EXPERIENCE WITH WORKSTATIONS

R. Brun

CERN, Geneva, Switzerland

ABSTRACT

Workstations have been introduced at CERN in 1982. After 5 years of experience, it is now possible to draw some conclusions on their impact and weaknesses. The requirements for new workstations and their integration in more conventional computing systems is analyzed.

Chapter 1

Introduction

Personal Workstations are now becoming a reality in the world of High Energy Physics. Low-end workstations with the capacity of a VAX 11/780 and a high resolution screen are now cheaper than a graphics terminal 3 years ago. Every month sees new announcements of still more powerful devices at lower and lower prices. But conventional time-sharing systems and mainframes are still largely dominating the world of computing in HEP. Many laboratories have massively invested money into these systems in the past years. Should we see workstations as an alternative ?

I have been involved in the past 5 years in the evaluation of workstations. My experience is limitated to the kind of work in which I am actively involved, development of large systems and interactive graphics. Moreover I have worked mainly with low-end workstations of the same manufacturer (Apollo) and my conclusions do not neccessarily apply to all kinds of workstations. In the first part of my lectures, I will describe the reasons why, as a privileged user, I became quickly so enthusiastic. In the second part, maybe moderating somehow my enthusiasm, I shall try to extrapolate my experience to the case of large physics collaborations, analysing potential problems and proposing some solutions.

Chapter 2

Experience of a privileged user

2.1 Getting started

At the end of 1981 the PERQ workstation was presented at CERN. Like all my colleagues attending the technical presentation, I was very impressed by the high quality of the screen and nice demonstration programs. A bee was flying on the screen when the system was idle and a finger moving on the edge of the screen was showing the line number in a file while editing. This was really new at the time. The device was left at CERN for tests and guinea-pigs were welcome. The famous game "Pacman" became a pole of attraction and was indeed a nice tool to understand the concept and performance of the so-called **raster-operations** (see I Willer's lectures).

I was, at this time, deeply involved in the reconstruction and simulation programs of the NA4 experiment. I was very lucky to have in my office a Tektronix 4006 giving me full satisfaction. The 4006 was not only used for event-scanning and program development for the NA4 analysis chain, but also for the HTV[1] package, an interactive version of HBOOK and HPLOT.

During the spring of 1982, together with Ian Willers, we tried to implement the HTV system on the PERQ but we failed. The PERQ was equipped with a small and slow 20 Mbytes Winchester disk and only 512 K memory. The system was connected to the central machines via a serial line at 4800 bauds. We had to split the HBOOK file (17000 lines) into 17 pieces and many changes were necessary in the code. The PERQ compiler was very restrictive. As the results were not coming quickly, the implementation of HTV on the PERQ became a very low priority.

In May 1982, representatives of a small company (Apollo) came to CERN to present their first products. I did not attend the meeting, the PERQ experience was enough. A few days later I was strongly encouraged to spend one week in Boston (Apollo headquarters) to implement HTV. I had been working very hard in the previous months, so a one week interupt was welcome. As I had never been in the States before, I accepted the proposition.

The week was very successful partly because I received good help from two competents persons in the company. On the first day, I could implement HBOOK, ZBOOK, ZCEDEX and a subset of KERNLIB (about 40000 lines of code). The second day I was able to run HPLOT and HTV with a very primitive emulation of the PLOT10/TCS package. I spent the rest of the week in improving the graphics interface, trying various implementations and learning more about the system.

When I came back at CERN, I was in the situation of somebody who had tasted some drug and has to try again. I had to wait 3 more monthes and then the DD division decided to purchase two Apollo workstations.

2.2 Hardware evolution

During these five years I had the opportunity to work with several types of workstations.

- The PERQ, as already mentioned;

- The Apollo DN420 in 1982. This machine was based on the MC68000. It had a 1 Mbyte main memory and a 60 Mbytes Winchester disk. Another DN420 has been running diskless attached to the DN420 for a few months. A 1600 bpi tape unit was connected to the Multibus unit. The communication with the central mainframes was achieved via a serial line at 9600 bauds. The CPU was equivalent to 0.25 MIPS.[1] The bitmap display had a resolution of 1024 by 800 pixels. The DN420 has been replaced in 1985 by more modern equipment.

