Fusion Engineering and Design 43 (1999) 265–273

The evolution of control and data acquisition at JET

J.G. Krom *

JET Joint Undertaking, CODAS Division, Abingdon, OX14 3EA, UK

Abstract

This paper presents the evolution of the JET control and data acquisition system. Some important points in the design of this system are highlighted and some future directions are indicated. The unexpectedly large growth of the amount of data collected is discussed. A few conclusions for the design of control systems for future fusion experiments are presented. © 1999 JET Joint Undertaking. Published by Elsevier Science S.A. All rights reserved.

Keywords: JET control; Data acquisition system; Fusion experiments

1. Introduction

The Joint European Torus (JET) is the largest single project of the coordinated nuclear fusion research programme of the European atomic energy community (Euratom). It was designed in the mid 1970s [1–3] and has successfully operated from 1983 onwards, culminating in the current deuterium–tritium experiments [4].

From the onset JET was designed to use a computer based integrated control and data acquisition system (CODAS). This system has evolved over the years, but many aspects of the original design are still relevant.

This paper will give a global overview of this evolution and then re-visit some specific areas in more detail.

* Fax: +44 1235 464404; e-mail: Jon.Krom@jet.uk

2. The early design

The early design was based on a modular, hierarchically organised, tree-structured control system (Fig. 1). This was implemented, during the JET construction phase (1979–1982), using approximately 18 Norsk-data (NORD [5]) minicomputers and employing CAMAC [6] front-end electronics [7–9].

This structure was selected, in order to have self-contained units, with minimal interaction between them. Or, in modern software engineering terms, to have units with a high internal cohesion, and a low external coupling. In this way, for example, it was possible to operate the vacuum control computer 'VC', while others were switched off, or to use one computer to replace another in case of failure.

This hierarchical structure was, and still is, reflected in the software structure. Three levels of control can be identified:

Level-1 Jet-wide supervisory control,

PII S0920-3796(98)00394-9

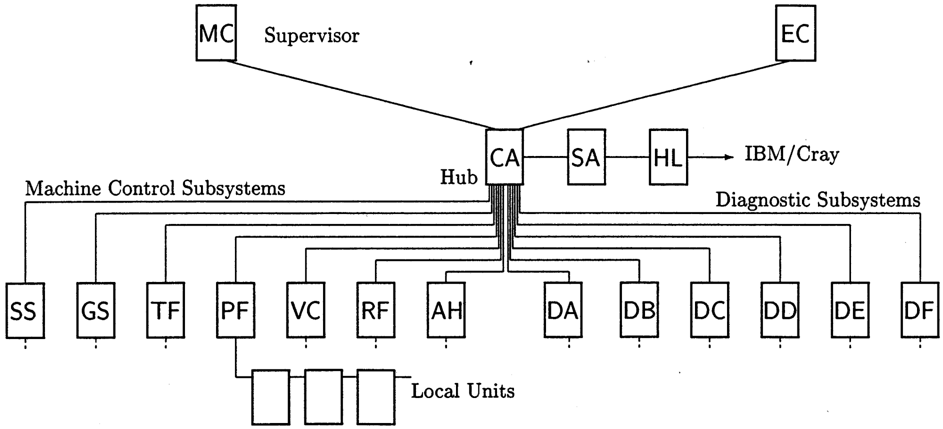

Fig. 1. The structure of CODAS around 1985 [9]. The JET subsystem computers are identified by a two character code.

Level-2 Subsystem supervisory control and
Level-3 Local unit or component.

Level-1 software runs on the 'machine control computer' ('mc'). Level-2 software runs on the subsystem computers, as does most of the level-3 software. In a few cases 'auxiliary crate controller' processors (CAMAC modules based on the TMS99000 micro-processor) are used for real-time level-3 applications in the front-end electronics.

At each level the software is highly modular, enabling the re-use of many software units.

The software in this early phase of the project was mainly written in Fortran. Some time-critical sections, in particular the CAMAC module drivers, were written in assembler and NPL (a nord specific 'computer oriented high level language').

For some tasks that did not have very strict real-time requirements (mainly level-2 tasks), a special control programming language was used. This language was based on the event-driven execution of compound statement blocks. Development of the language started in 1985, in order to provide the flexibility to handle rapidly changing requirements.

The 'man machine interface' (MMI) was implemented, during 1982, using an in-house developed 'point-and-click' mechanism. Each subsystem computer had, via special CAMAC modules, a small number of screens under its control. Each of these screens could, at any one time, show either a graphical representation of some plant status data (a 'MIMIC') or an alarm listing or a real-time trend-curve display.

