ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S01SRA07

The GSI Control System

U. Krause, V. Schaa, R. Steiner
Gesellschaft für Schwerionenforschung mbH, D 6100 Darmstadt, Germany

Abstract

The GSI accelerator facility consists of an old linac and two modern machines, a synchrotron and a storage ring. It is operated from one control room. Only three operators at a time have to keep it running with only little assistance from machine specialists in daytime. So the control tools must provide a high degree of abstraction and modeling to relieve the operators from details on the device level. The program structures to achieve this are described in this paper. A coarse overview of the control architecture is given.

I. THE GSI ACCELERATOR FACILITY

At GSI the heavy ion linac UNILAC runs successfully for 16 years. It produces beams of all elements up to uranium with energies between 1.4 and 20 MeV/u. In '89 the heavy ion synchrotron SIS and the experimental storage ring ESR started operation with ion energies up to 2 GeV/u.

A new control system has been designed for SIS and ESR which is adopted step by step also for the UNILAC.

The system has to handle quite different facilities,

— the UNILAC with 3 injectors and repetition rates of 25...100 Hz,
— the SIS with repetition rates of 0.1...3 Hz,
— the ESR with repetition rates of $\leq$0.001...0.1 Hz,
— transfer lines,

but has to provide an uniform operator access to all accelerators from one control room.

When the upgrading of the old components will be completed a total of 3000 devices scattered over an area of 250 × 250 meters have to be controlled. Complexity varies from simple DC magnets to ramped magnets (ramps generated from up to 900 samples) and diagnostic devices (up to 40kB of data). Fast switching of the machines allow time sharing between several experiments and injection to the next accelerator in sequence, all with independent beams.

II. OPERATING REQUIREMENTS

There is a need for several identical consoles for independent operation of accelerator segments and hardware redundancy.

A consistent "look and feel" has to be provided for the operator interface of all application programs.

To keep the control system serviceable and extensible it has to be a modular and uniform system with definite allocation of tasks. The size of the facilities suggests a decentralized, distributed and hierarchical system. Therefore real-time aspects are limited to the device handling level, data for these operations are stored there. By this means a clear separation between accelerator and device oriented aspects is achieved.

The operating level ("physics of accelerators"), located on powerful workstations, deals with the equipment as a whole and handles devices in a modeled form only. The device level ("physics of devices"), located mainly on microcomputers close to the devices, maintains and controls single devices. Both levels are connected by standardized device accesses using identical mechanisms for all devices.

III. CONTROLS COMPONENTS

On the operating level VAX 3100-VMS graphic workstations are installed forming a VAX cluster. Uniform software all over the system is ensured by cluster-disks. One to three VAXes are grouped as consoles for operator interaction.

The device level is equipped with two types of 68020 VME boards with a real time kernel. One is used as Master Processor (MP) for command evaluation. The other is equipped with an interchangeable communication interface and is used as Equipment Controller (EC) for device control and as Communication Processor (CP) for networking.

For communication between operating and device nodes ETHERNET is used. Devices are connected via modified MIL-STD-1553B serial bus with standardized InterFace Boards (IFB) as part of the devices. Besides this other interfacing like IEEE-488 is partly used by simply replacing the communication part of the EC boards. Typically 5 to 10 devices are driven by one EC.

Time signals for process synchronization are broadcasted from a central timing system to all ECs via serial bus (MIL-STD-1553B) and Timing InterFaces (TIF).

Each VME node is equipped with CP, MP, TIF and 1...9 ECs. A diagram of the components is shown in fig. 1.

Actually 33 operating computers are used in the main control room and in several local consoles close to experi-

Content from this work may be used under the terms of the CC BY 4.0 licence (© 1992/2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

Content from this work may be used under the terms of the CC BY 4.0 licence (© 1992/2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

