

Proposals on ATLAS' DCS software development

V. Filimonov, V. Khomoutnikov, Yu. Ryabov
Petersburg Nuclear Physics Institute (PNPI),
Gatchina, St.Petersburg, 188350, Russia

February 27, 1995

1. Introduction

This document is intended to sketch the general structure and main facilities of ATLAS DCS software and to propose basic principles of developing.

The two following sections outline a typical, for the time being, local area network (LAN) hardware and basic software functionality. Section 4 explains the correlation between DCS' central part and subdetectors' control systems.

We consider, that for a large control system developing, especially with rather long developing' and life' time, it is very important to collect and inherit solutions from simple prototypes up to the full system controlling the real object. At the same time the system should be able to accept appearing new hardware and software products. This can be provided more simple if they follow ground principles and use advanced technologies from very beginning of the developing. We emphasized here the principles and features which seem to be included to basics for system scalability (flexibility and expandability in functions and objects under control):

- Knowledge based approach which allows to describe the structure and control algorithms, diagnosis and recovery rules (for object of control as well as for the control system itself) making use of powerful means of knowledge representation languages. It provides the facility to improve the system functionality in many cases by "teaching" the system instead of new software developing.
- Object oriented style of software developing to collect and lightly modify software components, adopt them by simple way for new hardware elements,

The sections 5 and 6 are devoted to some details of those. From our viewpoint ignoring of these two things from the beginning of a developing can appear as the source of many difficulties while using these fruitful technologies on the further stages.

The proposals are developed to fulfill the general requirements for DCS [1], grounding on the experience of the PS' accelerator control system exploited successfully during years [2,3] and own the authors' experience of developing and implementing the knowledge based facility for initialization and diagnose of control equipment integrated now to that system [4,5].

2. Hardware overview

Basing on general request to ATLAS Detector Control System (DCS) [1], one can represent the DCS hardware structure as it's shown with Fig. 1.

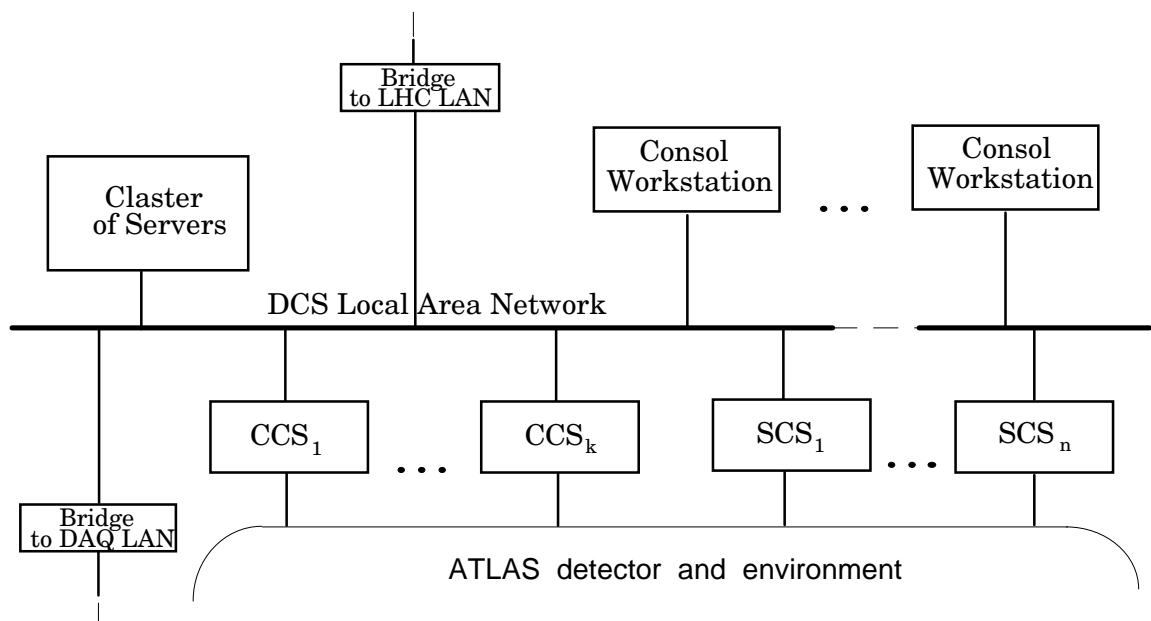


Fig. 1. General structure of ATLAS DCS hardware

Here a Subdetector Control System (SCS_i) is the control system of a subdetector while a Common Control System (CCS_i) is a subsystem (based, e.g., on VME crate with processor module) for direct measurements and control of common environment and for safety control of selected features of subdetectors.

