JET
A Large Physics Experiment

H. van der Beken

Joint European Torus, Abingdon, UK

Associate Member of EPS

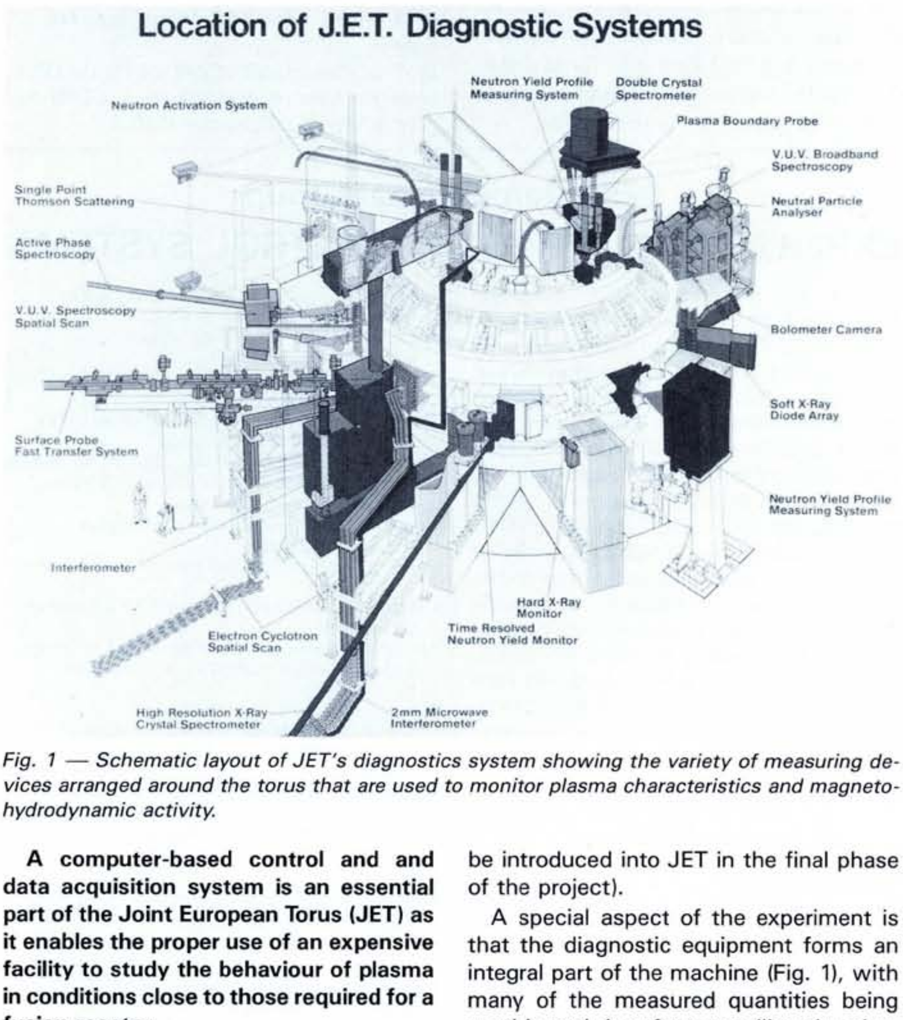

Fig. 1 — Schematic layout of JET's diagnostics system showing the variety of measuring devices arranged around the torus that are used to monitor plasma characteristics and magnetohydrodynamic activity.

A computer-based control and and data acquisition system is an essential part of the Joint European Torus (JET) as it enables the proper use of an expensive facility to study the behaviour of plasma in conditions close to those required for a fusion reactor.