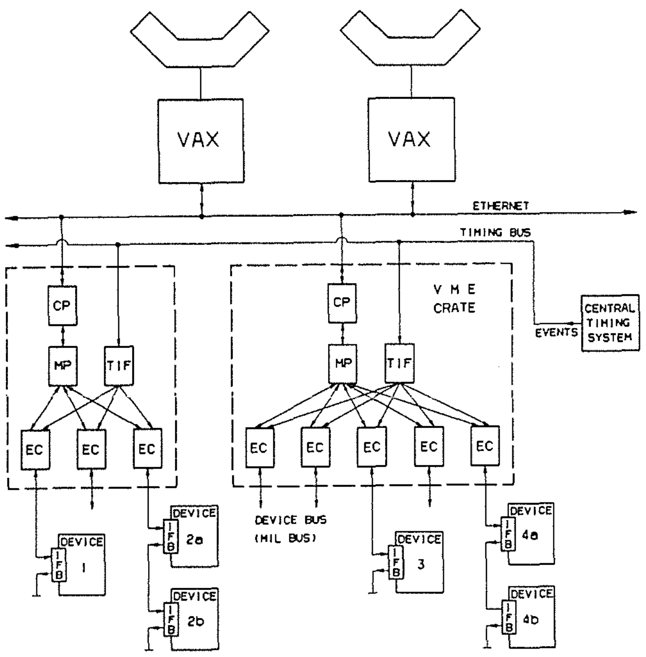

Figure 1: Hardware structure of the control system

mental facilities. For device controlling about 40 VME nodes with about 150 ECs are installed, driving more than 1200 devices.

IV. OPERATOR INTERFACE

For the variety of possible interactions with the control system, handling machines with a quite diverse nature, an operator console and operator interface has been defined combining every aspect of the operating modes.

The operator interface must offer a model of the process to be controlled. The process varies with the selected segment of the accelerator environment (SIS, ESR, UNILAC or transfer beamlines). Three software layers are present to support the selection mechanism to form the operating environment.

Layer 1 mainly consists of a code for selection of an accelerator segment. All computers forming a console are dedicated to the selected segment. Only codes which are usable in this context are provided, therefore all application, utility and service codes are grouped into categories. These categories are, e.g. usable at all consoles and accelerators, specific to an accelerator, to a beamline, a beamline segment or an experimental section. All specific codes serve only devices defined in the selected segment.

Additional codes are server for access to common databases for definition of application codes, accelerator and beamline segments, device names, properties, network addresses, error codes and help topics. Likewise alarm and error handling tasks and a logging facility for messages concerning starting and stopping tasks, abnormal conditions, failure of devices, are available in the console environment.

The second layer consists of two main process control codes, one for UNILAC and all beamlines, the other for ESR and SIS (SD[1], ZV).

— These application programs organize the access to devices using a number of specialized tasks. Each individual task uses the same context (i.e. same machine cycle, device setting etc.) and shares all process data.

— The acceptance of commands is acknowledged immediately, processed directly or passed to a subtask. All windows are organized by the central task and divided into areas with constant display of information and areas which are used by subtasks concurrently.

— The actual status of all devices in the selected segment is visible at a glance: device alarm (status or values changed) and appropriate refresh cycles guarantee an actual state of display. Icons reflect the current status of the device (switched on or off, positioned in beam or out, active or inactive etc.). Icons are unique for every device type. Additional windows are used for display of magnet read-outs showing deviations between actual and reference setting (see fig. 2), beam current profile and spill signals. History recordings offer an easy help to find a faulty element in case of beam-loss.

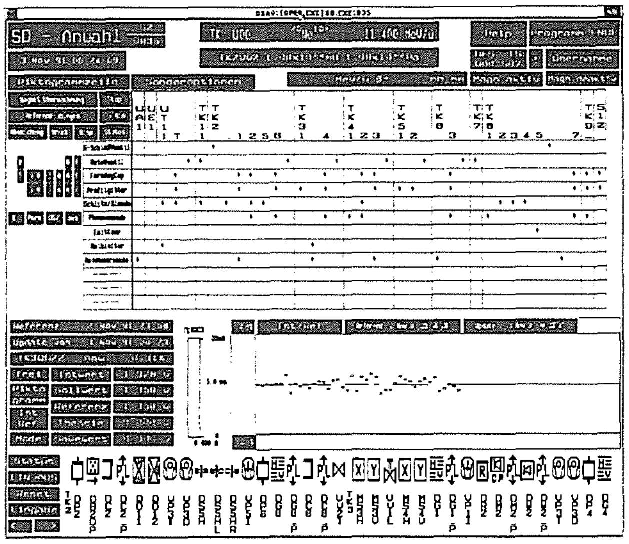

Figure 2: Process control for beamline

Additional codes of this layer are utility tasks, e.g. save magnet settings and restore saved or theoretical sets due to actual physical parameters, codes for manipulation of synchrotron devices according to algorithms, models and parameters, and for management and monitoring of the accelerator cycles. For supervision of the console environment (which task is running, on which node, network traffic etc.) a number of tasks is at hand.

