INFORMATION TECHNOLOGIES IN THE IHEP U-70 ACCELERATOR COMPLEX CONTROL SYSTEM

V. Voevodin, IHEP, Protvino, Russia

Abstract

Short descriptions of the U-70 accelerator complex treated as a fast cyclic technological process and its specifics from the controls point of view are introduced. The architecture of the currently used distributed computing and software in the Unix-like environment are presented. Main properties of unified informational scope for real time data processing are defined. The paper describes some current quantitative features of the U-70 control system and steps that were taken to homogenize the control system so as to be based on PC-compatible computers and Linux operating system only.

TIMING CONSTRAINTS

Currently the U-70 accelerator complex is prepared to work with light ions to be injected by liner accelerator I-100. So the last extension of the Control System (CS) was to include under control the beam transfer line (BTL) from I-100 to booster and its consol was installed in the I-100 Local Control Room (LCR). Figure 1 shows the current CS elements layout.

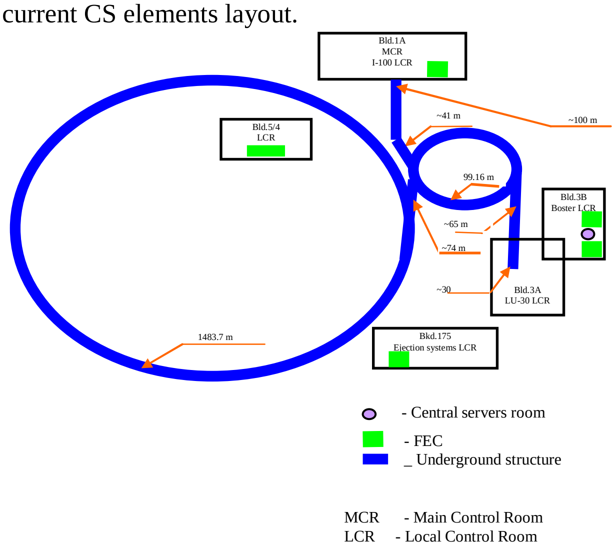

Figure 1: Layout of the control system elements

The control system embraces the technological subsystems of the next accelerator complex units:

- liner accelerator LU-30,
- booster accelerator U-1.5,
- main accelerator U-70,
- BTL from LU-30 to U-1.5,
- BTL from I-100 to U-1.5,
- BTL from U-1.5 to main ring,
- slow and fast ejection systems of the U-70.

The sequence of the units work consists of two stages. The first one is a booster cycles – one pulse of a liner accelerator, injection in the booster ring, acceleration in the U-1.5, ejection to U-70 or local physical experiment. Duration of the U-1.5 cycle is equal to 60 ms. To fill the ring of U-70 by particles is necessary up to 29 booster cycles – during this process U-70 runs as storage ring. After ejection of given number of pulses, we call it “booster package”, the U-1.5 stops and waits the end of a next stage.

Second stage is acceleration and ejection process of U-70 ring accelerator. Duration of this stage is near 8 s and duration of U-70 accelerator complex cycle is equal to 10s – we call it “super cycle”. The run of U-70 accelerator complex is continuous sequence of “super cycles”.

The duration of “super cycle” is general timing constraint for control system. It defines the rate and real time general scale of measurements, data processing and information renewal on the screens of all consoles. In addition there are two peculiar timing constraints.

If U-70 does not require the whole 29 booster cycles it is possible to supply a local experiment by beam from booster. In this case CS after finishing ejection to U-70 must during 20 ms change booster and liner accelerator set of settings values, to get new necessary beam parameters – energy, bunch form, direction of ejection. We call this procedure “programming inside package”. So, the control system attends two logical injection complexes (LU-30 and U-1.5) with different settings values – one for U-70, other for local experiments.

Due to some reasons, e.g. results of radiation measurements, it may be necessary to change U-70 complex regime in the nearest super cycle. In this case CS has 120 ms at the end of current super cycle to change the values of settings to start the new regime of the U-70 complex. We call this procedure “programming between packages”. So, the control system should to support different sets of settings values corresponding to U-70 accelerator complex regimes.

HARDWARE ENVIRONMENT

The U-70 control system was built in accordance with preliminary plans and project decisions [1], [2]. The three levels architecture was adopted for the U-70 CS. The upper level included two functional parts - Alpha Stations as specialized servers and X-terminals as user consol points to access the CS facilities. The Front End Computers (FEC) level was built on the base of VME crates, Single Board Computers (SBC) of Motorola and MIL 1553 Bus Controllers (RTI). The Equipment Controllers (EC) level included Multibus-I crates, SBC with Intel 80186 CPU. We have three categories of the interfacing means: I/O module assembly tailored for the EC tasks execution, the process equipment communicates directly with modules plugged in the SUMMA crates and EC controls the dedicated equipment through the Branch

Driver placed in the Multibus-1 crate, the specialized micro controllers (mC) are embedded in accelerator's equipment and linked with EC through RS-485 bus and RS-485 bus controller is plugged into Multibus-I crate.