A tree structured database was used to describe the CAMAC front-end hardware connected to each subsystem computer. A similarly structured database was used to describe the actions to be executed on and the data to be collected from this hardware during a JET pulse [10–12].

The collected data was transferred via a direct high-speed link to the computing centre of the host laboratory at Harwell, some 22 km away. This centre provided permanent storage and high-power analysis facilities. In 1987, these functions were taken over by an IBM mainframe owned and operated by the 'data management group' (DMG) within JET. This system was originally located at Harwell, but was re-located to the JET site in 1990.

The NORD implementation of CODAS was used, and further developed, for nearly a decade, but at the end of the 1980s it became too limited for JET's growing requirements. Ever larger amounts of data had to be shipped; ever more complex control tasks were required. More processing power and more modern software development facilities became available, but were not supported on the NORD architecture. Furthermore, the maintenance of the ageing NORD equipment became more expensive than a simple replacement with more up-to-date and faster hardware.

Similar arguments applied to the network technology and to the CAMAC interface systems. The large investment in CAMAC equipment and especially in its deployment meant that it could not simply be replaced. (It was estimated, in 1986, that about 30000 signals were wired into the CAMAC front-end systems [8]). However, it became more and more attractive to use new technology to interface to any new plant components or diagnostic systems.

3. Move to UNIX

During 1989, studies were made to choose a new hard and software platform. In 1990 it was decided that the optimum choice for the new system was a UNIX based computer system, with X11/motif based user-interfaces and Ethernet networking using the internet protocol (IP). At the same time both VME and PC based front-end electronics were introduced alongside CAMAC.

The new computers selected used the SPARC architecture. One of the main attractions of these computers was that Sun's modern SVR4 based operating system, SOLARIS-2, would be available soon after delivery of these systems. However, the state of that product was such that JET had to start with the older, BSD-based SUNOS-4 (Section 5).

A new, VME based card, and associated UNIX device driver, were developed to interface the CAMAC serial highway loops [6] to the SPARC's SBUS [13].

The original hierarchical tree structure was maintained. It provided a good way of managing the complexities of the control system and a good framework for extensions. This reflects, of course, the good control of cohesion and coupling that is possible in such a structure [14]. It also reflects the initial good design of the system.

During the move, we had to port more than 41000 lines of NPL and assembler to C. This was done using a commercial assembler to C translation tool: XTRAN [15]. This tool was, at our request, adapted to NPL. About one quarter of these lines required additional attention during this translation. The NPL had to be pre-processed to remove some specific hand-coded devices, and the C produced had to be post-processed to adapt it to the new libraries. Most of the latter was done in a semi-automatic way, using a collection of AWK scripts. Almost all this code is still in daily use.

It has been estimated that 500000 lines of Fortran had to be ported. The small number of NORD extensions were mainly translated automatically to Sun's Fortran dialect.

Some very specific NORD operating system features had to be emulated under UNIX. The 'RT-index' [5] or task name registration system, with its associated task scheduling system were originally emulated with a UNIX pseudo-device driver. This has since been re-engineered as a UNIX daemon.

Following the preliminary deuterium–tritium experiments in November 1991, JET went into a major shutdown phase during most of 1992 and 1993. This opportunity was used to move the whole of the CODAS system over to UNIX. Starting in the summer of 1993, monthly CODAS countdown rehearsals were held to check the operational completeness of the control and data acquisition systems in the new environment. In January 1994 JET operations successfully restarted under control of the new CODAS system.

4. Some current issues

Over the last few years, operating in a UNIX environment, several new issues arose. Some of our software has evolved beyond recognition, while other software has had to be introduced to handle new problems.

4.1. Level-1 software

The level-1 control software has evolved into a comprehensive, highly-integrated, high-level interface to the whole of the JET machine control and data acquisition system, without reducing the advantages of the modular setup. We can still operate successfully with any subsystem (or subtree below it) disabled.

The operators of the JET device (session leaders, engineers, physicists, diagnosticians), use this level-1 software extensively to prepare and execute JET shots. It allows 're-dialling' of all parameters of previous shots and can perform consistency checks of a wide range of operating parameters; preventing accidental operation of the JET machine with unsafe parameters.

4.2. Level-3 systems

As indicated earlier, before the move to UNIX, JET used 'auxiliary crate controller' processors (CAMAC modules based on the TMS99000 microprocessors) for a number of real-time applications. After the move, much larger amounts of processing power became available in the front-end systems.

