WEDNESDAY WELCOME BY JOSEF HORMES

J. Hormes, CLS, Saskatoon, Saskatchewan

Abstract

This is the welcome power point presentation presented by Josef Hormes, Executive Director of the Canadian Light Source.

PRESENTATION ONLY

WEDNESDAY WELCOME BY ELDER MATIAS, PCAPAC 2010 CHAIR

E. Matias, CLS, Saskatoon, Saskatchewan

Abstract

This is the Welcome presentation made by Elder Matias on Wednesday, October 6 welcoming all the delegates to PCaPAC 2010 held in Saskatoon, SK.

PRESENTATION ONLY

CONTROL SYSTEM STUDIO WORKSHOP REPORT

J. Hatje, DESY, Hamburg

Abstract

This is the Control System Studio Workshop Report as presented by Jan Hatje (DESY / MKS-2)

PRESENTATION ONLY

USE OF THE CELL ACCELERATOR PLATFORM FOR SYNCHROTRON DATA ANALYSIS

J. Qin, M. A. Bauer, N. S. McIntryre
The University of Western Ontario, WSC-143, UWO, London, ON. N6A 5B7, Canada.

Abstract

The analysis of synchrotron-based Polychromatic X-ray Microdiffraction (PXM) data has been used by scientists and engineers to understand elastic and plastic strains in materials on a micro or nano scale. Such experiments generate hundreds or thousands of images where the analysis of each image often entails intensive computations- a challenging task. As well, in the past, the speed of such computations has made it difficult to obtain feedback on the experimental results in near real time. This has constrained researchers from making critical decisions on direction subsequent experiments should take based on the results in hand. In order to improve the analysis performance of PXM images, we have investigated the use of parallel analysis schemes. This paper reports on the design and implementation of accelerated PXM analysis software. It has been developed on IBM PowerXCell 8i processors and Intel quad-core Xeon processors. A substantial improvement in processing speed has been obtained to the extent that it should be possible to obtain results at the same rate as they are produced on the VESPERS beamline at the Canadian Light Source (CLS) Synchrotron (~1 Hz).

INTRODUCTION

The development of high-energy PXM as a non-destructive method to determine elastic and plastic strains has been ongoing for the past decade [1-5]. The data generated in PXM experiments can consist of a large number of 2D digital images. Once these images have been generated from an experiment, ideally, it is expected that data can be processed at a same speed level as data is collected.

There are three major procedures involved in PXM data analysis, including peak searching, indexing and strain calculation. Briefly, *peak searching* attempts to extract useful information about intensity points (peaks) from an image to be used as input for the next two procedures. The *indexing* procedure takes the output from the *peak searching* procedure and generates the structural information about the sample material, e.g. the orientation of a crystalline lattice plane from which a diffraction spot is generated. Based on the indexing results and peak information, the strain analysis procedure then produces strain tensors in the sample. Based on the indexing results and strain tensor information, an orientation map and a strain map can be generated for the entire scanned area from which all PXM data were collected.

There are some existing software packages for PXM data analysis, such as the 3D X-ray Micro-diffraction Analysis Software Package at APS in Chicago which was developed at ORNL[6], and X-ray Micro-diffraction Analysis Software (XMAS) at ALS in Berkeley[7]. The common feature of these two packages is that they both are Windows-based software with a frontend interface implemented in Interactive Data Language (IDL) [8] and some backend procedures implemented in Fortran. Both can process a large amount of PXM data sequentially, i.e., step by step and one by one in sequence. This is a very time consuming process, and it usually takes days to finish processing a set of data collected from one PXM experiment. However, synchrotron time is valuable and it is often difficult to get a scheduled beam time. Data analysis using existing software means that researchers must complete the analysis following their time on the synchrotron. Faster analysis could help researchers make decisions on subsequent experiments during their synchrotron session and gain significant insight into the materials that they are studying.

In this paper, we introduce the development of an accelerated software for PXM data analysis, so called Fast Online X-ray Micro-diffraction Analysis Software (FOXMAS). It has been developed on a Cell accelerator platform comprised of Intel and IBM Cell processors. The software developed and the system it runs on makes it possible for PXM data to be processed in "near-real time", that is, nearly as fast as it is being produced. A description of the platform, the development approach, some performance evaluations, conclusions and future work are reported.