- The Apollo DN460 in 1985. Based on bit-slice technology, the CPU is equivalent to 0.9 MIPS. The display is the same as the DN420. The initial configuration had only 1 Mbyte main memory, expanded later to 2 Mbytes, plus a 160 Mbytes Winchester disk and a Multibus interface.

---

[1] 1 MIPS (Million Instructions Per Second) is equivalent in our benchmarks to a VAX 11/780 with a floating point accelerator.

- The IBM PC/RT (RISC technology) in 1986 with 2 Mbytes main memory and a 40 Mbytes Winchester. The display unit was a low resolution terminal. The experience with that machine has been very short, as I have not been able to edit even the shortest of all my files.

- The Apollo DN3000 in 1986. Based on the MC68020 running at 12 MHz plus the floating point coprocessor 68881. This is the workstation which is currently on my desk. It is equiped with 4 Mbytes of main memory and a 75 Mbytes Winchester disk. The CPU is equivalent to 1 MIPS. The display unit has a resolution of 1280 by 1024 pixels. The DN3000 is the low-end workstation offering today the best ***HEP Fortran style*** environment. This is the workstation which is currently on my desk. My comments in the next sections are essentialy based on this workstation.

- The VAX station 2000 in 1987, based on the micro-VAX chip, running diskless and with 5 Mbytes main memory (ALEPH configuration). This machine has a capacity close to 1 MIPS. The display unit has 1000 by 800 pixels.

About 60 Apollo workstations and 30 VAX stations 2000 (or GPX) were installed o.ı the CERN site in September 1987. The number of stations is increasing rapidly.

2.3 Description of Workstation facilities

2.3.1 A multi-task operating system

In contrast with the PCs, workstations have a multi-task operating system. The VAX stations run VMS. This is a very important advantage for users who already have VAXes. The large majority are running UNIX or UNIX-like operating systems. Even Digital seems to put more and more effort into ULTRIX.

On the Apollos, up to 56 processes can be run concurrently. Each process may have an address space up to 16 Mbytes and even more on the high-end stations. I can run on my workstation programs which do not fit in the central mainframes. There are typically 3 or 4 processes running in the background (servers) and it is very convenient to run 3 to 5 additional user processes. For example, while compiling a file in one process, one can edit another file (under the control of the display-manager process) and run an interactive program (eg. PAW [3]) in a third process. Each process runs in a different window.

2.3.2 The Display Manager

The Display-Manager (DM) is a high priority process which takes care of all the Input/Output (I/O) operations in any of the existing windows on the screen and also growing, shrinking and popping these windows. It may have its own I/O windows (eg. at the bottom of the screen on the Apollos) or it may provide a user interface based on pop-up menus

2.3.3 Graphics

Workstation manufacturers provide a **native** graphics system. The first programs run on the Apollo were based on the GPR package which is simple to use and efficient. The Minimal GKS (MGKS) package[2] has been developed on the Apollo and then made available on other computers. It has been used until recently by most of the graphics application.

The GTS-GRAL package (GKS/2D and GKS/3D) are now available on Apollo and many other workstations. Applications like PAW are now based on GTS-GRAL. This version of GKS is considerably better than MGKS, both in functionality and speed. One of the main problems remains with the size of the metafiles. Metafiles can easily reach several megabytes and they must be transported to the mainframes to get access to the plotting devices.

The **native** graphics packages are very attractive for applications running only on the workstation. They provide facilities specific to bitmap displays and they are at least 10 times[2] faster than GKS.

2.3.4 The SHELL

I have always used the AEGIS shell (UNIX like). The command names are somehow clumsy, at least not as user-friendly as VAX/VMS. For example to copy file1 to file2 one has to type **CPF file1 file2.** Just typing **CPF** does not issue a prompt but an error message. This is the only negative point when one compares AEGIS with VMS.

