DEVELOPMENT OF PHOTON BEAMLINE SOFTWARE AT DIAMOND

N.P. Rees[1], W.C. Pulford[1], M.A. Norbury[1], P.J. Leicester[1], E.L. Jones[1], M.T. Heron[1], P.N.Denison[1]
[1]*Diamond Light Source, Chilton, Didcot, United Kingdom*

ABSTRACT

Diamond Light Source is pursuing an ambitious programme of opening 8 beamlines to the user community in the first few months of 2007. All the control software for the beamlines is being developed using a common framework of EPICS for lower, engineering-level control and GDA (Generic Data Acquisition, a Java/Jython based system developed at SRS Daresbury) for higher level sequencing and the scientific user interfaces. We are introducing a development process based on the ESA small software project standards to help provide focus and a level of quality assurance and, in doing so hope to improve the requirements capture process thereby giving a clearer direction for the system realisation.

Overall, we are striving for as much commonality as possible across all Beamlines, by consistently applying EPICS as the interface to the technical systems and providing the Beamline Scientist and Users with full control over beamline operations through the GDA system. This presentation touches on all these areas, but focuses on the lower-level EPICS development.

INTRODUCTION

As illustrated in Figure 1, Diamond Light Source [1] has an ambitious development schedule for beamlines, with 8 being available within the first 4 months of operation, and a further 14 being commissioned over the following 4 years. Beamline development is managed by a team consisting of a Principal Beamline Scientist, a second Beamline Scientist, plus about 50% each of a Mechanical engineer, Electrical Engineer, Controls Software Engineer and Data Acquisition Scientist. From a controls perspective, 3 of the first 8 beamlines are nearly identical Macromolecular Crystallography beamlines and this simplifies the development, but not commissioning, process.

We have created a development process loosely based on common system engineering processes (for example, the European Space Agencies ECSS-E-40 Part 1B) to manage the balance between risks and requirements (see Figure 2). After the Technical Design Review (Roughly equivalent to ECSS-E-40 Detailed Design Review), development is split into two areas – external development managed through tender exercises and internal development by in-house teams. Most of the controls work is developed internally, and central to this is a concept of “feature sets” which is a simple set of enumerated requirements that are developed in conjunction with the beamline scientists. These encapsulate the essential features of a particular beamline, without the need for extensive background descriptions.

2006 will be a busy year for Diamond with the first half of the year dedicated to commissioning the storage ring and hardware integration and testing of the first 8 beamlines. The second half of the year will bring further refinement of the machine whilst the beamlines are being commissioned, ready for the first scientific users to arrive in early 2007.

BEAMLINE SOFTWARE ARCHITECTURE

We have adopted a two layer architecture for the beamline software, the upper layer being largely Java based, and the lower level C based using the Experimental Physics and Industrial Control System (EPICS)[3][4]. The interface between the two layers is defined in terms of XML files which are generated from information in the EPICS database, and which, in turn, instantiate Java objects in the upper layer software using CASTOR[5]. Whilst there is a certain amount of duplication between the layers, the upper level user interfaces are designed for the typical scientific user, and the lower level ones for engineers and Beamline Scientists performing engineering functions such as calibration, alignment and fault diagnosis.

