Representation and Usage of Knowledge for Initialization of Accelerator Control Equipment

J.M. Bouché [a)], G. Daems [a)], V. Filimonov [b)], V. Khomoutnikov [b)], F. Perriollat [a)], Yu. Ryabov [b)]
[a)] CERN, CH-1211 Geneva 23, Switzerland
[b)] PNPI, Gatchina, St.Petersburg, 188350 Russia

Abstract

A knowledge based application, called SETUP, to initialize and diagnose the CERN/PS accelerators' control equipment is described. The object model and the general features of control algorithms are presented, together with their relation to the knowledge description of the setting up of the system .The different ways of the integration of the SETUP in the control system are outlined.

I. Introduction

The purpose of this paper is to present the main ideas and solutions as well as the experience in the use of a knowledge based facility (called SETUP) implemented in the CERN Proton Synchrotron control system (Fig. 1) for initialization and testing of accelerators' control equipment.

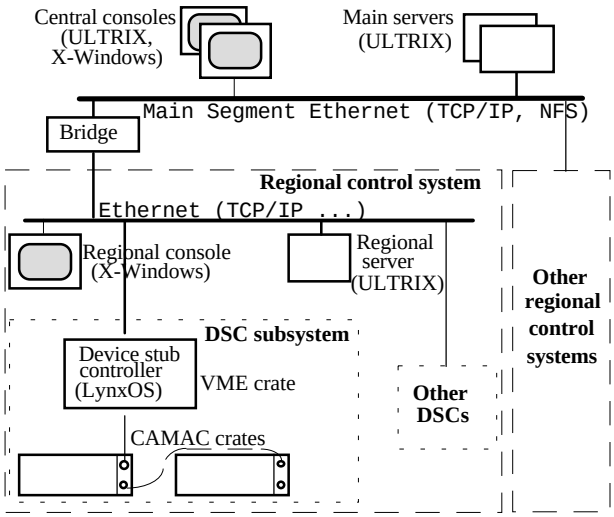


Fig. 1. Layout of CERN/PS control system

The evolving nature of the rejuvenation project of this system [1,2] requires new flexible methods and tools to append and to change easily the set of objects under control and their initialization algorithms. For each of the objects the tools should be capable to perform the following functions:

- Initialization (from zero) of hardware, setting up of the initial characteristic values, then testing of the equipment.
- Non-destructive test of the different objects.
- Information about the current state of the objects under control., in real time, during action execution.
- Diagnose of unsuccessful actions.
- Print, on demand, details of the set-up protocol (list of elementary actions and tests, results messages, etc.)

The solution was chosen to provide requested scalability with two modern technologies: knowledge based description of application domain combined with object oriented approach .

II. Knowledge Representation

2.1. Basics

The three following basic points were considered to select the set-up knowledge description method:

- Well-defined algorithms for setting up any type of control equipment present currently in the system.
- Presence of CERN/PS Equipment Interface library [3], providing the basis of the object oriented approach of control system description.
- Existence of Procedural Reasoning Object System for Control (PROSC) [4] implementing the idea of procedural knowledge (seems to be very suitable for control purposes) [5] combined with object methods of knowledge representation and real time interpretation.

The PROSC language provides the objects (classes) description, representation of their control algorithms as oriented graphs of actions and real time execution of those procedures when prescribed conditions arise. The possibility of forward chaining rules prescription for reaction on asynchronous events is implemented as well (e.g., start of a set-up procedure for an object is the reaction on a command issued by a user). The PROSC facility is described with more details in [4,6].

2.2. Knowledge Base Structure

The source knowledge base of the SETUP facility is a description of the system objects with their control rules. It consists of the class definition section and of the instantiated object description section.

The first one contains the descriptions of object classes, corresponding control rules and operational algorithms. This section is developed by specialists of domain and varies only when a class should be added/suppressed or an algorithm should be changed.

The second section describes the concrete object list, i.e., the instantiations of object classes of the current system. This section is built by program from the data of the general

purpose database of the control system. Insertions or deletions of equipment units, hardware modules, crates, etc., are hidden for users (for the case only when this doesn't demand a correction of the first section). The section is updated automatically after any reloading of the main database followed by the SETUP facility restart.

