FROM A UNIX TO A PC BASED SCADA SYSTEM

F. Momal, CERN, Geneva

Abstract

In order to facilitate the development of supervisory applications involved in slow process control (such as cryogenic control), the LHC/IAS Group (Equipment Controls Group) opted, a few years ago, for an industrial SCADA package which runs on UNIX platforms. However, to reduce costs and following market trends, it has been decided to move over to a PC-based package. Several processes relating to the testing of the prototypes of the LHC magnets are already controlled in this way. However, it was still necessary to provide all the services previously available to the users, for example, data archiving in central databases, real-time access through the Web, automatic GSM calls, etc. This paper presents the advantages and drawbacks of a PC-based package versus a Unix-based system. It also lists the criteria used in the market survey to arrive at the final selection, as well as, the overall architecture, highlighting the developments needed to integrate the package into the global computing environment.

1 A UNIX SYSTEM

Several years ago, when we were asked to install systems to supervise and control the tests of the LHC magnet prototypes, we looked to industrial SCADA (Supervisory Control and Data Acquisition) packages[1]. They are usually composed of some kind of real-time database (RTDB) holding the last known values of all the process variables one wants to monitor. Around this RTDB, a set of programs executes the most common tasks requested by the operators: PLC (Programmable Logic Controller) protocol drivers, trending and archiving tools, alarm handling programs, animated graphical representations of the process, etc. Choosing an industrial package would release us from the burden of having to develop and maintain our own software[3]. However we could only use these systems if they could be smoothly integrated into the overall CERN environment and if our different needs could be answered[2].

Firstly, and this is perhaps the most important requirement for these kind of systems, the software and its environment must be as stable as possible. Since several parts of the program were to run at the same time, for example, the PLC protocol driver and the MMI (Man Machine Interface), the operating system of the machine had to be multitasked. We wanted the archives to be accessible from all machines and, for some data, inserted into a central Oracle database. We wanted some alarms to be sent to the Unix based CERN Alarm Server. It should be possible to connect remote consoles to the system. At a time when Microsoft Windows 3.1 was still in its youth, we did not even think of another alternative than UNIX. SCADA packages offered in this environment had answers for all the major requirements. Apart from the stability that was undisputed, this OS was made to be multitasked. The use of X Windows gave us a convenient way of having multiple remote screens connected to the same machine. This was mandatory since several operators working in different control rooms were to use the program. Offering an access from everywhere to the data was simple thanks to the standard TCP/IP environment. All the archives and configuration files were stored on AFS. Using the API (Application Programming Interface) which comes with the software, we developed a lot of specialised programs ranging from complex sensor calibrations to Web interface. We ended up with a system which offered all that the process specialists were asking for and which offered us a reliable and solid environment.

Paradoxically, the main limitations we encountered were coming from the used technologies. We found the frequency of the operating system updates too fast. Due to incompatibilities, a new release of the Unix system was generally leading a new release of the SCADA package. We lost a considerable amount of time installing new versions of the used tools, running patches and converting applications. The level of expertise needed to resolve the problems forced us to invest largely in courses and time. X Windows which we found so powerful was too demanding in terms of network bandwidth to allow the use of home or portable PCs to be used as occasional consoles. Finally the combined price of the machines, the maintenance and the software was relatively high.

2 TOWARDS A PC-BASED SYSTEM

A couple of years ago, we decided to launch a new market survey on SCADA packages. This was based on two main facts: Europeans systems now existed and the market trend seemed to be drifting towards Windows NT. This last point became particularly crucial when we learnt that the company that was selling us our current SCADA had taken the decision to stop the development on UNIX. It seems more obvious daily that software companies are turning away from Unix, apart from very large and very expensive systems. Even more alarming, standards such as OPC[1] are emerging which are coming from the Windows world.

Firstly, we had to make sure that the previously described major requirements could be answered. Windows NT 4.0 seemed stable enough. The PC machines are nowadays as powerful as our Unix workstations. File servers such as Novell servers are offering the universal remote file access which we were seeking. Multiscreen cards offer the operator all the viewing comfort he asks for. However, we still had to find a way to provide the remote services people were expecting. This was an important point of our market survey.

