THE COMPUTER CONTROL SYSTEM OF THE SPS

M.C.Crowley-Milling
European Organization for Nuclear Research, CERN, Geneva, Switzerland

When the construction of the 400 GeV accelerator was approved early in 1971, it had already been recognized that new ideas and techniques would be required to meet the formidable task set by the control requirements. It would be necessary to deal with many thousands of measurements and controls and tens of thousands of bits of status information, spread out over a site of 15 square kilometres, and yet make it possible for two or three operators to set up and run the accelerator to supply high energy protons round-the-clock to experiments set up by groups of physicists from all over the world.

It was clear from the start that computers would have to be involved to assist the operators in their task, and to organize the multiplexing of many signals on to relatively few cables, as it would have been out of the question to run cables from each individual equipment to a main control room. Even with the use of multiplexing wherever possible, the total length of cables required has exceeded 1500 km. Having accepted that computers must be used it was necessary to determine their number and size, the means to be used to connect the equipment to them, how the operators would communicate with them and, probably the most important question of all, how the software was to be provided.

In solving these problems, a number of special requirements had to be taken into consideration. It was intended that the accelerator equipment should be designed for computer control from the start, with many of the conventional manual controls left off, so the computer system had to be available for the testing and commissioning of the equipment well before it was needed for running the accelerator. This meant that the design and initial implementation of the control system had to be carried out in a little over two years, in such a way that progressive additions could be made without having to make changes to the work that had gone on before.

In addition, a particle accelerator of this type is not a piece of apparatus that when finished remains unchanged. Not only is the accelerator used as a source of particles for experiments that may change their requirements with time, but also about ten percent of the operating time is spent on trying to develop the accelerator to give higher intensity, higher energy or even to accelerate different particles, so the control system must be flexible enough to allow changes to be made both to the equipment itself and the way it is used, without difficulty.

The third special condition was that most of the accelerator components, magnets, power supplies, r.f. equipment, vacuum system, injection and extraction apparatus, etc., had to be designed and constructed specially, often at the limits of available technology. The detailed methods of controlling these together could often only be determined when they were commissioned.

These requirements meant that the software system must be extremely flexible, and that the writing, debugging and modification of programs be made very simple.

In most computer control systems the programs are written in the assembly language for the computer, or in some high level language such as FORTRAN, or CORAL, by professional programmers and compiled into object code, which have to be linked into the operating system, loaded and run. It is very unlikely that a new program will run correctly the first time, and it may be difficult to find out why, especially as, with the comparatively simple operating systems of the computers used for control, a program error can cause the whole system to crash. Even when the fault has been found, the program has to be edited, recompiled, linked and loaded. The chances of errors and the difficulties of debugging are increased enormously in a multi-computer system. If programmed in the normal way, an array of programmers would have been needed to provide the facilities required for the SPS, even if it had been possible to provide all the information in sufficient time for the work to have been carried out.

A new approach to programming was clearly called for, and this was based on the use of an interpreter. An interpreter is a program which takes each statement of the source program, checks it for syntax, and links together subroutines to carry out the action called for. A single statement can be executed at a time, in a fully interactive manner, or statements can be linked into programs. Because of the checking carried out at each execution, and the "insulating" effect of a correctly designed interpreter, it is almost impossible to crash the system by making errors in the source programs. The provision of a simple but powerful command language (see below) has made it possible for the engineers designing the various parts of the accelerator to write the programs to test and commission their equipment on its own, and then to modify and extend them to use the equipment in conjunction with the rest of the accelerator. This reduced the number of professional programmers required and also avoided the usual difficulties of communication between programmers and equipment designers and users.

