OUTSOURCING, INSOURCING, AND INTEGRATION OF CONTROL SYSTEMS IN THE AUSTRALIAN SYNCHROTRON

Richard Farnsworth, with
Bryce Karnaghan, Wayne Lewis,
Andrew Starritt and Mark Clift
Australian Synchrotron.

Abstract

The Australian Synchrotron was built in less than four years and under budget with many subsystems outsourced. This presentation discusses some of the issues involved. It discusses the reasons for outsourcing, the approach taken, and some of the technical issues involved, including open source versus proprietary software, testing, training, collaboration and source control. The importance of a solid engineering approach, specification, interface, systems design and in-house ability are discussed. A discussion of engineering standards, both hardware and software, is presented. A balance of the positive and negative elements of the approach is put forward, and some suggestions for future projects run on similar lines are made.

INTRODUCTION

The Australian Synchrotron is a 3GeV machine with a storage ring circumference of 216 m and a current of 200 mA. The Australian Synchrotron achieved first light in June 2006 and has been open for users on five beamlines since April 2007. Construction commenced in July 2003 on a literally green field site with the awarding of a Building contract. One author (Farnsworth) is now the longest serving project employee and commenced in late April 2003. There was no local expertise and no major scientific or educational institution to provide staff or significant resources other than the land on which the Australian Synchrotron which situated. The decision to build the Australian Synchrotron was result of many years of lobbying by the scientific community. The Victorian state government funded the accelerator construction and a state government department, Major Project Victoria, the authority normally responsible for large-scale development and construction projects such as hospitals, stadiums, markets, large cultural centres and the like was nominated to build the Synchrotron.

The Major Projects approach was to take the significant components and outsource as much as possible, while building a small in-house team to manage the technical contracts and form the nucleus of the later operational facility. This approach works well for standard civil works and seems to have translated to the Australian Synchrotron successfully, although the tradeoffs are discussed in this paper.

Figure 1: Australian Synchrotron from the air.

MANAGEMENT APPROACH – STRUCTURE AND TEAM

The project deadlines were set by the Project and Technical directors before any other staff were hired. As the project was sponsored by the state government, the State departments set up a “Client” team which was responsible for media and communications, publicity, funding and all of the government functions. The project team were therefore insulated from the politics and able to concentrate on the project deliverables without distraction.

The project was run along with broad guidelines of

- Minimise staff;
- Maximise contracting and outsourcing;
- Meet tight deadlines;
- Defined acceptance criteria;
- Form small “Project/product teams; and
- Minimise and outsource risks.

These goals translated for the Australian Synchrotron

- A team of less than 50;
- The major subsystems outsourced;
- A lot of contractual support work;
- A need to integrate separate subsystems;
- Low risk solutions – leading edge, not bleeding edge; and
- Defined but not prescriptive standards

Outsourcing

All of the mechanical works, all the building, the injection systems, Linac, Booster and RF systems were outsourced as “turn-key contracts”. Like some, but unlike many other similar outsourcing ventures [1], controls were included in the scope of supply. This meant that contract standards for interfacing the control systems were required. A delicate contractual balance between specifying the exact required components and allowing the contractor freedom to produce a cost effective solution while assuming the risk for its correct operation. For example if we were to specify a brand of programmable logic device, or an operating system then the risk for ensuring that it is actually fit for purpose reverts to us.

Work and System Breakdown Structures

The various components of the synchrotron were analysed and a work breakdown structure was derived for all major subsystems. For the accelerator, this included storage ring RF, DC magnets, diagnostics, power supplies, vacuum equipment, and injection kickers. The controls work breakdown was then aligned with these systems and a corresponding system breakdown was derived. This was then used in ensuring that all of the systems were complete, especially in the case were the system spanned multiple work packages.

Controls Focus

Controls was recognised in having a special focus on integration. As a result the Controls team for a while was the largest team on the project reaching eleven members at its peak. This included the delivery of the accelerator and five beamlines. For a total project team size of just over fifty, this was a large team, but by comparative terms with other facilities it is small. This caused its own tensions.