On the other hand, doing the transition to a Microsoft Windows based system brings many advantages. The price of the different components of the system, whether it is hardware or software is smaller. Maintenance also is much cheaper. People are often used to the Windows environment and, although the solving of a complex problem requires an expertise as important as before, simple problems may now be handled by the users themselves. The quantity of programs that can be used to extend the system is enormous. As far as an API is offered by the SCADA, powerful and user-friendly compilers may be used. Several drawing programs exist to draw the MMI screens and many products can be used to analyse, chart and archive the data. Finally, if the new product follows the Windows interoperability standard, it could be easy to tightly link different products with the SCADA package.

3 A MARKET SURVEY

The market survey was launched together with a list of criteria which was to be used to sort the SCADA systems. The package should:

- Run under Windows NT.
- Have a client-server and modular structure. This means that while the RTDB is running on some server doing the PLC acquisition, the MMI can run on other client machines.
- Implement the mainly used automation protocol standards (at least Profibus, FIP, and OPC).
- Offer a complete API.
- Have an object-oriented RTDB.
- Use a standard computing language as the integrated programming language.
- Be automatically programmable using an external description source. This mainly means that the configuration files used in the different parts of the SCADA can be editable in ASCII.
- Follow the Microsoft interoperability standards (known as OLE, ActiveX and DCOM).
- Export the archives in a standard format.
- Be European. This is due to internal CERN constraints.

We did not ask for an Internet access through a browser due to the fact that we thought that the market was unprepared for a good solution in this area. However, we checked that the developers were making provision for this.

After a general pre-selection, four systems answering several of these criteria were analysed thoroughly and one was selected.

4 A GENERIC ARCHITECTURE

Before putting the selected system into operation, we had to set-up a global computing architecture that could offer the services previously described.

The base of the architecture is composed of a data server. This machine is dedicated to acquiring the values of the different process variables to be monitored. This data server contains the RTDB of the SCADA system. The connection with the PLCs is done through a specific network card from Applicom™. This card holds on board the PLC protocol driver. The data server is sending, whenever there is a change, the values of the process variables to the remote clients that have subscribed them. A second PC, identical to this data server, serves as a hot back-up. If the data server, for any reason, breaks down, then the back-up automatically takes its role. By dedicating a PC to data acquisition and distribution, one stabilises the load on the machine.

A client machine handles the archives[2]. On its hard disk, a few days of data can be held. Being a module of the SCADA system, it allows MMI clients to quickly access the recent values. For long term archiving, the data are sent to a central database running on a UNIX machine.

Another client machine, which is also a SCADA module, takes care of the alarms. In a unique location, it manages the handling of the process alarms by the operators. By connecting a specific program to this client, automatic messages are sent to the GSM phones of the operators who are on call. This machine serves also as a gateway to the CERN's central alarm server, if needed.

MMI clients located near the process, in a central control room or in the office, are also connected to the data server. All the graphical animations are done on the clients and thus the network load between the MMI client and the data server is very low. Only the values of the subscribed process variables are sent, on change, by the data server to the MMI clients. This allows having a full client that runs on a computer at home, the modem's bandwidth being sufficient. This solution proves to be more efficient than the X Windows approach. A network

[1] OLE for Process Control

[2] For small systems, all the clients may run on the same machine than the data server.

failure between the client and the server no longer affects the data server.
All the configuration files are available on a file server, allowing a central maintenance of the versions.
Internet access to the data is done through another program tightly linked to the SCADA system. This program offers Internet oriented interfaces, such as a Java API and JavaBeans. This interface is used mainly to have a remote synthetic view of the process rather than for a full remote control.
The main underlying concern that led to this architecture is modularity. Ideally, it should be possible to replace any part of the system with another one without jeopardising the overall architecture. If, for example, we were to use another SCADA package, we would not have to change any of the hardware parts (neither the PCs nor the acquisition cards). We could still use the long-term archiving system, the alarm gateway and the Internet access.
Developments needed to integrate the SCADA package in the architecture and to connect it to the different services (archives, alarms, and remote access) were not very important. This is due to the fact that we managed to find compatible software. We only had to do some coding for the connection with the archiving system and on the Java MMI used to access the data through the Web.