2.3.5 The file base

The UNIX hierarchical file system is extremely powerful and simple. It is easy to structures files within applications. The VM/CMS minidisks look old-fashioned and hard to use in comparison. With the ACL (Access Control List) facility, one can specify the Read, Write, Execute permission at the file or directory level. There is nevertheless one drawback in the system: it is not easy to specify disk quotas per user. This is not an important restriction as, in general, a disk is allocated to one single user.

2.3.6 The Editor

Editors such as the Apollo screen editor are simple to use. One can edit several files at the same time, each in a different window. One window may display up to 80 lines of code[3] (See Figure 1). In this way one has a global view of an algorithm and printed listings are less frequent than with conventional editors on alphanumeric terminals.

The editor is not particularly fast. For example, to edit the **GEANT.CAR** [7] file, it takes about 10 seconds to point to the last line and about the same time to save a new version. Note that the same test on the PERQ would have taken at least 10 minutes.

A very interesting feature of the editor is the ability to search in the transcript pad of each process and to copy parts of this pad into a normal edit-file.

---

[2] benchmarks run with GPR or GMR on Apollo and UIS-DC on VAX stations

[3] The number of lines depend on the selected character font for the window

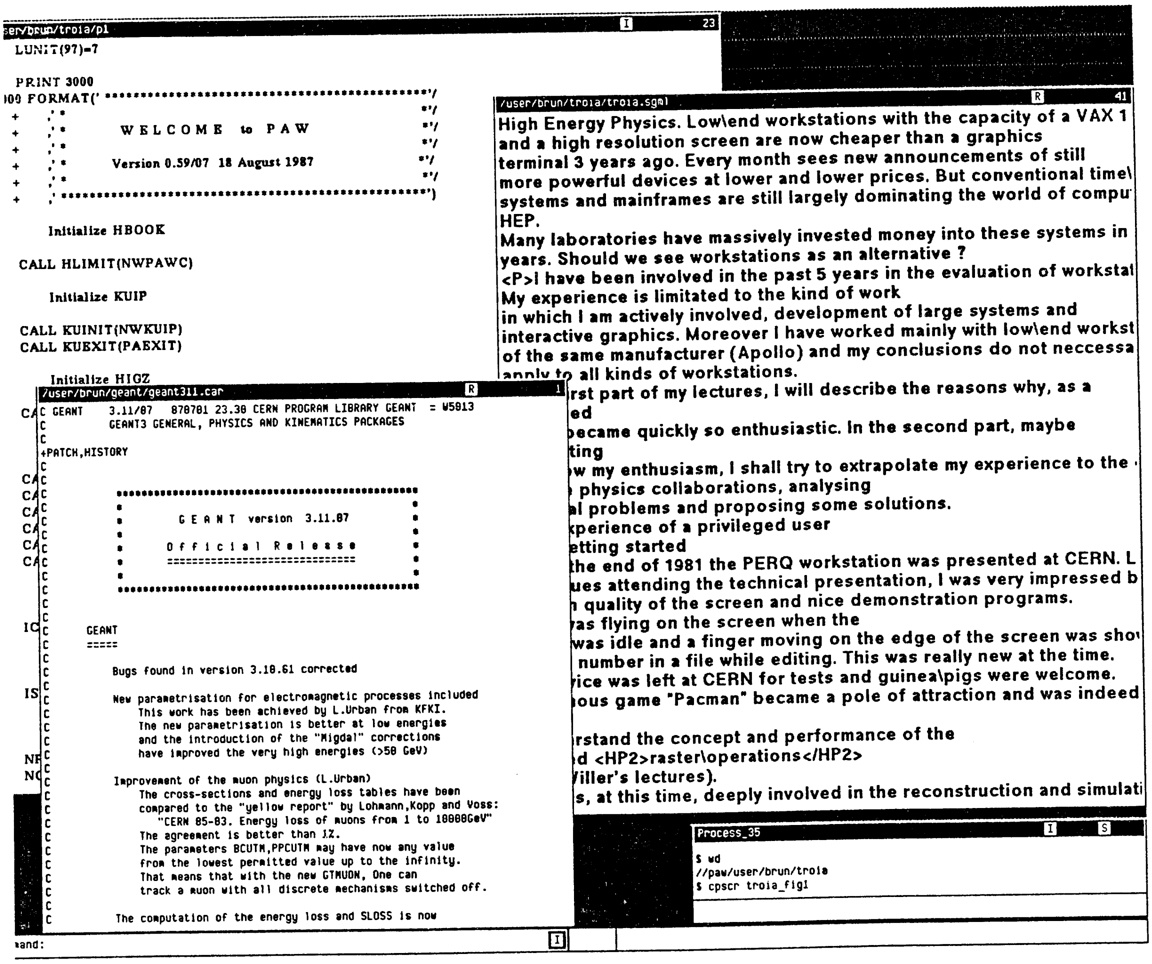

Figure 1: The Editor. Examples of several edit-files on the same screen

2.3.7 Compilers

Good Pascal, C and FORTRAN compilers are available. The Apollo FORTRAN compiler has many extensions like the VMS compiler. The speed of the compiler is a function of the selected options, Debug, Optimization,etc. Eight minutes are required for GEANT (1 minute to run YPATCHY). As it is possible to do something else (edit for example) while a file is compiled, the speed is in general not a problem.

I recompile frequently large pieces of code while I am thinking, or debugging a program, or during a coffee break. I cannot do that on a time-sharing system if I want to be fair to the other users.

The Apollo FORTRAN compiler can be considered as very good compared to the other Unix systems.

2.3.8 Linkage editors, Librarians

Apollo has a good loader and linkage-editor (the so-called BINDER). It is possible to create pre-linked modules and **Installed-Libraries**. We succeeded in making the CERN Library modules KERNLIB, PACKLIB, GEANTLIB and GKS **Installed-Libraries**.

