A USERS VIEW OF THE SPS AND LEP CONTROL SYSTEMS

R. Bailey
CERN
CH-1211 Geneva 23, Switzerland

Abstract

Every accelerator has a control system; at present the SPS has two, both of which are needed to run the machine. Consequently a user of the SPS / LEP complex has to be concurrently familiar with three control systems. While this situation brings problems it allows, even forces, comparison between the different systems, which in turn enriches the user viewpoint.

This paper assesses the SPS and LEP control systems from the point of view of the user, who may be an equipment specialist, operator, accelerator physicist or combinations thereof.

1. Introduction – what the accelerators do

Exploitation of the two large accelerators at CERN is a varied business. For the SPS in 1991 this amounts to running as a fixed target machine for over half the year, providing either protons (during 21 weeks) or sulphur ions (during 6 weeks) to the physics community. In conjunction with this the SPS acts as an injector to LEP, providing leptons in an interleaved repetitive supercycle. Furthermore about 15% of the fixed target running time is given over to machine development periods, when the SPS is required to run in some non-standard way, mostly as a testbed for the LHC. Finally, the SPS is also used in the other major mode of operation, as a proton-antiproton collider, for about 5 weeks.

In parallel with all of the 27 weeks of SPS fixed target running, LEP is taking beam either for Z^0 production or for a substantial machine development program, the latter amounting to about 30% of the total LEP running time.

For both machines, although mostly for LEP, installation and testing of new equipment is carried out throughout the year.

This diversity of operations and machine improvement is carried out from a common central control room, with the same teams being responsible for both the SPS and LEP. In particular, one group run the SPS in a variety of modes of operation throughout the year as well as running LEP. This means that these personnel have to be familiar with the different control systems used to interact with the accelerators. The same is true of the personnel responsible for equipment commissioning.

2. Overview of control systems available

From 1975 the SPS has been controlled, either exclusively or partially, via a system based on Norsk Data ND100 computers connected in a TITN star configuration [1]. The computers run SINTRON and the programmers are provided with the NODAL interpreter, libraries of graphics primitives and data modules and a means of calling FORTRAN executables [2].

From 1985 the major new requirement for SPS to provide beams to LEP meant a complete rewrite of the applications software. This was undertaken in a UNIX environment on an Apollo network, with C as the main programming language and Apollo-Dialog for the user interface. In the first instance access to the hardware was via a gateway into the existing TITN system. More recently the possibility exists to access some equipment completely independently from the TITN system, using the same overall Token Ring architecture as for LEP (see below).

Presently the SPS is run using a mixture of purely TITN (30%), Apollo via the gateway into TITN (50%), and purely Apollo Token Ring applications (20%) (see figure 1).

LEP applications also run on an Apollo network, with C as the main programming language and Apollo-Dialog for the user interface. All the Apollos are connected on a control room Token Ring, with communications out into the field through a bridge to a machine Token Ring running around the accelerator [3]. At several points around the ring there is a further bridge or gateway into either a regional Token Ring or an Ethernet network. Connected to these local area networks is a variety of configurations, allowing access into the hardware via several different equipment control assemblies, mostly using the MIL-1553-B standard (see figure 1).

