SPS/ACC/Note/85-8

8th March, 1985

THE INFRASTRUCTURE FOR THE MAIN CONTROL ROOM FOR e+ e-

J. Altaber, P.S. Anderssen, R. Bailey, M. Flückiger, I. Gjerpe, R. Lauckner, C. Saltmarsh, D. Thomas and I. Wilkie

1. Introduction

By mid 1987 electrons and positrons will circulate in the SPS ring. This will require a multi-cycle capability, which is a major change in the SPS operation. The operation group feels strongly that it implies a complete rethinking of the way the control room infrastructure has been designed and constructed ten years ago.

There already exists papers[1,2] that have expressed concern about the stringent limitations of the SPS control system as the accelerator evolves into a more and more complex machine. The SPS division now has to address the problem of how to approach a major rewrite of the control software. A well conditioned project of such magnitude does not happen by chance. The problem must be properly analysed, a structure carefully and rigourously designed, and then and only then can the software modules be coded, tested and incorporated. The importance of the various phases should not be underestimated, cutting corners in the early stages of the project will undoubtely produce serious consequences later on.

It must be stressed that discussions which have taken place so far have only been concerned with one aspect of the overall problem, namely the environment in which the work is to be done and which is the subject of this paper. It is of course true that this is an important aspect. It takes a lot of experience and research to decide which computing facilities, network, operating system and languges are and will continue to be appropriate. But these are only the tools. The major part of the task[3,4] remains how the people who design and write software will use the tools to produce control software versatile enough to fulfil the requirements in the coming years.

The proposal described in this paper preserves the SPS control system as it is to-day and which is compatible with its progressive evolution in the direction of the LEP architecture. In addition, it integrates smoothly the LEP control system thus opening the possibility to operate the LEP-SPS complex from a unique main control room.

This goal has an important bearing on the boundary conditions of the problem. Firstly, on time scale: we design and start the implementation of a new main control room in 1985, it must be available early 86 and in operation in 1987. The combined LEP-SPS operation will start late 1988. Secondly, the architecture must be sufficiently flexible to provide a progressive implementation and a possibility to integrate new products which are bound to appear during this four-year period.

Last but not least, it is highly desirable that the facilities which will be implemented for the main control room are also available as a large subset all around the field of the LEP and SPS machines.

2. The SPS NODAL system and its future evolution

2.1 The SPS NODAL system

The overall SPS control system is constructed around the NODAL interpreter. Very significantly its description (CERN yellow report 78-07) is entitled "The NODAL system for the SPS"(5)..

The NODAL language provides features for:

- data gathering and control through a distributed architecture,
- mathematical and string processing,
- man-machine interaction,
- the implementation of utilities strongly linked to the single cycle operation.

Taking into account these very broad objectives, the underlying software has been constructed with a strong focus on real-time response. It consists of a small resident real-time kernel named Syntron, an efficient file system minimising all mass storage access time, an optimised network package which carefully makes use of every single bit of the potential bandwith.

Despite the speed penalty which is inherent to an interpreter, the overall NODAL system behaves well and has been used very successfully without major modifications all along the ever changing past of the SPS machine.

2.2 The evolution

Since the pioneer conception of the NODAL system, many things have happened both inside the SPS and in the outside world. On the SPS side the operation and the machine study teams have gathered an extensive knowledge of the SPS machine but are also struggling with the limits of the system. On the other hand, in the outside world, a large amount of integrated technology housed into a number of commercial products is available off-the-shelf. Furthermore, the importance of the software part of the control system is increasing. The time has now come to move into infrastructure for the main control room. As a result of the flexibility of the present architecture, this new generation can be adequately bridged with the former thus preserving its essential features.

NODAL has two outstanding and so far unexcelled properties: interactiveness and networking; these qualities have often been described as "extended eye and arm" facilities(6). It provides very suitable tools for the commissioning, testing, development and learning of a complex machine; it may even be sufficient for the operation of a number of subsystems. Some improvements are necessary and the new NODAL interpreter written in Modula-2 will contain enhanced on-line editing and debugging features(7).

