Abstract

The new CPS controls system was successfully put into operation on 19 November, 1980, when the 800 MeV Booster, after switching over to the new system, was started up for particle physics. A subset of the new system had already been installed on the Antiproton Accumulator in Spring 1980 and has reliably served several months of engineering run. The Booster machine and control of the CPS cycle generation are now routine ly operated by the new system which is based on distributed processing, extensively applying microprocessors in the interface. The problems encountered in converting existing accelerator controls, the control switch-over itself and operational experience gained up to the date of the conference are described.

Introduction

The objective of the CPS controls project is to provide an integrated and homogeneous controls system that will allow operation of the CPS accelerator complex from one central control room. Involved are Linac(s), PS Booster, 28 GeV Synchrotron, Accumulator for antiprotons now, later for electrons and positrons, transfer lines between these and to destinations like SPS, ISR, 25 GeV physics and LEAR. Service aspects and maintainability are strongly emphasized.

The diagram is self-explanatory. Details are in the references. Features highlighted are:

1. The CPS controls project is one in which the required operational2 and accelerator engineer's aspects were systematically synthesized and included in the specifications with a high weight.

2. The CPS accelerator complex and its controls, are unique in that interleaved cycles3 of different particles and beam properties are produced and transferred to different destinations. Sequences and properties of these cycles are created on the control consoles and are controlled by a computer-based system, the Program Line Sequencer (FLS), which sends out coordinating messages to controls system and process hardware.

3. The new CPS controls system makes extensive use of distributed intelligence.4 There is a microprocessor-based Auxiliary Crate Controller (ACC) in most CAMAC process interface crates. The ACC's tasks are (i) cycle parameter refreshment, (ii) buffering fast data bursts from beam instrumentation and (iii) some pre-processing; by-products are unloading of process computers and local autonomy, giving graceful degradation. TV. The system makes use of a modular, telephone exchange-like analog multiplexer network5 for central console display of key analog signals from the whole accelerator complex. The bandwidth is up to 25 MHz, channels are computer selected from touch panels.

V. Crucial for the service and maintenance aspect and indispensable for the conversion, is the local access to the controls resources at different levels.6 This provides the means for multiple parallel checking, essential for switch-over and maintenance during accelerator stops. It also allows flexible and selective online diagnostics and patching.

The table illustrates the size of the system.

<table>
<thead>
<tr>
<th></th>
<th>now</th>
<th>later</th>
</tr>
</thead>
<tbody>
<tr>
<td>process variables (many modulated at 1 Hz (PPM))</td>
<td>4·10^4</td>
<td>10^4</td>
</tr>
<tr>
<td>application program modules</td>
<td>5·10^5</td>
<td>10^5</td>
</tr>
<tr>
<td>console software, expressed in lines</td>
<td>4·10^4</td>
<td>4·10^5</td>
</tr>
<tr>
<td>systems software, expressed in lines</td>
<td>5·10^5</td>
<td>5·10^5</td>
</tr>
<tr>
<td>general purpose consoles (central)</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>autonomous microcomp. + terminals</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>16 bit microprocessors</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>16 bit microprocessors</td>
<td>35</td>
<td>100</td>
</tr>
<tr>
<td>CAMAC (or CAMAC related) crates</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>CAMAC (or CAMAC related) modules</td>
<td>2000</td>
<td>4000</td>
</tr>
<tr>
<td>8 bit microcontrollers</td>
<td>1300</td>
<td>2600</td>
</tr>
<tr>
<td>multiplexers for analog signals</td>
<td>300</td>
<td>600</td>
</tr>
</tbody>
</table>

Conversion, Switch-over, Start-up

Conversion of existing accelerator controls meets two major obstacles: (i) every piece of equipment is affected and every individual is more or less involved. This implies historical "faits accomplis" and discussion partners with good controls competence but of heterogeneous opinions. (ii) the constantly running accelerators leave no time for tests with process hardware except in long annual stops; then too much must be done in too short a time which must be shared with urgent maintenance activities.