This processing power has been used to implement several hard real-time controllers [16,17]. Communications between the main subsystem computers and these front-end systems is via dedicated Ethernet links. A number of the machine operations groups outside the CODAS division of JET, have implemented their own front-end controllers [18,19].

Blackler [20] details some examples of PC based data acquisition subsystems. Further discussion of these systems falls outside the scope of the current paper.

4.3. System load issues

The main user-interface technology used at JET is based on X11/motif, but quite a few of the programs still present the 'look and feel' of the original programs from the NORD era. This was to maintain continuity in the user-interfaces. One of the main examples of this is the MIMICs package, used to display plant status data in a graphical way.

In the NORD based system there would, at the same moment, only be one or two MIMICs being displayed per subsystem computer; but in the current setup, it is possible to display several MIMICs on each work-station. It is not unusual to have many tens of copies of the same MIMIC being displayed on X-terminals all over the JET site.

This large number of extra user-interfaces had to be supported. The actual user interface programs normally run on special MMI computers. This caused serious load problems on these computers and users displaying MIMICs in offices to follow operations, started to affect the operation of JET.

This was addressed by changes in the hardware— and the software architectures:

- Alongside each subsystem computer, a so-called 'mate' computer was installed. All attempts to read the signals on a subsystem, that originate from any other computer, are channelled via these 'mate' systems. The 'mate' would only read the signal once (or at an appropriately low rate) from the subsystem proper and send a copy of the value to all clients that had registered an interest in this signal.
- A uniform interface mechanism, the 'object monitoring system' (OMS), has been created. It is a client/server mechanism based on the 'uniform resource locator' (URL) scheme that has become familiar from its use in the world-wide-web. E.g. the state of a valve in the vacuum subsystem could be read, using the following URL: 'plant://vc/hv-valve:17/read'. This OMS mechanism is now used as a general, event-based data-collecting layer under many of our display utilities (MIMICs, trend-plotting programs, etc.).

4.4. Networking and security

As indicated earlier, JET uses mainly the internet protocol (IP) over an Ethernet network. Some PC based diagnostics use the so-called 'NetBEUI' protocol for file server access.

At the IP level all computers at JET including the operational control subsystems, the development computers and the workstations in the offices, were originally connected as a single network. Soon after the establishment of this network, JET was given a class B address and the network was split into a number of subnets with secure routers.

The underlying Ethernet network has evolved over the years with the growth of traffic. Initially it was based on small segments, organised around the hierarchically related subsystems and their servers, bridged to a couple of backbones. It has now been updated to use several new technologies: switched Ethernet, fast Ethernet and FDDI. To avoid interference and expensive interfaces, we try to keep the main data transfers (JET pulse data collection, control activities and operating system file server traffic) to separate physical paths. Many of our computers have two or more network interfaces.

In the run-up to the current deuterium–tritium experiments, this style of networking was considered too insecure and a scheme was devised whereby only staff in the JET control could control any plant equipment, and staff outside the control room could only read data from the operational machines.

This was obtained, by implementing an ‘IP gap’, i.e. by completely separating the IP traffic in the operational ‘on-line’ networks from any computer or network not directly involved in operations. A relatively small cluster of computers (the ‘core’ cluster) straddles this gap and can communicate with IP addresses on both the operational and the off-line networks. These computers will not allow any IP traffic between the two worlds.

To provide the off-line world (within JET) with some, controlled read access, a proxy-server was written to operate in this core cluster. This proxy-server handles the client-server traffic caused by off-line OMS clients accessing data on operational servers (Section 4.3). The proxy-server will only transfer ‘read’ actions; ‘write’ or ‘control’ type operations are refused. Staff can thus display MIMICs, trend-curvers etc. in real-time on the workstations in their offices, but are unable to execute control actions which would affect operations.

4.5. Control room organisation

Another area of interest is the evolution of the lay-out of the JET control rooms. In the early designs, two separate control rooms were foreseen, one for machine control and one for experiment control. This was implemented, with the two control room separated by a glass wall. It was intended that any communication between the staff in these two rooms would only be via formalised channels.

It appeared, however, that there was so much need for informal communication, that around 1986 a large sliding door was installed in the glass wall. This door was almost always wide open.

When the move to UNIX required a complete re-design of the control rooms, it was decided to create, essentially, one large control room, in which there are areas with more emphasis on machine control and areas with a more experiment control or diagnostic nature.

