W. Kehrer/PS

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN/PS/92-57 (CO)
November 1992

THE CERN CONTROL PROTOCOL FOR POWER CONVERTERS

I. Barnett, G. Benincasa, O. Berrig, R. Brun, P. Burla, G. Coudert,
J.G. Pett, R. Pittin, J. P. Royer, N. Trofimov*
* CERN and IHEP Serpukov

CERN, CH-1211 Geneva 23, Switzerland

Reported By G. Benincasa and P. Burla

Invited Paper at the 1992 Nuclear Science Symposium,
Orlando, USA. 25-31 October 1992

and submitted to Transactions Special Issue of the IEEE Transactions on Nuclear Science (TNS), 1992

Geneva, Switzerland

THE CERN CONTROL PROTOCOL FOR POWER CONVERTERS

I. Barnett, G. Benincasa, O. Berrig, R. Brun, P. Burla, G. Coudert,
J.G. Pett, R. Pittin, J. P. Royer, N. Trofimov*
* CERN and IHEP Serpukov

CERN, CH-1211 Geneva 23, Switzerland

Reported By G. Benincasa and P. Burla

Abstract

The Control Protocols provide, for a class of similar devices, a unique and standard access procedure from the control system. Behavioural models have been proposed for the different kinds of Power Converters and the corresponding functionalities, with their parameters, variables and attributes have been identified. The resulting data structures have been presented using the ISO ASN.1 metalanguage, that permits universal representation independent of any computer environment. Implementations in the UNIX-based CERN accelerator control systems are under development.

I. INTRODUCTION

Standardization and uniformization of equipment access is not a novelty: usually specialists responsible for the controls of a substantial number of similar objects (power converters, vacuum systems, beam instrumentation devices, etc.) introduce a standard. Often standardization is based on the actual implementation of a particular series or in a particular context.

For a more general standardization that could be to a large extent independent of actual implementations, an approach based on the use of operational behavioural models has been proposed at CERN and is described in this paper [1].

The work has been carried out through two complementary phases:

A) a Conceptual Design Phase where static and dynamic models have been proposed and subsequently analysed to identify common activities.

B) an Implementation Phase where a unique software access procedure has been studied and developed in the common CERN Accelerator Control System .

The result of these two activities is called the Control Protocol.

In this paper, we describe in more details the design phase for of a Control Protocol applied to the power converters; a description of the implementation phase, such as it is presently developed at CERN in a cluster of 120 power converters, is also reported.

II. CONCEPTUAL DESIGN PHASE

The Design Phase has been divided in a sequence of interdependent activities:

- Design of appropriate models for the different kind of Power Converters.
- Identification of the Functionalities (see below) composing the models.
- For each Functionality, identification of the parameters with their variables and attributes.
- High level representation of the resulting data structures.

Where possible and convenient, International Standard (ISO rules) have been to some extent introduced in the design to ease the integration of products developed elsewhere.

II-1 Behavioural Models [2,3,8]

A model is an abstract representation of the behaviour of an object, in our case Power Converters: no details of actual implementations are necessary to design a model. Implementation intricacies should be hidden, as far as possible, inside the device and only those characteristics necessary for the operation of the power converter should be externally visible: this feature corresponds also to the definition of the Virtual Manufacturing Device (VMD) given in the ISO-MMS specifications [4].

The operational behaviour of most of CERN power converters families (dc, pulsed, ramped, driven by function generators, etc.) has been analysed, together with that other equipment often necessary for the accomplishment of their activity: for example, the function generators and the triggering and sequencing systems.

During normal operation a power converter can be found in one of a limited number of stable situations called States, or in unstable situations called State Transitions. State Transitions are provoked by voluntary (e.g. an operator's command) or involuntary actions (e.g. the trip to a faulty situation).

All the relevant States and voluntary actions have been identified, listed and precise rules issued to describe their meaning and their behaviour. Involuntary actions (also called exceptions) have been listed (grouped in classes) and the given rules concern essentially the recovery procedures.

An appropriate graphical representation of States, State Transitions and actions constitutes the so-called State Diagram. A modest number of such State Diagrams can cover any operational situation.