General purpose workstations are consoles of operational and maintenance staff and serve for observing status of the detector and environment as well as for initialization of control sequences. This should be provided by graphic user interface and by dedicated application programs.

Each server provides one or more general services such as access for database, data logging, control knowledge handling, etc. To provide a higher level of system reliability some servers could be doubled.

Access of the DCS for DAQ system and for LHC control system (as well as in inverse direction) is provided via bridges between local area networks with prescribed permissions of access level (typically, on "client - server" scheme).

3. Functionality and software architecture

The basics of functioning ATLAS' DCS can be expressed quite transparently with data flow diagram shown with Fig. 2, which can give an impression on software types and location. Some details of that figure are explained below.

Lower layer software. The software for detector data continuous monitoring as well as control drivers reside at CCS/SCS. We consider here the data processing as on-line data handling such as filtering and fitting, calculation of means and trends, detecting the range of scale been crossed, preparing data for logging, etc. A certain part of that software can reside at CCS/SCS also, while other part is a common data server. Some application, devoted mostly to special task of certain subsystem, can reside at CCS/SCS also.

System servers. The center of the system functionality is the collection of common program service tasks running on LAN servers and/or on the CCS/LCS. They are the following:

- Data server mentioned, which provides, except the functions named before, delivering data portions and views requested by applications or other servers;
- Display processor (the "Monitor driving" node on Fig. 2) that drives on-line data representation with real time monitors displaying current system status (tables, graphic images, etc.); One of its most important destinations is to show alarm signals in cases of some faults in detector' and environment' functioning to provide safety and prevent damages;
- Processor of decisions on system control, i.e., estimation of current state of controlled objects and automatic control (in the part outside of the scope of lower level

subsystems), checking operator command, etc.; the practical proposals on implementation of this server are outlined in section 5;

- Generic application implemented as the servers with shared access of other application programs.