It is clear from experience that there are indeed many informal contacts between these areas. If, for example, a neutron diagnostic operator notices a short dip in his signals, it is quite likely that he or she quickly checks with the neutral beam heating operators to see if the beams perhaps had a short trip around that time.

5. The future

The JET CODAS system has continued to evolve, and it seems reasonable to expect that it will do so in the future.

- JET is currently planning to move to SOLARIS-2 (Sun’s version of SVR4 UNIX). This operation system is essential in order to allow us to use more powerful SPARC hardware. It is also anticipated that the current versions of SOLARIS-2 will be more efficient than SUNOS-4. This move will involve at the very least a re-compilation of all software products, and in a few cased, most notable the UNIX device driver for the CAMAC loops, a serious redesign.
- Effort is being channelled into faster data collection schemes, in order to maintain our pulse rate, while collecting ever more data.
- New network technologies are being studied. An ATM network segment is being tested; reflective memory is going to be used in a real-time control network and ‘virtual LAN’ could be used to reduce the interference between the different network usages.

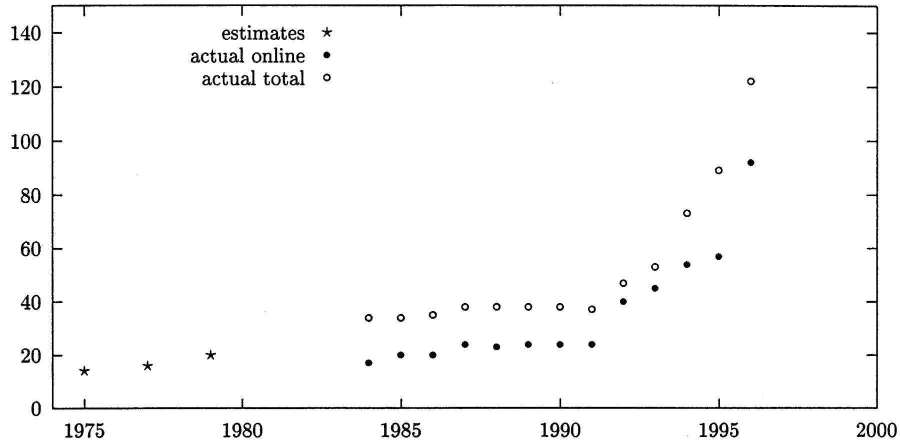

Fig. 2. The number of control computers within CODAS.

- Web-technology is being used more–and–more, especially in the electronic distribution of (status) reports, log-books and operations overviews.
- The usage of VME and PCs is still growing strongly, but CAMAC is also still in use (approximately 2700 modules in about 210 crates). To support this CAMAC investment, a module is now being developed based on Xilinx ‘field programmable logic arrays’ (FPGA) which can, by loading appropriate software, mimic several of the older CAMAC module types.

6. System growth

There are a few topics in the evolution of CODAS that should be looked at in more detail; one such area is the number of computers involved in control and data acquisition (Fig. 2).

In the first designs in the 1970s, it was foreseen that JET would use about 15 computers for this purpose ([1], p.226). One large mainframe would be provided by, and shared with, the host organisation for analysis and data storage purposes. It was expected that these would be all the computers that JET would ever need.

This was a reasonable viewpoint in the early and mid 1970s. However, the developments in diagnostic technology, driven by the developments in the electronics industry, out-paced this plan. When JET became operational it already possessed 34 NORD minicomputers, of which some 20 where classified as ‘on-line’.

At the end of the NORD era, this had grown slightly to some 40 computers, of which 25 were ‘on-line’. By this time a (single) DIGITAL-EQUIPMENT VAX computer had been employed an PCs had made their first appearances; the–big versus little–endian data problem had reared its ugly head!

After the move to UNIX the number of computers started growing rapidly. The generally increased processing power, in front-end electronics, main subsystem control, analysis and desktop workstations, makes classification, and counting, of the JET computer systems far less obvious than it was before. However, if we limit ourselves for the moment to control levels 1 and 2, where the original NORD computers were employed, JET currently[1] uses 122 SPARC computers, of which, at any one moment, 92 are on-line.

This collection of on-line computers, consists of the main subsystems, the MMI computers, the on-line file servers and the ‘mate’ machines (Section 4.3).

The early software designs allowed for growth of the total system size, but not sufficiently for the growth we have actually experienced. A data structure in common use at JET encodes the

[1] This is actually the number of SPARCs at the end of 1996.