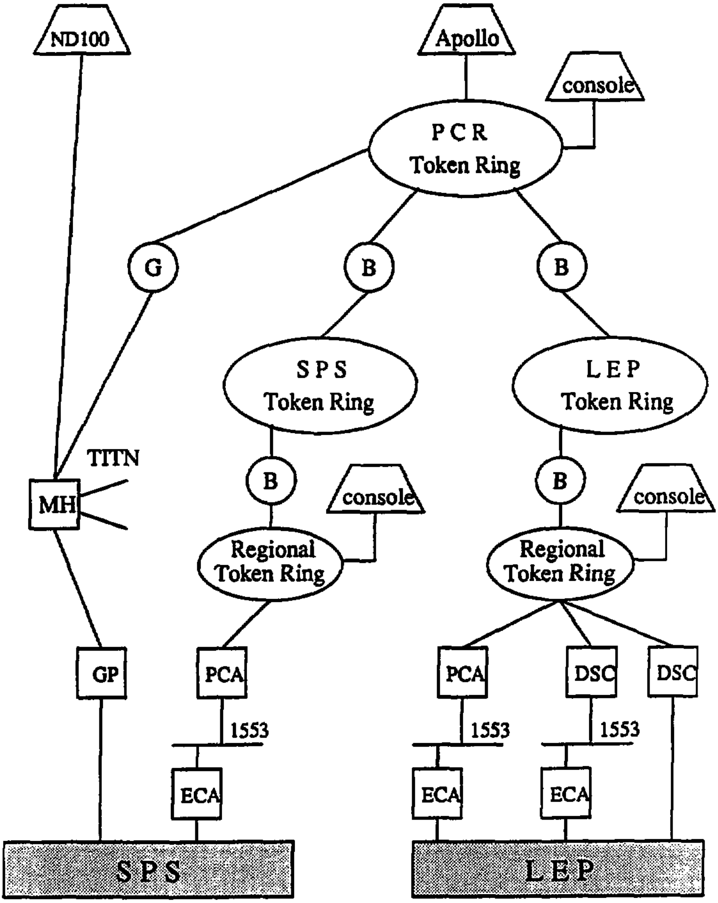

Figure 1
Logical schematic of the networks

3. Different types of user

The control systems of SPS and LEP are used at different times by a variety of different personnel. These largely divide into three categories; operators, accelerator physicists and equipment specialists, each of whom have somewhat different requirements for the control system. These requirements are not only for the underlying architecture (network, operating system etc), but also for the applications that run on top of it. In other words, the user here is seen as the person who runs the applications programs, rather than the person who writes them.

All types of user of course need reliable network communications, with good diagnostics when things go wrong. An adequate speed across the network from console to equipment is also generally required.

3.1 Equipment specialists

Equipment specialists need to access a diversity of accelerator hardware, setting and reading a multitude of parameters that are not of interest to other users of the control system. In many cases they also need to do this locally, in order to closely monitor the effects on their equipment. This means that they need to run specific programs both in the central control room and in the field, the latter requiring local console facilities. They may well want to run locally when the network is down. Most of these programs are written by the person who will run them, or at least by a close colleague, and as such the reliability of the application is not of great importance.

In many cases the amount of equipment accessed is far more than during normal operations, in order to thoroughly test a system, for example. For this reason the speed can be of prime importance to the equipment groups.

Key requirements;

local console facilities
execution speed

3.2 Operators

Operators rarely work on individual pieces of equipment, but rather on combinations of accelerator systems or even on the accelerator as a whole. In performing this work they prefer to see a high level of standardisation across the different applications and across the different accelerators. The applications also need to be easy to use, with the operator being presented with all the information that he needs but not swamped by auxiliary data that he rarely uses. Online documentation is a big help, particularly when the applications are new.

Since many tasks have to be performed at the right time in a sequence, the applications that perform them need to be highly reliable. Since operations is a long and repetitive process, it is essential that the speed of execution of programs is adequate, which generally means completion of the task in a matter of seconds. Good error reporting is also very important.

Key requirements;

ease of use
stability
standardisation
execution speed
error reporting

3.3 Accelerator physicists

Accelerator physicists have essentially the same requirements as the operators, except for the important addition of flexibility to allow new, non-standard applications to be used. Indeed since machine development periods usually involve doing several unusual things, standardisation and error reporting are not so important.

Key requirements;

flexibilty

ease of use
stability
execution speed

4. Comparison of the different control systems

Table 1 summarises the results discussed in more detail here.

In all three cases the speed and reliability of the network is adequate. However when there is a problem, it is much easier to pinpoint on the TITN system than on the Token Rings, which have become extremely complex.

Local facilities are also better on the TITN, where much of the equipment data is stored locally rather than in a central database.

