TWO CASE STUDIES OF INDUSTRIAL SYSTEMS IN FTU

M.Panella, G.Buceti, C.Centioli, F.Iannone, C.Torelli, V, Vitale, ENEA, Frascati, Italy

Abstract

FTU (Frascati Tokamak Upgrade) started operations in 1989. The experimental session on FTU is a pulsed activity where a shot every 30 minutes should be done. So the control system is asked to manage for all the subplants a startup phase, hardware-controlled plasma shot, and a shutdown phase. The development of the control system has undergone two phases: in the first one a commercial company has been involved into the development of an ad hoc control system, in the second phase a commercial package from Digital has been customized to be used on FTU machine. Both solutions have shown limitations. A full account of the two experiences will be given and further modifications will be described.

1 INTRODUCTION

When the Frascati Tokamak Upgrade (FTU) project started about fifteen years ago [1], reliability plus easy development and integration seemed far from what any commercial software product offered, and building our system from scratch would have entailed a relatively large team of in-house computer analysts.

Since we were very few people (about five professionals), we had to involve a software house in the development of our system.

Just to give an idea of the complexity of the FTU plant, we may recall that the control system has to monitor more than 6000 data points, acquire more than 2000 channels to produce about 15 MB of data per shot and has to run about 130 different types of mimics to monitor the plant status.

The basic requirements for the control system were:

- a computer-based plant control based on concepts of process control;
- give real-time information about plant status and variable values by means of mimics;
- automation of the main experimental phases (fig. 1) and plant management procedures;
- use of a real-time database;
- handle asynchronous events;
- handle alarms;
- archive, access, and process the acquired experimental data;
- man-machine interface based on graphic monitors to improve plant visibility and facilitate enhancement and maintenance;
- development of an ad hoc control system through a commercial company.

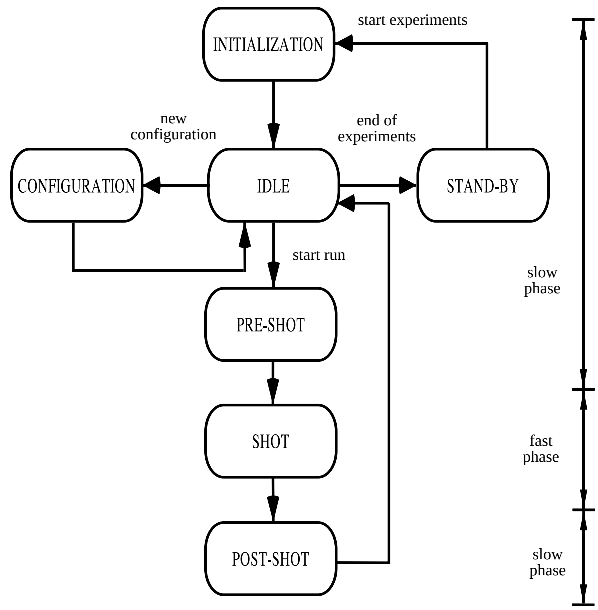

Fig. 1 - Typical Tokamak experiment State Diagram

2 THE FORMER FTU CONTROL SYSTEM

2.1 System Architecture

The system architecture (fig.2) was completely software based, in comparison with the previous way of conceiving a control system, i.e. hardware based [2]:

- logic-functional division of the system into plants (electrical power supplies, machine plants, radiofrequency, data acquisition, supervisor) and sub-plants (vacuum system, cryogenic, RF modules, etc.);
- dedicated software with a real-time database and a tailored man-machine interface;
- a computer for each plant connected via Ethernet;
- plant interface based on several PLC's and PLC supervisors, to transfer only data changes to the database for the slow monitoring phase;
- the fast monitoring phase was run by means an hardware timing system, via CAMAC modules;
- integration of the existing data acquisition system into the control system;
- several command procedures to be executed in a pre-defined order. This was achieved with a language interpreter and some other utilities provided by the supplier.