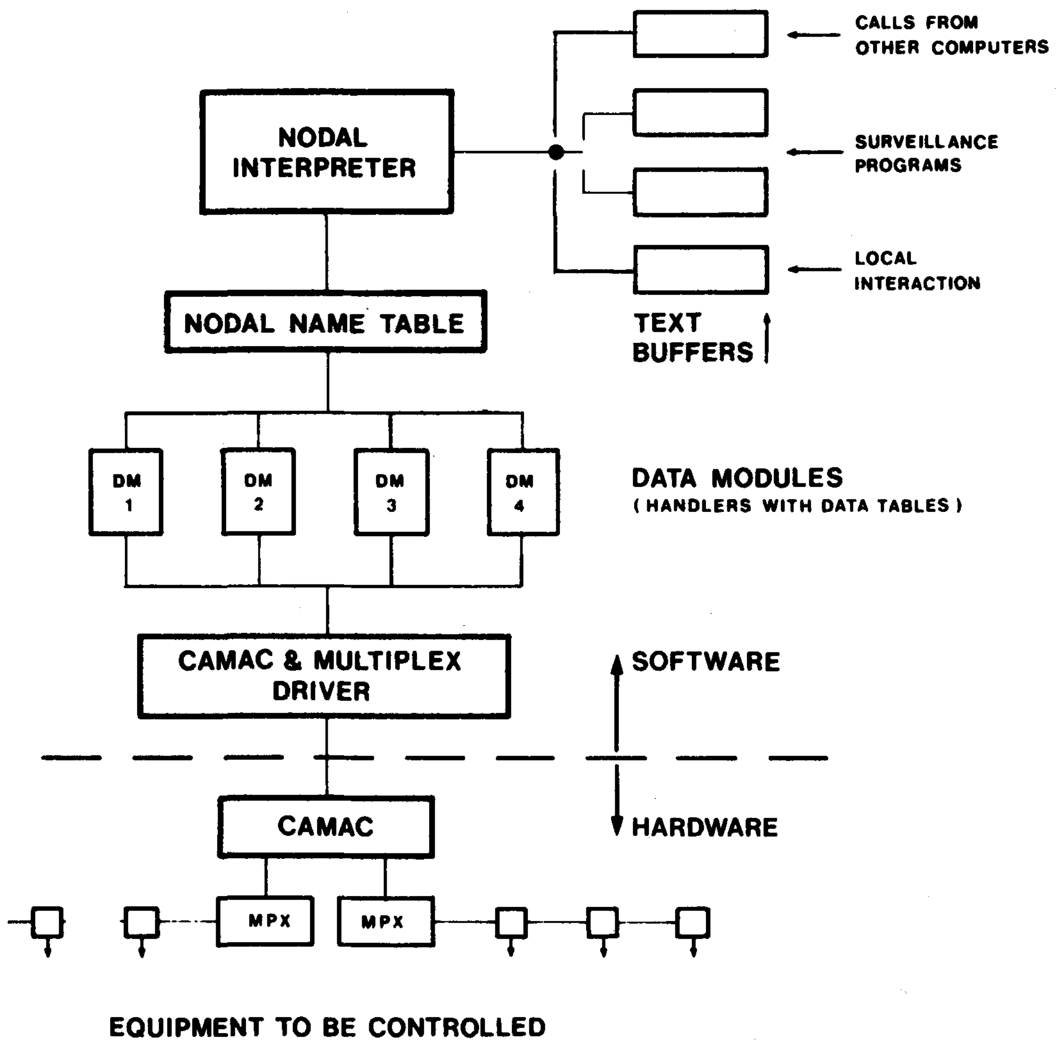
As stated earlier, an interpreter is basically a linker of subroutines; some of these are the functions normally provided in any interpreter to carry out mathematical operations, etc, but to use an interpreter in a control system other subroutines are required to interact with the hardware. These must operate on some form of data base which contains the information such as hardware addresses, conversion factors, status, etc. It was decided to have a fully distributed data base, with separate data tables and drivers for each basic type of equipment, such as pump, power supply, fan, stepping motor, etc. Although this involves some duplication, it means that each of the subroutines, called data modules, could be designed and implemented as soon as the elementary actions to be performed on the equipment had been defined. If a common data base had been used, it would have been necessary to wait until all the equipment had been defined, and any change in one part might have affected others.

The operation of the software system can be seen from Fig. 1. Multiprogramming is carried out by having a number of text buffers which are scanned according to priority by the re-entrant interpreter. On reading a statement involving a named variable, the NODAL name table is scanned to determine which of the data modules to call to perform the action required. When the parameters are passed, the data module uses the universal hardware driver to access the hardware through the interface system, which uses a combination of the international standard CAMAC and a specially developed multiplex system.

