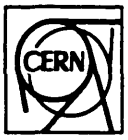
Sec .



EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN - PS DIVISION

CERN/PS 93-46 (CO)

CONTROL PROTOCOL
LARGE SCALE IMPLEMENTATION AT THE CERN/PS COMPLEX
A FIRST ASSESSMENT

H. Abie, G. Benincasa, G. Coudert, Y. Davydenko(*), C. Dehavay, R. Gavaggio, G. Gelato, W. Heinze, M. Legras, H. Lustig, L. Mérard,T. Pearson, P.Strubin, J.Tedesco

(*) CERN and IHEP Serpukov, Russia

International Conference on Accelerator and Large Experimental Physics Control Systems
Berlin, Germany, October 18-22, 1993

Geneva, Switzerland
15/11/93

CONTROL PROTOCOL : LARGE SCALE IMPLEMENTATION AT THE CERN PS COMPLEX - A FIRST ASSESSMENT

H. Abie, G. Benincasa, G. Coudert, Y. Davydenko(*), C. Dehavay, R. Gavaggio, G. Gelato, W. Heinze, M. Legras, H. Lustig, L. Merard, T. Pearson, P.Strubin, J.Tedesco
CERN, European Organization for Nuclear Research, 1211 Geneva 23, Switzerland

(*) CERN and IHEP Serpukov, Russia

The Control Protocol is a model-based, uniform access procedure from a control system to accelerator equipment. It was proposed at CERN about 5 years ago and prototypes were developed during the last years. More recently, this procedure has been finalized and implemented at a large scale in the PS Complex. More than 300 equipment are now using this protocol in normal operation and another 300 are under implementation. This includes power converters, vacuum systems, beam instrumentation devices, RF equipment, etc.. This paper describes how the general, unique procedure is applied to the different kinds of equipment. The obtained advantages are also discussed.

I. INTRODUCTION

Uniform equipment access for different kinds of accelerator devices has been investigated at CERN, in collaboration with other Laboratories, during the last few years.

The first basic ideas of this study have been published in 1989 [1], whilst the results obtained with the first prototype implementations have been reported in 1991 [2]. Since then the various investigated options have been frozen and a unique software structure has been proposed [3], ready to be implemented in any generic control system and, in particular, in the new CERN accelerators Control System [4].

Large scale implementations represent , in general, the best way of testing the validity of any new technique: the rejuvenation project of the PS control system (see a paper [8] in this Conference) has provided the ideal frame for such large scale test. In this project, hundreds of pieces of equipment of various type have to be interfaced to the new control system using new hardware and software.

The characteristics of the control protocols, both in the general design phase and in implementations, are exhaustively reported in References. However, for sake of efficiency, we provide the reader with a short overview in the next chapter.

II. SHORT DESCRIPTION OF CONTROL PROTOCOLS

An essential goal of any process control system is to provide users with a uniform and efficient access procedure to equipment. Uniformity and efficiency have been achieved at CERN through two separate, but complementary activities : modelization and definition of a general software structure.

II-1 Models [5]

Eminent specialists of the CERN accelerator equipment have produced static and dynamic models of the most common families of devices : power converters, vacuum systems and beam instrumentation. These models, independent of any hardware environment, permit to describe the devices as a collection of functionalities that are common to a family of equipment. Each functionality is subsequently decomposed into its essential parameters having variables (dynamic data) and attributes (static data).

At last, the resulting amount of information has been described using an appropriate data structure: this data structure is unique for a given family of devices. At this point, any control on a device belonging to one of the considered families can be performed by adequately exchanging messages based on this data structure.

II-2 A general software structure

The essential elements composing such software structure are represented in Fig. 1. The structure is based on a clean separation between common activities (top of the figure) that are general for a family of devices, and specific activities (bottom of the figure) that are peculiar to each single equipment. In actual implementations this means that the first ones could be written only once per family, whilst the second ones, containing all the hardware and operational intricacies of a device, must be specific to

each equipment or group of similar equipment. The two entities communicate each other by exchanging command and acquisition messages. The messages contain the data structures defined in the models with their data values.