The NODAL system is kept for the machine network transaction. Above it, an operation and programming environment will provide the necessary features for a new generation main control room.

3. Programming and operation environment

This title covers all the working conditions for an operator who either develops a control program or operates the machine with already existing control programs. This includes the ergonomy of the main control room which will not be treated in this paper and the software/hardware facilities which will be available and which is our main subject.

3.1 The architecture

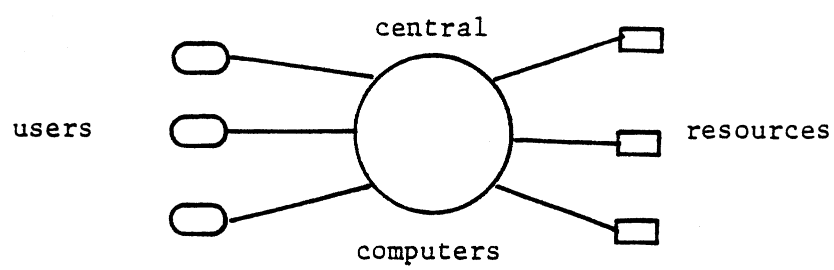
The operating system is the key component of the environment as it provides the "glue" for all the facilities which constitute it and thus it has a strong impact on the architecture. Until recently the main task of an operating system was to share resources amongst many users. There are basically two types of such resources:

- resources which are expensive and which are worth sharing. Such resources are CPU, memory, mass storage, printer,...

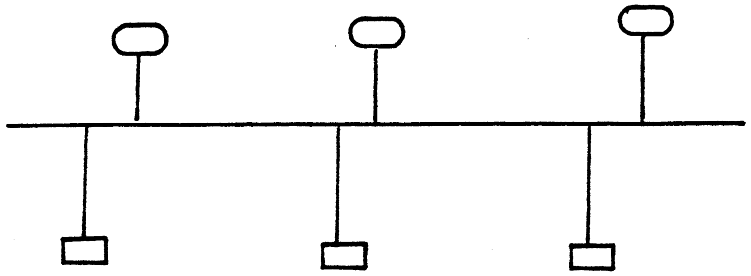
- resources which must be shared because users work on a common task. An example of such a resource is a file.

A major change in this field is happening nowadays, following the drastic fall in the price of personal computers. The personal computer provides a powerful micro CPU, large memory as well as medium size mass storage at such a price that sharing these resources amongst several users is not worthwhile any more.

Nevertheless, there are resources which have to be shared either because they are still expensive (printer/plotter) or because they are a meeting point between users (files). But here again, things are moving very rapidly. Until now the only way to share was to concentrate users and resources on a central point, the computer.



The availability of a Local Area Network (LAN) is changing completely the picture, the sharing is not provided centrally anymore, but through a distributed architecture as shown below.



Each element of the distributed system is given enough hardware/ software to execute almost autonomously. The previous main task for the operating system in a centralised computing facility is nowadays vanishing either because the resource has become cheap or because the sharing has gone on the LAN. Distributed architecture is definitively taking over centralised architecture, a trend that the SPS control system has pioneered 10 years ago, and that is pushed even further for the LEP control system[8].

3.2 <u>The components</u>

The personal computer will be an essential component of this architecture for the main control room. Large memory, medium scale mass storage, fast micro-CPU are the major ingredient of the receipe. A very interesting point is that personal computers can be enhanced as a powerful man-machine interface, a subject on which manufacturers have been very active.

We have already mentioned the necessity to share files and a large global mass storage together with a back-up facility which has been and remains a vital component of the architecture; the mass storage attached to the personal computer will only be used to contain the system and private files.

Hard copy devices such as a printer/plotter, possibly in colour, will be a necessary component as well.

The Master Timing Generator (MTG) is another good candidate to be included in the main control room components, as it should be accessible from all "consoles".