An other part of the knowledge base, containing the real facts relative to the corresponding object, appears during run time. It is created at compilation time of the source knowledge base and it exists during the whole life time of the SETUP. Its content reflects the result of inference processes and the status of external events. Those databases are accessible via the user interface facilities.

III. Object Model for CERN/PS Control System

3.1. General Scheme

The object model developed follows the control system structure. The basic object classes, defined to describe the structural elements of the system, are the following:

- ***MACHINE*** - accelerator control subsystem;
- ***DSC*** - Device Stub Controller (VME crate) based subsystem;
- ***LOOP*** - loop of serial CAMAC;
- ***CAMAC_Crate***;
- ***VME_Module***;
- ***CAMAC_Module***;
- ***Equipment*** - as defined for CERN/PS control system [3];
- ***Working_Set*** - predefined set of *Equipment*.

The ***EQP_Aggregate*** class is declared as a template of equipment aggregations where correlation of their states is possible and should be checked.

3.2. Object Inheritance

The scheme shown in Fig.2 demonstrates the definitions of the different object class levels. It allows an efficient distribution between parents and children of the setup algorithms and system element properties.

So, subclasses of the *Equipment* class can be either a simple equipment as defined in [3] (RCAMC, PTIM) or can belong to a composite equipment and inherit the rules of the *Composite_Equipment* class and the test algorithm(s) of *EQP_Aggregate*. The last one starts checking only after finishing all the started initializations .

CAMAC_Module_with_coupled_EQP and *QUAD_vacuum* inherit the property of *CAMAC_Module* of the included equipment and the mentioned algorithm of *EQP_Aggregate*. Moreover they have their own algorithms of checking and recovering included equipment.

Differences of *CAMAC_Crate* subclasses are caused by presence or absence of inserted control modules and by their position in the loop. Nevertheless all of them inherit the property of *CAMAC_Crate* to include RCAMC equipment representing the crate controller.

A *DSC* object has different set-up algorithms depending on the hardware lay-out (VME, CAMAC or both) of the corresponding control subsystems.