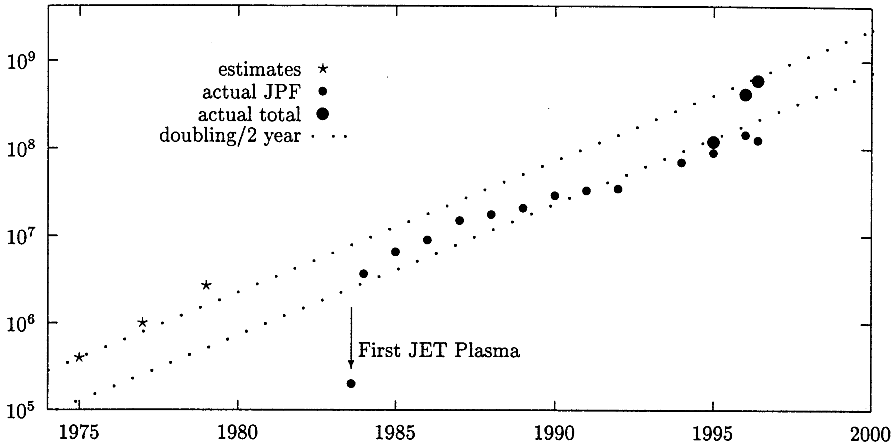

Fig. 3. The amount of data collected per JET shot (bytes).

subsystem in five bits, allowing a maximum of 31 subsystems. This was clearly more than enough at the design phase, based on the agreed specification; it is probably the most serious limitation in the current JET software structure.

7. Data growth

Another topic worth discussing is the amount of data acquired by the JET CODAS system (Fig. 3).

In the original 1975 design document [1], it was foreseen that JET would collect between 10^5 and 10^6 bytes per JET shot. This is the first estimate marked with a star in Fig. 3. The very first few shots collected well below these numbers, but once normal operations were established 1984, JET routinely collected four times as much as these original estimates.

The amount of data collected per JET shot has continually grown. This amount approximately doubled every 2 years. In June 1997, several JET shots collected $> 6 \cdot 10^8$ bytes.

It is interesting to note that the early estimates are close to this growth rate line, suggesting that these predictions were quite accurate, given the state of the technology at that time. There is, however, no indication that this growth rate was expected.

This growth clearly poses some problems for the data handling. It means that the data collection time following each pulse grows in similar fashion, reducing the number of shots possible during an operational day. Effort is continually being made to improve the JET data collection speed: faster network technologies, specific networks for data collection only, collecting more and more data in parallel, separate collection of urgent an non-urgent pulse data, improved coding of critical software, etc.

The last three samples in Fig. 3 (1995, 1996 and June 1997) indicate the effect of separate collection of urgent and non-urgent data. The urgent data is collected in the classical 'JET pulse file' (JPF), while non-urgent data is now collected in the 'late pulse file' (LPF). The JPF data is collected within the first few minutes after a pulse, allowing a quick first analysis. The LPF data on the other hand might be stored in the front-end systems for up to a day, before it is collected and merged into the main pulse data database. This LPF collection and processing happens during relatively quiet periods, normally overnight.

The effect of all this effort is that as far as CODAS is concerned, JET can still operate at a rate of one shot per 15 min.

This growth is handled, in the JET data structures, in a better way than the growth of the number of subsystems. The main structural limitation is in the internal format of the JPF and

LPF files. Although there are some limits in these file structures (like 32 bit page numbers) that are not really open-ended, it should be possible to handle such a growth rate until 2025.

8. Name-space problems

Although the tree-structured organisation of JET CODAS proved beneficial, a problem related to this structure has affected JET. There is, in such a tree-structure, a risk related to the name-space re-use.

In the JET structure, the subsystem name ('VC', 'DA', etc.) was made part of the name of all signals collected on that subsystem. Once a signal was known to be in a particular name-space, moving it to another (i.e. moving it to another subsystem computer) proved to be traumatic.

For example, once the diagnostic known as KC1 was associated with subsystem computer DA, its signals would be stored under the name 'DA/KC1-xyz'. For load distribution reasons, it might become attractive to move this diagnostic to another subsystem, such as DB, but that would imply that the signals would be renamed to 'DB/KC1-xyz'. Such a change would mean that the analysis software had to know that the signals for this diagnostic had one name before some date and another name after that date.

