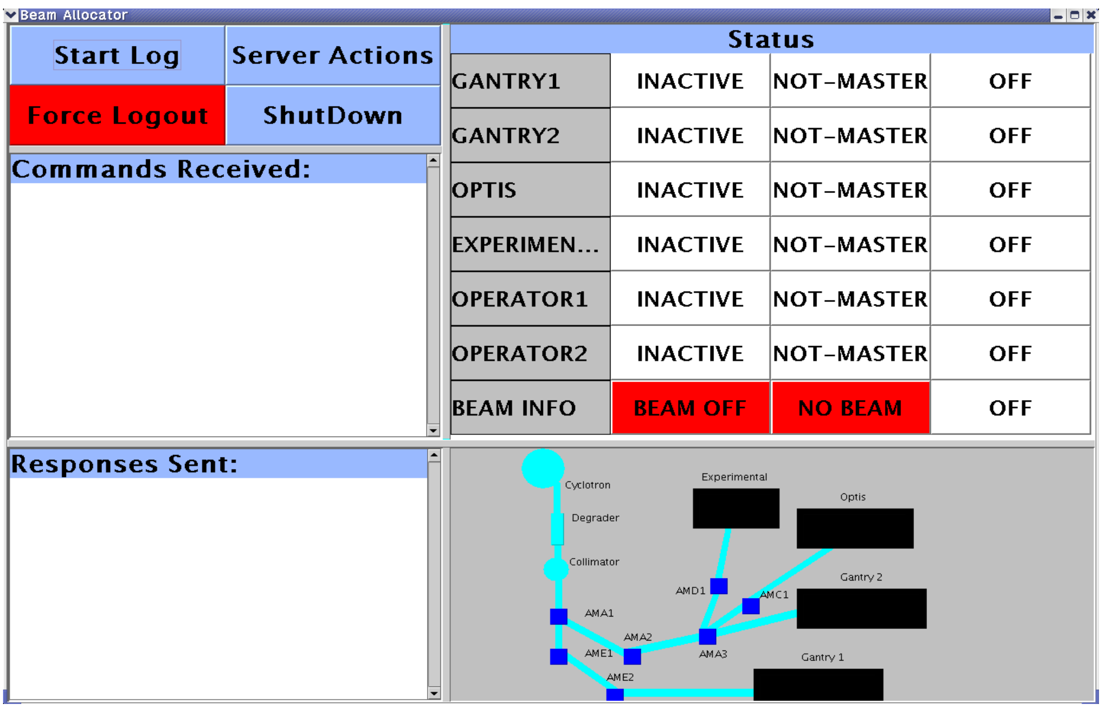

Figure 3:The Beam Allocator System status

The BeamAllocator is one component of the BeamAllocator System. The BeamAllocator starts individual 'Listeners' to handle and control the link with each new UserClient as they establish contact. The BeamAllocator controls the communication and the state set by the UserClient throughout its life cycle. The state of all the beam lines is shown all the time, continuously. The active beam line, the MASTER, the SUBMASTERS, and the Users and their capability levels are always visible (Figure 3). Device and line ownership information is also continuously shown.

To provide this ability the development cycle had led to two possible types of operations, one giving fully automatic operation and the second providing state maintenance and with operator intervention in case of a break.

With the fully automatic version, should there be a break or loss of communications, the BeamAllocator would disengage gracefully, cleanup and recover safely. This had involved multi-threading and multi-processing on the client side as well, so a version without the need for the client to be multi-threading has been developed. On the use of the single thread client, abnormal situation requires operator intervention for recovery. Client state information is maintained should there be a loss of communication.

The BeamAllocator does the Tune setting by using the component MachineConfigurator for high speed response. The commands from the UserClients come through the BeamAllocator and the Tune demands go through to the MachineConfigurator. Frontend status information and error status is routed back to the clients. The requested Tune is processed and the devices are configured through the IOC's.