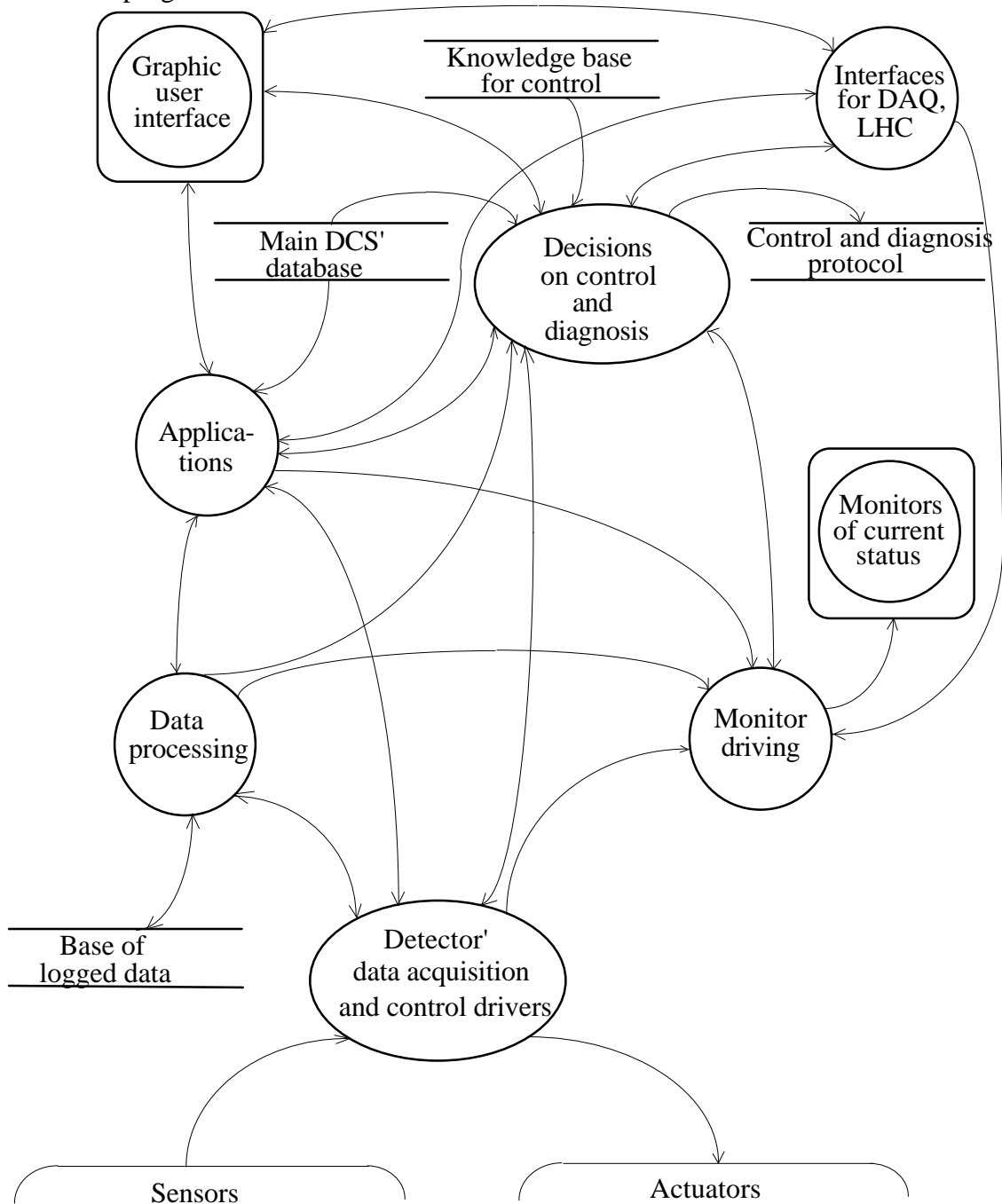


Fig. 2. Data flow diagram of DCS

The upper layer is composed of graphic user interface and dedicated application programs running on workstations and delivering full control facilities for operational and maintenance staff (and for other possible users).

The data structures needed for the DCS are quite evident. We consider that there is, except ones mentioned with Fig. 2, a base of row data hidden inside data acquisition subsystem. Those data are sent then to data server to arrange, handle, display them when necessary and for logging. After that, the data become accessible (via the same server) for any application of the system. It doesn't exclude, of course, sending of some data by acquisition system itself: emergency signals for displaying, messages about changes of element/subsystem state for decision on control, any available data after request of an application.

Main DCS' database contains all the data describing the detector' and environment: the physical and logical units, their characteristics, the parameter values for different operational modes, etc.

Control protocol is an ordered list of system' and its elements' transitions from one state to another together with fault diagnosis if any.

The control system has to have information on control composed of initial set-up rules, operational algorithms to achieve each mode of detector usage, control feedback for the cases of alarms and other incorrectness of functioning, rules of fault diagnosis, etc. Generally saying, these data could be prepared in any form suitable for the control system implementation. It's proposed here to represent this information as a formalized text of knowledge description in a syntax prescribed by the knowledge handling means (e.g. [6]). Some more details on this subject see in sections 5, 6.

4. About subdetector control systems

The correlation between subdetectors' and the complete detector control as well as between subdetectors themselves are inevitable, at least, from viewpoint of safety. Therefore, processors for decisions on control of central DCS and of "peripheral" ones have to collaborate strongly. It would be very essential to do this with homogenous software facilities and even to develop a kind of distributed" facility for this subject.

There is no reason to prevent constructing a subdetectors' control systems following the same basic principles and scheme, especially, as far as it concerns to "large" sophisticated subdetectors. The concept doesn't force, nevertheless, to use the only way mentioned. A subdetector control system, as concerned to its interior, may be developed

with any tools and reasonable approach of constructing. The only general top level requirements should be fulfilled:

- Homogenous hardware at the subsystem level facing to central DCS;
- Standardized for all systems protocols of networking;
- Compatible UNIX-like operating systems;
- Homogeneity of user interfaces;
- Making use of the same or compatible database management system;
- Providing access to own acquired data for any collaborating system with a way unified for the whole DCS;
- Operating any object outside own scope of a SCS is available under control of the central system.

