# SOFTWARE DESIGN OF THE SCHONLAND 6MV EN-TANDEM ACCELERATOR CONTROL SYSTEM

R.D. Maclear, S.H. Connell, A.H. Andeweg, J.P.F. Sellschop, SRCNS, Johannesburg, South Africa
M.E. Hogan, NAC, Faure, South Africa
H.F. Weehuizen, Massey University at Wellington, Wellington, New Zealand

## *Abstract*

Currently the 6MV EN-Tandem van der Graaff accelerator at the Schonland Research Centre for Nuclear Sciences is being upgraded for computer control. A distributed, network based control architecture has been implemented and successfully installed to control two sections of the accelerator. A graphical user interface (GUI) is being designed which is both didactic and easy to use. The entire control system software has been written to use OS/2 and exploits the multi-threaded and real-time nature of the operating system. Source code has been written in C and C++. A client-server type architecture is used to control electronic components to provide dynamic prioritisation of control instructions and monitoring events. A relational database is used to store all system and control variables to facilitate upgrading needs. The modular nature of the control system allows minimal down time when installing new sections for computer control. We present here the design concepts, a status report and first results of the accelerator control system.

## 1 INTRODUCTION

The Schonland Research Centre for Nuclear Sciences operates a 6MV EN-Tandem van der Graaff particle accelerator. There are two ion sources capable of producing both light and heavy particle beams. The Duoplasmatron ion source is a gas ion source producing proton and alpha ion beams. The 860A sputter ion source is used to produce heavy ion beams of almost all elements of the periodic table. The Centre is also in the process of developing and installing an Accelerator Mass Spectrometer facility. The Centre is involved, not only in physics experimental work, but also in routine geological, biological and medical sample trace analysis using an Ion Microprobe [1]. Any facility that is involved in this broad spectrum of work requires a stable, user friendly accelerator control system. A multiple PC-based control system is in the process of being developed and installed for the accelerator facility. The control system has been designed to require low maintenance during and after installation. A Graphical User Interface (GUI) has been designed for control of the accelerator with both ease of use and education in mind. This allows non-specialist users and students the opportunity to learn from using the facility.

Working on a limited budget, a variety of decisions had to be taken. It was decided to use OS/2 Warp 4 as the base operating system as the National Accelerator Centre in South Africa had already implemented a network broadcast system written for OS/2 [2]. OS/2 is also a stable operating system with excellent real-time and multitasking capabilities. All control software has been written in-house, and interfaces with a commercial SQL database sever.

The control system has been designed to be distributed, making extensive use of databasing and networking. A Variable Table Program is used as a standard communication system for broadcasting across the network, while distributing the control system reduces maintenance and installation down-time, and allows for local intelligent control.

## 2 THE DISTRIBUTED CONTROL SYSTEM

The control system was designed to be as homogenous as possible with very little maintenance. This facilitates easy upgrading and installation of future sections of the accelerator, and requires little expert computer knowledge. The architecture of the system is based on the Standard Model for accelerator control [3]. Software interfaces and controls are on two levels, while hardware control resides on a third. Fig. 1 illustrates the architecture and logical layout which is also explained below. The software of the system can be divided into four parts, while a computer network is used to distribute the system.

### *2.1 The Network*

The network used is an Ethernet 10BASE2 network which was chosen because the product was mature and readily available. The network has been separated from the Local Area Network (LAN) of the Centre to avoid unnecessary traffic. Since there are only a few computers connected to the network (a maximum of 7 is foreseen), loading is kept to a minimum, thus reducing network collisions. This also reduces the transmission delay of the Ethernet system.

The communication system is designed to be event driven. In event-driven systems, traffic is proportional to the number of events. Since the control system is a reasonably small system, with an average of 30 control variables per control node, the number of events on the network is almost always at a minimal level.