The third layer consists of a number of specialized tasks which are managed by the main process control codes. Two groups are discernible, first codes used for supervisory control and update of device information. These tasks do not need output devices, servicing purely all other, sharing the

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S01SRA07

same process data structures. The second group consists of interactive tasks like emittance measurement, isotope and charge spectra, beam profile display, etc. needing input/output devices as there are graphics, knobs, beam current control monitors.

All application programs provide a consistent "look and feel" by a standard color scheme to present device status and action. There is an unique screen layout management for programs on graphic screens and terminals. Selection fields performing similar actions are standardized in colors, position and names. An uniform access mechanism for all devices and a transparent handling of processes and functions guarantees that the operator does not need the knowledge about the actual control sequences.

Beside the operating environment, for troubleshooting, debugging, test and running-in NODAL [2] is widely used by maintenance people.

V. DEVICE MODELING

To ease handling of the big number of devices they are classified by types (DC magnet, pulsed magnet, beam position monitor, rf-structure, ...). All devices of the same type are represented identically to the operating level and use the same software on the device level. Different device parameters like maximum current of power supplies or number of wires of a position harp are considered by device specific constants in databases on device level.

The set of relevant features of one device type is called Equipment Model (EM) of which actually 42 are supported. They are modeled and described in a standardized way to allow the same access mechanism for all devices.

Every device is identified by an unique name of 1 to 8 alphanumerics, called nomenclature. Each feature of a device of every EM is represented by so-called property and property class. The properties are described by a name of 1 to 8 alphanumerics, the property class as designation of data exchange (R read from device, W write to device, N no data exchange), and a description of the data associated with the property: the number of data to exchange and their representation (INTEGER, REAL, BITSET, ...).

For a magnet the feature "set value of output current" would be described as

property:	CURRENTS (S indicates set value)
property class:	W (data send to device)
data count:	1
data type:	REAL

Properties are EM dependent, but standard ones like mains switch (POWER), device status bit pattern (STATUS), reset (RESET), etc. are identical for all devices.

Accessing devices from the operating level means simply calling the associated property/property class with exchange of data for a device represented by a nomenclature. For greater flexibility several ways of calling are supported like execution with or without response (at least acknowledge) from the device level, synchronous (wait for response) or asynchronous (response may be read later), automatic command execution periodically, etc.

Before sending a command to the appropriate controllers the operating representation is changed to the device representation. This includes the transformation of the device nomenclature to addresses used for identification on the device level, property/property class to the identifier of the associated software module (see section VII.) and data from operating to device format since both may be different, i.e. REAL on operating level, INTEGER on device level.

Data in the responses to a command are converted from device to operating representation and then the response is supplied to the user or buffered for later reading.

Device accessing is handled by one universal software module, using the same mechanism for all devices. All EM and device informations needed are extracted from a Central Data Base (CDB).

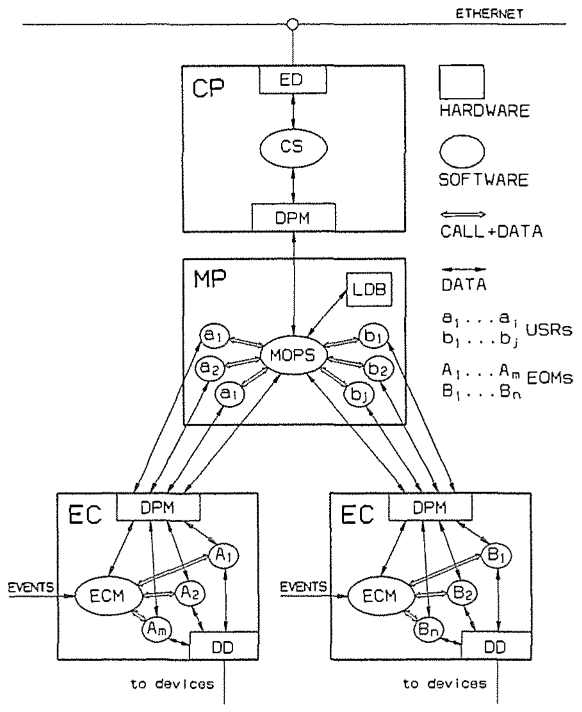

Figure 3: Software structure on the device level. ED: ETHERNET driver, DD: device bus driver, DPM: dual ported memory, LDB: local data base

VI. DEVICE CONTROLLER LEVEL

Device controlling in the closer sense of generating the command sequences for remote control is realized on the EC only. This is the only level where real time requirements have to be fulfilled. To enhance the power and to facilitate fast reaction with well defined response time the task of the ECs is limited to device driving only without interruption by higher levels. All data needed by the device are stored on the EC and it runs completely autonomously. Communication with higher levels is by dual ported memory on the EC and thus may be carried out asynchronously.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 1992/2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

Content from this work may be used under the terms of the CC BY 4.0 licence (© 1992/2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

On the EC is coded *what* to do. Timing requirements (*when* to do) are managed by the central timing system which delivers 255 different time signals, so-called events, to all ECs.