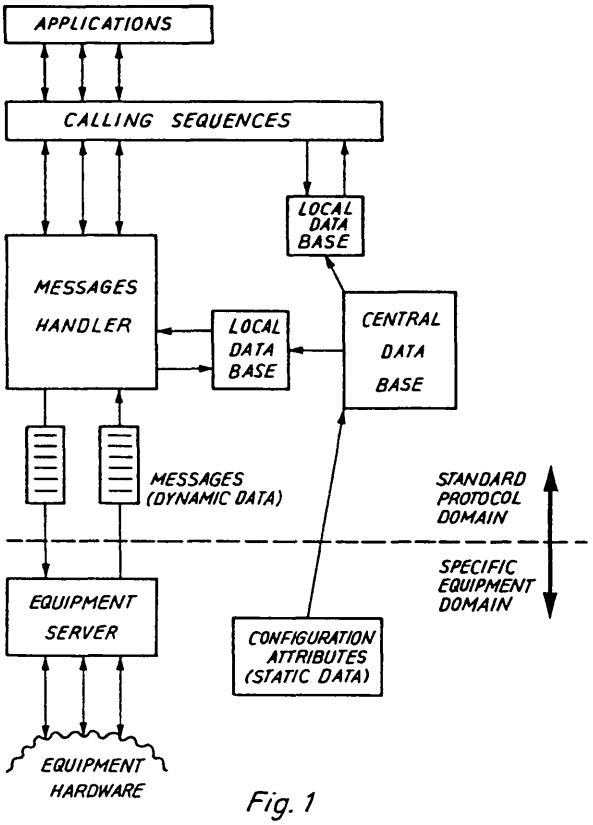


Fig. 1

GENERAL IMPLEMENTATION SCHEME

The heart of the common activities is the so-called Message Handler. It receives commands ("what" to do) from the upper layer (calling sequences) and produces (or receives) standard messages for (or from) the bottom layer (equipment server).

The Data Base contains the necessary information to drive all the devices connected to the control system. This means both the common information needed by a family of devices (essentially the data structures of the messages), but also the information specific to each single device (limits, number, type and formats of the variables, etc.). This second kind of information is originally housed at the level of the specific activities (Configuration) and is subsequently introduced into the Data Base.

For its size the Data Base is usually remotely housed: to speed up the access operations at the run time, a relevant part of it can then be transferred into a local Data Base, in the environment of the Message Handler.

At the top of the common activities are the Calling Sequences: they represent the external visibility of the devices and provide the necessary interface with the application programs. A Calling Sequence usually contains, amongst others, four essential parameters:

- Device_Name, a unique reference to a physical device,
- Action_Name, expressing "what" to do on the device,
- Conditions are usually process or real time events governing the Action,
- Data are usually associated to each Action.

The use of models has permitted to define, for each family of devices, a precise set of Actions with their associated Data structures. To appropriately encode this information for the Message Handler, the Calling Sequence software usually needs some information contained in a local data base.

At the bottom of the control chain (specific activity) is the Equipment Server. As already mentioned it is in charge of all specific control activities for a given device: it should not follow any particular constraints in its treatment, provided it accepts and produces the standard messages.

III IMPLEMENTATIONS AT CERN PS COMPLEX

The CERN PS accelerator control system has a three levels architecture, that can be considered as "standard model" for modern control systems [4]:

- The User-Interaction level contains powerful workstations (DEC) all running ULTRIX. Other central servers are used at this level for general services. This level is connected to the next one via an Ethernet network using the TCP/IP protocol.
- The Front End Processor (or FEP) level contains VME crates with 32-bit Motorola processors or industrial PC's : both are called DSC for device stub controller. Both processors run a RT UNIX operating system, Lynx OS, and are diskless: local servers are then used for file storage. "C" is the standard language.
- The Equipment level usually contains intelligent apparatus of various types (mainly G64) connected to a FEP via a field bus: the most popular are MIL 1553 and Camac, but other connections are also used. No hardware or software standards are fixed at this level.

In this control architecture the equipment access is realized using a software structure called Equipment Module (EM), described in Ref. [6]. The EM provides to the application programs a standard Calling Sequences that includes, amongst others, the following essential parameters: Family_Name, Member_Number, Property, Cycle_Number, Data for which Property indicates the selected action and Cycle_Number of the specific machine cycle.