As an example we report in Fig. 1 the Static State Diagram for a generic power converter. The circles represent the three identified Static States, ON, OFF and Stand-By, and the bold arrows interconnecting the circles represent the action (and then the State Transitions) required to pass from a State to another one. For sake of completeness, we have also represented in dotted lines the occurrences of two kinds of exceptions called Resettable Fault and Unresettable Fault, with their recovery rules.

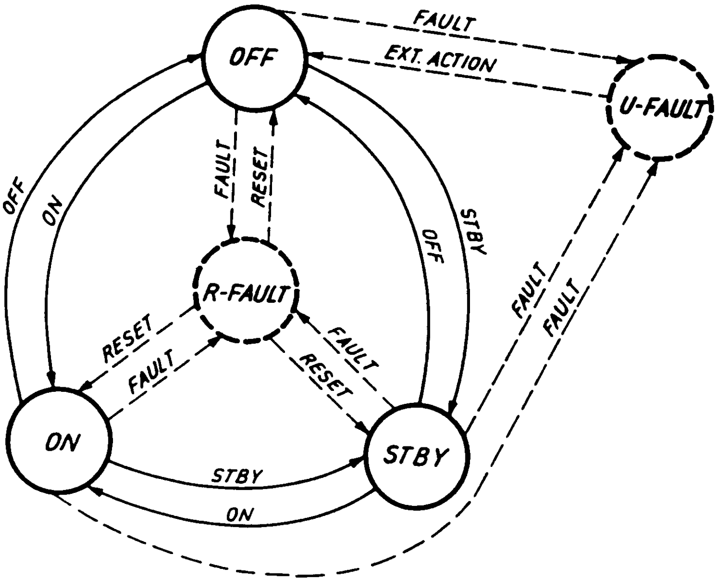

Figure 1

II-2 Functionalities

Direct implementation based on the identified models (i.e. one protocol per model or per State Diagram) is not suitable for, at least, two reasons:

- they would produce too many procedures, and our goal was to strongly limit them,
- the risk that, at each slight difference in a new device, a new model could be invented, with an endless proliferation of procedures.

Despite their inherent differences, the models are always composed of sequences of States, Actions and State Transitions that can be adequately represented by a limited number of sets of parameters.

This has naturally brought the idea of defining Functionalities as the sets of all those activities that have a common goal in the operational sense.

Any operational situation and consequently any behavioural model can be described by using an appropriate set of Functionalities. Five Functionalities have been identified from the power converter models:

- Status Controller
- Setting Actuator
- Measurement Actuator
- Trigger Sequencer
- Function Generator

At a certain extent the meaning of Functionality joins the definition of VMD Capability, given in the ISO-MMS rules [4].

Each Functionality is described by an exhaustive list of parameters, variables and attributes:

- a parameter describes a particular activity inside the Functionality,
- a set of variables is associated to each parameter: they contain values changing during operation,
- attributes contain fixed data of the device, or data changed very seldom by specialists.

II-3 Data Representation

The significant amount of inter-related data produced in the previous representation requires a universal data structure representation tool. We have used the ISO standard Abstract Syntax Notation One (ASN.1) [5]. ASN.1 uses a simple metalanguage and permits data structure representations independent of any computer environment. Moreover, if wanted, coding rules exists, producing a binary translation (Transfer Syntax) ready for communication network.

As an example, we show the ASN.1 representation of the Functionality Status Controller. We briefly recall that ASN.1 essentially defines data structures as description of data Types: simple Types (primitives) are expressed in capital letters and new Types are defined by user (underscored) at any degree of nesting (definitions inside other definitions), to the last Type containing only primitives.The symbols -- are used for comments.