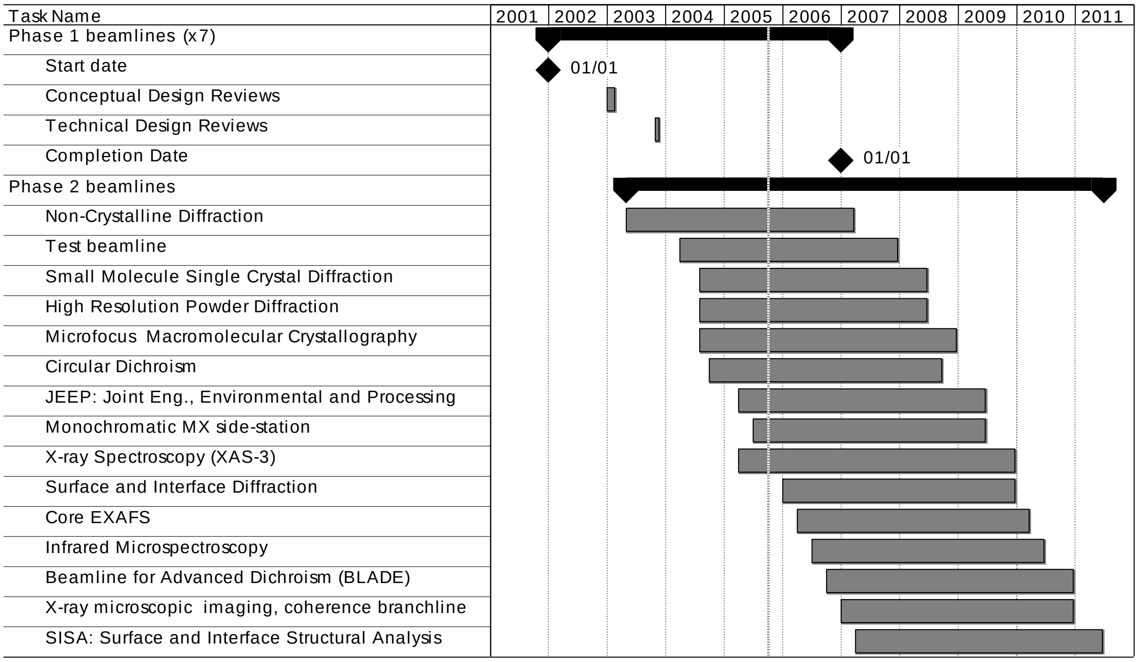

Figure 1: Schedule for development of Diamond beamlines. The 7 phase 1 beamlines are all on the same schedule and their start, review and completion dates are shown. The Phase 2 beamline bars show the start and completion dates.

Upper level architecture

The upper level software not only handles normal beamline science operations, but also user liaison office functions (application processing and time allocation etc) user authentication, data processing (including GRID interfaces), data management and archiving. However, the core of the operational system is based on the Generic Data Acquisition (GDA) project[6][7] originally developed at the Synchrotron Radiation Source (SRS) Daresbury, and now being jointly developed by them and the Diamond Data Acquisition Group (DAG). The essential items of the GDA architecture are illustrated in Figure 3. Users have access to a variety of user interface clients (only a scripting IDE and traditional controls user interface is shown here, but Matlab, IDL and other interfaces also exist). These clients interface with two servers on each beamline

1. A scripting server to coordinate beamline operations. This executes code in the Jython variant of the popular Python scripting language and this handles all beamline and experimental co-ordination.
2. An event server to monitor asynchronous events coming back from the hardware. This can forward data to any of the clients.

There is also Data Handler to accept the incoming bulk detector data and ensure it gets to disk with the correct metadata and beamline state information obtained from the event server. All the inter-module communication utilizes the Common Object Request Broker Architecture (CORBA) and its object management is by a central object server.

Much of the recent work in this system has been modularizing the architecture to pull out the core server modules so that multiple clients can interact with single instantiations of the servers, and also enhancing the EPICS interface. This work is now complete and the focus is turning to generating clients for our specific user requirements.

Lower level (EPICS) architecture

The lower level system builds on the work that has already been done by the EPICS community throughout the world – particularly that of the Diamond machine control system[8] and the synApps software developed at Argonne Photon Source[9]. We have also been assisted by commercial input, with a number of companies undertaking to provide hardware with EPICS control systems. In particular, Oxford Danfysik (along with their sub-contractors, Cosylab) has provided a complete