Once a library is installed, it can be mapped to the virtual address space of a process. This is done once per session and takes only a few seconds. The **Binding** operation is then extremely fast and the image module generated small. The table below shows the linking time for PAW and GEANT on Apollo and for comparison on VAX station 2000, VAX 8800 and IBM (VM/CMS).

	Apollo	VAX 2000	VAX 8800	IBM
PAW	5s	164s	62s	48s
GEANT (GEXAM1)	3s	182s	67s	55s

- Time to link PAW and GEANT(example 1) on various systems -

Note that on the VAX 8800 and IBM, the link time may vary (in the bad direction) by large factors. On VAXes the sharable library for GKS was used to link PAW.

I have found this fast linking to be a very important feature in the program development phase, where the cycle EDITOR, COMPILER, LINKER must be repeated frequently.

2.3.9 The Symbolic debugger

The Apollo symbolic debugger can be compared to the VMS debugger. There are two important features missing :

- SET WATCH variable;
- CALL routine (very useful in interactive applications).

The macro facilities to set conditional breakpoints are very powerful, and the possibility to see large pieces of code with an **arrow** pointing to the statement being executed is wonderful.

2.3.10 Communications

The Apollos at CERN are connected to ETHERNET and CERNET. Transfer of files to/from VAX and IBM is better done using the FTP (TCP/IP) program. Reasonable transfer rates can be obtained: 15 to 22 Kbytes/s for text files and up to 50 Kbytes/s for binary files.

A big problem remains with FTP. The user interface is different on Apollo, VAX and IBM. The transfer of binary files is not always possible and results may be unpredictable. This situation is not acceptable in the medium and long term.

2.3.11 Terminal emulation

More and more sophisticated tools for remote-login[4] on central or other machines are coming. It is sometimes difficult for a beginner to choose the right one.

- **EMT**: Terminal-Emulator enables remote login via a serial line. With EMT many types of connections are possible, mainly via the INDEX system. During the development of GEANT, I have used this sytem to communicate with a VAX at Carleton University in Ottawa. I could transfer text files, just by listing them on the screen, at a time when no DECNET or BITNET facilities were available across the Atlantic.