5 PRACTICAL EXPERIENCES

We have already installed Windows NT based SCADA systems on several processes. It took some time to acquire the required Windows NT expertise and anyone wishing to do the same transition should not forget it in the planning.
We have had no problem yet with the stability and everything seems to behave as expected. We can only recommend not trying to limit the PC hardware components. SCADA packages are resource demanding and a lot of memory, a large disk and a fast CPU are welcomed.
The only complaint we had from the users of our systems was regarding the change in the SCADA graphical screens. In fact, we took the opportunity to try to upgrade the quality of the MMI, but as often in these kind of applications, stability is one of the most important points.

REFERENCES

[1] F. Momal, D.Bienvenu, D.Brahy, D.Lavielle, R. Saban, B. Vullierme, L.Walckiers, “A Control System based on Industrial Components for Measuring and Testing the Prototype Magnets for LHC” , EPAC’94,London,1994 (http://mish231.cern.ch/docs/papier2.htm)
[2] F.Momal, D.Brahy, R.Saban, P.Sollander, "Integrating a commercial Industrial control System to the Accelerator Control System", ICALEPS'93 ,Berlin
[3] R.Saban, P.Ciriani, A. Guiard-Marigny, H.Laeger, M.Rabany and A.Swift, "Equipment Industrially Controlled ",ICALEPS'93 Berlin

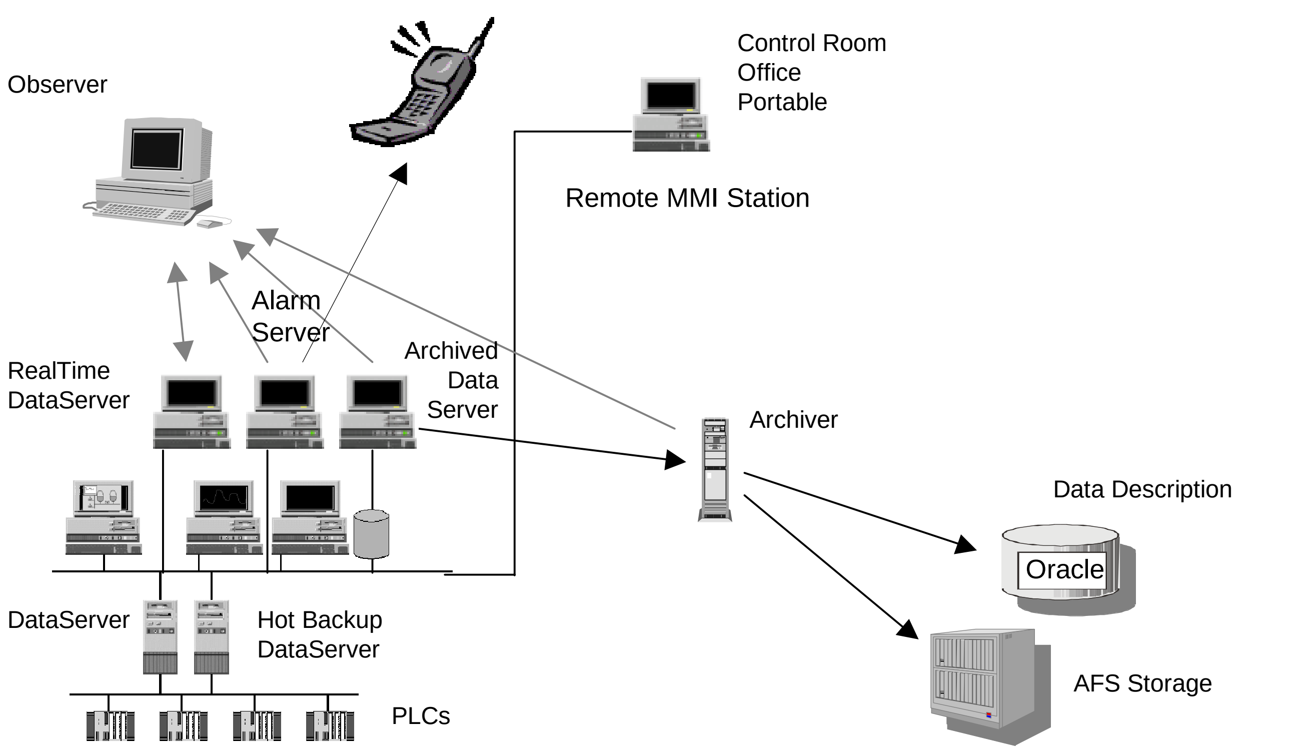

Figure 1: The general architecture