The JET experiment is based on a large tokamak with a major radius of 2.96 m, minor radii in the horizontal and vertical planes of 1.25 and 2.1 m respectively, and a toroidal field of 3.45 T at the centre of the plasma maintained inside the vacuum chamber. Plasma discharges are generated for some 20 s every 20 to 30 minutes and plasma currents of up to 7 MA have been achieved. A fusion power in charged particles of 100 kW has been produced with deuterium plasma containing He^3 as the minor constituent: the fusion product has reached $7-9 \times 10^{20}$ m^{-3} keVs in deuterium-deuterium plasma. This is equivalent to about 12 MW of fusion power [1] for a deuterium-tritium plasma (tritium will be introduced into JET in the final phase of the project).

A special aspect of the experiment is that the diagnostic equipment forms an integral part of the machine (Fig. 1), with many of the measured quantities being used in real-time for controlling the plasma. For instance, measurements of the plasma density made using various interferometric and Bremsstrahlung devices are fed to a front-end microprocessor which generates a validated signal for the density. This signal is used by another microprocessor to control systems that introduce gases to establish the required density. The basic experimental quantities from the devices are also used at a later stage for off-line data analysis.

Process control and data acquisition are thus much more intertwined than in other large physics experiments (*e.g.* accelerators) and this led to the implementation of a fully integrated control and data acquisition system called CODAS [2, 3].

CODAS

The detailed design of CODAS started four years before JET began operating in 1983 and an experiment could only be run right from day one by using the system [4]. Although the structure was classical in concept, it took advantage of the best developments existing at the time. Moreover, CODAS had to be delivered on schedule and on budget while avoiding significant technological risks. Although it has evolved over the years, deviations from the original design are insignificant [5]. Accurate records do not exist, but it is estimated that more than 100 man-years of effort have been invested up to now in developing the software (consisting of about two million lines of FORTRAN code and some 100 000 lines of code in lower level languages) with around 50 man-years for the hardware.

The overall system was divided into 20 subsystems, each covering a "logical" part of the machine, in order to accommodate the construction, operation and development of JET. Sub-systems include those for the vacuum, the poloidal field (coils, power supplies and instrumentation), the RF power used to heat the plasma (32 MW of power in the range 25–55 MHz), and for the neutral beam (some 20 MW of neutral particles with energies in the range of 80–140 keV are introduced into the region of the vacuum chamber where the plasma is produced). Each is supported by its own computer and is interfaced to items of equipment through a CAMAC serial communications highway (see page 40) with fibre optic links.

The time-critical real-time applications run on front-end microprocessors [6] and the man-machine interface is based on CAMAC-driven operator consoles. Each subsystem can be included or excluded from centralized operation depending on the experimental programme — a modular aspect of CODAS that allowed JET to start operation with only "basic" subsystems while others were under construction or being commissioned.

CODAS interfaces to some 50 000 input and output channels, both analogue and digital, and has remained essentially unchanged since the machine started operation in June 1983. Fig. 2 gives a simplified representation of the overall structure of the system. At present, more than 20 Mbyte of data are collected during each pulse from the various subsystems and concatenated in a single JET Pulse File (JPF). The JPF's are archived for further analysis on the JET IBM mainframe computer where we have accumulated more than 200 Gbyte of data [7].

Data Acquisition and Control

A computer-assisted control and data acquisition system is not an expensive gadget, but a necessary tool for all large scientific installations as it allows an expensive facility to be exploited correctly in a research environment, where it is inconceivable to present operators with a multitude of dials, knobs and switches.

Henri van der Beken is Head of the CODAS Division at JET, Abingdon. He obtained a master's degree in electronic and nuclear engineering in Grenoble before working at CERN and at LAMPF, Los Alamos, USA.