A new trend is appearing through proposals[9] to introduce modelling capability in the control system. So far the problem was not sufficiently mature and has not been implemented but things are moving fast in this field and we have to foresee a component which can provide operator assistance by modelling capability. The events[10] to take into consideration are:

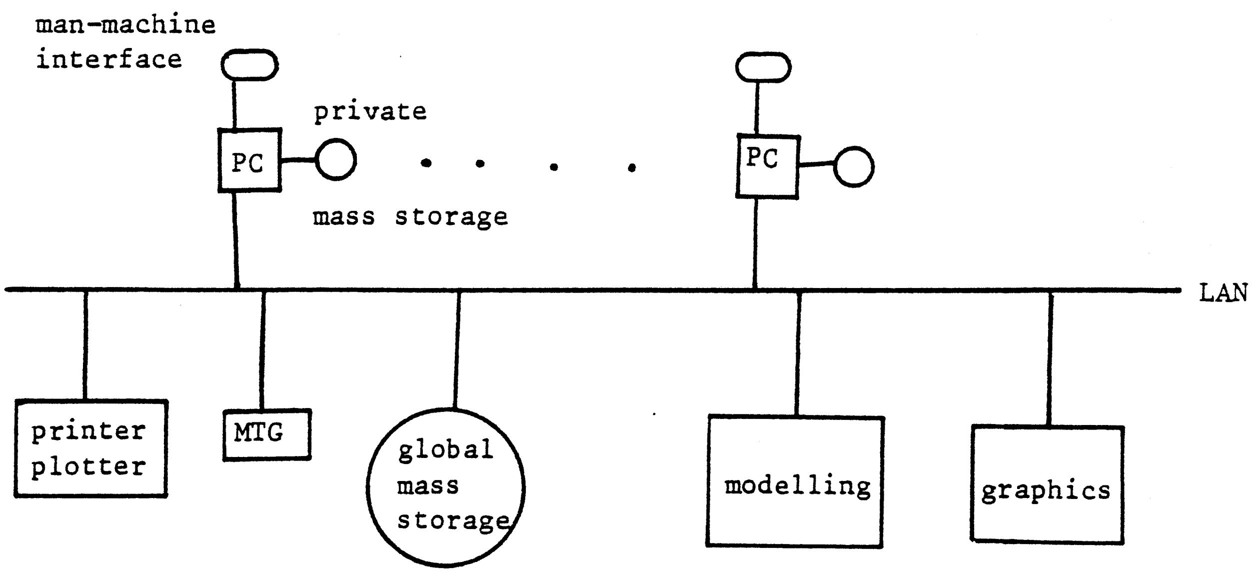
i) a new modelling program called MAD is in its polishing phase and execution time has been significantly reduced from the former AGS program,

ii) an interactive environment called MADNESS is being developed and will be a significant improvement from the previous "patchy"-like off-line access,

iii) the presentation of the results through powerful interactive graphic devices is being experimented at Brookhaven and, if successful, will make a significant improvement in the usage of modelling,

iv) experiments are currently under way in the SPS for data acquisition and analysis.

From this list of components we can refine the picture of the main control room into the diagram below.



The important point to underline about this diagram is its open architecture. If we choose the network and engineer the software properly, it will be possible, and may be easy, to introduce new ideas or products in the future and in addition, it offers the possibility of a progressive implementation.

4. The personal computers

4.1 The operating system

In the distributed architecture described above the operating system has lost most of its important function of being a means of sharing. Its main duty remains to integrate the software packages which can be run under its supervision.

There are two essential criteria for selecting an operating system: the command language and the software packages available. The command language may be a question of flavour; the main argument about it is that programmers are very reluctant to switch from one operating system to another, so that a portable operating system like UNIX* is very attractive.

A large choice of software is becoming available in the field of compilers, graphics, database. A widely used operating system guarantees the availability of numerous packages at a low price.

The file system is a package which is higly integrated into the operating system. Quite rightly, the hierachical file structure pioneered by UNIX is the most popular nowadays and has been implemented for many other operating systems.