- **EMTG**: This is a modification of EMT used as a Tektronix 4014 emulator. This has proven to be very useful when debugging the GEANT graphics system on the VAX. In this mode of operation, two windows are open: one contains the normal alphanumeric output, and the second one emulates a subset of the 4014. A 4014 emulator is also provided by Apollo, but graphics and alphanumeric output get mixed in the same window.

- **TELNET**: Remote-login facility running with TCP/IP. This is now the best tool to communicate with VAXes at CERN or any other system running TCP/IP.

- **TELNETG**: A modification of TELNET to run with graphics applications. It is similar to EMTG[5] and slightly more powerful.

- **EM3278**: This is a terminal emulator using SNA for communications with the IBM systems. Unlike the other systems not mentioned here, the mapping of the function keys is correct, but the system remains sometimes blocked for unknown reasons. EM3278 does not work for graphics applications.

All these facilities, not yet mature, are nevertheless extremely useful. Most of the time, when working on my Apollo, I am connected via TELNET to VXCRNA/VXCRNB and via EM3278 to the IBM (See Figure 2).

---

4 Without remote-login tools a second terminal would be required

5 EMTG and TELNETG have been developed in collaboration with Ben Segal

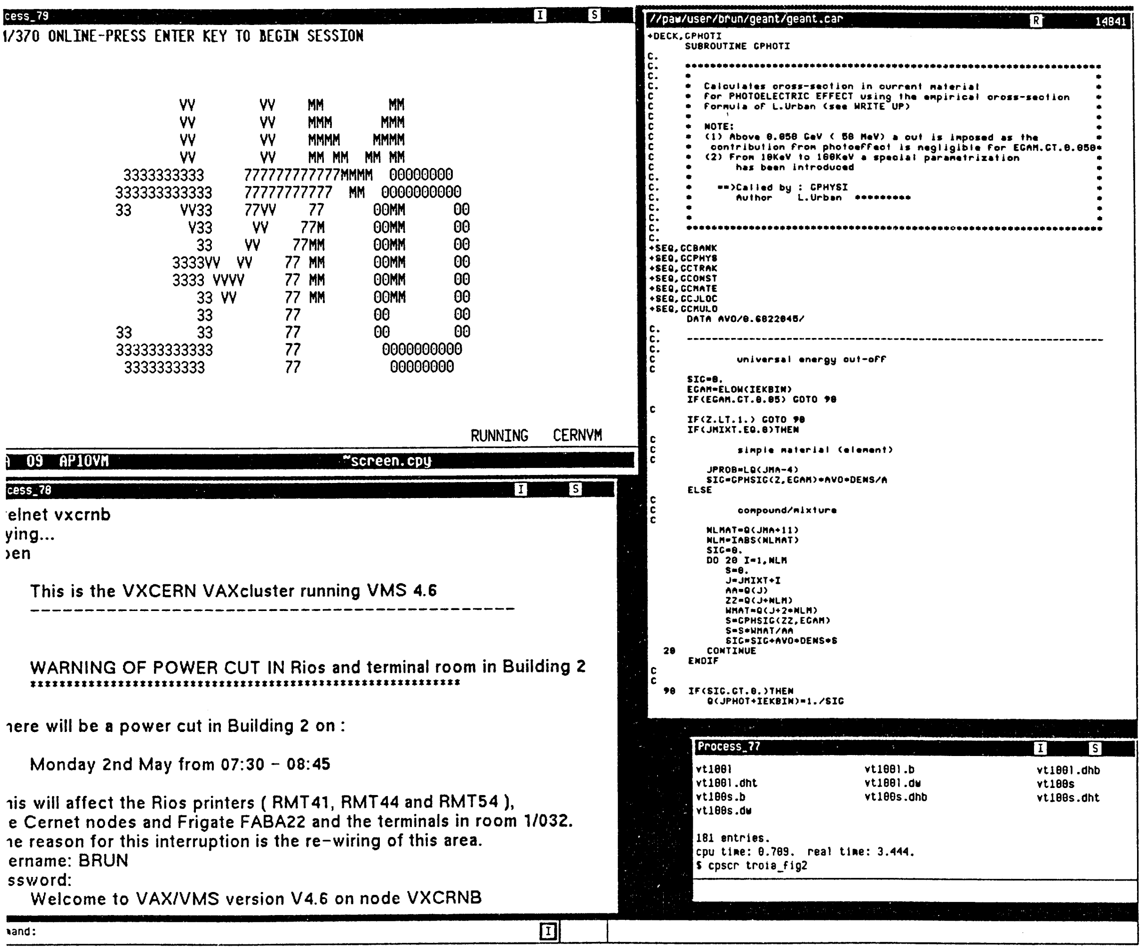

Figure 2: Remote login. Using TELNET and EM3278 to connect to VAX and IBM respectively