Table 1 Comparison of observations

	SPS old	SPS new	LEP
Network			
Speed	●●	●●	●●
Reliability	●●	●●	●●
Diagnostics	●●●	●	●
Local facilities	●●●	●●	●●
Applications			
Execution speed	●●	●●	●
Stability	●●	●●●	●●
Error reporting	●	●●●	●
Standardisation	●	●●●	●●
Flexibility	●●●	●	●●
Ease of use	●●	●●●	●●●

Key The more blobs the better

●	poor
●●	adequate
●●●	good

4.1 SPS old

A key feature of the NODAL based control system is flexibility. It is extremely easy to produce a working application program, communicating with the machine and displaying data to the user. While this is an excellent feature, particularly for equipment testing or for one–off applications, as operations become more complicated it becomes more difficult to control the overall coherence of the system.

In the SPS the operational applications grew out of equipment commissioning programs, essentially on a system by system basis, and in an iterative way. As an example quadrupoles, sextupoles, octupoles etc were all controlled by different suites of programs all essentially doing the same thing. Adding a new system involved adding a new suite of programs to control it. Apart from the obvious problems of duplication of effort, this has also led to a certain diversity of the way similar functions had to be performed in different applications, which is very confusing to the user and makes it difficult to remember how to drive the different programs.

Because it is so easy to write or modify programs in this environment, in the absence of any real software management the stability of the applications is never fully achieved, and maintenance is consequently very difficult.

The very limited memory available in the control room consoles meant that most of the applications had to be kept small, and as a direct consequence of this error reporting had to be kept to a minimum, as did commenting the code.

Finally the speed at which the applications run has been found to be adequate. Since no online database exists the individual programs do their own data management, and though this brings its own problems it tends to be fast. Consequently the speed is determined by that of the NODAL interpretor and that of the TITN network. As a benchmark, sending a 100 point amplitude vs time function to the accelerator takes around 1 second per hardware address, which is considered acceptable.

4.2 SPS new

There were two significant differences between the way the new SPS applications were developed as compared to the old. Firstly the overall functionality of the software needed to operate the accelerator was analysed in detail before any design was considered, and secondly the underlying data structures were completely determined before any implementation was undertaken [4]. By its very nature this kind of software development leads to software that needs little change once implemented, and results in a very stable system. The highly modular way in which the applications were designed allowed an easy and standard way of handling errors, and the error reporting is excellent.

Knowing the detailed functionality led to a high uniformity, not just at the level of the operator interface [5] but more generally in the facilities the different applications shared. As examples there is only one function editor, one dataviewer and one application that is able to send to the equipment anything from a single function to the settings for the whole machine. This has contributed greatly to the ease of use of the software, and this is enhanced by a standard online help facility describing how to drive the applications.

Having a sound definition of the data has allowed the applications to be largely data driven, giving coherence to the different accelerator systems and allowing new systems to be integrated without writing a single word of code.

The major disadvantage of this approach is that the software has been produced specifically to operate the SPS in the various modes forseen over the next ten years. Any novel running of the accelerator during machine development sessions invariably requires new features which are very difficult, sometimes impossible, to accomodate. Up to now these problems have been overcome by exploiting the high flexibility of the old TITN system.

The speed of execution of tasks is similar to the old system, but in this case database access times and the TITN network are the determining factors. The reliability of the gateway into the TITN is not good but problems are easy to spot and rectify.

4.3 LEP

Before the construction of LEP was complete, an analysis of the software required to run the machine was made. Naturally the emphasis was put on the software needed to commission the accelerator, and for the startup of LEP the controls and equipment groups provided a suite of powerful utilities for sending settings to and acquiring data from the hardware. These utilities exist as commands on the control room consoles and provide a means of quickly making script programs to do standard or non-standard things to the accelerator. Much use of this facility has been made during the commissioning phase, and more recently by accelerator physicists during machine development sessions.

The applications used today in operations also make heavy use these utilities for accessing the hardware. While this may be convenient for the programmers it invariably introduces overheads in the execution speed. The speed is further reduced by the underlying online data organisation, since the structuring of the data does not reflect the way in which we now want to run the machine.

The development of the operational applications has not followed an integrated approach, which has brought low coherence and a very variable level of error reporting.

Uniformity across the user interface applications has been achieved to some extent. Following the standards of the SPS has ensured a look and feel of the individual applications that is liked by the operators, and most programs are now easy to use.