CELL ACCELERATOR PLATFORM

The target Cell accelerator platform, called *Prickly*, is one of the clusters in SHARCNET [9]. It is a heterogeneous High Performance Computing (HPC) system consisting of one head node for hosting user logins and a chassis with 12 Linux cluster blades providing total 160 computing cores. Among the 12 blades, four blades are Intel blades and the other eight are IBM Cell blades. On each of the Intel blades, there are *two* quad-core Xeon E5420 processors running at 2.5GHz with 8GB of memory. Each of the Cell blades contains *two* PowerXCell 8i processors, so called Cell processors, running at 3.2GHz with 16GB of memory. Blade interconnection is achieved through Gigabit Ethernet.

Unlike traditional multi-core processors which are homogenous, such as those on Intel blades, the Cell processor itself has heterogonous multi-cores [10]. It employs two types of cores optimized for different kind of tasks. Each Cell processor has nine cores, i.e. *one*

*Work is part of Science Studio project supported by CANARIE http://www.canarie.ca/

FAST ORBIT CORRECTION AT THE CANADIAN LIGHT SOURCE

C. Payne, Canadian Light Source, Saskatoon, Saskatchewan, Canada
D. Chabot, Brookhaven National Laboratory, Upton, New York, USA

Abstract

Correction of the electron beam orbit in the storage ring at the Canadian Light Source has historically been implemented using a correction system capable of only moderate update rates. Over the past several years work has been undertaken to reduce orbit perturbations and improve end user synchrotron beam quality by reimplementing the correction system and enabling orbit corrections several orders of magnitude faster. This paper will describe the implementation and migration of the orbit control software from the slow correction system to the fast system.

CLS ORBIT CONTROL HISTORY

The present orbit control system in use at the Canadian Light Source (CLS) is described in [1]. This system is an intermediate step between the previous orbit control system [2] and the system described in this paper. The design limitations of the current system as impetus for change are worth mentioning and will be briefly discussed below.

DESIGN LIMITATIONS OF THE EXISTING SYSTEM

There are several key limitations inherent to the orbit control system in use at the CLS at the time of writing.

Update Rate Limitations

The main motivational factor to migrate to a new system is maximum possible update rate. The present system is only capable of quasi-static update rates on the order of 0.1Hz. Although this has been successful at sufficiently maintaining the orbit of the CLS Storage Ring (SR), faster corrections rates are desired to further reduce orbit perturbations.

Serial Application of Orbit Corrections

The Matlab [3] program, CLSORB [4], applies corrections in a sequential manner. This results in undesirable orbit perturbations as the corrections are applied one after another around the storage ring. Distribution of corrections from CLSORB through the Experiential Physics and Industrial Control System (EPICS) [5] produces additional non-deterministic behaviour due to network and computer latencies. Delivering corrector magnet setpoints in this way also has the effect of accruing hardware delays on a per-channel basis, instead of per power-supply controller. This adds significant delays to the process of setpoint distribution, and is a major factor governing the achievable rate of the orbit control system.

HARDWARE

The hardware involved in the orbit control system is shown schematically in Figure 1. The hardware consists of:

- An Industrial 3GHz x86 PC IOC with 1GB or RAM running Real-Time Executive for Multiprocessor Systems (RTEMS) v4.10 [6]
- Four (4) Versa Module Eurocard (VME) Crates [7]
- Four (4) pairs of Struck Innovative Systems (SIS) PCI/VME 1100/3100 cards, for connectivity [8]
- Four (4) Analog to Digital Converter (ADC) VME cards (ICS-110BL sampling ADC)
- Eight (8) Digital I/O Modules, model VMIC 2536 D-I/O, 2 per VME crate used to control corrector setpoints

In addition, hardware independent of the fast orbit control software system:

- Beam Position Monitors (BPMs) which produce analog signals in proportion to the position of the electron beam passing through them. [9]
- Bergoz BPM Modules which sample the BPMs to produce analog x-y coordinates of beam position. [10] These signals are then digitized by ICS-110BL VME modules and the data processed by the RTEMS IOC.
- OCM Power Supply Controllers, VME based devices interfaced with the OCM power supplies. There are 48 vertical and 48 horizontal orbit correctors, contained in a bulk IEPower [11] chassis. It should be noted that although setpoints are via the fast, VME interface, the power supply feedback is exclusively via serial interface.