Activities are coded in EM specific **EQ**uipment **M**odules (EQM) as the smallest independently executable units on ECs. They may contain complex actions consisting of several device commands. EQMs are connected to events with an identical connection for all devices of an EM. Execution of EQMs is on reception of the associated event. This guarantees systemwide synchronous operation of otherwise independent devices if only connected to the same event.

Besides event driven EQMs other ones may be called by direct request from higher levels signaled in the ECs dual ported memory. Execution is not immediate but at command events always delivered in the gaps between accelerating cycles when real time reaction is not required. For pure DC devices no events are evaluated, all actions are executed directly when initiated by operating request.

EQMs cannot be interrupted so execution on the EC is strictly serial. If an EQM could not be started in time since a previously started one is still running, an error is signaled allowing EM dependent handling.

Event reception and calling of EQMs is done by the Equipment Control Monitor ECM, installed on every EC. Furthermore it supervises the status of the EC and the connected devices.

Process control by the timing system enables an easy way of fast switching between different accelerator cycles. To do so for every device 16 datasets for different accelerating cycles are stored on the EC. Selection and thus establishing the generated cycle is by a dataset number as part of the delivered event code. So the timing system determines the sequence of accelerating cycles with switching from pulse to pulse. This allows alternative supply of several experiments, each with different beams, including the injection to the next accelerator in sequence.

For a detailed description see reference [3].

VII. COMMAND LEVEL

The task of all components between the operating and the EC (implemented in hardware as well as in software) is to link the operational representation of devices to the device handling on an EC. ECs are linked to the operating by the MP to not burden the EC. The additional communication processor CP only manages the ETHERNET handling and is mainly transparent.

The MP has to evaluate commands from the operating level and to send back the responses.

EM specific command evaluation is coded in modules, called **U**ser **S**ervice **R**outines (USR). Every existing combination of property/property class/EM is represented by one USR on every MP where the EM is realized. Calling a property/property class simply results in execution of the associated USR. An USR may do any combination of data processing like evaluation of measurements, sending data to EC or receiving data from it, calling an EQM or another USR and performing consistency checks of exchanged data.

USRs are called by the **M**aster processor **OP**erating **S**ystem (MOPS). Since commands are sent in standardized form, analysis, dispatching and sending results to the operating level is the same for all devices and can be handled completely within MOPS. The handling of connected commands, i.e. periodically calling an USR, is implemented in MOPS, too. MOPS supervises and displays the status of MP and assigned ECs. Extensive multitasking allows quasi-parallel execution of commands on the MP level.

VIII. AUTOCONFIGURATION

Nomenclatures, device numbers and property description are stored on device level in local data bases, one per MP. From these databases the CDB, needed for device accessing, is automatically updated:

— At startup of an operating VAX it collects the local databases of all MPs to form the CDB,

— At startup of a MP the local database is sent to all control VAXes for update of the CDB.

A hierarchical still-alive supervision is implemented on every level of the controls system.

IX. CONCLUSION

The control system is in use since commissioning of the new facilities three years ago. During this period no severe problems were encountered. Routine operation of the machines now shows a reliable system. This experience confirms the structural concepts of the system being flexible enough to implement additional requirements and features.

Major effort now is to upgrade the old facilities and to install additional beam diagnostics both resulting in adding a lot of devices to the system. On the operating level main focus is in clarifying and unifying the complex handling and thus enabling a simpler and more reliable routine operation.

We would like to take the opportunity to thank all our colleagues who have contributed to the success of this control system.

X. REFERENCE

[1] V. Schaa, G. Fröhlich, M. Balloff, Jun-Ye Wang, "A New Generation of Operating Programs for the Accelerators", GSI Scientific Report 1987, p. 357

[2] M.C. Crowley-Milling and G. Shering, "The NODAL System for the SPS", CERN 78-07 (3 September 1978).

[3] R. Steiner, C. Riedel, W. Rösch, "Real Time Aspects of Puls to Puls Modulation", this conference.