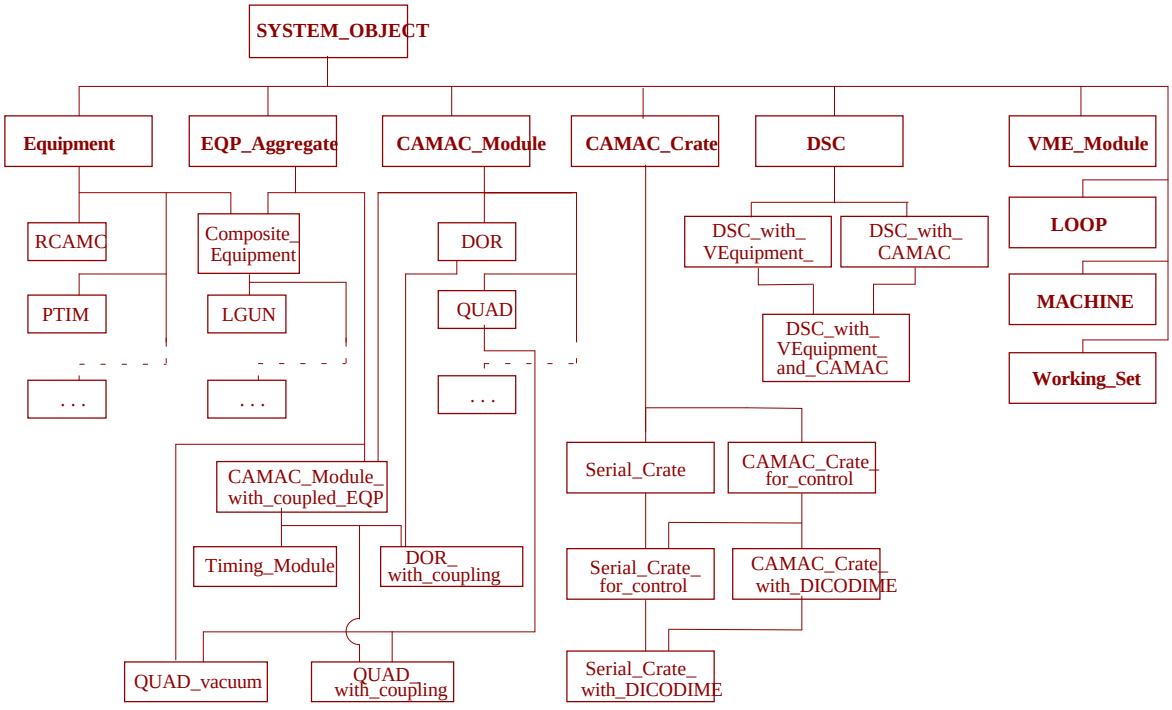


Fig. 2. Simplified scheme of class inheritance

3.3. Inclusions of Objects

It is considered that the objects needing set-up are:

- All the CAMAC crates
- The *Equipment* belonging to one of the equipment classes mentioned in the SETUP facility knowledge base
- All other objects including at least one of the previous two

If, for example, a CAMAC module includes only equipment that do not need any set-up actions, the CAMAC module will not be included into the system structure description. This has to be taken into account studying the definitions of the objects' inclusions.

An object of the classes *MACHINE* and *LOOP* includes each object of the next class in the list (one step lower in the real system structure). A *DSC* class object can include VME based equipment and/or loops of CAMAC crates.

A *CAMAC_Module* and *VME_Module* object include a set of the *Equipment* objects representing control equipment (only if that equipment is not a lower level member of another equipment).

A *CAMAC_Crate* object includes a set of *CAMAC_Module* objects representing all the CAMAC modules inserted to it, as well as *Equipment* objects included to those *CAMAC_Modules*. Each *CAMAC_Crate* object includes also the *Equipment* object representing the crate controller.

An *Equipment* object includes other *Equipment* objects if the latter ones represent its lower level members.

IV. OBJECT CONTROL

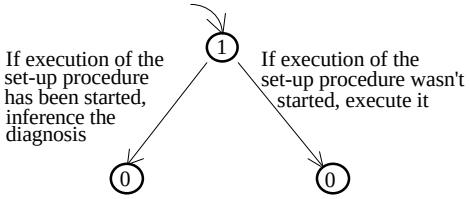
The functions listed in chapter 1 are provided for each object included in the current structure description in accordance to the rules and procedures prescribed for the corresponding class. So, two basic rules defined for super class SYSTEM_OBJECT declare that the procedure ***Full SETUP*** should be executed if the ***Start Setup*** command has been issued and the procedure ***Nondestructive SETUP*** should be the reaction on the command ***Test Equipment***. The definition of SYSTEM_OBJECT class and general scheme of both procedures are shown with Fig. 3.

The remarks near the arcs describe the action and condition of its execution. Therefore, first the right arc will be attempted to execute. The node marked with 0 is the final node of any procedure graph. It is achieved if the set-up procedure is executed successfully. If this arc has been rejected and therefore the condition of the left arc has become true, then the diagnostic procedure will find the reason of the set-up fault.

The procedures of setting up as well as of diagnosing are defined differently for each class, but PROSC selects the proper procedure according to the class of the handled object. *Init_Command* and *Test_Command* in the class definition are the facts been established by the user interface due to the issued user commands mentioned above.

```
CLASS SYSTEM_OBJECT
 HAS PROPERTY
  Full Setup IF Init_Command;
  Nondestructive Setup IF Test_Command.
```

a) Definition of SYSTEM_OBJECT superclass



b) Scheme of Full Setup and Nondestructive Setup procedures

Fig. 3. Definitions for root class SYSTEM_OBJECT

Figure 3 can be considered also as a general illustration of combination of control rules expressed via procedures and direct rules of forward chaining.

Other rules of the reaction on an event for different object classes are listed below (the corresponding class is indicated in parentheses):

- Test the host equipment if one of lower level equipment is attempted to initialize (*Composite_Equipment*).
- Check all the equipment included if one of them is attempted to initialize (*CAMAC_Module_with_coupled_EQP*);
- Recover all the vacuum control equipment if one of them is attempted to initialize (*QUAD_vacuum*);
- Test the previous crate in the loop (its crate controller) if a crate hasn't responded (*CAMAC_Crate*);

Among the "rules in procedures" we note that a set-up procedure for a composite object provides for its elements (included objects) the execution of the same action, i.e., *Full Setup* or *Nondestructive Setup* procedure, additionally to the actions for the host object itself. Figure 4 demonstrates the scheme of the command "distribution" during the initialization of a DSC.