Point (i) implies that human communication is a substantial part of the job (a) for reaching workable compromises between cost and homogeneity, and (b) for defining what the system should do and look like for operation. Aspect (a) was dealt with by a cumbersome but ultimately successful exercise which yielded accelerated process interfacing standards, a complete set of interfacing modules and performance specifications for each module. After prototyping the non-off-the-shelf material in 1978 and rendering, production and delivery started in 1979. In a second round of discussions late 1979, with specialists of the Booster, timing, operations and software, a detailed layout of the Booster and PLS interfaces was agreed upon. It implied negligible restructuring of existing equipment, either for homogeneity or for new requirements. Aspect (b) was dealt with by including a permanent operational aspects section in the project group: their task was to draw up comprehensive and consistent specifications for future operation. This yielded agreement on the means of interaction (consoles) and relevant input for definition of the software. The result is a highly modular software layout. In addition to the modules specific to one application each, there is the systems and console software, and the so-called applications skeleton which coordinate and arbitrate their execution. The
skeleton was designed and standard protocols were defined in 1978 so that production started in 1979.

This is where point (ii) comes in. After pure software tests and tests with the interface, further progress can only be made with process equipment and, for beam instrumentation, with the beam itself. Between Spring 1979 and Summer 1980 (the long stop meant for switch-over), there were only 2-day stops every five weeks, during which a combination of the development time (with beam) available for the project. Careful preparation allowed efficient use of these short periods of availability. Late in 1979 the complete applications skeleton and a representative sample of specific modules were successfully tested from the first console down to the process. This was a crucial milestone confirming the viability of the system and giving the confidence that it was now "only" a matter of time and hard work. A second encouragement came when in Summer 1980 the Antiproton Accumulator started up, reliably served by the standard interface and lower level software modules," supplied as a fringe activity to the Booster conversion.

Of the four months (mid-July to mid-November 1980) switch-over shut down, the first 1½ months were taken for hardware switch-over and checkout, the second 1½ months for overall system tests including software. The last month was used for beam tests. Since time was short, applications people (the last in the chain, with a fixed end date) worked incredible hours. The beam tests proved to be the most effective period, since all relevant people were available simultaneously and the beam reveals bugs otherwise tedious to find. The month of beam tests was exceedingly hectic. For outsiders the situation was disconcerting; few people had the overview, and their confidence level at that time remains a secret. As by a miracle the situation fell into shape the weekend preceding the scheduled start-up date and - after some hesitation - it was decided to go ahead. The start-up was successful and thanks to a strong presence of all participants the run until Christmas attained about 92% of its planned operation time with only 1.6% downtime attributed to the new controls system.

**System Software**

The topology and the tight timing requirements imposed a weighted coexistence of interpreted software (consoles) and compiled code (process computers), so the systems software had to provide a stable environment for both interpreted and compiled applications programs. Thus for the operating system, the network package, languages and libraries, methodologies had to be sought between performance and protection, and maintainability had to be maximized.

The network (adapted from SPS) was running using the computer manufacturer's operating system, and the Nodal interpreter (also adapted from SPS) was available some two years before beam date. Libraries and facilities for compiled programs were added progressively, and real-time Pascal one year before beam date. Thus, a stable base was provided for applications development. For the three months encompassing the start-up of the new PSB controls, systems software modifications were restricted to corrections considered essential; there were nine such corrections. Help for users of the system software, i.e. the applications programmers, became a full-time job during this start-up period.

Performance and response-time constraints (delivery of several hundred data points to display screens every 650 ms cycle via the network) dictated the use of relatively unprotected mechanisms to access serial CAMAC, the data links and certain files. Relevant drivers reside in a global common area and applications using the links or CAMAC run on a high privilege level. In all other cases, conventional operating system protection mechanisms are used as much as possible.

Experience during the start up period confirmed that the system performance is approximately as expected, provided that enough physical memory is available to eliminate disc thrashing. File access time is in certain cases excessive, so a rapid-access file package which has been implemented will be put into service shortly. Still, occasional unexplained systems crash must be interpreted as the price of sacrifice of protection for speed.

**Consoles**

During start-up, two groups of users must share the various resources: (i) the builders of the different software and hardware elements, (ii) the machine operators who must prepare the machine.

Three access levels to the control system are provided: (i) autonomous access to the process hardware using mobile microcomputers and terminals, (ii) access to the process through mobile terminals connected to the process computer, (iii) access through main operator consoles in the control room. These last two are directly involved in the start-up.