The Interactive Language

The high level interactive language NODAL used for the control of the SPS was developed to simplify the programming of real-time multi-computer control systems. It was designed to be interpreted, and combines many of the best features of FOCAL and BASIC, with the additions needed for real-time and multi-computer use, and has the extensive string-handling facilities of SNOBOL4 /2/.

Commands can be executed in the immediate mode, or built up into programs using the group and line structure. A program can call for the execution of one or more lines or groups in another computer, with the remission of the results. Commands are provided to ensure synchronization between programs and subroutines in different computers.



SIMPLIFIED SCHEMATIC OF SOFTWARE SYSTEM

Fig. 1.

A simple program that demonstrates some of these facilities is given below.

1.1	LOAD MYFILE	Get "MYFILE" from the library (MYFILE contains the array "AR")
1.2	EXECUTE(GP6) 3 AR; WAIT(GP6)	Execute group 3 of this program in computer GP6, passing the array "AR". Wait for a reply from this computer.
1.3	FOR I=1,10; TYPE "THE POSITION AT" I "=" PS(I)!	Type out the lines giving the position of the beam at each point, as given in the array "PS".
1.4	SAVE LOG PS	Save the array "PS" on the library file "LOG"
1.5	END	
3.1	FOR I=1,10; SET MAGNET(I)=AR(I)	Set the currents in the ten magnets to the values given in the array "AR".
3.2	DIM PS(10); WAIT-CYCLE(6)	Create the floating point array "PS" and then wait for a specific event in the machine cycle
3.3	FOR I=1,10; SET PS(I)=DET(I)	Read the values of the ten detectors and set them into the array "PS".
3.4	REMIT PS	Remit the array "PS" back to the calling computer.

The Computer System

The layout of the computer system was partly determined by the physical constraints. Most of the equipment to be controlled is in the nine auxiliary buildings and the two experimental areas. In some cases a complete subsystem (radio frequency, injection, extraction, or main magnet power supply) is housed in one building and a computer can be dedicated to that subsystem, but others are distributed all round the accelerator. For the latter, there is one general purpose computer in each building which deals with the local parts of all such subsystems.

At the time the system was designed, the few multicomputer control systems that existed had used satellite computers as slaves of a large central computer, and multicomputer operating systems were not available from any minicomputer manufacturer. Since it was necessary to develop a new system, a considerable effort was put into making an assessment of the likely requirements for the computers