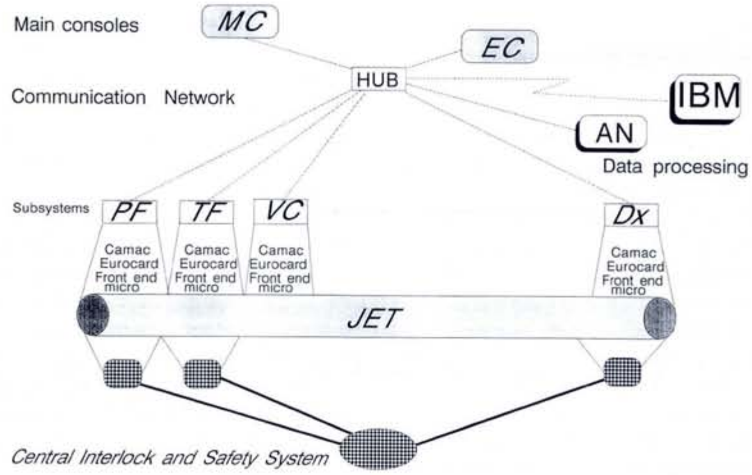

Fig. 2 — Structure of JET's fully integrated computer control and data acquisition system (CODAS). Each of the 20 subsystems (PF – poloidal field, TF – toroidal field, etc.) is supported by its own computer interfaced via a CAMAC serial highway to front-end microprocessors, with Eurocard packaging, that receive diagnostic information and runs application programs in real-time to control the machine. Man-machine interfaces are provided at the main consoles (MC, EC) connected through a communications hub (a mainframe computer) to the subsystems. The entire system has 50000 input and output channels and generates 20 Mbyte of data during plasma discharges lasting 20 s.

JET's wide spectrum of diagnostic techniques moreover provides fundamental information on plasma current (and current profiles), position, shape, density and temperature profiles (ions and electrons), impurities and fusion reactions [2]. The spectrum also covers measurements of magneto-hydrodynamic phenomena which are fundamental to the behaviour of the plasma and its control. Improved knowledge of the underlying physics is expanding the application of front-end, real-time digital processing and feedback techniques to this diagnostic information. The successful implementation of these methods requires a strong coupling between diagnostics and controls as well as flexible and powerful tools that programmable facilities can offer.

The flexibility inherent in computer software thus permits the adaption of operating "tools" to the progress made in understanding complex physical phenomena. Applications of these new methods are expanding with operational experience [6], and they now extend from applications preventing the development of abnormal plasma behaviour to feedback systems allowing operating conditions unattainable in pre-programmed modes (*e.g.* adjustment of the plasma position to obtain a preset coupling resistance of the RF antennae which introduce power into the plasma).

Collaboration

When comparing CODAS with other major computer-assisted control systems one finds large parts which are similar in both even though little was "imported". This shows that we talked to our colleagues, were aware of each others' developments, helped each other without restrictions, while all the time continuing to work hard on "our" system. Most of us ended up with very similar "sets of tools" in our separate laboratories but were unable to borrow much software. Consequently, if the development of CODAS had been split into 20 projects, one for each subsystem, the total effort would have been significantly larger. Collaboration gave savings in time, effort and money so one wonders why it did not develop at a wider level.

Computer technology has evolved at an accelerating pace over the last 20 years so each project often introduced new computers to take advantage of the best hardware that was available. Different instruction sets and operating systems and, for a long time, the need to use proprietary low-level languages made the exchange of software close to impossible. Another barrier, even with compatible hardware, was the scarcity of application programs that did exactly what we wanted. Instead of trying to understand and modify not too well documented software, it is much more fun to re-invent the wheel: a serious review of the requirements was seldom carried out as these rarely existed in a clean form. Compatibility was also often only in appearances as imprecise and incomplete standards, together with the willingness of users to exploit the enhanced capabilities offered by commercially available applications software, made porting — the transfer of software application programs to other control systems — an art (it is not yet a science in spite of all the seminars on reusability).

Present Situation

Today's picture is less bleak: there are indications that changes are occurring and that computer-assisted process control and data acquisition may become easier to assemble from standard building blocks which can be tuned to the specific needs of each installation. This is because important progress in hardware usually makes a low-level language unnecessary (even "real" programmers no longer write code in hexadecimal). More people also realize that designing a new menu facility (a graphic display to help an operator interact with the experiment) is a waste of time. Attention should be directed instead to ensuring requirements are agreed upon and understood. Computer-assisted software engineering (CASE) tools, although not yet fully mature, are being used more and more.