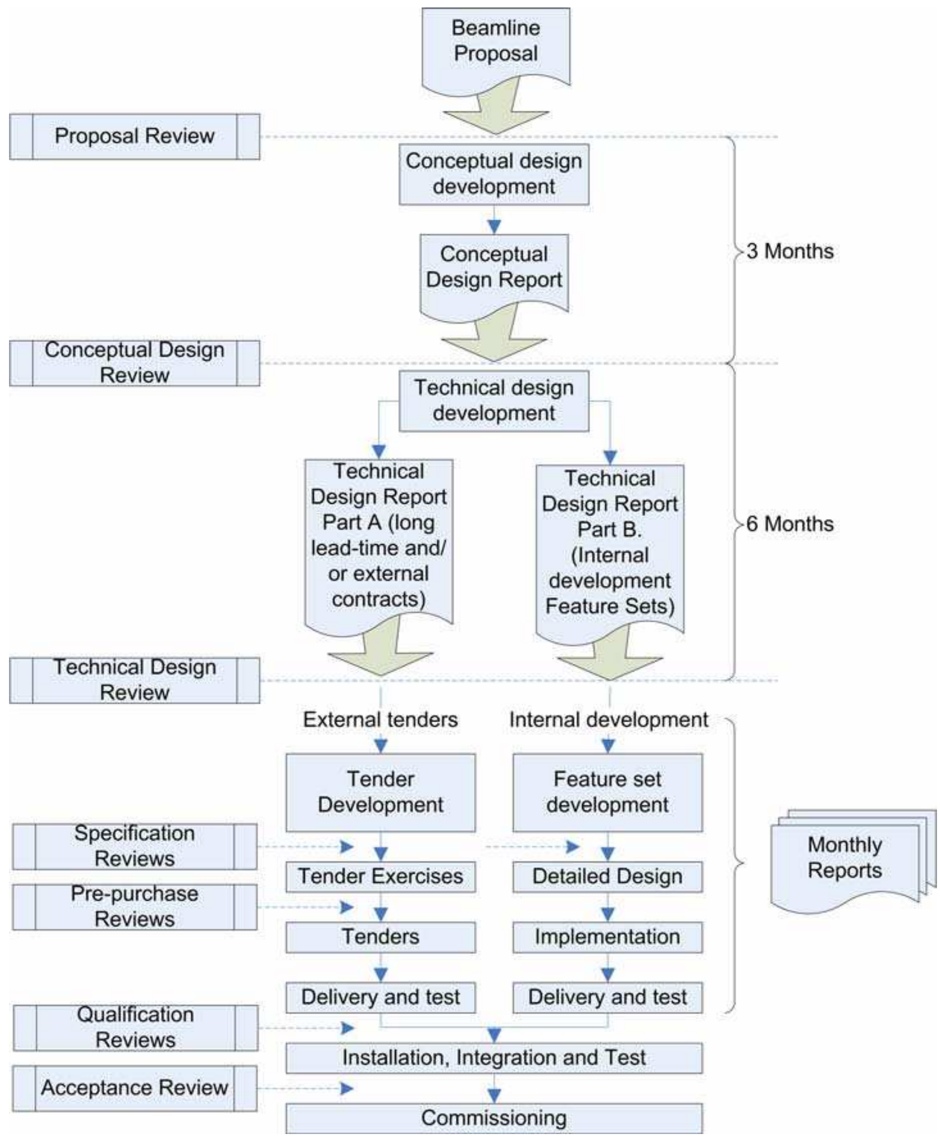

Figure 2: Outline of Diamond Beamline development process

solution, including EPICS control system, for the optical components of the three Macromolecular Crystallography beamlines (see Figure 4).

The basic aim is to provide detailed user interfaces for every signal available on the beamline, plus the normal EPICS archiving and alarm handling clients. The development is coordinated across all the beamlines and so we are developing common modules for slits, Diamond Diagnostic Units, focusing mirrors, Double Crystal Monochrometers etc. The goal is to have prototypes for all these modules developed and complete beamline simulations available to the Beamline Scientists by late 2005. This will allow them to give feedback on the interface design, and the functionality offered before hardware integration starts in early 2006.

EXAMPLES OF EPICS DEVELOPMENT ON DIAMOND BEAMLINES

Embedding of metadata

One problem that has confronted EPICS developers in the past has been how to coordinate the development of databases alongside device structure, alarm handling and archive configuration. We have decided to embed client metadata in the EPICS database definition files and have a number of scripts to extract these data in order to build client configuration files. This keeps all database configurations in one place and the client configuration files can be generated as part of the build

