5.2 SPS Distributed Control System: The Emergence of Local Area Networks

Pier Giorgio Innocenti

The control system for the SPS incorporated a host of innovations. These included the first truly distributed multi-computer control system, where any action could be carried out from any part of the system, the first multi-computer operating system, the first large scale use of an interpretive language for the applications programs, the introduction of the Data Module for isolating the applications programmer from the hardware details, and the provision of simple-to-program colour graphical and other interfaces to the operator [22,23].

In the early 1970s, when the SPS design was being discussed, computers had been applied to issues relating to control in a number of accelerator laboratories, but mostly in an *ad hoc* fashion. When more than one computer was involved, they either performed independent control on different parts of the accelerator, or were organized in a master/slave relationship. For the SPS, given the large size of the ring, with its six access shafts and equipment buildings, and the need for local commissioning, it was mandatory to have the computer system distributed around the accelerator. Each of these distributed computers had to be able to operate autonomously. Instead of having them report to a single large central computer, as was normal practice for such a system, they had to be connected to a network that would allow any two computers to communicate independently of the rest. Such a network is now called a Local Area Network (LAN) [Highlight 9.3], but no such system existed at the time. Even Ethernet, for which it was early days (not yet proposed as a standard), would not have satisfied all the requirements, e.g. long range operation with cable runs of up to 6 km. Moreover, high level network protocols ensuring the cooperative working of several computers were not available. A packet-switching network was designed for this application, the specification drawn up and a contract placed with industry for its implementation. The network featured star topology, with a message handling node at the centre. Links of twisted pair cables were operating at 750 kbit/s, with repeaters every 2 km. Computers were installed in each of the six service buildings, in the experimental areas and in the control centre: in the service buildings to control the machine components, in the experimental areas to control the secondary beams and to interface with experiments, in the control centre to drive operator consoles, provide data storage for the machine data base and for program development.

A crucial decision was to use interpreted rather than compiled applications programs. It was clear that developing the applications programs for the operation of the SPS would require a very large effort in a relatively short time. The typical solution would have been for the equipment experts write detailed specifications and to engage a large number of programmers to turn them into code, debug and modify them as the experts changed their specification. This was not the route followed, because it was thought preferable to provide a user-friendly and safe computing environment, such that the accelerator experts could write most of the applications programs themselves. The key was the use of an interpreter, a program which executes instructions in a high-level language directly, rather than compiling them and then executing the resultant machine code. Since no interpreter for a multi-computer system was available, a new one was developed, called NODAL [24]. This was the first true multi-computer operating system. Commands could be sent from any computer to a target machine for executing locally a subroutine (procedure) and returning the results to the caller: it performed what was to become standard practice — Remote Procedure Calls (RPC).

Precise timing and sequencing required for accelerator operation could not be executed by the computer network: a separate timing network distributed clock and coded "events" with microsecond precision. The clock and events were used to trigger accelerator components directly and to start programs in computers.

Another innovation was the 'Data Module'. Until then, in most accelerator control systems using computers, the applications programmer was expected to incorporate direct references to individual hardware interface devices in his programs and carry out any necessary conversions. However, as subsequent changes to the interface might involve changes to many programs, the risk of introducing errors was large. A data module dealt with all the low level interaction needed for the control of a given type of equipment, e.g. power supplies, taking into account variations between different types. The application programs could then call the data module, only having to give the name of the power supply, the action to be carried out and values in engineering units. A change to an interface, or a modification to a power supply, only required the data module to be modified, leaving the application programs unchanged. The Data Module combined data and associated actions (methods) into a logical structure (object): it was the first example of what is now called Object-Oriented Programming (OOP).

Innovative devices were developed for presenting information to, and receiving instruction from the operator in the control room: Programmable 'touch buttons' on a display [Highlight 5.3] and a track ball pointing to regions of the display itself, as done today with a mouse, touchpad or joystick. In addition, a computer programmable knob could be dynamically assigned to functions more easily performed by turning such a knob. All computers in the network could be equipped with a colour graphics system which allowed computer generated displays to be incorporated in a common manner.