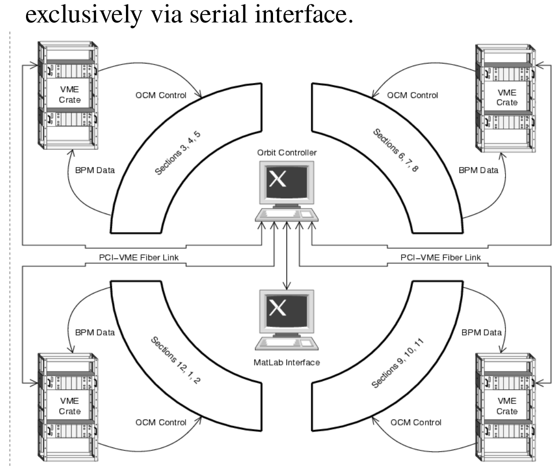

Figure 1: Hardware Overview

HIGH-LEVEL APPLICATION PROTOCOLS

N. Malitsky, BNL, Upton, Long Island, New York

Abstract

The report presents an overview of the different communication and serialization approaches in the context of the high-level client-server accelerator application environment. The list of examples includes CORBA, CDEV, Google's Protocol Buffers, DDS Extensible and Dynamic Topic Types, EPICS PVData, and others.

PRESENTATION ONLY

WHAT'S BEHIND AN ACCELERATOR-CONTROL SYSTEM?

Rüdiger Schmitz, Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany.

Abstract

A control system has a lot of features, some essential: e.g. a set of application programs. The infrastructure they need in order to run so that the operators at least be able to switch the accelerator on and off.

Graphical User Interfaces, intelligent control algorithms or data acquisition methods are obvious, but other features (not as obvious) also require considerable manpower and should not be underestimated. They have a major impact on the availability of the control system. I call these features the ‘meta-control system’.

This paper describes the efforts made by the control systems group at DESY to provide a reliable tool for the operators, minimizing the downtime caused by control system failures. It reviews this aspect of computer based accelerator control dating back to the late 1970s when the accelerator PETRA went into operation, controlled entirely by mini-computers from Norsk Data [1].

Both the computer with the supporting technology and the control system group are essential to an accelerator’s success.

INTRODUCTION

MCS -the machine control group at DESY- has built, maintains and improves the control systems of all current DESY-accelerators: The preaccelerators LINAC II, DESY II and PIA, the light sources DORIS III and PETRA III and the free electron laser light source FLASH II. Since the decision to switch off the proton-lepton collider HERA II in 2007, DESY changed its scientific profile from a predominantly high-energy physics laboratory to a synchrotron light research centre. This had a major impact on the required reliability and availability of the control systems:

- The top-up mode for PETRA III does not tolerate any failure in the accelerator-chain for more than 5 minutes.
- The cramped schedule of the beam line experiments at DORIS III, PETRA III and FLASH II may well leave behind an unhappy user if part or all of the requested beamtime is lost.

OPERATOR VIEW

A control system is most visible at the operator-console. Nowadays this is an assembly of monitors and input devices such as mice, knobs or keyboards connected to computers. The operator console is the place from which all available functions of the accelerator in its different phases of operation can be controlled: user run, maintenance periods and machine studies.

The technical implementation differs a lot from control system to control system, but nevertheless the look and feel is not much different. (FLASH and PETRA have different ‘control systems’ but for some areas like vacuum and sequencer there is hardly any difference.) Differences arise from the different age of the accelerators and also from the skills and preferences of the constructor and operator.

Application programs in operator consoles may be rich clients written in Java and Visual-Basic or they may be operator panels generated for example by jDDD and web-based-applications running in a browser such as Web2C [2].

At the other end, there is the accelerator which will be directly affected by actions initiated at the operator-console or by automatic processes running independently of operator interaction. The diagnostic- and machine-protection systems will necessarily report any malfunction of control system procedures.

In between we have what I call here a ‘communication cloud’, i.e. something allowing communication between the operator console and the accelerator. This leads to the simple operator view of an accelerator control system shown in Fig. 1.

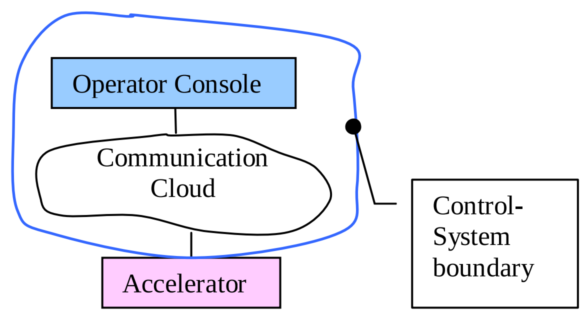

Figure 1: Simple operator view of a control system

CONTROL SYSTEM PEOPLE VIEW

Looking more deeply into a control system one can identify the different hardware building blocks in the ‘communication cloud’: Computers, networks, field buses, diagnostic systems and turnkey systems.