The StatusClients for data and information display are handled on the same basis. The StatusClients can also be broadcast site wide.

The interlock aspects interact with the BALL through communications with the Run Permit System.

The TuneManager, Figure 4, is used to develop and manage the Tunes and their relevant files. The TuneManager's file editor allows, the file selection, management and interpolation functions. Individual or multiple devices can be set, and on selection, a particular Tune can be set to configure the beam characteristics. Individual or multiple magnets cycling facility is also available. The magnet cycling profile is specified using a cycling definition file. For magnet cycling the device maxima and minima are extracted from the Configuration Database [3] of the control system

The selected Tune files are loaded in the shared memory area for the MachineConfigurator to access at high speed for Tune setting.

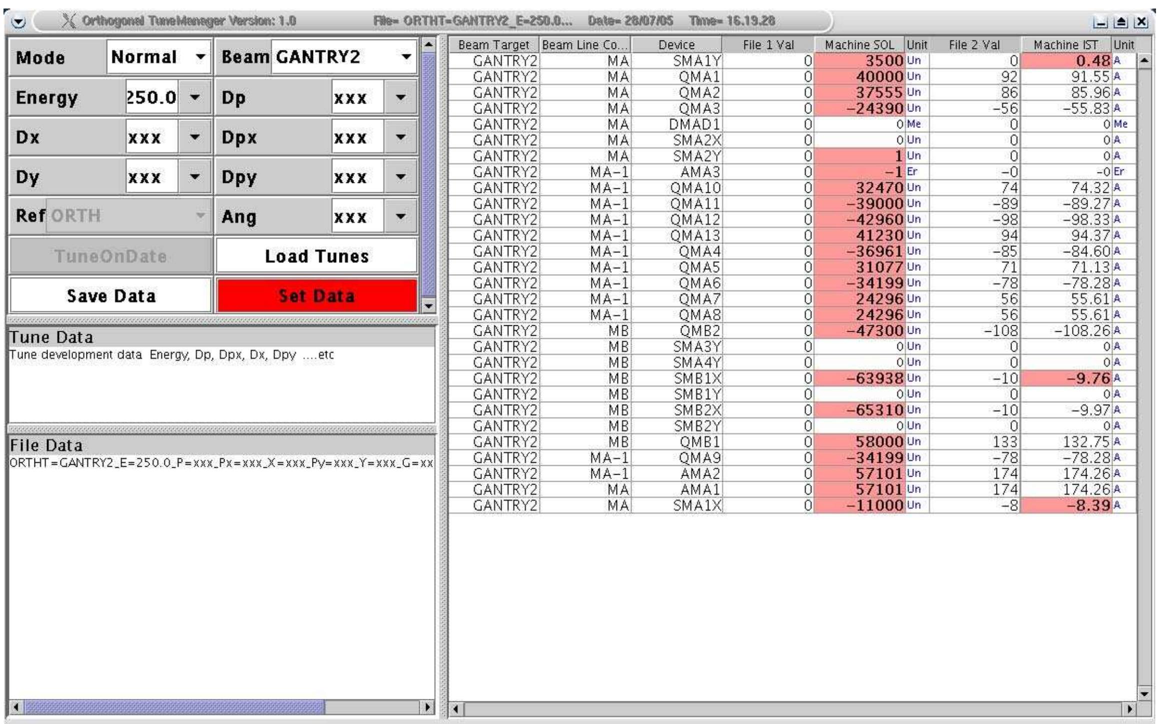

Figure 4:The Tune Manager

There are two main modes or types of tune files. One is in the Reference form, where there is one xml Tune file contain all the devices affecting the setting of the tune parameters. The Tune parameters being: the Energy, the Dispersions, the Emissivities, and the Gantry angle. The Second mode is the Orhtogonal Tune files, where the assumption is that the Tune parameters are orthogonal and setting devices for one particular parameter does not affect the other settings. For this reason, these orthogonal tune files are formed for each of the Tune parameters, and an Orthogonal reference file is formed for the combined parameters of a particular tune, if necessary. The TuneManager has been formed to process both the Coupled and the Orthogonal tune setting needs.