The operational software relies heavily on servers running at all levels of the network, from the control room Apollos down to the front end computers. While communication between these servers is normally transparent to the user it often involves passing through several bridges. If one of the bridges or servers dies it is sometimes difficult to diagnose which one, and in many cases a procedure of sequentially restarting one after the other is required.

Furthermore many applications are dependant on certain computers to be up in order to run. There are presently around 10 such critical nodes on the control room token ring, the failure of any one of which would affect operations to some extent, in many cases seriously.

These two implementation details directly affect the overall stability of the software needed to run the machine.

5. Alarms

Quite apart from the application software used to drive the accelerator, there is another area of the control system that is of great importance during routine operation of the machine, namely the surveillance system.

Ideally this should work on the simple principle that software, running without operator intervention, should check that all elements required to be ON are ON, that those that should be OFF are OFF, and that all settings stay within a tolerance acceptable for operations. This software, running frequently, should report any abnormal findings to an alarm system for processing prior to presentation to the operator as a new alarm on his screen [6].

In practice the viability of such a system depends very much on other parts of the control system. It is imperative for such a system to have available a definitive source of data reflecting the way the machine is actually supposed to run at the time. Furthermore because most machines run in several modes of operation, each requiring a different configuration, this image of the machine has to be dynamic.

It has already been mentioned that the LEP operational applications have not been developed in an integrated way, and one consequence of this is that there are several different ways of storing the actual machine settings. This makes it very difficult to provide standard surveillance programs; in reality each set of equipment has to have it's own program, a situation which is of course very difficult to administrate. So while for LEP the central alarm server works well, the amount of useful information reaching the operator is presently rather limited.

The same problems were encountered with the alarm system running on the SPS TITN network. Again there was no coherent image of the machine, and it took several years before the alarm system was providing information of sufficient credibility for the operators to use with confidence.

It is ironic that the new SPS operational software, which is driven from a central online database, does not yet have any kind of alarm system. Indeed we are experiencing problems due to this as more and more systems are migrated from the TITN to the Apollo-based software, since there is presently no means of surveying them. The aim is eventually to use the same system that is presently in use for LEP, but with simple surveillance programs comparing measurements with settings in the online database.

6. Conclusions and remarks

In the case of both the SPS and LEP, the network and control room utilities proved adequate during the running in of the machine. As testimony to this, beam was circulating in LEP one or two days after first injection, and the first Z^0 was reported within a month.

However, remember also that machine commissioning is done by specialists and over a limited period of time. When it comes to building the complex, integrated software packages that are required in routine operations, it has proved difficult to do so from the utilities provided. What is needed is a review of the operational requirements and a corresponding rewrite of the application software. Furthermore it is very difficult to determine these operational requirements in advance of getting hands-on experience of the accelerator.

The new SPS software is a good example of what can be done. It was based on 10 years experience of running the SPS in a variety of modes, and the software produced satisfies most of the operational requirements. The same thing now has to be done for LEP, this time after 3 years experience but drawing on the lessons learned in the SPS.

7. References

[1] M.C. Crowley-Milling, "The Design of the Control System for the SPS", CERN 75-20

[2] M.C. Crowley-Milling, "Experience with the Control System for the SPS", CERN 78-09

[3[R. Rausch, "Real Time Control Networks for the LEP and SPS Accelerators", in *Control Systems for Experimental Physics*, Villars, Switzerland, September 1987, pp. 244-247.

P.G.Innocenti, "The LEP Control System", in *Accelerator and Large Experimental Physics Control Systems*, Vancouver, Canada, 1989, pp 1-5

P. Lienard, "The SPS and LEP Control Network Architaecture", in *Accelerator and Large Experimental Physics Control Systems*, Vancouver, Canada, 1989, pp 215-220.

[4] R. Bailey, J. Ulander, I.T. Wilkie, "The New Applications Software for the CERN SPS", in *Control Systems for Experimental Physics*, Villars, Switzerland, September 1987, pp. 158-163

[5] A. Ogle, J. Ulander, I.T. Wilkie, "Experience with Workstations for Accelerator Control at the CERN SPS", in *Accelerator and Large Experimental Physics Control Systems*, Vancouver, Canada, 1989, pp 437-441.

[6] M. Tyrell, "The LEP Alarm System", these proceedings.