A) Control Parameters

```
Actuation          ::= ENUMERATED
       { A.Off(1)
         A.Stby(2)
         A.On(3)
         A.Reset(4)  }
```

B) Acquisition Parameters

```
StatusAcquisition   ::= SET
{ext-physical-status  PhysicalStatus --ISO-MMS standard--
availability          ExternalAspect
internal-state        StaticState
state-qualifier       StateQualifier --exceptions--
busy-time             REAL --in seconds--
status-detaill        IndicatorList --details of exceptions-- }
```

Each new underscored Type must now be described

```
PhysicalStatus      ::= ENUMERATED
   { Operational(1),
     PartiallyOperational(2),
     NotOperational(3),
     NeedsCommissioning(4)  }

ExternalAspects     ::= ENUMERATED
   { NoConnection(1),
     Local(2),
     Remote(3)  }

StaticState         ::= ENUMERATED
   { S.Off(1),
     S.Stby(2),
     S.On(3)  }

StateQualifiers     ::= SEQUENCE OF BOOLEANS

IndicatorList       ::= SET OF IndicatorRecord
```

IndicatorRecord ::= SEQUENCE
{name VISIBLESTRING, -- name of the exception--
indicator BOOLEAN.--exception present or not--
last-change Date --last change of exception-- }

Date ::=VISIBLESTRING --YYMMDDhhmmss --

Attributes for the Status Controller Functionality are not reported here.

II-4 Configuration

The identified Functionalities, with their corresponding parameters, variables and attributes, are sufficient to design general access procedures for most actual power converters. In many cases, a given equipment does not require all the Functionalities (e.g. no Function generator) or, inside a given Functionality, could not use a certain number of parameters or variables (e.g. not have the STBY Static State). Moreover the values of attributes must be specified for each single device: for example, the exception messages (see before) must be specified, etc. The list of all used possibilities and of the attribute values for a given, actual power converter, is called its Configuration. The Configuration transforms a general purpose data structure into a specific tailored tool to control this or that single device.

III IMPLEMENTATION PHASE AT CERN

Equipment access procedures based on the proposed models and data representation can be easily implemented in any control environment: we briefly describe an implementation in a typical, three level control architecture, such as the CERN Accelerators Control System [6].

The three level architecture is represented in Fig. 2.
The first level is based on workstations (Dec or Apollo workstations running Unix) and provides facilities for MMI and for Applications.

The second level, called Front-End Processors or FEP, (VME crates housing 680xx processors running RT-Unix) contains equipment-oriented facilities.

The two levels are interconneted through an appropriate LAN (Ethernet or Token Ring).

A dedicated server houses an interactive Oracle database containing, amongst other things, an adequate description of all controlled equipment.

The control hardware is directly housed in the FEP or, as for the most of power converters, a third less expensive level is provided, based essentially on G64 crates running simple OS. A MIL 1553 field bus is used in this case to connect a cluster of such crates to a single FEP.

In this controls architecture (Fig. 2), the Control protocol has provided a strong standardization at both ends of the control chain: the equipment call in the workstations, and the interface with the power converters device in the FEP.

It can be described as the sequence of three software entities [6] [7]:

- the calling sequences
- the messages handler
- the equipment server

III-1 The Calling Sequences

Application programs running in the Workstations access any Power Converter using a unique calling sequence having standard parameters:

EQUIP(Name, Action, Event, Data, Error)

-- "name" is a unique name assigned at CERN to every power converter;

-- "action" is the name of a specific invoked service: all the parameters and variables (or groups of variables) previously described in models have appropriate services to access them;

-- "event" indicates (if used) the conditions governing the execution of the requested action.
They could be a specific trigger, a given accelerator cycle, a type of particle etc.

-- "data" contains (if used) the information values associated with the specified action (single value, array of values, structures).
Each action has assigned its own structure of data derived from the model: in certain cases the structure is unique for all the power converter classes, in others it is assigned to each instance and is defined in the Configuration (see II-4).

-- "error" contains, at the end of the call, all reported exceptions, using an appropriate hierarchy.

When EQUIP is called, first it collects the necessary information from local or external databases (device address, Configuration etc.) using the parameter "name" as a search keyword. This information is subsequently packed in network format and it is sent to the appropriate FEP using Remote Procedure Calls (RPC).

