CONSYS – A NEW CONTROL SYSTEM FOR ASTRID AND ELISA

J.S. Nielsen, K.T. Nielsen, and T. Worm
ISA, University of Århus, Denmark

Abstract

A new PC/Windows NT control system has been developed for ASTRID [1] and ELISA [2]. The system is network based (Ethernet), with distributed front-end and client/console computers. The system consists of three parts: The kernel, devices, and client programs. The kernel, common for all computers, handles all communication between devices and client programs, be it locally on the same computer or across the network. The devices store the values of the parameters on the system, and handle all the input/output communication to the hardware under control through device drivers etc. For interaction with the operators, a number of client programs have been developed, of which the major one is the Console. The tripartition of the system allow very easy addition of new devices and client programs, as new types of hardware needs control, and as new needs for utility programs arise. The computer-code is highly object-oriented reducing code size and development time. The system is fully software configurable with all addresses, conversions, and display properties stored in an ODBC compliant database.

1 INTRODUCTION

In the mid ninety's it was realised that the old control system for the ASTRID storage ring would become obsolete, and it did not have room for further upgrades. Therefore a new control system had to be designed. The development of the new control system, ConSys, was started at ISA in the beginning of 95. The aim of the project was to develop a general site and machine independent control system. Taken into considerations was also the possibility of new facilities in Århus.

Although UNIX workstations are widely used for control systems elsewhere, it was preferred to base the system on the PC platform. At ISA everything apart from the central control computer was based on PC's. That implied that there were a lot of PC cards for the ISA-bus, which one wanted to keep. The PC's also had the power needed for a control system, and it was therefore decided to stay on the PC platform. The price of PC's compared to UNIX workstations also influenced this choice.

As operating system, Windows NT was chosen, as it fulfils the requirements for an operating system for a control system. It is a multi user environment with good reliability, good network support (TCP/IP), good network security, and it runs concurrent processes with different priorities. Furthermore, it has a powerful graphic interface and good development tools. From the start it was the intention to use commercial software wherever possible.

2 HARDWARE MODEL

The system is based on the standard model for present days control systems - an Ethernet based system with distributed front-ends and console/client computers. There is a central server for domain management, file server and configuration database. The control system is connected to the in house network, so parameters can with the right software be accessed from everywhere. Since front-ends, as well as console computers, are PC's running Windows NT, it has been possible to use the same control system core software on all computers.

3 SOFTWARE MODEL

The overall model follows the so-called publisher/subscriber model. When a client application wants data from or control access to parameter on a device, it first has to subscribe to the parameter. When the value of the parameter is changed, the new value is automatically transmitted to the client.

The code is highly object oriented, with a high priority of code reuse and flexibility. Central objects are Data values, Address labels, Request, Data servers, and Devices.

4 SOFTWARE COMPONENTS

The system consists of the following components.

- **Database:** Used to store all machine/hardware dependent information, like parameter information, parameter addresses, conversion data, and display information.
- **Clients/applications:** Applications for machine operation like the general-purpose console. See section 8 for more detail.
- **Kernel:** Contains all the common, central parts of the control system. It is the same for all computers, be it front-ends or client computers. It includes the base class definitions for all ConSys components, a transport layer, and an interface to the database.
- **Data Servers (Acquisition Agents):** Responsible for transmitting the data values from the devices to client programs. Different filters are available, so values, for instance, are only transmitted if the data value has change more than a specified value, and/or a given time period has elapsed.

- **Devices:** Serves as the storage device for parameters and as the interface between ConSys and hardware drivers. Virtual devices are a special subset of devices, which do not have direct hardware access. They instead serves as storage for information parameters, or for intelligent data manipulation/reaction (for instance calculation of lifetimes, or automatic choice of multiplexer values, as viewers or cups are taken in or out of the beam. It is an important feature of ConSys, as opposed to many other control systems, that devices also can be a client, i.e. subscribe to any parameter on the system.
- **Drivers:** The NT drivers for hardware devices. Today most new hardware for PC's come with drivers for Windows NT.

5 DATABASE

The database is used to store all machine/hardware dependent information, like parameter information, parameter addresses, conversion data, and display information. Any ODBC compliant database can be used, and at ISA we use the Microsoft SQL Server.

The database is build by a number of related tables. The object-oriented structure of the control system cannot be represented directly in a relational database. Therefore, many tables representing a class hierarchy has some common fields used to initialize the base class and a number of generalized fields to initialize the inherited objects. A number of helper tables describe the actual meaning of the generalized fields for a specific ConSys class. With the aid from the helper tables an object oriented database editor has been created.

Most ConSys requests to the database use the same joint parameter request, build from a series of central database tables. The joint data request includes almost all information available for a given parameter, like addressing, interpretation/conversion information, data type information and display information.

6 FRONT-END HARDWARE

Today most parameters for the ASTRID storage ring are controlled through the small-scale bussystem G64. Originally a CERN standard the G64 system has been further developed in the computer department at the Institute for Physics and Astronomy (IFA) at the University of Århus. The system consists of a Z80 processor running at 4 MHz communicating with up to 10 IO cards via a 8 bit backplane. The IO cards available at IFA includes digital IO card, ADCs, DACs, and special processor controlled cards for autonomous function generation (synchronised ramping of parameters). The G64s are communicating with the PCs via a homemade (by the IFA computer department) PC card (called the PCDoct), with an onboard 80186 processor. Each PCDoct can serve up to seven G64s and the communication here is a 19200 baud serial connection. To interface with high level code a Windows NT driver has been developed.