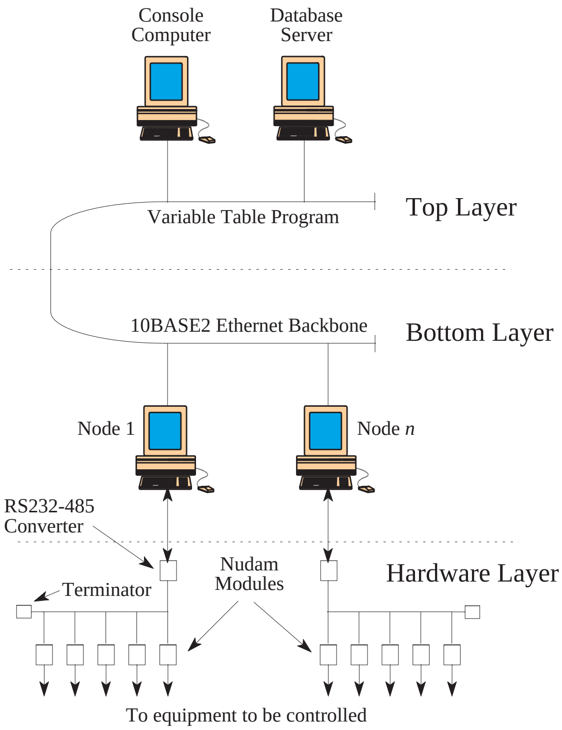

Figure 1: Logical layout and architecture of the control system.

## 2.2 *The Variable Table*

The Variable Table Program [2], written by the NAC control group, is used as the base message broadcast system of the control system. Each control variable has it's own entry in the table with fields for actual and reference values, and actual and reference status. Other fields in the entry are used for general house-keeping and variable linkage status. The variable table serves as a common interface between each control node and console node on the network, broadcasting state and variable changes to each node. A local and global interface is available. The local interface is used on controlling nodes so that only variables associated with that node can be accessed. The global interface is used on the console node for access to all variables. In this way, an operator can change any variable in the table, while only the controlling node for that variable is able to respond.

## 2.3 *Control Nodes*

Control nodes are used to respond to operator requests and convert these requests into actual controlling instructions. The hardware used for electronic control are Nudam modules [4] which accept ASCII instructions via the RS232 port of the PC. Since there is always more than one controlling application on a node, a COM port server had to be designed to coordinate COM port access. A priority queue is used between the controlling clients and the server. All write instructions have a higher priority than read instructions. When a variable is being controlled by the operator, both the reading and writing priorities for that variable are dynamically increased, thus this variable will have the highest priority in the queue, causing the server to process these instructions immediately.

When an operator initiates an instruction, the variable table notifies the corresponding client via a network-wide broadcast, which in turn processes the value change into a command which is sent to the server. The server removes the command, writes to the electronics and returns the result back to the calling client. The calling client processes the result and performs error checking. Fig. 2 illustrates the software layout for a controlling node. When no instructions are issued by the operator, the clients perform polling for the variable through the server. Any changes in the variable value are written directly to the variable table.

The nature of the controlling electronics ensures that the written instruction gets a return value or state before the next instruction is performed. This removes the problem of determining which result from the electronics corresponds to which instruction, and therefore no unnecessary processing is needed by the server.

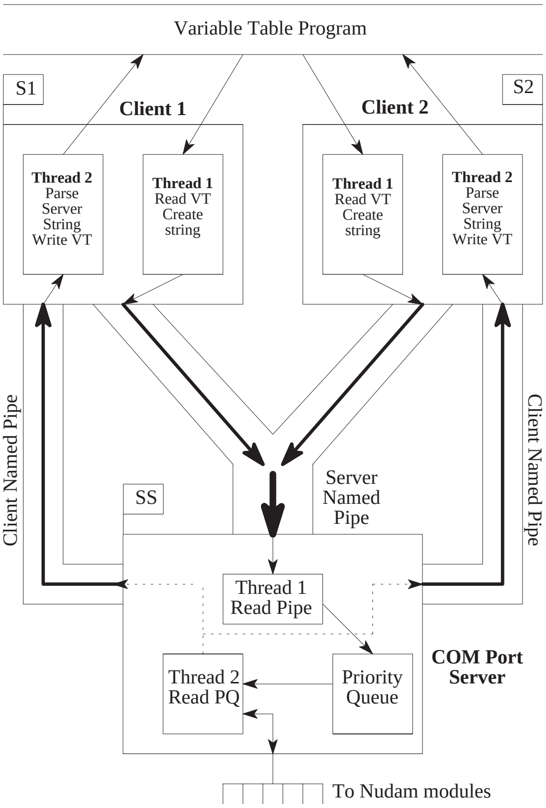

Figure 2: Typical software layout for a local controlling node. Each client and the server have a semaphore (S*n* and SS respectively) which indicates whether the program is running or not.