4.2 The man-machine interface

The man-machine interface for the SPS is quite sophisticated. It is constructed in the CAMAC system thus these modules, although subsequently used in many other places, remain expensive compared to commercial equivalents. This is one of the reasons why we have few consoles in the main control room and why the man-machine interface in the field is mostly reduced to a VDU like terminal. This in turn leads to programs which are different whether they are run in the main control room or in the field.

*) UNIX is a Bell Lab. trademark.

Discussions with the operation group and visits to other laboratories like Fermi Lab. have shown that streamlined man-machine interfaces can provide the necessary comfort and can be considered as a simplification for programming and for operation. In fact, having the choice between several overlapping functions does confuse the operator who has to remember or find which devices have been used in each interactive program.

The operation group is willing to evaluate simpler man-machine interfaces such as a keyboard, colour screen(s) and a mouse; all these elements can be bought from industry and can be part of the personal computers.

The software package which goes together with these devices will preferably conform to the ISO GKS recommendation.

4.3 Languages

This is probably the field where many users have felt heavy frustration. The current control system is restricted to the NODAL interpreter and the possibility to call Fortran sub-routines which are produced off-line and packaged into the NODAL data structure.

A very good indication on where we are on the language issue is given by considering the current status of the working space in the console computer.

In the console system today the total working space is 46 Kbytes slit into 18 Kbytes for NODAL program and 24 Kbytes for defined functions which are mostly written in Fortran.

The conclusions from this is clear:

i) make other language insertion into NODAL easier by removing the fixed partition between the NODAL programs and Fortran defined functions, by providing larger working space and by offering a more comfortable development system,

ii) introduce other languages as a language for writing control programs.

It is certainly worthwhile to allow the use of the structure of a programming language like Pascal. The minimal set of languages which can be run in a personal computer are:

- an interpreter: NODAL,
- a mathematical language: Fortran,
- a structured language: Pascal.

The NODAL interpreter will be provided by the ACC group. As far as other languages are concerned, they will be commercial products which fulfil the ANSI standards.

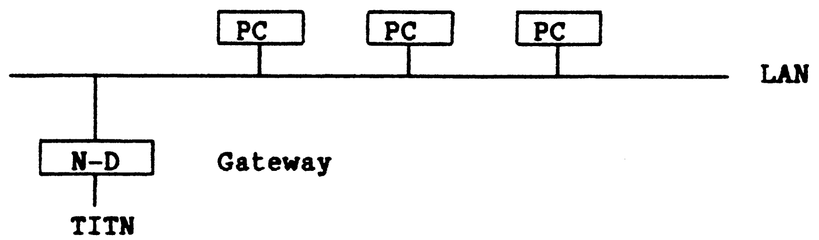
4.4 The bridge

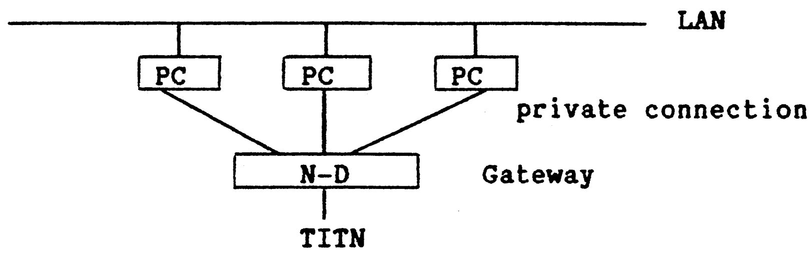
This paragraph deals with the bridging between the programming/ operating environment and the machine network. These two elements cannot be merged into a unique one. In fact, the main characteristic of the former is the flexibility and the confort it provides to the user, while the later deals with response time and data security of control programs. These characteristics are very difficult to combine in a satisfactory manner as illustrated by the two following examples:

- UNIX is an excellent operating system for program development. It has however no real-time features although several attempts have been made in this direction.

- the NODAL system for the SPS offers real-time control facilities but it has shown to be weak on the users environment features.

Physically speaking this connection can be made through the LAN or through a private connection, the NODAL control system hook being an N-D computer gateway.





The first scheme is more elegant but as opposed to the second one it merges on the LAN the control traffic with the N-D gateway and the development traffic between the PC's. The main difficulty with the second scheme is that the private connection which can be used are limited in bandwidth.