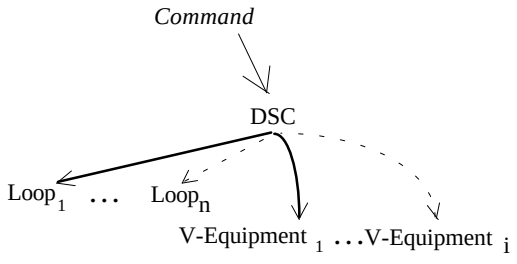


Fig. 4. Scheme of DSC initialization

V. SOFTWARE STRUCTURE

The SETUP facility software is composed of the knowledge base together with three program parts: the knowledge interpreter, the interface facilities and the current

structure generator. General structure of software with its environment is shown in Fig. 5 as a data flow diagram.

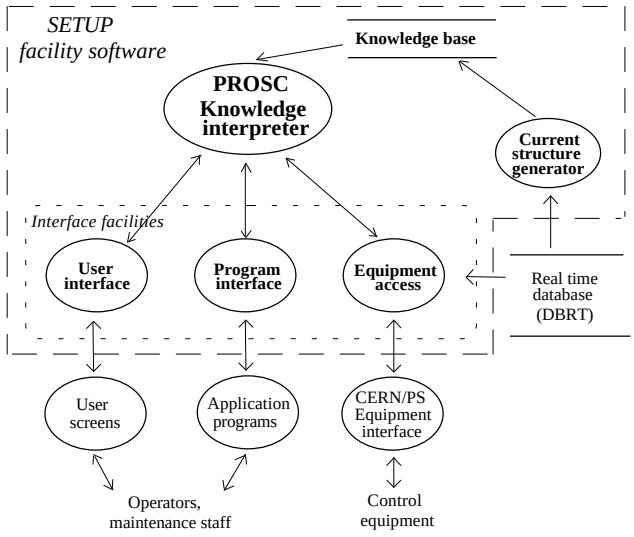


Fig. 5. SETUP facility software structure

The PROSC ***Knowledge interpreter*** is an event driven inference machine that compiles the knowledge base immediately after starting, starts the Equipment access interface and then operates with objects in accordance of events occurring (facts been established) and of the rules prescribed with the control knowledge. It is an applications independent product and needs therefore an interface layer to deliver facts and to execute real control actions.

Each interface facility is a separate process (a group of processes) in order to access a group of control system elements via a standard system facility. Communications of the *knowledge interpreter* to/from interface facilities are based on UNIX socket mechanism. The communications are completely asynchronous.

The ***Equipment access*** interface provides the accelerators control via standard PS Equipment Interface [3]. This facility communicates with different DSCs via parallel processes.

The ***Program interface*** facility is a library containing the functions to connect to the PROSC knowledge interpreter as a program server in a local area network and to access a graphic user interface in X-Windows/OSF Motif environment. The functions have to be linked to a user task intended to use the facility.

The ***User interface*** is a OSF Motif widget that can be called from any application via a program interface facility for user control of the set-up process i.e. selection of a set-up action, observing in real time the progress of the set-up process, then looking through and eventually printing the control protocol and all the facts related to the selected objects.

The ***Current structure generator*** is the only “off line” part of the facility. It prepares in PROSC language the list of currently presented objects under control via control system database containing all the configuration data.

The knowledge interpreter together with equipment access facility runs on one of the control system servers. The current structure generator runs there before starting the knowledge interpreter.

User interface facility can be arranged (via program interface) on the base of any application running on a workstation or X-terminal capable to communicate with the PS control system. An example of such implementation is outlined in the next section.

An autonomous Motif based application calling the SETUP widget and a program able to connect to the PROSC from alphabetical terminal have also been implemented.

The software environment:

- Operating system: ULTRIX.
- Programming languages: C, PROSC.
- Networking: TCP/IP.
- Graphic user interface: X-Windows / OSF Motif.