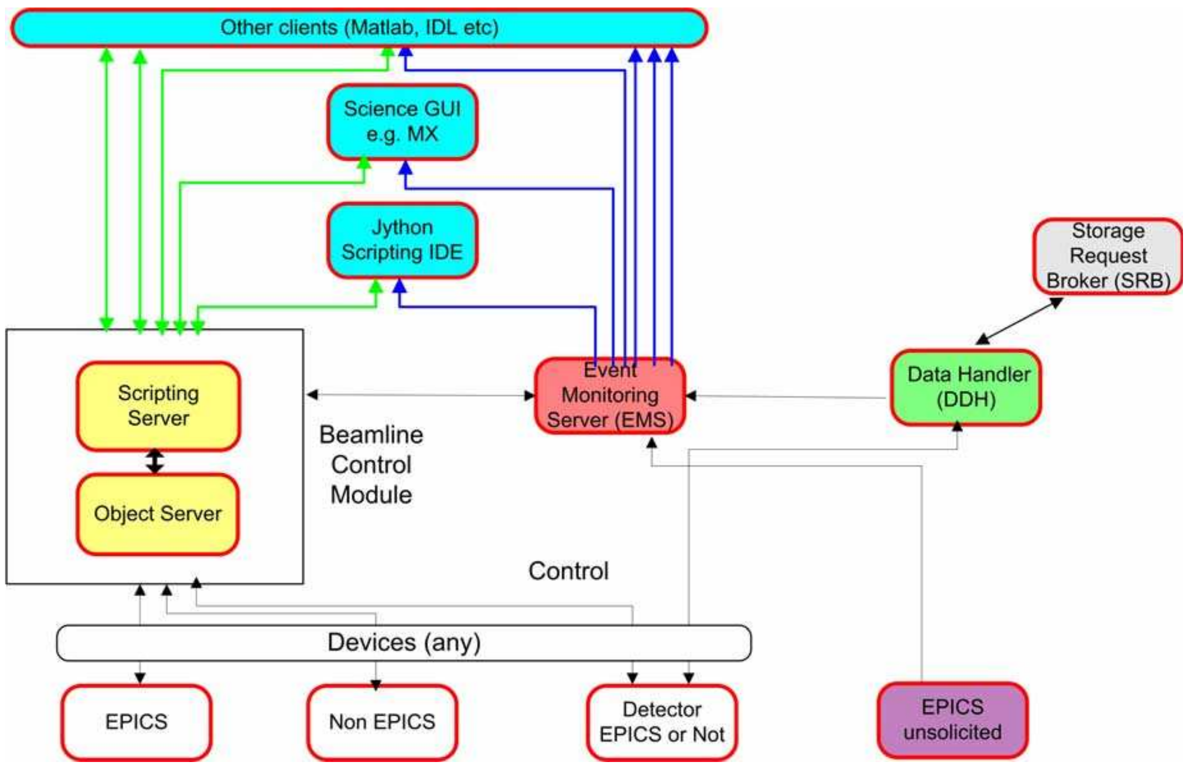

Figure 3: GDA Architecture

process. This process has been pioneered in the machine controls system, but has been extended by beamlines to provide the additional information for the GDA interface files. At the moment the information is embedded in comments, but we would like to work towards defining this metadata structure more formally, so that developments such as these can be coordinated throughout the EPICS community.

Motion control, EPICS motor records, and synchronization

Diamond has adopted mostly Delta Tau motor controllers on beamlines, instead of the OMS controllers used on the machine. The primary motivation for this was the extra capability of the Delta Tau controllers required for some aspects of beamline control. One requirement that we have from a number of beamlines is the need to synchronise multiple systems during data taking. Examples are continuously scanning the insertion device and monochrometer in EXAFS experiments, or synchronising detector readout with diffractometer motion. In particular, scanning the monochrometer in a smooth calibrated energy scan requires a controller that can both move multiple axes in synchrony, and also implement lookup tables in one axis dependent on the position of another axis (for example, to take out pitch and yaw errors in a mirror whilst translating it perpendicular to its surface). Unfortunately, EPICS has no standard solution to these sorts of problem, principally because of problems in the motion control interface.

EPICS motion control has either been via the motor record, which is a standard interface, but not extensible, or through custom written drivers interfacing to primitive records. Neither solution provides a generic interface to complex motion control requirements, and the high level of functionality available in modern motor controllers. Hence, we are implementing a C API below the motor record which motor control device drivers can support, which also supports primitive records and which can be extended to provide enhanced functionality, if and when required. We hope this will alleviate the long standing problems EPICS has had in this area.

Diagnostics, Vacuum, Personal Safety and Equipment Protection Systems

Virtually all software in these areas use EPICS modules that have been developed for the standard Diamond systems for the Diamond accelerator – the exception being slit diagnostics which have no

