RELOCATION AND RECONSTRUCTION OF THE JÜLICH NEUTRON SCATTERING INSTRUMENTATION - CHALLENGES AND PLANS

M.Drochner, H.Kleines, F.Suxdorf, ZEL, FZ Juelich, Germany

Abstract

The Jülich research reactor (DIDO) will be shut down in 2006. It is planned to relocate a significant part of the instrumentation to the new neutron source in Munich (FRM-2). On this occasion older control systems will be replaced. Due to manpower shortage and time restrictions this comprises a serious challenge.

We are using a highly modular framework based on PCs and industrial control equipment which got its first tests at some of the instruments at Jülich. For deployment in Munich, we particulary attach importance to remote maintainability.

INTRODUCTION

A significant number of neutron scattering instruments has been established at the research reactor Jülich (DIDO, after a reconstruction called “FRJ-2”). Each one of them comprises a significant material value — construction of one instrument costs some millions of USD or EUR. In addition, the complexity of such a device and the conditions of operation within a neutron beam need specific know-how which has taken years to accumulate at Jülich. It was decided recently, for non-technical reasons, that the reactor will end operation in 2006.

On the other side, a new research reactor (FRM-2) was just constructed near Munich, as a much larger and powerful replacement for the aged “FRM-1”. For that reactor, instruments are needed to make use of the more intense neutron beam.

So it was decided to relocate a number of instruments from Jülich to Munich. This should of course happen as fast as possible, to be able to continue measurements soon after the reactor shutdown at Jülich. Since the manpower to do the work is limited, and the budget is too small to engage external help, some compromises must be made.

RELOCATION PLANS

Jülich’s instruments at FRM-2

The following instruments will be moved to the new Munich reactor:

- KWS1: Small angle scattering
- KWS2: Small angle scattering
- KWS3: Small angle scattering / reflectometer with focussing mirror
- DNS: Diffuse scattering, Time-of-flight
- Reflectometer (formerly HADAS)
- NSE: Spin echo
- RSSM: Backscattering Spectrometer
- SV-29: Time-of-flight

Timeline

The Jülich research reactor will be shut down in May, 2006. All but one instrument will be used up to that date. Then, deconstruction and transport will take place, followed by reassembly at the destination place. Assembly and cabling will (mostly) be done by technicians from Jülich, due to the limited budget, as mentioned above. The following is the estimated timeline for the individual instruments:

- Decomposition of instruments:
  - NSE first, starting fall 2005
  - Other instruments will run until reactor shutdown
  - KWS1, KWS2, DNS starting mid-2006
  - Reflectometer and SV-29 starting mid-2007
  - KWS3: no plan yet
  - RSSM is newly constructed, thus no decomposition
- Assembly of instruments at FRM-2:
  - RSSM: in progress
  - NSE: end-2005
  - Other instruments: until 2007

After the mechanical construction is ready, the control systems can be installed and tested. After that, software installation and commisioning can start. If everything goes well, the relocation will be complete in 2009.

Changes to instruments

The height of the neutron beam (relatively to the floor) is smaller in Munich than at Jülich, by about 30cm. This requires some serious changes to the mechanical construction to compensate for the missing space. Also, the neutron guides will be completely reconstructed to adapt to the different arrangement of the instruments. In addition, many devices controlling the neutron beam, as shutters, polarizers and choppers need to ne rearranged or replaced.

Because the instruments at Jülich are grown over decades, parts of the electronics, control systems and data acquisition are just outdated. In particular, there are control systems running on Motorola-m68k based VME processors on two instruments (NSE and SV-29) which will be replaced by PC technology, as described later. To be able to maintain that number of instruments with limited supplies of manpower, we are going to use the same control system hardware and software in all instruments. Where not already done, the control frontends are changed to use industry standard Siemens-S7 equipment.

In the process, the overaged power supplies in the NSE spectrometer will bw replaced too.

Detector systems must be tuned or replaced to handle the higher neutron flux delivered by FRM-2.

Where not already done, the control system software will be changed to follow the “Jülich-Munich standard”. This affects the instruments NSE, SV-29 and partly the Reflectometer. Also, changes for local networking, administration and data storage facilities are necessary to adapt to the environment at FRM-2.

It is important for us to allow remote maintainance of the control systems from Jülich, using the internet. Since the network at FRM-2 will not allow direct network connections from the outside, for obvious reasons, we will have to setup secured virtual private networks. For the PCs running Linux remote login and control is easily accomplished. For the S7 PLCs, special ethernet adapter modules will be used which allow the development tools provided by Siemens to access PLC programs and data.

CONTROL SYSTEM HARDWARE

The following key components were chosen due to experiences gathered in previous research and industrial activities in our laboratory. These universal building blocks are used wherever appropriate to save developer time, benefit from existing know-how and other synergy effects.

PROFIBUS is one of the major players in the fieldbus business, probably the leading one in Europe. The most widely used physical layer is a simple 2-wire copper cabling with RS485 signaling. Media access is managed by a token passing scheme. Rates from 9.6k up to 12M bit/s are defined. The maximum distance depends on the baudrate in use. With (commonly used) 1.5Mbit/s 200 metres can be achieved. Physical segments can be bridged or coupled by optical fiber connections to a multitude of this. The addressing scheme limits the number of connected stations to 126.

The protocol profile DP (Decentral Periphery) is designed for communication with simple slave devices which can even be implemented without a microprocessor. The peripheral devices define a logical input and an output data area (image), each up to 240 bytes in size, which are transparently mirrored into the master by the protocol engine. A major advantage of fieldbusses is the robustness designed into the protocols. For widely used busses as PROFIBUS various diagnostics tools are commercially available to help in tracking down problems.