However, it is also clear that the implementation of standards is governed by powerful market forces. Aside from wishful thinking, we users have not come together to specify requirements, although the achievements of the EPS control systems group (EPCS) constitute the essential first step in a sharing of expertise.

Porting

CODAS is presently being modified to assure its availability, reliability and performance for the years to come by porting to a UNIX operating environment. The move to UNIX (V.4) for the operating system, Ethernet 802.3 for communication, VME, Fortran 77 and C as programming language and workstations for the man-machine interfaces has highlighted the importance of software portability because the effort to implement the move estimated to be as large as 35 man-years.

It is beyond the scope of this paper to justify the above choices: some were dictated by the need to minimize the porting effort and budget; others followed from making CODAS as independent as possible of the fortunes of a single manufacturer. For the sake of compatibility with our old 200 Gbyte of data, and to leave the door open to a mixed system (old and new), we were also obliged to choose the the larger vendors [8]. We are also unable to use CASE tools as we are moving more than two million lines of existing code rather than designing a new system.

The advent of *de facto* standards since 1979 and the availability of certain tools has made some parts of our software not only obsolete but unnecessary. We are therefore using, whenever feasible, public domain software although we shall have to continue to choose between products coming from competing groups (*e.g.* the Free Software Foundation and UNIX [9]) and between differing approaches to high level tools for workstations.

Conclusions

I am not tacitly advocating that a universal standard should apply to all computers and all computer-assisted control systems as I have been in research too long not to realize that progress comes from multiple approaches. Market forces, moreover, will continue to force changes and existing standards will be superseded. Nonetheless, we shall still have the opportunity to minimize the effort expended on design and implementation because useful and well-supported standards are now available.

Strong managerial support of the EPCS's efforts to rationalize some of the development of control systems for physics experiments could reduce the costs of "closed-shop" projects and provide better tools and procedures to all laboratories. But to reap its full benefits, this support will have to be substantial and of a continuing nature.

REFERENCES

[1] Gibson A., Proc. EPS-8 General Conf., Amsterdam, 1990 (to be published).
[2] *Fusion Technology* **11.1** (1987) 1.
[3] JET Joint Undertaking Progress Report (1989).
[4] van der Beken, H., *IEEE Trans.* **NS-34** (1987) 742.
[5] Bombi F. *et al.* — The JET Project: Scientific and Technical Developments 1977-1978, *EUR-6831 EN/EUR-JET-R10* (European Atomic Energy Community) 1980.
[6] Steed C.A. *et al.*, Proc. 12th Symp. on Fusion Engineering, Vol. 1 (1987) p. 673: *IEEE Cat. No. 87CH2507-2.*
[7] van der Beken H. *et al.*, *IEEE Trans.* **NS-36** (1989) 1639.
[8] Cohen D., *Computer* (Oct. 1981).
[9] *GNU Bull.* (The Free Software Foundation, Inc., Cambridge, USA).

Accelerator Control Systems

P.N. Clout

Vista Control Systems, Inc., Los Alamos, USA

Fig. 1 — Control rooms of CERN's accelerators. (a – left) For the PS machine in the mid-1960's where operators were obliged to move between racks of equipment in order to operate and adjust the machine.

(b – right) For the SPS machine in the late 1970's where the controls of the computer-based control system are grouped within reach of one operator around each of the several operator stations with their video screens, alphanumeric displays, oscilloscopes and devices for selecting and adjusting components of the accelerator.

Charged particle accelerators were first connected to a computer in the early 1960's. The integration of computers with these research machines has expanded rapidly ever since, enabling considerable increases in size and performance.

Interest developed from what was probably the first reported use of a computer for data acquisition in nuclear physics by Edwin Norbeck in 1962 [1]. The report, and the rapid reduction in price of small computers, caught the attention of accelerator builders.

On entering an accelerator control room in the mid-1960's (Fig. 1) one would have