For the two schemes the bottleneck resides in the N-D gateway with its unique CPU. If this shows to be a problem we can duplicate the gateways. As far as the LEP control system is concerned there should not be any problems as the gateway will be made from a VME multi-processor assembly.

4.5 Access to the machine network

From the software end, the connection can be made in terms of messages which are structured in NODAL form. In fact, the NODAL data structure is well suited to make a bridge between different programs running in different CPU's. This property has been fully exploited for network operation between two NODAL interpreters by means of EXECUTE and REMIT statements. To illustrate this let us take the example of a Fortran program running in the personal computer and which need to measure the orbit. The program will call a subroutine which is part of the subroutine library and which may look like CALL ORBIT (HOR, VERT). The program which calls this routine expects to find in the arrays HOR and VERT the 108 horizontal and vertical positions of the SPS beam. As a result of calling this routine a NODAL program will be sent through the control system gateway. Studies are under way to decide what is the best way to implement this. A virtual terminal protocol will be available from the personal computer to the control system gateway.

4.6 The photo-fit picture of the personal computer

From the above we can now give a description of the personal computer that we need:

Hardware

- A widely used bus architecture.
- A CPU at least comparable to the N-100 + floating point.
- A range of graphic screens, interactive device like a mouse.
- 0.5 to 3 Mbytes of RAM.
- A range of hard disks (up to 100 Mbytes) with back-up facilities.
- Open networking following ISO 8802 recommendations.
- At least 8 free slots, e.g. for graphics interface, TG3, memory extension, network interface...

Software

- A widely used operating system with hierachical file scheme.
- User access to the hardware features described above.
- A range of compilers including at least ANSI Fortran 77 and Pascal. A Modula-2 compiler will be an asset.
- Full implementation of the GKS package.
- Database management system.
- Server functions.

5. The IBM PC-AT

The upper range PC-AT of the IBM personal computer family appears to suit pretty well with the above photo-fit picture(11). It makes use of the latest Intel chips, the 80286 for the CPU and the 80287 floating point co-processor. It is constructed around a well defined bus for which over 200 manufacturers are producing compatible hardware. This is an outstanding advantage for constructing MTG and its timing module TG3. A host of software packages exists and several of them have been under evaluation on a PC-AT which we have had on loan for the last two months.

The price range of such a personal computer is low enough to simplify the maintenance problem by having a set of spares.

5.1 Operating system

There are two operating systems available; PC-DOS from IBM and XENIX, a UNIX version of Microsoft. PC-DOS is a single user operating system under which most of the commercial software/hardware packages can be run. It does not offer multitasking capabilities which can be a limitation in a LAN environment; in addition it accesses only 1 Mbytes of memory. This memory space is split into an area of 640 Kbytes for the resident system (less than 50 Kbytes) and the user space, the area above 640 Kbytes containing the basic input/output system which drives and buffers the peripheral devices.

XENIX offers multi-user multi-tasking and can address up to 4 Mbytes of memory but it does not support the wide range of commercial software/ hardware packages as the PC-DOS does.