2.4 Reliability, Backup, Maintenance, Miscellaneous

During these five years, I had to report three hardware problems:

- The power supply for the DN420 had to be changed (machine unavailable for two days);
- the monitor of the DN460 has been replaced (machine unavailable one day);
- the Winchester disk of the DN3000 became too noisy and was changed at my request (machine unavailable for three days).

I have never lost files (I have anyway frequent backups on the central machines). However one of my colleagues had a serious disk crash and lost his files two years ago. We have encountered many problems this year with a 500 Mbytes disk attached to a file server DSP90. Users are responsible for the backups of their disks, either on 60 Mbytes cartridge tapes or via the network. To my knowledge, no problems with the CPUs or the memory boards have been reported.

With the exception of the DN3000, all the other stations cannot be installed in a normal office. They are too noisy and they can replace the central-heating system. This remark in particular applies to the high-end stations.

2.5 Conclusions of a privileged user

I had the opportunity to test the first generation of workstations. I knew, when I started, that many facilities were still missing, in particular in the area of communications. Being the first guinea-pig, I had to solve many problems which may look trivial to-day. I am convinced that it was a good investment. I do not see how large packages such as GEANT or PAW could have been developed in the same time-scale without workstations.

Chapter 3

Trends

In the first parts of this lecture, I have concluded very positively on the usage of workstations. But I had three workstations in five years, the first two being very expensive. This model cannot be applied to the case of several hundred potential users. Economical parameters such as the hardware life-time, must also be part of the analysis. In the following sections, I discuss a few points which are becoming more and more important.

3.1 The operating system

As already mentioned, UNIX will become more widespread on workstations but also on mainframes. The various existing SHELLs will, I hope, converge to something more user-friendly. Modern user interfaces based on the portable **X-Windows** standard will be coming shortly. It is therefore important to invest into UNIX systems now.

3.2 The Program Development environment

There are no doubts that workstations will continue to provide a better environment than time-sharing systems. Computer Aided Software Engineering (**CASE**) is already offering tools for **Visual Programming** systems based for example on the SADT methodology. Such systems, still in the infancy, will play a major role in three to five years from now.

Symbolic Debuggers and Performance Coverage Analyzers will continue to improve.

3.3 Text processing

A profusion of text-processing systems exist on workstations. They are either very sophisticated and non-portable (INTERLEAF) or the quality of the output is not suitable for publication.

The **SGML** system[8] is now introduced at CERN on the VM/CMS system. It enables text and graphics (GKS) to be mixed and it contains a very powerful and "simple to use" formula-processor **EQN**. The high resolution bitmap displays are perfect for fast preparation of documents, the laser-printer being used only for final printing. It seems that the implementation of SGML on a workstation will require several man years of effort. A new version of the INTERLEAF system coming soon is interfaced to SGML. This solution (or equivalent) would present two major advantages:

- possibility to use a WYSIWYG (What You See Is What You Get) system on the workstation;
- possibility to send the results across networks for processing by the non-interactive versions of SGML and vice-versa.

The **TEX** and **LATEX** systems are available on all workstations. A very good and fast LATEX viewer is available on the Apollo. LATEX, being a ***de facto*** standard in most of the HEP laboratories, is certainly the best interim solution until cheap and efficient versions of SGML become available.

3.4 Workstations with Super-minis and Mainframes

It is likely that the life of Super-minis is coming to an end. Even when running the very popular VMS (and may be because of that) they are always saturated. Multiplying their capacity by a factor 10 will not change the situation. A user of GEANT will run ten events interactively instead of one.