The body of an EM is composed of a Data Table (DT) containing all necessary information to drive a device and a series of routines (called Procos for Property Code) each one dedicated to the execution of a particular Property. Where requested, a RT Task has been added to this scheme, permitting to execute the command and acquisition actions in synchronization with the process.
The protocol software functionalities of II-2 have been implemented using the EM tools. In the Calling Sequences, the Device_Name corresponds to the couple Family_Name and Member_Number, the Action_Name is the Property and the Condition is the Cycle_Number. The necessary information contained in the Local Data base is housed in the Data Table. The DT is created from a central Data Base (ORACLE) housed in a dedicated server. The messages have been implemented as C language structures.
A message is composed of an Header and a certain number of fields, each field containing the information related to one of the functionalities identified in the models [5]. The Header is always present with fixed format and structure; the other fields can be present or not, their structure is fixed but their contents (number of parameters, format etc.) is reported in the DT for each device or family of devices. The Header contains fundamental information for the Equipment server such as the identification of the device and the requested service (see also VII).

Basically, three implementations have been produced, each one best adapted to one specific family of devices, but all based on the described software structure.

IV CONTROL OF POWER CONVERTERS AND ASSIMILATED EQUIPMENT [9]

About 130 Power converters are already operational in the proton Linac, and 70 more are under development for the lead ion Linac. Moreover, it resulted from a posteriori investigation that the same control protocol used for power converters could also be used for other kind of equipment for which no specific behavioural models were provided. This is the case for about 25 RF Cavity stations and about 55 stepping or DC motors used to actuate various kind of mechanical devices: all this equipment uses similar hardware and software layout as for power converters that is reported in Fig. 2.

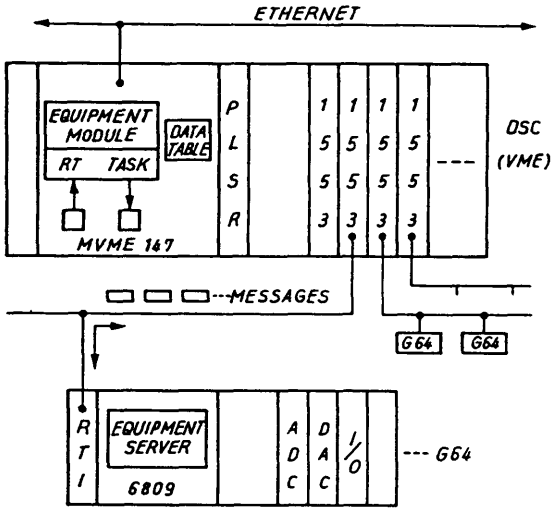


Fig. 2
CONTROL OF POWER CONVERTERS

The concerned power converters constitute a large cluster of devices having very similar characteristics each other and not demanding huge or sophisticated computations. For this reason the economical solution implemented in the PS is housing the specific hardware modules in G64 crates connected to the DSC with the MIL 1553 fieldbus. In the G64 crates, simple but adequate 8-bit 6809 processors, programmed in Pascal, provide all requested actions on the specific hardware. In this way one quite expensive DSC controls an entire cluster of about 70 power converters using four 1553 interfaces. A unique EM (called POW) is housed in the DSC, in charge of all power converters.The messages pass through the 1553 fieldbus and are sent and received using an appropriate driver (QUICKDATA, developed at the SL Division). The previously mentioned RT task provides synchronization with the PS cycles(~1.2 sec). The information permitting to associate the appropriate setting of the values to the present machine cycle is contained in a hardware module called PLSR (program line sequence receiver). At each machine cycle, the RT task, using the PLSR information, selects in the Data Table the appropriate message to be sent to the G64; in the complementary way, the RT requires also from the G64 an acquisition message that is stored in the Data Table

in a position corresponding to the current machine cycle. When an operator (using an application program and an appropriate calling sequence) sends a command or requires an acquisition for a given machine cycle (Condition), his request is stored into, or his data are taken from the Data Table which is regularly up-dated by the RT task.

