THE DESIGN OF THE CONTROL SYSTEM FOR ANKA

S. Avsec, B. Jeram, G. Mavric, M. Plesko, M. Smolej, J. Stefan Institute, Ljubljana, Slovenia
e-mail: mark.plesko@ijs.si

Abstract

ANKA is a low budget light source. It has to be built with minimal cost. This includes a short construction period to keep the personnel cost low. The use of ANKA as an industrial light source on the other hand places tight constraints on the reliability of its components and its operation. In the creation of the control system one has to minimise in-house development and maximise the use of commercial-quality products. Here, the term commercial-quality applies equally to off-the-shelf-products and to products in the public domain or from other accelerator centres that are field proven and are documented at a level equal to commercial products.
It has been our goal to search for the simplest model and thus the most reliable one. Our extensive investigations have shown that none of the actions and responses of the control system for ANKA need a real-time operating system. Therefore, we have disposed of the whole middle layer of the control system including VME crates in favour of a fieldbus network which connects the individual devices directly to the console computers. The consoles will be PCs running Windows NT. The fieldbus candidate is LonWorks from Echelon which provides plug-and-play and requires almost no additional development. On the software side we have opted for TACO, an object oriented control system environment developed at the ESRF.

1. DESIGN CONSIDERATIONS

The 2.5 GeV electron storage ring ANKA[1] is being constructed at the Forschungszentrum Karlsruhe in Germany and will be used primarily for industrial applications. Such a light source operates for 10 years and more. This asks for a control system that is upgradeable, scaleable and simple to maintain. Therefore, the following considerations have been given top priority in the design of the control system.

§1. Employ standards that will last over the lifetime of the project and solutions that will be supported for a long time.

The control system architecture should allow heterogeneity if necessary, but should not be based on it in its basic form. Simplicity requires that there is only one operating system and a uniform and consistent developing environment:

§2. Use a uniform operating and development environment.

The same is true for the control system electronics. It should not be adapted for each possible device, but rather such devices should be chosen that easily support standard electronics:

§3. Minimise the number of electronic boards.

The cheapest solutions are commercial "off the shelf" solutions. They are well tested and therefore reliable, well documented and thus simplify development. Their price is heavily outweighed by savings in personnel time and cost.

§4. Minimise in-house development and maximise the use of commercial components.

Here, the term commercial applies also to products in the public domain or from other accelerator centres that are field proven and are documented at a level equal to commercial products.

2. THE COMPONENTS OF THE CONTROL SYSTEM

2.1 PCs, Windows NT and Java

Although PC hardware and in particular PC software is significantly cheaper than their workstation equivalents, there are other more important reasons in favour of PCs:

- PCs are used for accelerator control in several laboratories (CERN[2], DESY[3,4], KEK[5])
- PCs have become equivalent to workstations in features if not in power
- PC hardware has become sufficiently reliable
- excellent development tools exist at low price
- commonly known graphical user interface (GUI) tools, e.g. Visual Basic, allow for easy prototyping and testing
- rugged industrial packages exist, if needed in an accelerator environment.

For the operating system, the obvious candidate is Windows NT, because:

- it has all attributes of a modern multitasking operating system
- it supports all standard network protocols
- it is stable and reliable
- it has built-in graphical elements (mouse, events, objects, etc.)
- users are already familiar with it from before
- through the object linking and embedding (OLE) mechanism it is possible to interface the control system into spreadsheets (e.g. Excel) and graphical programming tools (e.g. Visual Basic).

The programming language Java may look a bold choice at the moment, but we have to look into the future. Java, which may be described as the next generation object oriented language after C++, has been accepted by the

0-7803-4376-X/98/$10.00 © 1998 IEEE

whole industry. This year has witnessed an explosion of Java applications and development tools. It is just a matter of time that state-of-the-art development environments for Java, similar to Delphi or Optima, will be available on PCs and other platforms. We are currently evaluating several promising development tools.

Java allows furthermore for true portability and complete platform independence, even between UNIX and Windows. All investment in Java source code will therefore never become obsolete. Having been born for network computing, Java simplifies control over the network by being built into World Wide Web browsers. The fact that Java is an interpreted language is not a problem, as computing power is not such an important issue in accelerator control.