VI. Experience of Utilization

The operational console of CERN/PS control system includes a series of generic applications allowing the control and observation of all the aspects of the running machines. It was the most natural and suitable way to integrate the SETUP facility.

One of the most popular applications called ALARM panel [7] was selected as the first platform for the SETUP utilization. The ALARM system performs a regular checking of the control equipment and displays eventual faults. The SETUP application was integrated to it via the program interface. It provides the possibility of working either with the SETUP widget for human control or via the direct request to the knowledge interpreter to initialize a set of selected control equipment.

The ALARM application itself was improved during the SETUP exploitation to provide access not only to equipment units but also to crates, loops, etc.

The first operational version of the facility was installed on February 1994 for one machine, second version for 3 machines - on October 1994 [8]. Since March 1995 the list of objects under control includes 5 accelerator complexes, about 50 VME crates (DSC subsystems) with more than 150 modules, about 40 loops of serial CAMAC including more than 130 crates with more than 700 modules. About 3000 units of control equipment are based on the hardware mentioned. The current knowledge base contains definitions of 100 object classes and about 200 control procedures.

Two more (except the ALARM system) generic applications use now the SETUP facility.

The experience of the SETUP facility in operation during more then one year showed its efficiency in two important aspects:

- It became a popular tool for accelerator operators, used several times per day (much more during the startup after a shutdown).
- The facility allows the specialist of a domain to modify and reinforce knowledge easily as it was demonstrated during the extension of the knowledge base with the new equipment class definitions for the added accelerators (it has been done three times).

VII. CONCLUSIONS

The SETUP facility presented is the product of the collaboration of CERN/PS and Petersburg Nuclear Physics Institute that was started on December 1992. It was intended to provide a flexible high level tool for setting up accelerator equipment in the rejuvenated control system. This goal has been completely fulfilled and the implemented solution are checked every day in real practice.

To the authors' opinion the knowledge based system can be a good solution for a wide field of real time control systems.

Acknowledgments

The authors would like to express their gratitude to specialists of CERN/PS Control group J. Cuperus, N. de Metz-Noblat and C.H. Sicard for useful consultations and assistance during the implementation of the SETUP project.

VII. REFERENCES

[1] F. Perriollat, C. Serre, "The New PS control system overview and status", Proc. ICALEPCS, Berlin, Germany, 1993 (Nucl. Instr. and Meth. **A 352** (1994)), pp 86-90.

[2] G. Benincasa, G. Daems, B. Frammery, P. Heymans, F. Perriollat, Ch. Serre, C.H. Sicard, "Rejuvenation of the CPS Control Systems; the First Slice", CERN/PS/91-14(CO), April 1991.

[3] F. Di Maio, A. Risso "The CERN-PS Equipment Access Library. Software Specifications" PS/CO/Note 93-87 (Spec), February 94.

[4] V.M. Filimonov, V.P. Homutnikov, Yu.F. Ryabov, "An Approach for Description and Interpretation of Procedural Knowledge for Complex Physical Installation", Preprint PNPI 1828, St. Petersburg, September 1992.

[5] M.P. Georgeff, A.L. Lansky," Procedural knowledge", Proc. of IEEE, Vol. **74**, 10, Oct 1986. pp 79-100.

[6] G. Daems, V. Filimonov, V. Homutnikov, F. Perriollat, Yu. Ryabov, P. Skarek. "A Knowledge Based Control Method: Application to Accelerator Equipment Setup", Proc. ICALEPCS, Berlin, Germany, 1993 (Nucl. Instr. and Meth. **A 352** (1994)), pp 325-328.

[7] J-M. Bouché, J. Cuperus, M. Lelaizant. "The data driven alarm system for the CERN PS accelerator complex", Proc. ICALEPCS, Berlin, Germany, 1993 (Nucl. Instr. and Meth. **A 352** (1994)), pp 196-198.

[8] J-M. Bouché, G. Daems, V. Filimonov, V. Khomoutnikov, Y. Ryabov. "Knowledge Based SETUP Facility. User Guide", - CERN/PS/CO Note 94-81, November 1994.