Instant Friends

EPICS was chosen for the control system, being at once the most popular in the English speaking community and one of the oldest and therefore lowest risk. Because of its collaborative nature, entrance to the existing accelerator community was assured. It provided instant friends! The Australian Synchrotron is the major EPICS user in the country, although we did discover two other smaller users – one on a smaller accelerator and one on a observatory site. Training was shared with both the smaller users.

Shared Training

One of the best results was achieved by sharing training with the turnkey contractors. In this way we adopted similar standards ansd techniques without resorting to contractual arguments. It didn’t always work, so picking your contractors wisely also matters.

GUIs

The issue of Graphical User Interfaces (GUI) always seems to be the most problematical, maybe because it is the visible part of the control system. The Australian Synchrotron chose to work around this by defining standard tools for the contractors and to eventually work towards integrating these tools into a facility framework by slowly replacing them. To achieve this and to meet the accelerator physicists requirements, a commercial MS Windows based integrated development environment (IDE) was used and the resultant windows executable was run in a Linux environment using WINE. This actually worked much better than it sounds, response is fast, and functionality is essentailly identical across the major facility platforms which are MS Windows and Linux. Additionally the Linux operator interface machines are incredibly stable, in general only failing when power is removed. The advantage of coding the GUI’s in house, rather than using standard packages (EPICS EDM and MEDM for example) is that the GUI’s can be very specifically tailored to the user requirements. Additionally, some of the complexity of database can be placed in the easier to program GUI. Some sample screen shots are shown in figures 2,3 and 4.

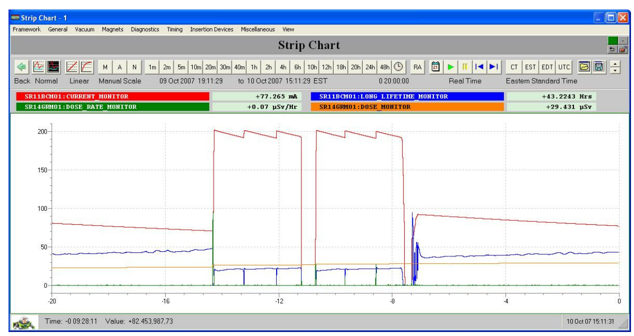

Figure 2: Integrated GUI Strip chart and archive viewer.

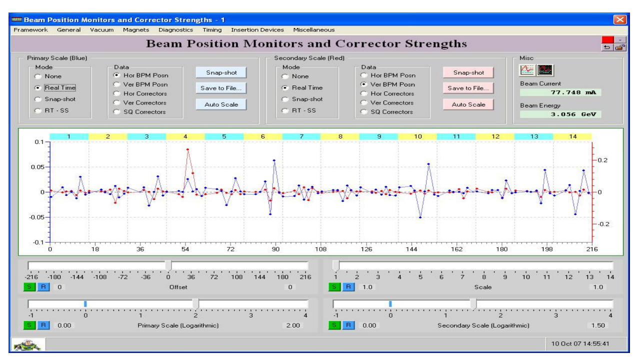

Figure 3: Beam position monitors.

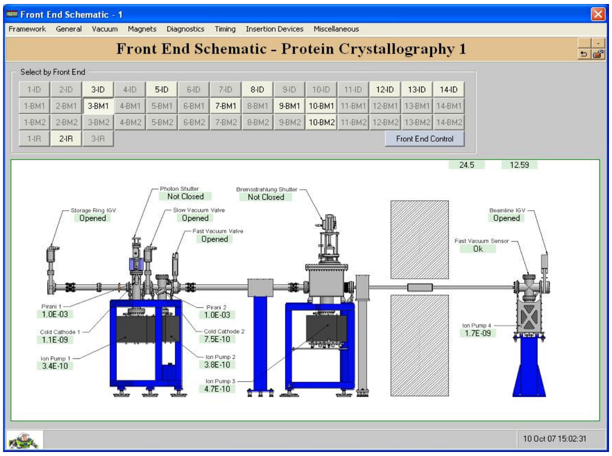

Figure 4: Front End schematic.