There are one Main Control Room (MCR), five LCRs and some number of stand alone consoles in technological rooms. Under service of the CS there are about 10000 various signals. Approximately it is 60% of total number of accelerator complex signals. Now 38 ECs, over 20 SUMMA crates and near 200 embedded mCs are installed.

SOFTWARE BASIC DECISIONS

The general software layout was described in [3]. The Alpha stations were equipped with DEC Unix and Red Hat Linux 4.1 and 6.2 operating systems (OS). The upper level software consists of three main relatively independent parts: Distributed Real Time Databases (DRTDB), Graphical User Interface (GUI) and Data Processing Applications (DPA). The GUI and DPA are pure data-driven software completely based on the data and descriptions stored in DRTDB.

Since the start of U-70 CS exploitation the DPA software ran on Alpha stations with Linux, GUI – on Alpha stations with DEC UNIX, DRTDB – on any computer with any Unix-like system. The X Window, Motif and XRT packages lay in the basis of GUI, which realizes the wide set of functions with the DRTDB information.

The FEC level software runs under control of LynxOS operating system and performs transformation and transmission of packets between Ethernet and MIL 1553 communication field buses. There are the following specialized application protocols on top of UDP and TCP transport protocols to access EC services: loading EC application tasks, loading configuration files, loading DRTDB tables to EC memory, EC virtual terminal, and remote access to data of the DRTDB tables in memory of EC.

The basic idea and main aim of system software design was to create Unified Informational Scope through all levels of CS architecture. This real time Unified Informational Scope should to have next properties:

- **accessibility** – any informational item are accessible from any point of the CS independently of place of the storage and placement of the request source;
- **security** – prevention of unlawful access to data;
- **presentation** – unified external presentation of data structures and logical organization to users;
- **addressing** – common system of coordinates to address any informational items from software.

Distributed real time database management system SSUDA was developed to provide Unified Informational Scope of the control system. It supports three-dimensional tables with possibility to replicate some data tables in the RAM of ECs. A table may have the only low level replication. Figure 2 shows the general software parts.

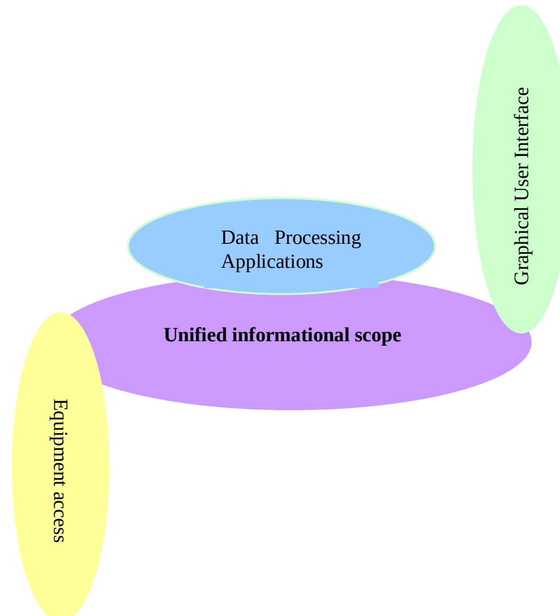

Figure 2: General software architecture

Low level equipment access software in general performs synchronized with accelerators events exchange of data between equipment and SSUDA tables in RAM of ECs. It guarantees external and temporal consistency of the Unified Informational Scope data. This software realizes the change of setting values for programming inside and between packages. So, Unified Informational Scope is an informational model, at any moment reflecting the current state of the U-70 accelerator complex. Now the Unified Informational Scope contains 2193 data tables and stores it in 12 DB files, which are distributed over 8 computers. Nearly 250 tables are replicated to memory of 38 equipment controllers.

CONTROL SYSTEM EVOLUTION

Some experience and steps for CS improvements were discussed in [4]. The main aim of the current evolution is fully homogeneous computers and operating systems of the CS upper and FEC levels. The great variety of configurations of the PC-compatible computers available on the market led us to decision to migrate this direction. Now we are using some number of PCs, PC-compatible servers and industrial PCs, replacing Alpha stations and VME crates. As a basic operating system we have chosen Fedora Core and now are using versions 3, 4, 6 and 9.

We replaced Alpha stations with Red Hat Linux, running DPA, by PC-compatible servers with Fedora 3. SSUDA DB binary files and sources of all applications were carried over to the new environment. To migrate from Alpha stations with DEC UNIX the new graphical user interface was developed on PC with Fedora 4 on base of GTK+ package. Control rooms consoles with dual monitors are successful. Newest consoles we equip with Fedora 9. Due to independence of GUI from data processing software now we have two separate GUI. The old one will be taken off operation next year. Both GUI realize the same set of actions on the same data using the same descriptions.