In the case of the two Linacs, only two synchronization events are necessary. The first interrupt is sent to the G64 software, just before the passage of the beam. The local software executes the necessary acquisitions, prepares the acquisition messages and stores them in the appropriate output buffer of the 1553 Remote Terminal Interface (RTI). The second one, after the passage of the beam, is used by the RT task in the DSC. The RT task, using the 1553 driver, retrieves the acquisition message and then sends the appropriate command message to the input buffer of the RTI. Given the simple operational needs of this family of power converters, the structure of the messages is also quite simple; for example, the command message, after the omnipresent Header, contains only the status control field and a maximum of 4 setting parameters. The whole message consists of 44 bytes and needs about 1.3 ms to be transferred to the equipment via MIL 1553.

As usual, the information permitting to adequately treat the various parameters for each specific power converter is contained in the Data Table. An important feature of the POW Equipment Module is the capability of accessing in a similar manner new (protocol based) and old devices. In fact, to preserve the important software investment represented by the application programs used in PS Complex to drive various hundreds of Camac based power converters, it was mandatory to implement an identical calling sequence in both cases. An appropriate mechanism permits to select the old or new implementation.

V CONTROL OF BEAM INSTRUMENTATION DEVICES

Beam instrumentation devices present the greatest variety of different implementations. It is certainly in this field that the control protocol has demonstrated its flexibility and adaptability.

The control protocol has been used, or will be used (in brackets) for the following devices:

- 33 (6) current beam transformers used for intensity measurements,
- 8 (56) pick-up stations used for position measurements,
- 1 (1) FFT (Fast Fourier Transform) based, Q-measurement system,
- (4) wire beam scanner, profile measurement system.

It should also be noted that, inside a same family of devices, great variety of developments exists: for example, the generic indication of "beam current transformers" hides at least half a dozen of implementations, totally different each. In contrast with the previously described power converters, the treatment of beam instrumentation devices requires, in general, huge computations and fast hardware access. For this reason it was decided that both the control hardware and the specific software (Equipment server) should be housed inside the DSC, as shown in Fig. 3.

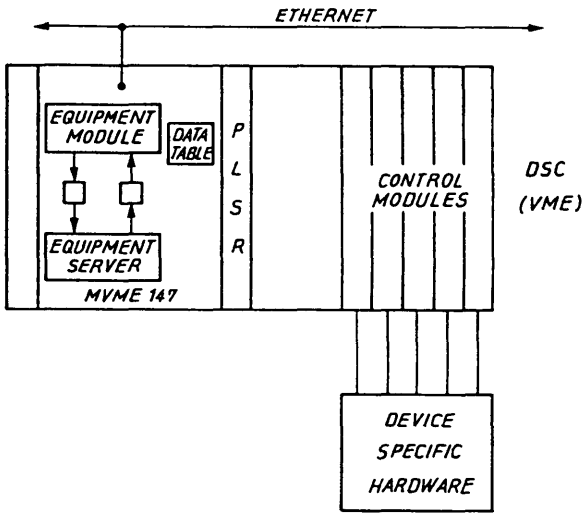


Fig. 3

CONTROL OF BEAM INSTRUMENTATION

In general, a separate DSC is dedicated to a single device or to a cluster of similar devices. The Equipment Module and the Equipment server share the same processor: they exchange the various messages using the Lynx OS message queue facilities. The activity of the RT task (see III) is now totally at the charge of the Equipment server that can easily access the PLS information.
Various interrupts and triggers permit the equipment server to execute at precice instants in the machine cycles the required control and acquisition activities. A unique basic version of an EM for instrumentation has been written, however, a separate compilation for each type of devices permits to obtain Equipment Modules having the Family_Name better adapted to the concerned devices: Trafo, Pikup, Qmea etc. Despite the mentioned variety of the considered

devices, the overall software structure has always fulfilled all requirements.
As already mentioned, the Equipment server is the only software part that should be appropriately written for each considered device; however, certain activities, such as the communications with the Equipment Module and the various actions depending on the real time constraints of the accelerator processes look very similar from one device to another. For this reason a group of device specialists [7] has produced some kind of frame where these common activities are already treated, leaving to a new application the care of introducing only what is specific to a single device. This facility has produced an important standardization also at the level of the Equipment server, has strongly reduced the development times and is now used in all new implementations.