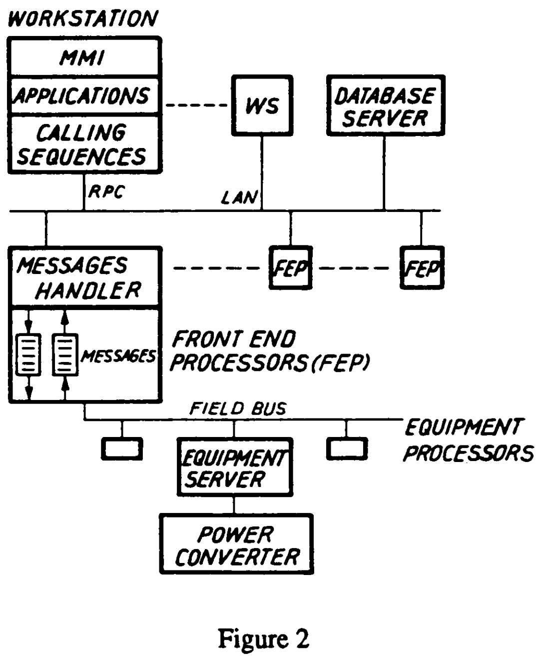

Figure 2

III-2 The Message Handler

A copy of a unique Message Handler software package is housed in each FEP: it receives the previously described calling sequence and produces standard message frames for equipment.

The previously described ASN.1 data structure has been easily translated into a "C" structure, the language used in our environment. A message frame is a template where the Functionalities with the values of their parameters and variables have assigned predetermined fields.

This somewhat rigid definition of frames derives from the willingness of serving with a unique protocol a large variety of power converters, some having very powerful processors and others with limited treatment capacity, and of providing portability in various environments.

All the power converters using the same set of Functionalities, parameters and variables belong to the same "type" and use the same message format (still contained in Configuration).

Upon reception of the calling sequence packet, the Message Handler first selects the appropriate message frame for this power converter, then it fills the field corresponding to the requested action, and finally it sends the complete frame to the equipment: several fields can be filled (at the extreme all the frame) before sending the message, to cope with operational requirements.

A similar strategy is adopted for acquisition messages going from equipment to the applications.

III-3 The Equipment Server

The Equipment Server is in charge of all control and acquisition sequences, specific to a given power converter: it receives standard control frames from the Message Handler and produces other standard acquisition frames for it. Once the rules of the protocol accepted and, in particular, once the kind of exchanged messages (Configuration and "type") agreed, the device specialist can write his server in the most appropriate way for the specific device.

IV. CONCLUSIONS

Equipment access procedures, as described in this paper, are under development for a first cluster of 120 power converters for the linear accelerator LINAC 2 of the PS Complex. Results are expected for the end of 1992.

Implementations based on the same procedures but in a slighthly different configuration (Equipment Server directly housed in the FEP) are under consideration for another cluster of power converters.

The described principles of Control Protocol have also been implemented for other families of devices: beam instrumentation devices, vacuum systems, etc. Various applications are already running (mainly beam instrumentation devices), thus confirming most of the Control Protocol advantages.

V. REFERENCES

[1] G. Baribaud et al., reported by G. Benincasa. "Generalities o the operational control Protocol", CERN Internal Note PS/CO 91-01 (1991).

[2] G. Bonthond et al. "Definition of a behaviour models of power converters", CERN PS/CO/ Internal WP 88-23 (1989).

[3] I. Barnett et al., reported by P. Burla, "Formal description of the parameters and variables of power converters for defining a control protocol", CERN Internal Note SL/PC 91-86 (1991).

[4] ISO/DIS 9506-1 and 9506-2, "Manufacturing message specification" (1988).

[5] ISO 8824" IPS-OSI - "Specification of Abstract Syntax Notation One (ASN.1)" (1987).

[6] The PS and SL Control Groups,."PS/SL controls consolidation Project", CERN PS/CO 91-09 and CERN SL/CO 91-12.

[7] G. Benincasa et al., Minutes of "Uniformisation of Software Access Procedures (USAP) meetings", CERN PS/CO/Min. 92-28, 92-38, 92-47.

[8] G. Baribaud, "Operational Protocol for Timing Events", CERN SL/BI Internal Note 90-5.

Distribution (of abstract)

PS, SL, AT, ECP Scientific Staff