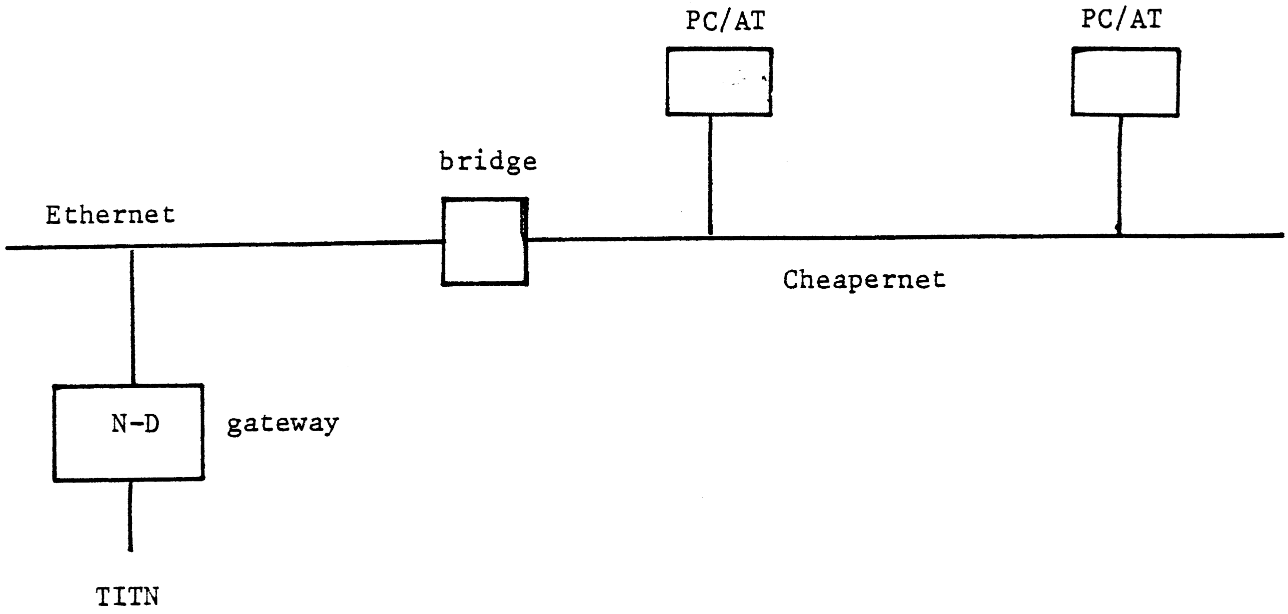
We have certainly to start our work with PC-DOS and continue our evaluation of other commercial software/hardware packages. Nevertheless we are severly lacking information on the evolution of this product which must overtake the limitations of addressing space and single tasking. We have indications that these improvements are "in the pipe" as the internal architecture of the Intel 80286 has all the necessary features for it.

5.2 Networking

On the LAN side we have investigated three possibilities:

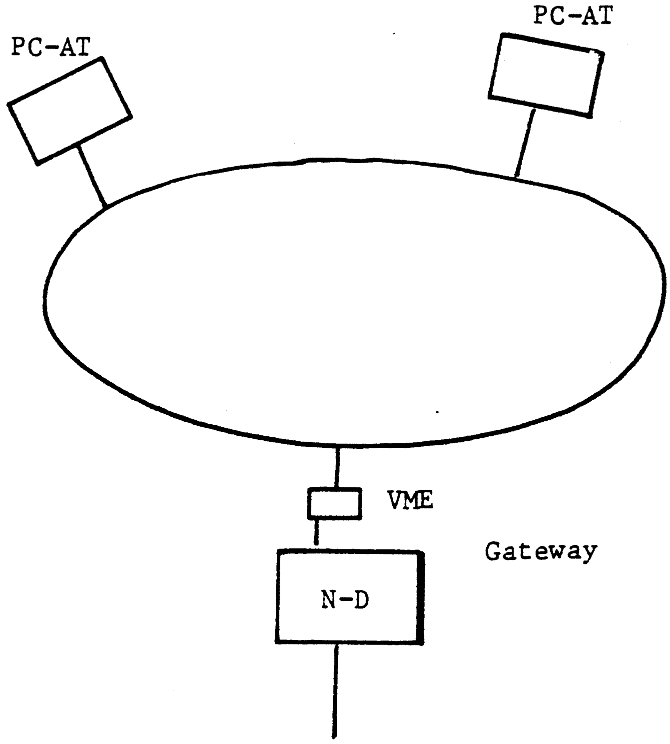
- PC-net which is available from IBM but which has been rejected because it is an IBM proprietary network.

- Ethernet, which is also available in its "cheapernet" version from non-IBM vendors, and for which a PIOC interface exists for connection to Norsk Data computers:



The main disadvantages of Ethernet are its price and difficult connectivity which is illustrated above. It permits relatively short distance (~ 700 m without repeaters) which may not be a drawback for a LAN limited to the main control room.