Hardware Selections

The selection of hardware for the systems was made with a knowledge of the size of the team, the availability in the local market of the hardware generally used on similar facilities and EPICS interfacing issues. At the time of selection, EPICS for Linux was recently released, thus making PC hardware viable. A proper port of EPICS to a commercial real time Linux seemed a sensible alternative to the traditional VME and VxWorks hardware/software combination.

An equipment protection PLC for inhouse works was standardised upon and the use of the Modbus EPICS driver as a simple ubiquitous interface solution where possible was encouraged. Safety PLC's were selected with this in mind also. This meant that most hardware was PC based, albeit much is server class. Where VME is required, PCI to VME bridge technologies allow PC class CPUs to drive ported software drivers. PC104 form factor hardware and solid state disks provide the reliability for certain other items.

Operating System Selection

The use of specialist real-time operating systems can be important; the traditional Synchrotron VxWorks dominance was being challenged by other operating systems such as RTEMS. In a somewhat bolder move, a variant of a Redhat distribution was settled upon for both non real time applications and real time applications. The real time work was provided by "Redhawk" propriety real time systems provided by a commercial supplier. Later non realtime IOC's on PC104 hardware and standard PC hardware was used. As a result the Australian Synchrotron is probably the first facility of its class to use Linux operating systems for all critical data collection purposes and all user machines. As it turned out, we probably could have avoided real time operating systems entirely and may do so in future.

Timing Systems

One of the hardest part of synchrotron real time control is the timing system. This has been manufactured in-house in the past, but relatively recently commercial off the shelf timing systems have become available. The Australian Synchrotron started with a commercial analogue trigger generator and a number of discrete delay generators, but recently has moved to digital event generators to reduce the timing jitter and allow the beamlines to receive signal as required. The master oscillator is a standard off the shelf unit as are RF distribution amplifiers.

Conventional Facilities

The conventional facilities include building controls, compressed air, HVAC systems, cooling water and the like. These were specified very early on by the architects and civil engineers before the controls team was established. They use conventional building industry techniques for their interfaces. The EPICS interfaces were not required as a part of those facilities, and the effort required to pass data to the EPICS accelerator systems has been substantial.

External Peer Reviews

There were several external peer reviews by controls leaders of other similar institutions and they provided useful guidance and confidence. They consisted of two experts each time– one from a European and one from a North American lab, and were able to assure management that the controls project was going to be successful. They also were able to provide minor corrections and good direction, especially for some of the trickier implementation details. After the project was nearing completion, a further review was made by an Australian controls leader from the Australian neutron source

Engineering Approach

A standard systems engineering approach, that of identifying the functional requirements, sorting our the interfaces, producing design documents, creating the code and sourcing the hardware, testing against the requirements was very important. An incremental delivery approach was taken; with the minimum required delivered first. This was known in the scheduling as the "bare-bones" approach. This was both consistent with the low risk approach and ensured that problems were dealt with as early as possible.

Interface Engineering

Identifying the interfaces was very important. Significant effort was put in ensuring the required flow of information was well known before the implementation started. A consultant systems engineer was engaged by the facility (for all disciplines, not just controls) and the interface design engineering methodologies were adopted by all teams. Techniques included block interface diagrams, sequence diagrams, collaboration diagrams, use cases and flow charts.

Formal Internal Peer Reviews

An important part of the engineering process adopted was that of formal peer reviews for the design of each system. They consisted of a “Pack” of information distributed amongst reviewers before a formal review. Typically these had the requirements, a traceability matrix, the sequence diagrams and design specifications. A peer review was held and formal notes taken for any corrective action. The reviewers were across all disciplines that may have an interest in the subsystem.

The Build System and Source Control

On of the more novel aspect of the project was the effort placed in creating a “Build” for the accelerator. It was decided very early on to make a single version of the accelerator software and to control very tightly exactly what software is deployed. This includes all EPICS code base, but also extends to PLC code. This level of control is very useful when delivering incrementally, and regularly all the changes and bug fixes for the previous period are collected and a complete compile, authorise and release cycle is made. For the accelerator this is weekly, for beamlines this is as required. Emergency changes are handled via a “patch” process. Authorisation is from a two or three person change control board, and every change can be reversed. The Australian Synchrotron accelerator has deployed over 125 builds to the accelerator (approximately weekly) .