A part of subdetector sensors and actuators has to be connected to a central control station(s) of the DCS. This is to fulfill the requirement [1] of possibility to shut down subdetector safely if it's necessary due to own control system fault or an external circumstance. Therefore, independently of the way of subdetector's control system implementation, some of the subdetector elements (objects) and control algorithms have to be presented at the central knowledge base.

5. Knowledge based control facility

In our opinion, the current state of computer science allows to use the knowledge base and expert system technology at real-time applications. The series of requirements to DCS of ATLAS detector laid open in [1] and listed below can be carried out with these methods.

- An operator has to be able to perform standard operations, such as the change of the global state of subdetectors (e.g., on, off, standby, etc.) or to identify problems arising, without in-depth knowledge of the detector hardware.
- Automatic corrective actions without operator intervention have to be implemented.
- Operator assistance should be provided by comprehensive help facilities and fault diagnostics.

Tasks corresponding to these requirements can be formulated with some logical system rather easy. It's possible to define any state of equipment with a set of facts (assertions) and to describe on this base the rules of transition from one state to another and reactions on event. Then a system of logical inference would be capable to give an exhaustive fault

diagnosis, to fulfill operational prescriptions after the operator command or automatically as a response for an event.

For real-time applications a control system has to possess the following additional features:

- Continuous embedded operation: System for process control has to operate continuously, unlike clear diagnostic system.
- Asynchronous interaction: System has to be capable to catch events or facts in any time. It is unreasonable to assume that events occur at designated times. The system has also to carry out actions in environment and to take into account temporal properties of equipment (delayed feedback, for example).
- Guaranteed response time: Appropriate actions have to be executed before it is too late. In other words, system must ensure that the response time is short enough to allow the response to be properly effected.
- Concurrent decision of problems: In process control, the occurrence of multiple problems is quite usual. It is thus necessary for the system to be able to handle multiple problems concurrently.

These requirements decrease very strongly the set of knowledge handling products though not up to the empty one. The authors have an experience of implementing knowledge based system in real-time environment [4, 5].

The Procedure Reasoning Object System for Control (PROSC), developed at PNPI [6], seems to be capable of being applied for real time application, at least, as the first solution of the problem for a prototype system.

The PROSC uses an object oriented architecture for knowledge representation and for reasoning about actions and procedures in a dynamic domain. Any kind of equipment is represented as a collection of objects with their properties. The behavior and control of those are described with a set of plans or procedures. Inference mechanism used in the PROSC names "Procedural reasoning" [7].

The most important advantage of the method implementation is the capability of system, based on it, to obtain facts (accept events) during an inference and to change the sequence of actions depending on the state of environment. The duration of an inference as reaction on an event is essentially reduced relative to "classic" rule based system due to consideration the only object (and including ones) that the new fact pertains to.

Owing to standardized interface of communications to other collaborating tasks the PROSC facility could be later changed easily for standard industrial software product if more suitable one appears.

6. Object oriented approach of developing

It seems to be very natural to think a control system element as an entity having its own attributes: parameters and control rules as well as a set of methods to operate them. The fruitful idea of object oriented approach of programming to encapsulate all own data and access methods in object's software module make it possible to implement the software for equipment classes (e.g., power supplies, pumps, valves, etc.) completely independent one from another, to hide on this way their redundant technical details from upper level applications delivering instead of that a general interface specification. This idea has been successfully implemented while developing some recent systems of accelerators' [2] and detector's [8] control. It was applied there to control equipment considered on one level. The corresponding libraries developed represent that equipment for external applications as objects with uniform access methods. In such a way, any changes inside an object software and hardware don't influence the applications unless an externally visible attribute has been changed.

We propose here to consider the DCS and its domain more generally as an association of different level's objects that can include one another or cross, have common elements. (It doesn't suppress an equipment access library for operations with basic objects like mentioned before.) The simplest examples of "a bit higher level" objects could be cycles of continuous regulations (for pressure, temperature, vacuum) including several sensors, pumps, coolers, etc. The lower level objects of that might be visible or hidden from outside in dependence of control needs. From other end of the scale of objects lie the whole ATLAS with environment and large subdetectors.