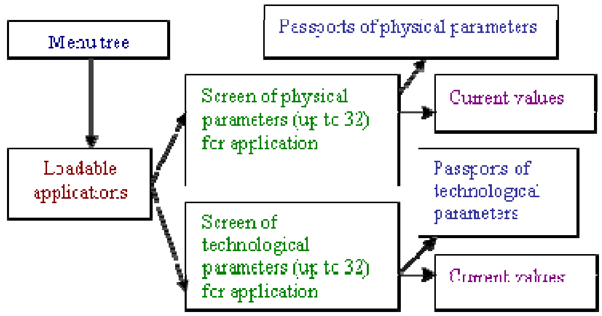

Figure 3: Structure of GUI descriptions in SSUDA

Figure 3 shows the structure of GUI descriptions. Any user session starts from selection of an 'operator task' in the menu tree. To solve the 'operator task' may be necessary to initialize some number of applications for distributed data processing. For each 'operator task' the window with current values of corresponding parameters will be created. The values are presented in form of editable table. It is possible to request graphical presentation of functional dependencies of the any parameters from Unified Informational Scope.

The menu tree has separate brunches to control different regimes and logical injection complexes. So GUI supports edition of settings values for programming inside and between packages. Now the menu tree contains 3893 points. There are 45 tables of screen descriptions for technological parameters and 51 tables for physical parameters presentation. 263 passport tables describe 13318 technological parameters. 361 passport tables contain 29537 records of physical parameters descriptions.

VME based Front End Computers with LynxOS we are replacing by industrial PCs with TA1-PCI MIL 1553 bus controller and Fedora 6 OS. We did over again all application software of FEC without any change of the software of upper and low levels of the CS. Under Fedora we use facilities, such as signals, semaphores, shareable memory, threads in accordance with POSIX standards.

ACCELERATOR START UP PROCESS

When an accelerator is switched off the Unified Informational Scope corresponds to the latest accelerator state. After switching on or pressing button 'Reset' of an EC the operating system starts in the EPROM. During initialization EC OS sends message about start via MIL 1553 bus. The start messages are received by special server computer of the control system upper level.

The server use descriptive information in the DB to define what tables with hardware codes should to be replicated in the memory of the equipment controller. For each table the server defines the upper level computer where the data are stored on a disk and sends request to this computer to do the table copy in EC memory. The computer receives the request and sends description and data of a table to EC. After all necessary tables are replicated the server starts to load corresponding application tasks to the equipment controller.

EC OS receives from the computer the table description, reserves memory block of required size, fills it by received data and registers the table as part of Unified Informational Scope. So, Unified Informational Scope is spreading to low level of the CS with the parameters values, corresponding to the last state of an accelerator. Then the application programs are loaded to EC memory and invoked. On start of execution the application may request the loading of required configuration file from upper level and during initialization it is obligatory to copy setting values from data tables to hardware.

So the CS automatically expands the Unified Informational Scope to low level and achieves consistency of the Unified Informational Scope and U-70 accelerator complex states. Thus on start up the U-70 is switched to the latest working state automatically.

CONCLUSIONS

Each super cycle inside of the control system 50 graphical files of standard presentation of general U-70 accelerator complex beams parameters are renewed and output to any consol by operator request. Usually during cruising mode no less twelve user sessions are opened, more 180 application processes are running. During U-70 accelerator complex investigations or field adjustments the control system working load is significantly higher.

The CS successfully works for a long time. It little by little gets renewed by modern homogeneous computer and software environment. Simultaneously its functionalities extend to new technological subsystems and CS processes new signals. The next step is more wide usage of industrial PCs with embedded electronic modules for direct processing of signals. In this case an industrial PC will combine functions of EC and server.

REFERENCES

[1] V.Komarov, A.Sytin, E.Trojanov, V.Voevodin, V.Yurpalov, "Upgrade of the U-70 complex controls", ICALEPCS'95, Chicago, Illinois, USA, 1995, V.2, p.930.

[2] V.Komarov, Yu.Milichenko, V.Voevodin, V.Yurpalov, "Draft Design Study for the Control System of the U-70 Complex", IHEP&CERN, PS/CO/Note 96-26, Switzerland, Geneve, 1996.

[3] V.Voevodin, "Software Architecture of the U-70 Accelerator Complex New Control System", Proceedings of the ICALEPCS'99, Trieste, Italy, 1999, p.457.

[4] V.Voevodin, L.Kim, E.Klimenkov, V.Komarov, V. Kuznetsov, "Experience with the new control system of IHEP accelerator complex", Proceedings of the ICALEPCS'2005, Switzerland, Geneve, 2005.