LONG TERM OPERATIONS

The Australian synchrotron has been in operation for over twelve months and delivering photons to beamlines for over six months. Users are starting to appear and the machine has achieved over 95% uptime over the last six months. The control systems have proved very reliable from day one, and the beamlines are likely to achieve a similar reliability.

Insourcing

The ongoing maintenance and support for the systems provided by others is a continuing effort. The acceptance tests in general included a compile and deploy test – so we could be sure we got all of the source code. As time progresses, the importance of having all source code in our configuration repositories increases. Bringing the various outsourced systems into a common environment is also important as it reduces the long term effort needed to maintain the code base. Some of the hardware differs from systems to system, and replacing the hardware with common components will decrease the spares count and increase the ability of staff to quickly deal with any hardware issues if and when they arise.

Source and Configuration Controls

We use Perforce [2], a commercial but free for Open source developers and Bugzilla [3] as the source control and bug/enhancement tracking tools respectively. While the use of Perforce is unusual for the community, it has performed extremely well and suits our needs. Bugzilla has similarly proved to be very worthwhile and allows tracking of all bugs and enhancements for later implementation and review.

Databases

The controls team has not made major use of relational database technology, but small database have been set up for cable, component and a few other assorted minor uses. We tried early on to use a component database. Our concept was to make it small enough so that each entire database could be stored and copied with each build thus eliminating the need for a database with an in built version control mechanism as it would be eternally imposed. As time progressed, the project team didn’t use the component database effectively and it is no longer maintained.

Testing and the Released Cycle

A full separation between development, which occurs in the staff offices and the testing environment, the Controls “integration” laboratory exists. For every release a series of regression tests are performed on existing code and a selection of hardware and simulators are used to ensure that each release is unlikely to cause problems upon release. Where the test equipment is unavailable in the lab, then sometimes a patch release is created and tested live in machine studies time. The GUI is an exception to this, typically a patch GUI is available to the operations room.

The Facitliy status is web enabled at ref [5].Figure 5 shows the facility status as released weekly.

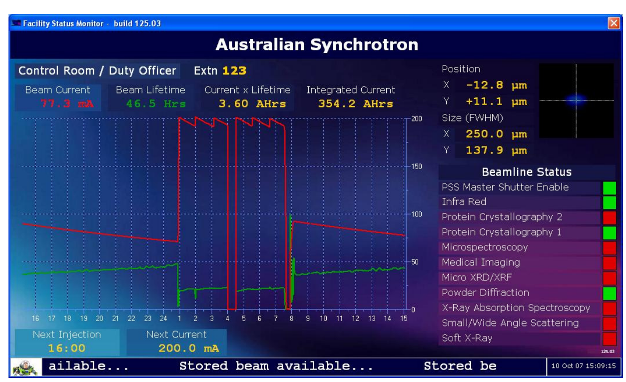

Figure 5: Facility status monitor.

SUGGESTIONS FOR FUTURE PROJECTS

The following suggestions are made for future projects

- Get controls in early as you can,
- Get conventional facilities as part of the systems
- Standardise before any contracts
- Copy instead of reinventing where possible
- Don't take too many risks, and manage those you do.
- Set realistic, milestones and goals
- See what works elsewhere
- Get good people and let them free.

Farnsworth Law of Control Systems

Finally, a statement of some experience on the way:

"Controls gets squeezed most as time runs out." So be as prepared as possible to minimize the pain with strong engineering and software design, configuration management and change control is important. Testing and release and reversion are critical. Always have a plan B and C and D, be prepared to use them, but hope you don't have to.

REFERENCES

[1] D.P Gurd et al., "First experience with handover and commissioning of the SNS controls systems", Proceedings of the ICALEPS2003 Gyrnongju, Korea October 2003 p. 23

[2] www.perforce.com

[3] www.bugzilla.org

[4] www.winehq.org

[5] www.synchrotron.com.au