Generally, the objects to be defined have to present the whole spectrum of subsystems and elements' aggregations needed to operate detector and environment in different modes (including automatic reactions on events) and by different person. Depending on the competence, an operator may be familiar with high level objects while an expert is able to investigate a problem "in depth", i.e. to operate with more~ detailed objects.

The sets of object states are to be defined expedient and valid. They are to be fruitful for understanding situation and for operation on any level of competence. This means that status description and possible diagnostic messages, might be issued during operations, should be expressed in natural (maybe lightly formalized)~ language.

To ensure the requirements for DCS [1] an object could have the~ following attributes:

- The list of lower level objects;
- The list of facts determining its status;

- The set of operator commands and other external messages have to be responded by object.
- The set of rules to operate after operator's commands or external events; the rules can depend on the states of included objects as well as of the own object status;
- The set of graphic images showing corresponding equipment in different forms and being able to reflect current status of object by changing color, positioning of some graphic elements etc.;

The scheme of the object's general structure is shown with Fig. 3.

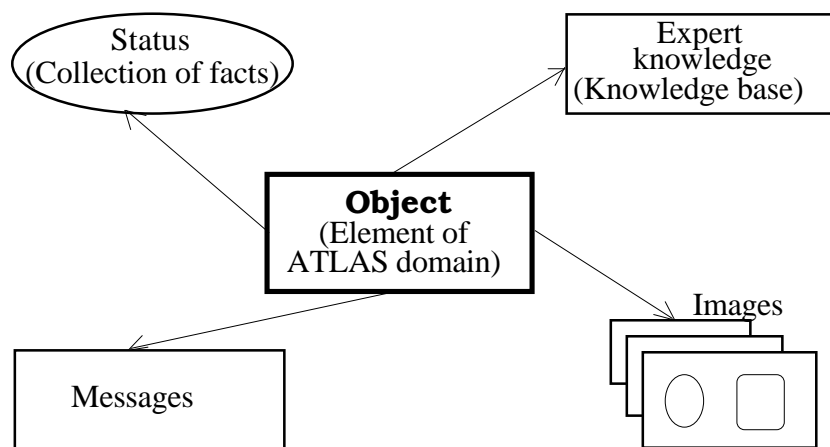


Fig. 3. Generalized scheme of an object with its attributes

The software implementation of an object can be different. Except equipment access object library for basic objects it can be an application from scheme on Fig. 2, or separately developed subdetector control system. In any case the object should be presented in knowledge base together with its control methods needed for correct operation from the DCS operator's level (commands, actions, correlation with states of objects related, etc.).

It seems to be reasonable to define the objects, sophisticated enough, on the level of knowledge. This would ensure the most simple and transparent means for establishing relations between them, for defining correlation of their states and available controls as well as for corrections of properties and control algorithms resulting their commissioning, changes, replacements and exploitation experience.

7. Conclusion

It is quite clear that to start a practical developing one needs to study the details of ATLAS detector as a controlled object. They are the set of object states and transitions, control functions needed, distribution of controls between central DCS and subdetectors' control systems, list of parameters with their descriptions, etc.

On this ground the following main directions of developing should be established:

- Developing object oriented representation of ATLAS applicable for construction of the DCS;
- Description of control and diagnostic rules in the knowledge representation;
- Developing of general software structure and tools of DCS;
- Definition of data structures needed and their distribution between subsystems, selecting suitable database management system(s).

The declared ideas and found solutions should be implemented in a prototype control system to check, investigate and improve them while test runs.

REFERENCES

- [1] "ATLAS Technical Proposals", CERN/LHCC/94-43, LHCC/P2, 15 December 1994.
- [2] The PS and SL Controls groups, "PS/SL Controls consolidation project", CERN/PS/CO 91-09.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.

- [7] M.P. Georgeff, A.L. Lansky, Procedural knowledge. Proc. of IEEE, Vol. 74, 10, Oct 1986. pp 79-100.
- [8] P. Mato et al., "The new slow control system for the ALEPH" experiment at LEP", Proc. ICALEPCS, Berlin, Germany, 1993 (Nucl. Instr. and Meth. A 352 (1994)), pp 247-249.