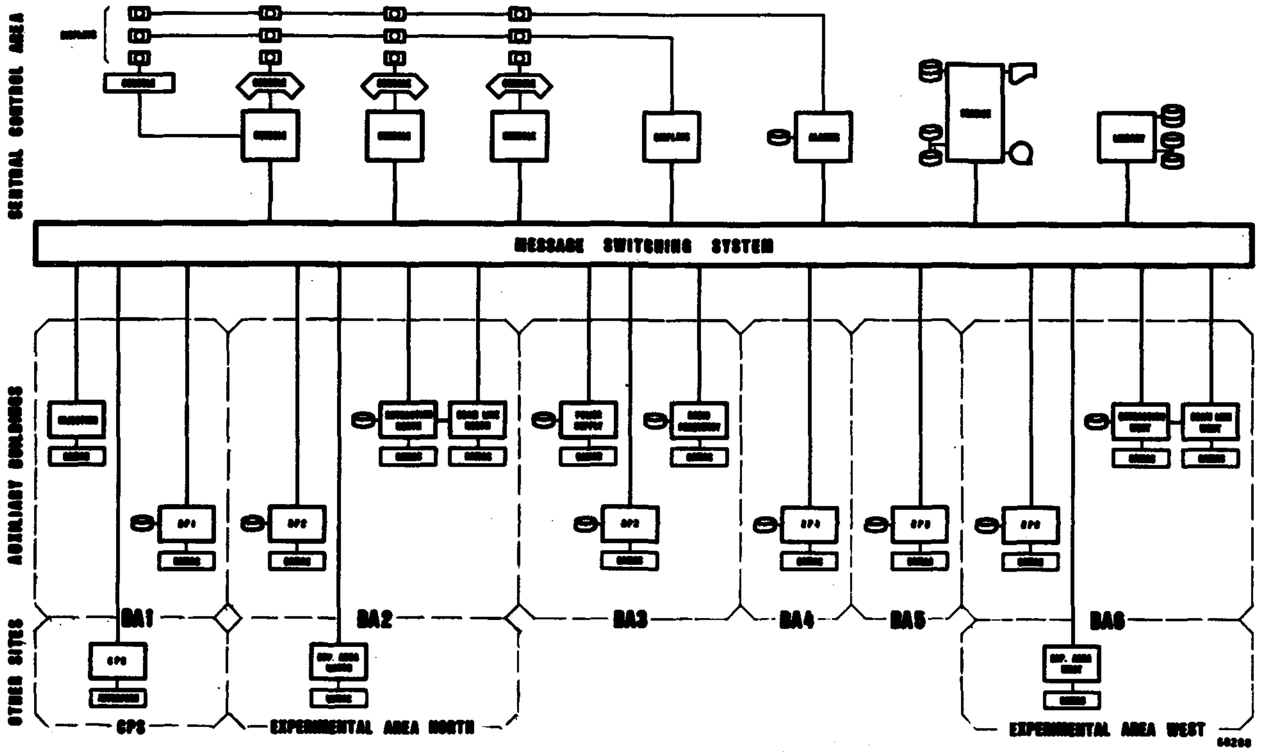


Fig. 2. Layout of the Computer System

and the intercommunication system. The outcome was a system of identical mini-computers, some acting as the satellites, and others performing the various duties which would normally be carried out by a large central computer. The layout is shown in Fig. 2; the computers are connected by data links to a message switching computer which forms the node of the network. Any computer can originate a message to any other one, and the messages are sent in the form of "packets" each package consisting of a 4-word "header" which gives the routing information, and up to 64 words of information. A special protocol was developed to minimise the software overhead on the transfer of packets, which travel on the data links at the rate of 500,000 bits per second (most of the packet switching systems now coming into operation use data links operating at a maximum of one tenth this speed).

Operator Interface

As can be seen from the computer layout diagram, the control consoles are in effect peripherals to the console computers, and there are no direct connections to the accelerator equipment. (The safety interlocks for access into the accelerator form the only exception and they are directly connected to a separate special console.) This has made it possible to have three identical consoles, each of which can be used to control any part of the accelerator. The selection of the part of the accelerator to be operated on is made in the software by means of sequential choices in the form of a push-button "tree", in which firstly the system is selected, then the sub-system, and so on. Instead of real push-buttons, button images are created on a TV monitor, the face of which is covered by a transparent screen on to which a number of capacitor grids are deposited. When touched by a finger, the local capacitance is increased, triggering an interrupt, which tells the computer which button has been chosen. The next layer of choice is then displayed, and the operator guided through the correct sequence of operations. Further interaction can be carried out using a rolling-ball to position a cursor on a colour display screen which can have alphanumerical characters or graphic displays on it, and selected values can be set with a knob mounted on an incremental encoder. The whole accelerator can be set up and operated using just these three interaction devices; the touch screen with its almost infinite variety of buttons, the rolling-ball and the knob.

Experience

The computers, and a stand-alone software system with the NODAL interpreter, were available early in the program, and they were used in various parts of the laboratories and workshops for the development and acceptance testing of various parts of the accelerator equipment, speeding up these operations considerably. The simple programs required could be written by all grades of staff, and no special software support was needed.

The next stage, with the computers installed in the auxiliary buildings, connected up to the message transfer system and used to commission the large assemblies of components, caused some headaches, mainly due to the sheer volume of equipment to be connected up and "debugged". Due to the short time scale, the fifty data modules required had to be written by quite a number of people, most of them not professional programmers, and since a mistake in one of these can crash the system, it is understandable that there were some difficulties. However it is clear that these difficulties were minute compared to those that would have been experienced if all the applications programs had been written in the conventional manner. The applications programs were written and tested by about 50 to 60 people, mainly as a part-time activity, most of whom had no experience beyond batch FORTRAN, and many had never written a program before. The insulation provided by the interpreter allowed these activities to go on without much mutual interference.