- Token ring for which a commercial PC-interface does not exist yet but we are trying to obtain information within the CERN-SPS/IBM collaboration on network. The main advantage is that the token ring will be used for the LEP and SPS machines in the future. Of course, the interface to the N-D does not exist commercially but the ACC group is busy developing it in a VME crate.



On the software side the choice is very limited. The ISO specifications stops at the Data Link layer of the OSI-ISO model[(12,13)] and the other levels will appear too late for our work. We are studying a Department of Defense specification named Transport Control Protocol/Internet Protocol (TCP/IP) which is widely used[14)] and for which an IBM implementation may be available.

We hope to set up the experimental Ethernet and token ring before June 85 and to evaluate the implementation of TCP/IP on these two LAN's.

6. Time scale

6.1 First half of 1985

Individual PC-AT's will be available. Two machines have been delivered and another two are expected in March and four in May. Most of them will be installed in the main control room building and will be available for the evaluation of commercial packages. The languages which are currently available are Fortran, Pascal, Modula-2, Basic, and a macro- assembler.

We expect the delivery of a high quality graphics display by April so that we will be able to provide a first implementation of a streamlined man-machine interface for evaluation. A prototype of a touch screen developed by IBM is available as well.

6.2 Second half of 1985

A LAN of PC-AT's will be installed without the connection to the control system.

Most of the PC-AT will be equipped with a first version of a man-machine interface. They will contain the new NODAL interpreter.

A mass-storage device will be connected to one of the PC-AT providing the global file server for which we aim at a capacity of at least 250 Mbytes of hard disk with streamer tape back-up.

6.3 First half of 1986

The N-D gateway will be available connecting the main control room LAN to the SPS-TITN network.

The finalized man-machine interface will be available.

A subroutine library system (4.4) will be in operation.

TG3 with its associated low level software will be available and MTG will be installed by the end of the period.

7. Conclusion

The main control room infrastructure for e^+e^- has ambitious goals and a very tight time scale; its success relies heavily on an extensive use of commercial products. Some of the products are not yet available in Europe but we expect to obtain pre-releases in the next coming months.

The proposal which has been described allows a progressive implementation so that many software developments can start now. In addition, the "openness" of the architecture both on the LAN side and on the internal structure of the PC-AT, guarantees the possibility of future expansion.

References

1) C. Saltmarsh, D. Thomas, SPS as a multicycle machine: software requirements, SPS/AOP/Note 84-14.

2) R. Bailey, Operations software for e^+e^- proton cycles, SPS/AOP/Note/84-13.

3) V. Hatton, Control room software for e^1e^-, note for the SPS Technical Committee of the 01.02.85.

4) Paper from AOP to be published.

5) M.C. Crowley-Milling and G. Shering, The NODAL system for the SPS, CERN yellow report 78-07.

6) J. Altaber, V. Frammery, C. Gareyte, P. van der Sto, An interpretive approach for interactive real-time control through network, IFAC/IFIP Workshop on Real-Time Programming, Schloss Rezhof, Liebnitz (Austria), April 1980.

7) Reports on Modula-2 NODAL and NODAL Editor, SPS/ACC/Note, to be published.

8) J. Altaber, M.C. Crowley-Milling, P.G. Innocenti, R. Rausch, Replacing mini-computer by multi-microprocessors for the LEP control system, 1983 Particle Accelerator Conference, Santa Fe.

9) P.S. Anderssen, Proposal for a ND-500 as a back-end to the MCR consoles, notes presented at the SPS Controls meeting 09.02.84.

10) F.C. Iselin, E. Keil, J. Niederer, Mad development plans and estimates, LEP/TH/EK/ek/84-56.

11) IBM PC-AT, Reference Manuals.

12) ISO/DIS 7498, Information processing system, Open Systems Interconnection, Basic Reference Manual.

13) ISO/IS 8802 series.

14) Transmission Control Protocol/Internet Protocol, Internet Protocol Transition Work-Book (March 1982), Network Information Center, SRI International, Menlo Park, CA 94025.