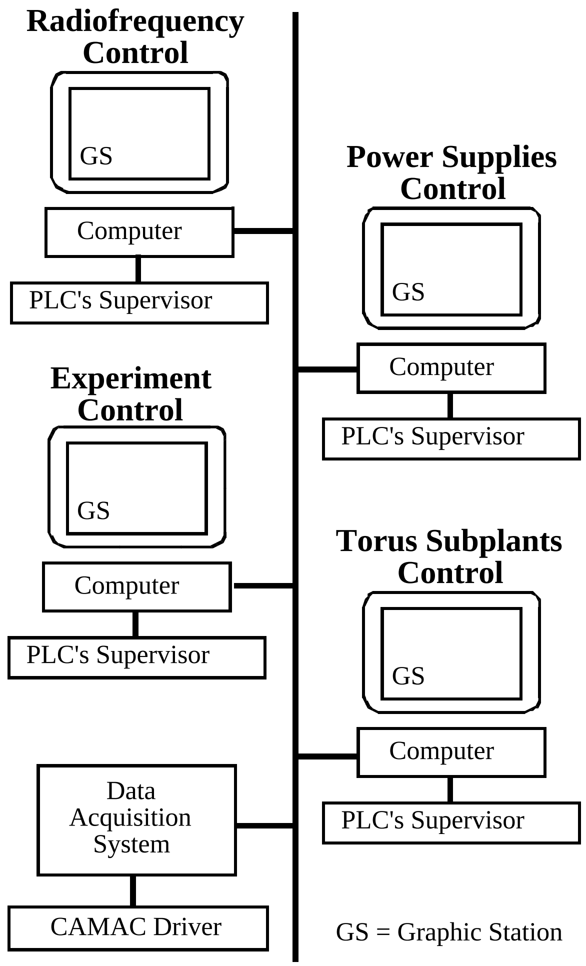

Fig. 1 - FTU control system: block diagram

2.2 Limitations

After several years of FTU operation, we realized that:

- since the local project team and the company developers team hadn't kept in touch from the very beginning, the control system came in a single delivery (i.e. without intermediate tests or releases); this led to architectural mistakes one couldn't get rid of;
- the company couldn't supply a software maintenance service for a long time, especially on the inner parts of the system. Moreover, even the attempt to maintain the code on our own, getting the program sources from the company had proven useless, because of the lack of knowledge transfer from the company to our professionals.

We found other limitations on the first release of the FTU control system:

- a large amount of code was developed and a long period of debug was requested to reach a stable configuration.
- due to the lack of operational knowledge of the machine, great emphasis – and thus big effort – was placed on the development of capabilities which were little use in everyday operations;
- the operational experience suggested many modifications and every change was a painful task, because the inner knowledge of the system was out of our group.
- a true availability of the system was never reached.

3 THE CURRENT CONTROL SYSTEM

3.1 Requirements

Because of a long "shutdown" of FTU, planned to improve the machine characteristics, we analyzed the system limitations and we decided to rebuild a new control system, meeting the following additional requirements:

- advanced and powerful man-machine interface, available on the market;
- easy servicing and updating of the software system;
- possibility to include important commercial software applications, such as database, historical recording of plant measurements, etc;
- possibility to integrate user applications, such as the existing data acquisition system.
- take advantage of the updated hardware available.

3.2 Choosing the software environment

Before replacing the previous software, we looked around for a commercial product. The Digital BasestarOpen package seemed to meet our requirements, because it's a distributed platform born for applications integration: implementations of BasestarOpen made for different operating systems (such as Digital UNIX, HPUX, VAX-VMS and WindowsNT) may be used to build complex systems on an heterogeneous network.

3.3 Hardware Architecture

The data flow between the PLC's and the database system is filtered now by a VME cpu under an OS9 operating system that transfers only the variations in data to the database by means of a TCP-IP protocol.

The VME CPU also controls all the commands that are sent to the plant (whether they are sent by mimics or applications).

Graphic representation of the plant is achieved by means of a network of X-terminals in the control rooms, rather than several dedicated stations point-to-point connected to each computer.

3.4 Software Environment

The current system is centered on the BasestarOpen database, which is the Digital Equipment Corp. (DEC) package [3] designed to facilitate the integration of automation applications with plant-floor control devices.