This was mainly a problem with the diagnostic subsystems, where one or more unrelated diagnostics had to share one subsystem computer. (A situation with low internal cohesion but high internal coupling). The machine control subsystems had much higher internal cohesion; the question of moving a component, for example from 'VC' to 'PF', hardly ever arose.

It would appear to be the case, that for the diagnostics at least, the subsystem name should not have been part of the signal name. The functional tree-structure and the implementation tree-structure should have been de-coupled.

9. Conclusions

From the experiences with the JET control and data acquisition system, the following conclusions can be extracted:

- A hierarchical, tree structured organisation, both in hardware and in software, is beneficial. Such a structure can withstand a major re-engineering of the underlying implementation. It encourages cohesion and coupling to be kept under control.
- Signal names should not be based directly on the implementation tree-structure, but on the functional structure.
- Experimental installations such as JET, should allow for large, continual growth and should use open-ended data-structures.
- Experimental installations such as JET, should allow for mixed configurations. (Such as mixed little and big endian data sources and software components produced by many different teams).

Acknowledgements

Most of the JET control and data acquisition system as described in this paper was implemented, and is managed, by the CODAS division within JET. Some of the areas touched upon are, however, implemented and managed in cooperation with other divisions. Most of the PC based front-end systems are developed by the data analysis and modelling division (DAMD). The JPF and LPF storage is realised in cooperation with the data management group (DMG). The magnet and power supply division (M&PSD) and other machine operation groups developed some of the VME applications.

References

[1] European Atomic Energy Community, The JET Project: Design Proposal for the Joint European Torus, EUR-5516e EN (EUR-JET-R5) 1976.

[2] European Atomic Energy Community, The JET Project: Scientific and Technical Developments 1976, EUR-5791e EN (EUR-JET-R8) 1977.

[3] European Atomic Energy Community, The JET Project: Scientific and Technical Developments 1977 and 1978, EUR-6831 EN (EUR-JET-R10) 1980.

[4] M. Keilhacker, Latest JET Results in Deuterium and

Deuterium–Tritium, 24th European Physical Society Conference on Controlled Fusion and Plasma Physics, 9–13 June 1997, Berchtesgaden, Germany, 1997.
[5] Norsk Data A.S., ND-100 Reference Manual ND-06.014.2A En, Norsk Data A.S., Oslo, Norway, 1984.
[6] Commission of the European Communities, CAMAC, Updated specifications, EUR 8500 EN, 1983.
[7] E.M. Jones, Status of the JET Control and Data Acquisition System CODAS, Nucl. Instr. Methods Phys. Res. A247 (1986) 58.
[8] H. van der Beken, et al., CODAS, The JET Control and Data Acquisition System, Fusion Technol. 11 (1) (1987) 120–137.
[9] M.L. Browne, CODAS A large scale multi-computer control system for the JET project, JET-P (86) 37, 1986.
[10] R.F. Herzog, et al., Interfacing between Database and Equipment, the JET Equipment Drivers, Proceedings of the 6th European Conference on Electrotechnics, Brighton, UK, 26–28 September, 1984.
[11] H. van der Beken, et al., Computer Control and Data Acquisition at JET, JET-P (87) 19, 1987.
[12] H. van der Beken, et al., 1989, Data Acquisition at JET-Experience and progress, JET-P (89) 27, 1989.
[13] B. Catanzaro, Multiprocessor System Architectures, Prentice-Hall, Englewood Cliffs, New Jersey, USA, ISBN 0-13-089137-1, 1994.
[14] Any Software Engineering textbook, e.g. R.S. Pressman, Software Engineering, A Practitioner's Approach, McGraw-Hill Inc., New York, USA, ISBN0-07-0507-83-X, 1987.
[15] Pennington Systems Incorporated, XTRAN User's manual, Pennington, New Jersey, USA, 1990.
[16] H.E.O. Brelén, An adaptive plasma density controller at Joint European Torus, Fusion Technol. 27 (2) (1995) 162.
[17] V. Marchese, et al., Detailed design, installation and testing of the new coil protection system for JET, Proceedings of the 18th symposium on Fusion Technology, 22–26 August 1994, Karlsruhe, Germany, 1994.
[18] M. Garibba, et al., First operational experience with the new plasma position and current control system at JET, Proceedings of the 18th Symposium on Fusion Technology, Karlsruhe, Germany, 22–26 August 1994.
[19] F. Sartori, et al., DSP control of the fusion plasma configurations of JET, JET-P (95) 29, 1995.
[20] K. Blackler, Large and small data acquisition systems at JET, Fusion Eng. Des. 43 (1999) 363–369.