Mainframes are still attractive. Centralized operations and I/O services are of good quality. They will provide in the next years the bulk of the CPU capacity for "production jobs".

One must therefore work on the hypothesis that workstations and mainframes will cohabitate. The fraction of a mainframe devoted to time-sharing should be kept to a minimum to ensure a better performance for number-crunching. The speed and the reliability of communications between the two worlds must be improved.

Tools, such as VAXACCESS (Apollo − VAX) or INTERLINK (VAX − IBM) are steps in the right direction. The Network File System(NFS)[6] or the Network Computing System(NCS)[7] will make possible the access to mainframes without learning the Control Langage for these machines.

Getting access to central data bases from programs running on a workstation will become essential. How can we transport data records in a transparent way between machines with incompatible floating point formats? Systems such as ZEBRA[4] are already used to transport data in a machine independent way. They will have to be coupled with facilities like FTP if one wants :

- to avoid having two copies of the same data set;
- to transport only a subset of a direct-access file;
- to provide a data-structure server.

---

[6] from SUN Microsystems

[7] from Apollo

In a similar way, graphics tools must be developed to enable programs running on mainframes to communicate with the workstation. Programs such as TELNETG must be considerably developed. They could be based on the emerging X-WINDOWS standard and on the user interface package KUIP[5] developed in the context of PAW. The HIGZ[6] graphics package which offers high-level graphics primitives could be interfaced to X-WINDOWS to provide more efficient communications.

3.5 Workstations for number-crunching ?

Workstations of 4 MIPS are now on the market. 10 to 50 MIPS Stations are expected in less than one year; they should be available at the cost of today's low-end station in a time-scale of three years.

Meanwhile, attached processors or compute servers (eg. Alliant FX8) are available. Arrays of transputers or other kinds of powerful parallel machines (up to 40 MIPS) are being integrated into networks of workstations.

It is possible and simple to use workstations in parallel. A LEP collaboration could for example run the simulation program GEANT on 10 workstations (one event per station). A capacity equivalent to central mainframes can be obtained at night time when workstations are not used for interactive applications.

Chapter 4

Conclusions

Workstations are now a reality. They will progressively replace in the next two or three years the terminal which is to-day on each Physicist's desk. They are invaluable tools for individuals or small teams developing large systems. For the new big collaborations, I see two possibilities:

- **Low-level integration**. Collaborations continue to use workstations as we do now. This solution does not require too much development. Communication facilities with the mainframes will be gradually improved. But these workstations may become very quickly obsolete (two years life-time) when new models are announced. They will become expensive terminal emulators connected to bigger machines.

- **High-level integration**. A substantial effort has to be undertaken in the area of communications, Remote-Procedure-Calls facilities, Graphics and User-Interfaces, access to mass-storage. The ratio of (What is executed on the workstation)/(What is executed on the Mainframe) can change with applications and within the same application as a function of the upgrades taking place on both sides. The life-time of workstations should increase (four to eight years).

We are just entering a new and fascinating era of **distributed computing on heterogeneous networks.** Workstations are going to play a major role in this scheme. This cannot be without consequences on the design of the HEP software.

References

1. R.Brun.

2. *MGKS Reference Manual,* CERN Program Library J550 *HTV Users Guide,* CERN DD/EE/80 – 5

3. R.Brun, O.Couet, N.Cremel, P.Zanarini *PAW Users Guide,* CERN Program Library Q121

4. R.Brun, M.Goossens, J.Zoll *ZEBRA Users Guide,* CERN DD/EE/85 – 6

5. R.Brun, P.Zanarini *KUIP Users Guide,* CERN Program Library I202

6. R.Bock, R.Brun, O.Couet, R.Nierhaus, N.Somon, C.Vandoni, P.Zanarini *HIGZ Users Guide,* CERN Program Library Q120

7. R.Brun, F.Bruyant, A.McPherson *GEANT Users Guide,* CERN DD/EE 84 – 1

8. A.Berglund *CERN SGML User's Guide,* CERN/DD 1987