Naturally, due to its industrial origin, PROFIBUS is well suited to connect drive control equipment as stepper controllers and position encoders. It is also a good choice to connect PLCs (Programmable Logic Controllers), since there are models available from various vendors with PROFIBUS-DP built in or simply to add.

To give a fair picture, it should be mentioned that the initial effort to get up and running a PROFIBUS is higher than for e.g. a system directly using PC bus (ISA or PCI) cards for periphery access. Once mastered, however, the PROFIBUS makes expansion and access to new devices easy, while the poor PC user still has to fight with interrupt conflicts and other scalability issues the PC architecture is plagued with.

We had to develop an own PROFIBUS master board with PCI interface because commercial suppliers did not provide a Linux driver nor technical documentation sufficiently to write one. It is based on a Motorola 68340 microcontroller. The PCI interface is a simple dual-port-RAM-style, with facilities to issue interrupts in both directions for handshake. The microcontroller is running a commercial PROFIBUS stack from Softing which was purchased as porting kit and adapted locally. The board exists in both PCI and CompactPCI formfactors.

There are hundreds of PROFIBUS-DP slave devices commercially available from a number of manufacturers. All these could be easily integrated into the control system. In existing applications, we are using the following:

- SIEMENS PLCs (S7 series). Critical logical interconnections are handled here, as mentioned above. A number of peripheral devices, connected either through the PLCs backplane bus or an independent fieldbus, can be controlled this way. This opens an access path to all industrial automation components supported as slave devices of the PLC.

- ET200 modular periphery system (SIEMENS), consisting of a base device with PROFIBUS connection and small I/O modules. This is a cost efficient way to implement analog and digital input/output ports. Stepper motor controller and encoder modules are available as well.

- Some high-end motor controllers directly attached to the PLC

- Angular position encoders by Heidenhain (using a gateway from EnDat Encoder Data to PROFIBUS-DP which is available from Heidenhain).

- EUROTHERM and JULABO temperature controllers

Being logically equivalent to the bus used on most of todays desktop computers, CompactPCI allows to deploy a variety of existing software in a mechanically and electrically more robust platform. Systems are available in 3U and 6U form factors. Since the primary purpose of the CompactPCI machine in our case is to house mechanically small interfaces like PROFIBUS, GPIB or serial lines a 3U system is typically sufficient.

The costs are still much higher than for a commodity desktop computer by about a factor of two , although they dropped significantly within the last years. The value one gets in exchange is a better quality of connectors and a design which allows the air flow to cool all components equally.

Only a fraction of the wealth of peripheral boards for desktop PCs is available in CompactPCI formfactor. The adapters we need are among them however, so this is not a serious limitation for us.

CONTROL SYSTEM SOFTWARE

A modular hardware construction set as described here will only develop its maximum gain if the software controlling it is similarly configurable. A client-server architecture is the obvious choice. High-level experiment control software is typically running on a workstation. It is adapted to the experiment setup and the principles of the measurement carried out. The programs dealing with the actuators and sensors on the other side are dependent on the specific hardware components used and their configuration. The client-server protocol connecting both sides should allow to maintain the software modules separately.

We have chosen the TACO framework developed at the ESRF (European Synchrotron Radiation Facility) in Grenoble (France) as the universal software interconnection. It is used for accelerator and beamline operation and for control of related experiments at the ESRF facilities and in some other projects.

TACO is a client-server framework built on top of SUN RPCs. RPCs are readily available on virtually every platform capable of TCP/IP networking today. TACO has been ported to a variety of operating systems like HP-UX, Solaris, OS-9, Linux, Windows NT, LynxOS and VxWorks. The hardware handling is done by so called device servers. Each device server implements a set of functions which depends on the kind of hardware it deals with. The set of individual functions, their (numeric) names, arguments and return values must be agreed on within a control system. (The ESRF provides some sample implementations which are useful as reference.)

To make its functions available to clients, the device server registers itself with the so called manager process. Clients then consult the manager to get the actual location (i.e. address of the machine it runs on). This allows for some flexibility in the distribution of services to frontend computers. TACO employs a simple database (the traditional BSD dbm which comes with most Unix systems) to store operational variables. Clients and servers can access the database (through an RPC connection, thus network-wide) to update the current state or to retrieve parameters.

To be integrated into a TACO environment, each piece of hardware has to be supported by some kind of TACO server. The level of complexity the servers functions operate on can be different from basic bus accesses up to complete experiment control commands like target positioning. The implementor is somewhat free to distribute functions between server and client layers. (TACO servers can also be clients of others, allowing more complex schemes of layered servers.)

On the client side, we are using two variants: Where flexibility is desired and no GUI is needed, the “Python” scripting language us used, with a small binary module implementing the TACO client. For rather static GUI applications, the “Qt” library is used, with TACO access provided by device specific C++ wrapper classes.

As operating system on the CompactPCI boxes, workstations and other computers we are using Linux exclusively.

CONCLUSION

We have established, together with the responsible persons in Munich, a modular toolbox which os powerful enough to control virtually all kinds of neutron scattering instruments. All high-level control tasks are fulfilled by PCs running the Linux operating system.

By limiting the number of different components in the various instrument control systems we achieve easier maintainability and an effective use of manpower.

So were are optimistic to meet the challenges of a relocation of eight neutron scattering instruments in a short timeframe.