### *2.4 Console Node*

The console node is used to interface to the entire control system. The operators use a GUI for control, which has been written to be as generic as possible. This is achieved through the use of object oriented programming and class inheritance. Instead of each variable being controlled separately, a dialog box is constructed to control all variables for the electronic device. For example, a standard voltage controlled power supply would have a variable for the voltage, current and digital state. This would be incorporated into one dialog box with a slider control for voltage, digital control buttons and feedback variables, instead of three separate dialogs for each.

Each controlling dialog box is created at real-time, based on information stored in the database. A set of generic digital and analog controls is used to construct each dialog box. This reduces the rewriting of any code when a power supply is changed. All that is needed is for the database to be modified and the dialog box will be created accordingly.

### *2.5 Database Node*

Since the control system is still in its development stage, and a large number of parameters are modified regularly, it is important that the system is readily expandable and flexible. This is most easily achieved through basing the entire control system on a database which is easier to maintain than redesigning any software. This also allows the system to be reconfigured and modified at one single point.

A relational, network-aware database, DB2/2, is used for storing all control and system variables. The variable table uses this database to create each variable entry, and each node reads this database at startup. The console node, using a variety of database objects, retrieves all the necessary information for system control and this is stored locally. The control nodes retrieve only the information that is necessary for that node, and stores it locally in memory. Storing the database information locally on each node, provides for any problems on the database node. This way, if the database manager develops some problems, the control system can continue to function, losing only some functionality. The database classes that are used are written to determine the database structure at startup, so that no code needs to be regenerated if the structure of the database is modified in any way.

The database consists of a variety of tables, but two main tables are used for controlling hardware. The Main table contains all variable definitions while the second contains all power supply definitions, cross referencing with the Main table. This way, any power supply object can be created directly from the database, allowing for easy modification.

## 3 INTEGRATION AND PRELIMINARY RESULTS

Software integration of the control system is a very quick and easy procedure. All that has to be done is placing a few entries in the database for each power supply and control variables. All other aspects are automatically updated. The hardware integration is a far more time consuming task, as the existing electronic controls are old and still use buttons and dials, and all this has to be replaced and new electronics interfaced. All electronics are tested off-line before installation and commissioning, reducing accelerator down time.

Preliminary results indicate that the software control is very reliable and error free, with no errors being reported apart from operator errors. The average response time for instructions to be processed from the client through the server, to the electronics and back to the client is 94 ms. This excludes the overhead of network broadcasting. The response time is acceptable for our control system, as there are no time-critical processes that need control. Analog fast response is done by faraday cup meters and beam profile monitors.

## 4 CONCLUSION AND FUTURE PLANS

The control system has been successfully installed for two sections of the accelerator. It has been functional for nearly 6 months now and has proved to be very successful. Some problems have arisen from operators who have become reliant on the old control system, but are slowly becoming accustomized with the new PC-based version.

Currently, the control system is used only for standard control. Future plans include logging, automatic beam selection, safety implementation, continuous variable monitoring and intelligent variable control. This would need a separate program to monitor a variable, and based on some preset physics laws and procedures, determine what the value of that variable should be.

The entire control system is envisaged to be installed and operational by the end of the year.

## 5 REFERENCES

[1] A.H. Andeweg, *et al*, “The Schonland Micro-Scanning Ion Beam Analysis Facility”, Nucl. Instr. and Meth. B130 (1997), 37-44

[2] H.F. Weehuizen, I.H. Kohler, M.E. Hogan, P.J. Theron, “The PC Based Control System of the NAC”, ICALEPCS’97, Beijing, China, 1997

[3] A. Daneels, “Behind the Scene of Experimental Physics Controls”, Nucl. Instr. and Meth. A352 (1994), 1-5

[4] A.H. Andeweg, S.H. Connell, R.D. Maclear, J.P.F. Sellschop, “Computerised Control of the 6MV EN-Tandem Accelerator at the Schonland Research Centre for Nuclear Sciences”, These proceedings