VI CONTROL OF VACUUM SYSTEMS

The vacuum system is principally composed of three kinds of equipment: pumps (of various types), gauges (essentially Pirani and Penning measurement devices) and valves. For the three types adequate models have been provided by vacuum system specialists.

The control of vacuum devices presents some peculiarities not found before. Firstly, there is often inter-dependency between various equipment: for example, to open a valve one must execute a sequence that includes the pressure check in adjacent sectors. Secondly, the used devices include industrial products for which no standard interfaces are provided in the PS control system. For this reason, neither the layout used for power converters (independent devices) nor that used for instrumentation (standard VME control modules) are adequate. The used layout is represented in Fig. 4.
To take into account existing hardware and industrial product, the RS-232 standard has been used instead of the Controls standard MIL 1553; the RS-232 interfaces have been multiplexed using X25 hardware that is accessed by the DSC.
The RS-232 connects G64 crates, controlling essentially pumps and valves, and industrial standards controlling Pirani and Penning gauges.

The Equipment server has here a particular configuration: it is in fact split into two parts, one resident in the DSC and another spread in the various G64 crates. The first one is principally in charge of: i) communicating with the Equipment Module using the standard protocol messages, ii) treating the before mentioned inter-connections between various equipment and iii) preparing and sending appropriate commands to each single device, using the required standard. The second one executes the appropriate control sequence for each single device.
Because the vacuum systems are static, a RT task, synchronized with the process, is not required as in previous cases: however, because the used hardware produces poor access times, a continuously running RT task (not shown in Fig. 4) has been added, that updates the Data Table.

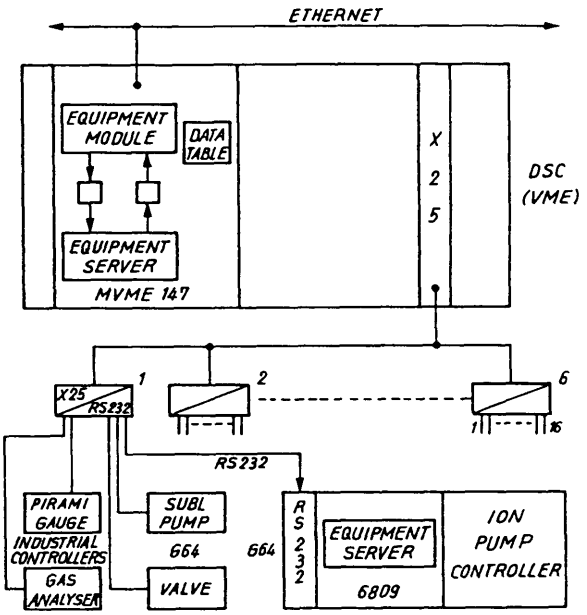


Fig. 4
CONTROL OF VACUUM SYSTEMS

The following equipment is already in use or (in brackets) is under installation:
- 45 (120) pumps of three different types
- 20 (48) Pirani and Penning measurement gauges
- 10 (18) sector valves.

VII COMMON FEATURES AND SOME ASSESSMENT

The reported implementations largely demonstrate that the Control Protocol provides an adequate unique tool to access most of the accelerators devices. We think that the reasons of this success principally resides in a balanced use of fixed rules (without them there is no standardization) and flexibility (without it the implementations in real world would be very limited).

VII-1 Fixed rules

The structure of the messages with their different fields and the data structure of the various parameters are the main rules which any implementation should comply with: they are reported in the specifications [3]. This does not mean that all the features described in [3] must be implemented in each device but that, if a feature is used, it must follow the rules. We want to recall three important standard features, not reported before:

i) Error treatement

Three error treatment levels have been implemented:

- A summary of all anomalous situations is provided in all acquisition messages: Physical status (operational, partially operational, not operational, needs commissioning) complies with the ISO rules; Operational aspects (no-connection, local, remote, locked).
- Status qualifiers (interlock, non-resettable fault, resettable fault, busy, warning); a Busy time indicates for how long (in sec) the device is busy. The meaning of these indicators is precisely described in the specifications [3].
- A second level called Detailed status is invoked with an appropriate service (see below): for each one of the status qualifier indicators, the specialist can define up to 32 detailed fault conditions. Their meaning is free and it will be described in the data base.
- At the third level a history of the faults is provided: the Equipment server can store with a date as many detailed status messages in a ring buffer as he wants. Appropriate services (see below) permit their retrieval.