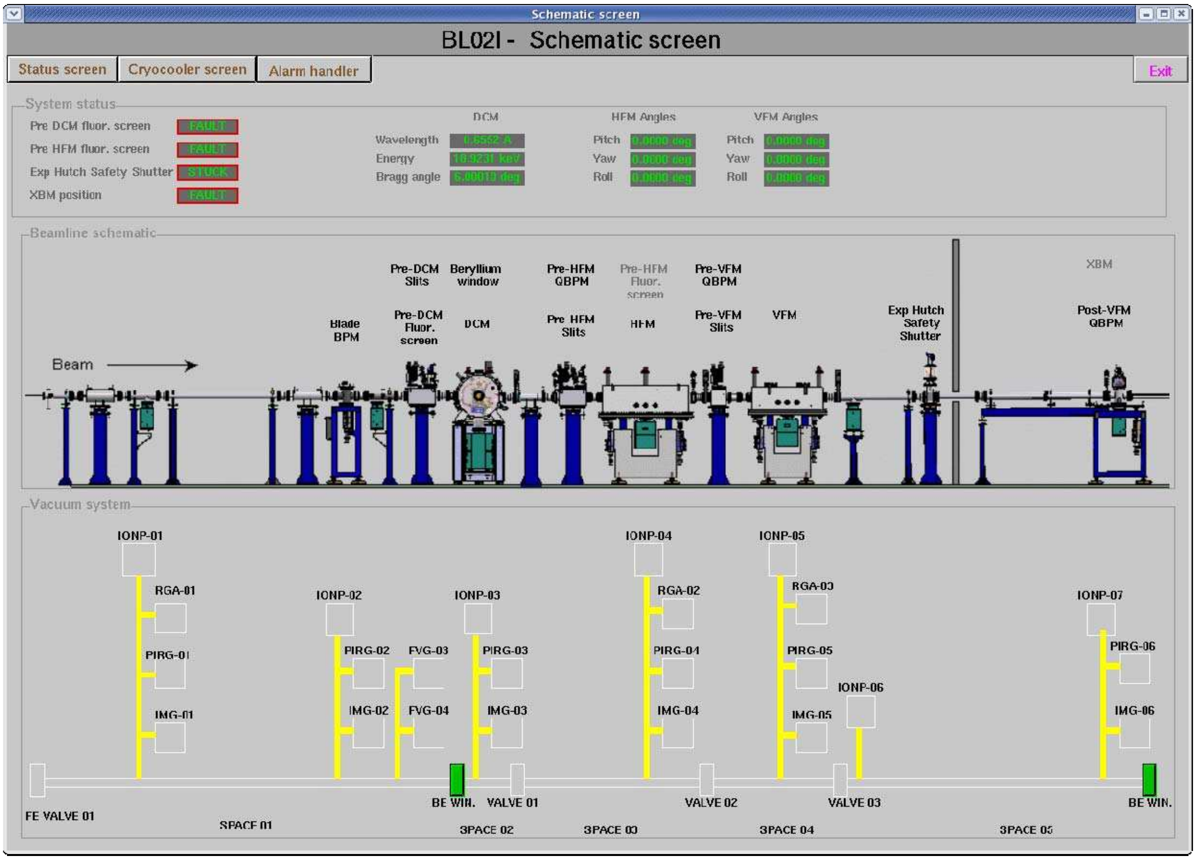

Figure 4: MX Beamline synoptic screen

real analogy in the electron beam. However, we use the same vacuum modules (using serial interfaces to cold cathode and Pirani vacuum gauges, residual gas analysers), equipment protection modules (utilising Omron CJ1 PLC's), IEEE 1394 (Firewire) video camera modules (Point Grey Research Flea cameras) and underlying Personal Safety Systems.

CONCLUSIONS

The Control Systems for the Diamond Beamline are progressing well, and should be relatively mature by the time the first users arrive at Diamond in 2007. We have used a combination of well tested techniques and tools plus targeted development in certain key areas to mitigate risks and ensure that the scientific requirements are met.

REFERENCES

[1] R.P. Walker, "Progress with the Diamond Light Source Project", EPAC 2004, Lucerne, Switzerland, July 2004.
[2] European Cooperation for Space Standardization, "ECSS-E-40 Part 1B Space Engineering Software – Part 1: Principles and requirements", Noordwijk, The Netherlands, 2003.
[3] L. Dalesio, J. Hill., M. Kraimer, D. Murray, S. Hunt, M. Claussen, C. Watson, J. Dalesio, "The Experimental Physics and Industrial Control System Architecture," ICALEPCS 1993, Berlin, Germany, Oct 1993.
[4] The Experimental Physics and Industrial Control System. http://www.aps.anl.gov/epics
[5] The Castor Project. http://www.castor.org
[6] M.J, Enderby & W.C. Pulford, "The Generic Data Acquisition Project", NOBUGS 2004, Villigen Switzerland, October 2004.
[7] The Generic Data Acquisition Project. http://www.gda.ac.uk/
[8] M.T. Heron et al., "Progress on the Diamond Control System", ICALEPCS 2003, Gyeongju, Korea, October 2003.

[9] Advanced Photon Source Beamline Control and Data Acquisition Group: synApps. http://www.aps.anl.gov/aod/bcda/synApps/index.php.