Other hardware can be divided into two groups: Instruments communicating with a standard interface (Serial, GPIB, etc.), and instruments which are or require special PC cards (PCI or ISA). For PC cards a Windows NT driver is needed, which is the case for most new PC cards. For standard interfaces, there exist many good products, which also support Windows NT. We have chosen to use National Instruments products for many of these applications, because we have found these products to work well under NT, and they have good support.

7 COMMERCIAL TOOLS

The policy has been to use commercial tools wherever possible. This has the benefit of a fast development cycle and a professional look. The disadvantages are that one often runs into limitations as one is bound to one solution. The following products have been used in the development of ConSys:

- **Microsoft Visual C++ 5.0:** The primary development tool. The control system has a strong bound to MFC, Microsoft Foundation Classes. The communication protocol is based on pipes, an intrinsic part of Windows NT.
- **Microsoft SQL server:** Used to store all configuration data.
- **Microsoft Excel:** Used as editor for the DAF-files (function tables).
- **Component Works (National Instruments):** Active X controls, used for plotting and technical controls.
- **CVI (National Instruments):** Can be compiled into the ConSys code. Used for device drivers and some displays. Also used as mathematical library.
- **WWW Browser:** Used for documentation, and for status displays. In the future possibly also for control (Java).
- **Ragnarok (RCM) [3]:** Used for version control of our source code.
- **Visual Basic, Excel, Borland Delphi Pascal, LabView etc.:** Is supported through the ConSys API (a Windows DLL) for user programs.

8 EXAMPLES OF CLIENT PROGRAMS

- **Console:** The main program to display and adjust parameters. It is completely machine independent – all console pages are fully defined in the database. It has three views: A list view, which list the parameters sequentially, a graphics view with parameter values on top of a bitmap, and two control bars with "analog" display of two parameters. The set values in the control bar can be adjusted by two digital potentiometers, so an "analog" feel is obtained.

- **ReSto:** A general-purposes program to store and restore parameters on the system. Uploads parameters from one or more machines (or machine parts). Which parameters to download are specified in a so-called sequence file, which have to be applied to the setup file before download.
- **RampControl:** A general-purpose program to perform slow (<1 Hz) ramping of parameters. Ramp floating point values in either specified time or at specified rate. Also have options to set binary values, as well as possibilities for synchronisation of different ramps, including wait for a specific time or a given parameter value to be reached.
- **Datalogger:** A general-purpose program to log run-time data values to the database. It is possible to set up different logging conditions, so logs are only performed if specific criteria are fulfilled.
- **DAFLoader:** A program to load DAFs (Dfi Autonomous Functiongenerators). Reads an ASCII file generated by Excel (or any other suitable program) with the vector description of the function-generator ramps and load the vector tables into hardware. Have options for pre-processing the vector data, for instance for parabolisation of bends, and small energy scaling.
- **ConSysManager:** The system set-up program. With this program a system set-up file can be made, which specify the system database, and other system parameters. Likewise it is also here, it is specified which computers participate on which systems, as well as specification of which devices to load on individual computers, and the set-up of the devices.
- **DatabaseEditor:** A system program to ease set-up and maintenance of the system database. It gives an object oriented editing of the ConSys tables, based on description tables.

9 STATUS

The system is running with all of the ELISA parameters, and many of the ASTRID parameters. The system has been tested with all of the ASTRID parameters, but we are presently waiting for an update of user data-taking software to work with ConSys. For the coming electron run starting august 1998 the system will be fully deployed.

The system core (transport layer, data structures, database interface) is fully implemented, as well as the core client applications (Console, Store/Restore, Soft Ramps, Datalogger). A few important auxiliary applications (General plot program, Orbit correction, and an Alarm and Surveillance program) are still missing.

At present there is 2602 parameters defined in our database divided into 750 groups. (One group is typically one supply.) There is four dedicated frontends, three combined frontend/client computers, 6 dedicated client computers, and a number of office computers participating in the ISA ConSys control system.

10 CONCLUSION

The overall experiences with the system development have been very good, and generally there have been a high satisfaction with the chosen tools. The stability of the Windows NT operating system (both server and workstation editions) has been very good. The stability of the ConSys system as a whole has been reasonable. For the dedicated front-ends, which do not have client applications, the stability has been quite good, but client applications may fail in some cases.

The performance of the system is very good. The data rates are most often limited by display capabilities on console computers. Typical update rates are several Hertz.

More information on the system can be found on http: //isals.dfi.aau.dk.

REFERENCES

[1] J.S. Nielsen and S.P. Møller: “New Developments at the ASTRID Storage Ring”, WEP05F, These proceedings.

[2] S.P. Møller: “Design and First operation of the Electrostatic Storage Ring, ELISA”, THZ01A, These proceedings.

[3] PhD project of Henrik Bærbak, Department of Computer Science, University of Aarhus. See http://www.daimi.aau.dk/~hbc/Ragnarok.html