ii) Global actions

They are simple atomic actions (no data) provoking well defined sequences in the device. Examples are the Status controler actions such as On, Off, Open, Close, Reset, Recovery etc.:

- Reset means to restore the setting conditions of the device to default values predetermined for each device by the specialist or by the operation crew.
- Recovery is similar, but the setting values are those of the last available operation.

iii) Date

Every command and acquisition message contains a date tag with defined precision and resolution: this permits, for example, to unambiguously verify in time the coherence of measurements executed by different devices geographycally spread in a large area.

VII-2 Flexibility

The general software structure presented in II-2 (Fig.1) can be developed in any modern control system: the best proof is that it has been implemented without major pain in the existing PS control environment, so preserving software investment and local traditions. Many other features exist providing further flexibility. We just recall two of them:

i) Services

Apart from the described command and acquisition services, a series of other useful services are defined in the Control protocol: they are all of the immediate command-response type. A short message containing only the Header (III) with the appropriate Requested service is sent to the Equipment server: in response one obtains the required message. A dozen of different services are already used (read back of control messages, detailed status information, history of errors etc.) but their number can be easily extended. When a new service is proposed and assessed as having a general interest for the user, it is simply added in a list and documented in [3].

ii) Specialist actions

By putting a code different from zero in a field called "Specialist action" in the message Header, the rest of the message is interpreted as a non standard one: the structure foreseen for the specific device is not used and the data are transmitted as row data to the Equipment server. This facility is strongly used by several devices (especially beam instruments): information such as calibration factors, offsets, scaling factors etc., are in this way introduced in, or extracted from the Equipment server without interference with the normal operation actions. In contrast with services, these actions are not reported in the specifications [3]: they are in general defined, known and used only by the specialists of each device.

VIII CONCLUSIONS

Evident benefits are produced by the use of the Control protocol in various key fields, such as simplification in the design of application programs, improved maintenability and reliability, overall software costs etc. These advantages are in general present in any kind of standardization already in use at CERN and in other laboratories, but they are often obtained per family of devices and are based on a standardization of the control hardware. On the contrary, the use of behavioural models and the flexibility of the described software structure permit to use the Control protocol for most of the

accelerator equipment. As already mentioned , most of the functionalities identified in the models are so general in the field of accelerator equipment, that the Control protocol has been used for families of devices not foreseen at the beginning of the project (see IV).

It is not easy to quantify the total economy in software costs, without falling in demagogy. We only recall that several hundreds of man-years are usually spent in a complex such as the PS for control software, a large fraction being dedicated to the equipment interface. By the experience already obtained with the present implementations, one can infer that the economy for the PS Complex will largely justify the initial investment in this project.

REFERENCES

[1] R. Bailey, G. Baribaud et al. , " Operational protocol for controlling accelerator equipment" ICALEPCS '89, Vancouver, Canada.

[2] G. Baribaud, I. Barnett et al. , " Control protocol: the proposed new CERN standard access procedure to accelerator equipment. Status report" ICALEPCS 91, Tsukuba, Japan.

[3] G. Benincasa, A. Burns et al. , " Final report on the uniform equipment access at CERN", CERN Internal Report PS 93-16, 1993.

[4] The CERN PS and SL Control Groups "PS/SL controls consolidation project", CERN Internal PS/CO Note 91-09 and SL/CO Note 91-12, 1991.

[5] G. Baribaud, I. Barnett et al. " Generalities on the operational control protocol" CERN Internal Note PS/CO 91-01, 1991.

[6] J. Cuperus, W. Heinze, C.H. Sicard, "Control Module handbook", CERN Note PS/CO 91-25, 1991.

[7] M. Legras, J. Tedesco, "Application typique du protocole pour l'instrumentation", CERN Internal Note PS/BD 93-02, 1993.

[8] F. Perriollat for the PS/CO Group. "The new CERN PS control system; overview and status ", ICALEPCS '93, Berlin, Germany.

[9] W. Heinze, "Power supply control for the Linac 2", CERN Internal PS/CO Note 93-34, 1993.