2.2. The TACO Control System Architecture

There are many implementations of control systems that were developed in-house by accelerator laboratories, which are too site-specific to be taken into consideration for ANKA. Of interest are general control system architectures. Among those that deal with the console and control layers and the communication between them are TACO and EPICS and maybe the commercial Vsystem.

We have evaluated the performance of TACO [6] (see Appendix A) and have found that it fulfils all needs of ANKA. TACO has been already ported to the hardware platform of our choice - PCs. This has been done both for the Linux and Windows NT operating systems, while the other two systems, EPICS and Vsystems don't support a full PC solution so far. Although richer in features, they rely on expensive hardware, VME and Digital, respectively.

TACO, developed at the ESRF, is a control system architecture that deals with the console and control layers and provides communication between them. TACO is object oriented and is being currently upgraded by the ESRF to support the newest developments in distributed objects technology like CORBA, etc. Other laboratories (SLS, Soleil) are seriously considering to join a collaboration based on TACO, therefore we can expect further tools and applications developed that we can use "out-of-the-box" for ANKA.

Due to the small size and lower control data flow of ANKA, TACO will run directly on the PCs. TACO relies on a client/server architecture. In the case of ANKA, both the clients and the servers will run on PCs, albeit not necessarily on the same machine. The device servers will collect the data gathered through the field bus or will send commands into the field bus. Each device server will be responsible for a certain group of equal devices, represented logically as objects. The clients will run on any operator console and will, through the applications programming interface (API), communicate with any device server running anywhere. This will allow that control system information will be available all around the ANKA intranet, be it in offices or on beamlines and even anywhere in the Internet. Security provisions will ensure that no dangerous action can be performed outside of the control room.

In a joint effort with the ESRF and the SLS to upgrade TACO, we will extend the communication model of TACO from RPC to CORBA. In addition, the API will provide true objects and will be available to Java with a Java-CORBA implementation. All controls will be done through Java applications or applets that will run in the enviroment of a WWW brower like Netscape or MS-Explorer. The client-side of the control system will be thus completely platform independent and due to its object-oriented structure portable to other acelerator facilities.

2.3. The Fieldbus

Our extensive investigations have shown that most of the actions and responses of the control system can happen on a time scale of 1s, with the exception of:

1. ramping: the power supply current has to be increased in 1 ms steps
2. feedback loops: signals have to be processed and acted upon in the 100 Hz range
3. knobs: fine tuning with knobs needs about 25 actions and displays per second

The first task can be solved by intelligent power supplies or by cheap micro controllers, one attached to each power supply. The second tasks requires a dedicated system which has to be developed on separate hardware anyway. There are no short term plans for feedback systems in ANKA, but if they will have to be built, they will run on their own in specialised hardware and just be connected to the control system for exchanging messages. Knob control may be very useful. But there are just two knobs that an operator can handle at the same time, therefore the necessary responses can be achieved within a non-real-time operating system task at high priority.

As there is no need for a real-time operating system, we can dispose of the whole middle layer of the control system including VME crates. For this we need a field bus network to connect the individual devices directly to the console computers.

There are two low-cost solutions on the market:

- CAN-bus[7], a fast, low-function bus for real-time network systems
- LonWorks[8], an intelligent decentralised network for automation and control

After first evaluations we have a preference for LonWorks, because it offers a complete network system in hardware and software with all the necessary development and network management tools. LonWorks already implements layers 1-6 of the ISO/OSI model and thus transparently connects controlled devices and control room consoles without any need for network programming. Many types of electronic boards are available from third parties (RS-232, ADCs, etc.), others can be easily assembled. LonWorks thus reduces field bus development practically to mere network designing.

2.4. The Interface to the Devices of ANKA

Following §3, we define a set of I/O boards that will exist and be supported in advance. To our mind, these should suffice to control all standard ANKA devices with the possible exception of dedicated diagnostic and instrumentation tools. But even there, one should try to fall back to the standard boards.