Below are shown the typical Orthogonal file names for each of tune parameters.

ORTHT=HE_E=570.0_P=xxx_Px=xxx_X=xxx_Py=xxx_Y=xxx_G=xxx_040505_103244.xml

ORTHT=HE_E=xxx_P=1.1_Px=xxx_X=xxx_Py=xxx_Y=xxx_G=xxx_040505_103244.xml

ORTHT=HE_E=xxx_P=xxx_Px=1.2_X=xxx_Py=xxx_Y=xxx_G=xxx_060405_112652.xml

ORTHT=HE_E=xxx_P=xxx_Px=xxx_X=1.2_Py=xxx_Y=xxx_G=xxx_060405_112652.xml

ORTHT=HE_E=xxx_P=xxx_Px=xxx_X=xxx_Py=1.2_Y=1.4_G=xxx_060405_112652.xml

ORTHT=HE_E=xxx_P=xxx_Px=xxx_X=xxx_Py=xxx_Y=xxx_G=250_060405_112652.xml

The Orthogonal files can be combined to a form a single orthogonal tune reference file.

ORTHREFT=HE_E=570.0_P=1.1_Px=1.2_X=1.2_Py=1.2_Y=1.4_G=250_060405_112652.xml

The Reference file for the Coupled Tune have a similar naming but with no corresponding individual tune files for each of the tune parameters.

Through the set of available commands, some of which are, Login, RequestMastership, ReleaseMastership, ReadDevice, SetDevice,. DoTherapy, SetOperationsMode, SetTopology, the user performs the required tasks. The beam is requested by using Beam settings for a particular 'Tune' with certain energy, intensity, emittance, gantry angle and other parameters. These are specified in XML type Tune files as mentioned before which have a particular name format which identifies a particular Tune and its characteristics precisely.

To facilitate the user interaction with the BeamAllocator, an application programming interface has been developed in C++. The function calls giving the response to each command are provided by the API. The response structure returns the message code, the message length, and the message. A standard Linux and a VxWorks version of the API has been formed.

The primary functionalities of each of the components have been developed and demonstrated in the development environment. Each component has been demonstrated individually. The TuneManager is being run and tested in the operational environment. The systems integration phase for the components working together has also been achieved and demonstrated. The whole system has now been transferred to the PROSCAN machine environment to establish run ability and to develop and establish a test environment and a formal release environment.

A Test environment and a suite of programs forming the TestManager have also been created, to be able to test various operational scenarios and also be able to do regression testing as changes and modifications go through the development cycles.

The TestManager suits are in Perl, C++, and Linux scripts. Single Client operation scenarios to multiple Client operations scenarios are managed and tested. The tests are automatically runnable and can form a daily or routine test sequence before starting normal operations. In any case the tests are run anytime a new version or modifications are put through.

The detail development of the system components is proceeding

REFERENCES

[1] A. C. Mezger, G. Dzieglewski, "Protection Mechanism for a high power accelerator", ICALEPCS'2005, Geneve, Switzerland, October 2005.

[2] A.C. Mezger, D. Anicic, T. Blumer, I. Jirousek, "Integration of a New Knobbox in the PSI Control Systems ACS and EPICS", ICALEPCS'2001, San Jose, California, November 2001.

[3] H. Lutz, D. Anicic, A.C. Mezger, M. Gasche, "The Configuration Databases for the PSI Proton Accelerator Controlsystems", ICALEPCS'2005, Geneve, Switzerland, October 2005.

[4] D. Anicic, A.C. Mezger, M. Gasche, H. Lutz, "PSI Proton Accelerator Control System Upgrade", ICALEPCS'2003, Gyeongju, Korea, October 2035.