Adequate programs were available by the time the accelerator was ready for the first beam tests to enable the power of the control system to be demonstrated and the tests speeded up, as has been reported in another paper.

The accelerator is now in full operation for physics, and the control system has been extended beyond the accelerator, to the secondary beam lines, enabling the experimental teams to control elements in their beam lines, and to record beam-line and accelerator parameters on their data tapes.

Conclusions

What can be learnt from the experience with the SPS that could be applicable to other control systems?

Firstly, interactive facilities for writing and editing applications programs on line are essential if the software for a large control system is to be provided on a tight timescale. Although incremental compilers can help on suitable systems, the full facilities can only be provided by an interpreter. One of the greatest time savers is the ability to "debug" a program on line, running it step-by-step until something goes wrong, then using the immediate command facility to find out the error and then editing the program to eliminate it. Because of the amount of work an interpreter has to do every time an applications program is run, it operates considerably slower than a compiled program to do the same job. It was thought in some quarters that this would be a serious disadvantage, but this has not proved to be so in practice, due to a number of factors. Many of the control actions are geared to the operator's speed of comprehension and reaction, or to the operating speed of valves, motors, precision analogue acquisitions, etc, and the interpreter is still fast compared with these. In addition, provision has been made for the incorporation of blocks of compiled or assembled code as functions of the interpreter, so that if an interpreted program is found to run too slowly, suitable parts of it can be encoded and called from the interpreter. This work is simplified by having the logic already tested in the NODAL program. Lastly, a number of frequently repeated actions or data manipulations can be incorporated as additional properties of the data moduleswhich are assembled into the system.

Secondly, that for any but the simplest, fixed-duty controller type of application, it pays to be lavish with the hardware where this can save software effort. Having a resident interpreter in every computer meant that the size of the fast-access store had to be increased, but this was negligible compared with the simplification of the program organization in the multi-computer system. This, together with mass-storage on all the satellite computers, meant that they could be operated as stand-alone systems for local tests and commissioning without having to reconfigure the software.

Thirdly, the use of the touch screen, rolling-ball and knob, together with displays, all of which can be programmed in the interpretive language, allows the use of a general purpose console which could be applied to any process without change. It has been accepted with enthusiasm by all those who have to operate the accelerator. The touch screen, designed at CERN, is now available commercially, and is finding applications in power station control and other fields.

References

1. M.C. Crowley-Milling, The Design of the Control System for the SPS, CERN 75-20 (1975).

2. M.C. Crowley-Milling, T. Hyman, G. Shering, The NODAL System for the SPS - 1974, CERN Internal Report, Lab II-CO/74-2 (1974).

ДИСКУССИЯ

H.Kumpfert : Could you please comment on the availability of CAMAC-modules? What percentage of your CAMAC hardware was bought from industry and what had to be built in your own laboratory?

M.C.Crowley-Milling: The CAMAC system was designed for data acquisition purposes from experimental installation. In this case there is no necessity in control but in our case we need to control a great number of systems. Quite often 1/6 or 1/2 part of the CAMAC module was used for control. In our lab we have developed some controller which easily enabled to construct a control system.

В.В.Комаров: Какова надежность системы? Каково время устранения неисправности?

M.C.Crowley-Milling: The system reliability is very high. To estimate it quantitatively is a matter of difficulty. The troubleshooting time depends on the location of the failure. If this is known, it takes just a few seconds to replace the relevant block.

В.Л.Серов: Производилась ли первоначальная наладка ускорителя непосредственно с использованием ЭВМ? Имеются ли системы местного ручного управления отдельными устройствами? Означает ли остановка ЭВМ лишение Вас возможности изменить режим работы устройства?

M.C.Crowley-Milling: Yes. There are systems which have a lot of units: vacuum system has 200 pumps, and it is difficult to control it without computer.

Yes, we have. No energy system can work without computer.

В.И.Нифонтов: Каким образом применение микропроцессоров могло бы изменить общую структуру Вашей системы управления?

M.C.Crowley-Milling: This is a good question to ask. We all are thinking hard about it. Certainly if we were to design the system again we would do it in another way.