- ADC/DAC (16-bit) + 16 digital I/O for power supplies
- 16-channel ADC (12-bit) for temperature, vacuum, RF, diagnostics, etc.
- 64 digital I/O or similar
- RS-232 interface
- GPIB interface

For the local control of beamlines, a few extra boards will be made available: timer/counter, stepper driver, shaft encoder, etc.

3. FIRST TESTS WITH LONWORKS

The purpose of the tests were to determine the operating parameters of the LonWorks network in a real environment and test the ability of a single Neuron module to control the power supplies of dipole magnets. Especially time critical is the ramping process, at which several modules should synchronously (within 1 ms time-jitter) increment their output value once per millisecond.

The tests were made at the synchrotron of the University of Bonn. In the short available period of time, we have done the following:

1. Measurements and comparisons of the network transmission rate and Neuron chip board behaviour in a non-noisy and a very noisy environment were made;
2. Writing a prototype application to ramp a power supply and set and read its current;
3. Creating a set-up where machine parameters can be tuned through mechanical knobs and checking response times at very high network loads;
4. Attaching a micro controller board to a Neuron chip and programming the micro controller to either drive or simulate PS ramping and control;

Several types of communication between Neuron-chip and peripheral (custom made) devices have been tested (Neurowire, Muxbus, bit shifting and serial RS232 communication) and the response and data throughput times measured.

Our tests have not shown any severe limitations of the LonWorks and so far in no case exclude the idea of using them for the control system. In opposite: the tests have proven, that the LonWorks are relatively simple to handle, operate and maintain, and easy to connect among each others or with peripheral devices.

LonWorks simplify the tedious work of programming the network communication, putting all of its essentials in form of firmware into the Neuron chip. The powerful Neuron C language enables the user to quickly build and program a large and complicated communication network, which can easily be extended or modified. The integrated debugger is a useful tool to test and follow the events in the system.

4. CONCLUSIONS

Building an accelerator control system is no art anymore. It has become possible to take a few industry solutions and just work on the implementation and the interfacing. All existing accelerator control system models do their job, which has been proven by the fact that all accelerators work. It has been our goal to search for the simplest model and thus the most reliable one.

Due to the relatively small size of ANKA, we have found that instead of having relatively dumb devices and a strong middle layer, we bring intelligence to the devices and avoid the middle layer completely. This is possible also because most equipment has already some intelligent control. It is just necessary to combine them with a proper field bus.

Maximal portability and longevity has been achieved by choosing tools and products that will remain alive even after a new generation of PCs or a change of operating systems. Such an approach reduces cost and saves development time, too.

5. ACKNOWLEDGEMENTS

During our work, we enjoyed in the friendly cooperation with the ANKA team at the Forschungszentrum Karlsruhe. Furthermore, we thank prof. Husmann and his collegues for kindly allowing us to run tests at the Bonn synchrotron and for giving us all the necessary support there. We thank the company Echelon Munich for loaning the LonBuilder development kit for the time of our tests.

6. REFERENCES

[1] H.O. Moser et al., ANKA Conceptual design, Forschungszentrum Karlsruhe, June 12th, 1995.

[2] Ivan Deloose, PS/CO Note 96-53, Sep. 20th, 1996.

[3] P. Duval, R. Schmitz, Controlling DESY Accelerators with PCs, PCaPAC Workshop, DESY Hamburg, 1996.

[4] C.M. Fu, M. Clausen, G. Csuka, PC Hardware in Realtime Controls, PCaPAC Workshop, DESY Hamburg, 1996.

[5] A. Abe, private communication.

[6] A. Götz et al., TACO: An object oriented system for PC's running Linux, Windows/NT, OS-9, LynxOS or VxWorks, presented at the PCs and Particle Accelerator Control (PCaPAC) Workshop, DESY Hamburg, 1996.

[7] Road Vehicles - Interchange of Digital Information - Controller Area Network (CAN) for High Speed Communication, Document Iso/DIS 11898, International Standardization Organization (1991).

[8] The '95-'96 Echelon LonWorks Product Databook, Echelon Corporation, 1995.