BasestarOpen contains a predefined set of objects:

-data-point objects to represent machine and plant status;

-event objects to represent significant events, e.g., high temperature alarm;

-trigger objects to generate automatic notification of data changes;

-enbox objects to allow users and applications to receive notification of events;
-activity objects to represent user-written applications.

3.5 Software Architecture

Our control system consists of a set of processes centered on the BasestarOpen database and distributed over three Unix machines and a VAX (OpenVMS) that allow remote supervision of the plants and the experiments.

Each mimic is based on the SL-GMS graphic tool and operates through a dedicated process (named BaseStar Graphic Enabler) that obtains information from the BasestarOpen database and allows the technical team to develop any mimic easily.

Plant status can be obtained from a real-time database, updated by communication processes and, whenever necessary, the operator can transmit commands via the mimics to the plant structures and/or hardware devices, such as the Camac modules.

In addition to sending simple commands from a mimic, the operator can send a prefixed sequence of commands that have their own control logic for carrying out specific activities (experiments, machine cooling, etc.)

We developed in-house all system processes utilizing BasestarOpen database tools, particularly the event mechanism that is based on objects such as event, enbox, and trigger, also for the database access and inter-process communication,

4 LIMITATIONS

Compared with the previous experience, the adoption of the Digital BasestarOpen package was an undoubted improvement of control system.

Nevertheless, several serious limitations arose from this experience:

- the supplied documentation has not been sufficient to investigate all the aspects of the package, as, for example, the system messages interpretation;
- because of the pulsed nature of our experiment, we configured many processes to startup and shutdown for each experiment (on average twenty per day). We discovered too late this is not the proper use of the package and some extra bugs arose;
- due to the relationship with a commercial company, we were free to change some application parameters, but we had to wait for a new package release to fit the system on further plant needs;
- as for the previous system, we exploited the suggestions coming from the company experts (i.e. Digital) to customize our application, but the lack of deep knowledge of our needs brought often misleading indications;
- other users of the same package are out of the scientific world, so we didn't have the possibility to share our experience with other laboratories.
- we experienced situations not shared by other users, so the company didn't consider our problem so important to be solved employing new resources ;

Some additional problems derived from our strategy:

- a strong emphasis was placed on the centralized nature of the system, underestimating the advantage of the architecture where separate modules and/or subplants may run autonomously;
- we had just one year to release the current system, then we had no time to redesign the global architecture
- the startup phase took advantage of the 'ready-to-use' nature of the package but a lot of time has been afterwards spent to identify and correct the bugs.

5 CONCLUSIONS

The tokamak systems are in practice experimental as regards both their intrinsic complexity and the constant modifications necessary for physics investigations. What is important in our context is that within a single environment, different objectives have to be reached simultaneously: plant control in seconds time scale (via the PLC) together with the study of phenomena in milliseconds (via the Camac); continuous monitoring of the plant together with the control of the "pulsed" phases of experimentation; the simultaneous transmission of single commands or complex operational procedures by the same, relatively specialized, operator. Finally, a high level of control reliability with flexibility (easy reconfiguration according to experimental requirements) is required.

Now, the experiments on FTU proceed and the control system is reliable in a way it was asked to run for the experimental sessions.

However, we believe the whole project can be adjusted and optimized. In fact, we wish to improve the architecture moving some functions on the plant computers and simplifying other parts. This will make the subplants more autonomous and then more reliable.

Our results bear out the opinion that it is worthwhile purchasing a commercial package and adapting it to provide a system capable of sophisticated control requirements.

Nevertheless, while a commercial package reduces in-house software engineering, one cannot expect from this robustness and reliability as from a software product dedicated to process control, without extra care in the optimization.

REFERENCES

[1] R.Andreani et al., “The Frascati Tokamak Upgrade: Electrical Mechanical and Thermal Features”, *Fusion Technology* 218,1991.

[2] B.Berardi et al., “Development status of FTU control system”, *Symposium On Fusion Technology*, Avignon, 1986.

[3] M.Panella et al., “A commercial real-time manufacturing platform for the new control system on FTU”, *Fusion Engineering and Design* 43,1999.