The mobile terminals (ii) available since the setting-up of the CAMAC loops, proved extremely useful for: (a) verifying that all hardware is running properly, (b) verifying, by direct access to the process variables, that programs file in the correct order, (c) interactive access for debugging the microcomputer programs in CAMAC. The mobile terminals have proved indispensable at run-in. During routine operations of the accelerators, they are locked out.

Three main operator consoles were in use during start-up; two were available for operations, the third was reserved for software testing and debugging. The great care put into the inherent security of the interactive software tools proved to have been absolutely essential and effective. For any conceivable interactive manipulation, deadlock situations are avoided. Moreover, the provision of features to unblock application programs has turned out to be very useful.

The use, in the control room, of a terminal connected to the console computer, enabled us to check the proper functioning of a complete control activity and to make patches. Not initially foreseen, it allowed quick deblocking of many an awkward situation.

**Process Interface**

The process interface was structured to keep coupling between subsystems weak to provide sufficient autonomy. This proved very helpful during the step-wise installation and tests of subsystems. One aim, graceful degradation, was convincingly demonstrated when proton production continued even beam property modulation while the process computer was down.

Relieving the process interface of real-time constraints by distributed processing and separation of data highways from channels transmitting time-critical events, proved effective when, during tests, the cycle time was reduced to 0.3 s and no sign of saturation of the process interface was found.

Providing generous numbers of analog signals on the consoles proved crucial also as fall-back in case of other impediments. Due to the bandwidth of the system and geographical distribution over the whole site, maintaining a high signal to noise ratio in all branches was no mean task.

Much emphasis was put on diagnostic tools, hardware and software, on-line and off-line. They became available in the prototyping stage of the standard modules, giving benefits in all phases: prototype testing, production in industry, acceptance at CERN, installation
and on-line debugging. Diagnostic procedures were a crucial element in the comprehension between designer and producer. Diagnostics are possible from all access points to the system, it became an early habit to start diagnostics of the process interface right from the main consoles. Now operators can coarsely locate failure sources or even cure some before the expert gets in.

Thorough quality control of incoming hardware and a one-week temperature cycling followed by an exhaustive test eliminated most failures otherwise occurring during the first months of operation. This intense care proved justified in the widespread application of microcomputers where hardware and software failures superimpose. Early crashes due to hidden software faults occurred, but after debugging the reliability has increased to a level exceeding that of the process computers.

**Applications Software**

The applications program production effort for the Booster and the cycle program generator was estimated at 40 man-years, while little more than 1.5 years were available. Accelerators were continuously in operation and the 4 months installation shutdown had to be shared with other activities. This left effectively 2.5 months for application software tests during this shutdown including 1 month from test. The planned start-up date was met, but only at the cost of an extreme effort of the applications people involved working long overtime for months on end.

Though testing was correctly estimated at 70% of the total production effort, i.e. with the interface and equipment, first without beam (10% = 4 man-years), then with beam (10%), the elapsed time was too short. Even though the software skeleton and its major control functions, including beam property modulation, had been tested with the corresponding hardware earlier in the year, the bottleneck was clearly the tests with beam. Only the actual beam tests could bring all people together at a sufficient alert-level for efficient systems debugging in an environment where tracing of bugs to their root causes was excessively difficult. Indeed, for many months on end, it became an early habit to start diagnostics of the process interface right from the main consoles.

**Operational experience**

Three months after start-up it is too early for balanced appraisal of a complex new control system: first, because of inevitable bugs and teaching problems; second, since some attractive facilities are still due; third since the users have not fully made the conceptual switch from old to new controls.

The feasibility of the following guiding concepts was proved: (i) control from one central room, (ii) general purpose (non-dedicated) consoles, (iii) trees structure for accessing relevant accelerator parts, (iv) parameter reservation in multi-operator environment, (v) different access according to operational mode, (vi) comprehensive alarms, (vii) working parameter sets reflecting virtual machines, (viii) direct analog signals at one's fingertips.

**References**

2. J. Bollriot, M. Bouthéon et al, this conference.
5. S. Battisti, P. Heymans et al, this conference.
8. I. Boillo; J. Moulin et al, this conference.