There is no precise definition as to where control systems boundaries are drawn. What belongs and does not belong to the control system is defined in different ways by different people. But at least one needs all major subsystems interfaced to the control system.

An even deeper view will bring us to the software. But at this point the system cannot be understood without yet further information, information which cannot necessarily be found in the control system.

To get information about the underlying principles and concepts of a control system one should ask the control system group. They will use a lot of buzzwords or

TANGO COLLABORATION NEWS

J. Meyer (ESRF) on behalf of the Tango community
ALBA, DESY, ELETTRA, ESRF, SOLEIL

Abstract

During the last years, the Tango collaboration was and is still growing. More and more users are requesting new features and developing new tools for Tango. Decisions whether the requested features will be implemented and whether new tools will be part of the Tango distribution need to be made. The organizational aspects of the collaboration need to be clarified as well as the decision making process for new developments.

This paper will explain the collaboration, its organization and the decision making process as well as the latest facts and features around Tango.

Some ongoing developments are the new code generation tool to allow inheritance in the Tango class structure, the new event system for high bandwidth event distribution and the Tango packaging to allow installation with a few clicks.

WHAT IS TANGO?

Tango [1] is a control system tool kit developed by a community of institutes. It is object oriented with the notion of devices (objects) for each piece of hardware or software to be controlled. Tango classes are merged within operating system processes called Device Servers. Three types of communication between clients and servers are supported (synchronous, asynchronous and event driven).

But Tango is not only the software bus which handles the communication between device servers and clients. The Tango tool chain offers software from the hardware interface to the graphical user interface for several programming languages.

Tango utilities are available, with the basic installation, for code generation, device configuration and testing and for administration and survey of a whole Tango control system.

An archiving and a configuration snapshot system usable with Oracle or MySQL are also available.

Table 1 : Available Tango Modules

Module	Description
Core Libraries	Client/Server communication libraries for C++, Python and Java
Device Classes	About 300 hardware interface classes are available to download [1]
GUI Frameworks	Available for C++ and Python using QT, for Java using Swing and a web interface written in PHP
Client Bindings	LabView, Matlab and IgorPro
Tools	Pogo – Code generator for device classes in C++, Python and Java Jive – Configuration and testing tool Astor – Administration and survey of the Control system
Archiving	Archiving and snapshot system with GUIs and web interface. Usable with Oracle and MySQL
Alarm System	Event driven alarm service
Sardana	Framework for experiment control : Interface standardization, configuration, sequencing, command line interface

COLLABORATION HISTORY

Tango development started in 1999 at the ESRF. SOLEIL joined as the first partner in 2002, ELETTRA and ALBA joined in 2004 and the DESY (beamline controls) in 2008.

For every new member a new memorandum of understanding was signed by all collaboration partners.

We meet twice a year to discuss all ongoing projects. In case of lack of consensus, we tried to find a solution, all collaboration partners could agree upon.

A coordinator was named in each institute for all organisational, but also technical requests concerning Tango.

A mailing list is available for all questions and propositions to the whole Tango community.

A GROWING COMMUNITY

Since last year we have two new institutes requesting to join the collaboration: MAX-lab in Sweden, FRM-II in Germany. Tango is also used by other laboratories, for example LMJ (beam diagnostics) in France. Industrial companies are evaluating Tango, due to outsourcing requests from new projects.

The number of software development projects around Tango is increasing. To package the system and to keep the source repositories clean, we have to decide which projects will be part of the Tango distribution and which ones will be add-ons.

With the growing community, the increasing number of users and the foreseeable number of new developments around Tango, we have to find a new organisational form, to be sure, to take decisions on development priorities and strategies within a reasonable delay.

THE NEW ORGANIZATION

Taking into account the increasing number of users, we will reduce the frequency of Tango meetings to reduce organizational effort and cost. Specialised meetings on particular development projects are encouraged.

THE TINE CONTROL SYSTEM PROTOCOL: HOW TO ACHIEVE HIGH SCALABILITY AND PERFORMANCE

P. Duval and S. Herb, DESY/Hamburg

Abstract

Over the years the TINE control system [1] has implemented numerous strategies for achieving high efficiency data transport within a distributed control system. This was essential for controlling a large machine such as HERA [2]. Our recent experience with controls for the PETRA3 and FLASH accelerator complexes at DESY has revealed new scalability issues. The principal problem has been in limiting the communications load on the front end servers and network in the presence of increasing numbers of client applications, many of which are written by 'part-time' developers who prefer simple API calls, or use development platforms which support only such calls. A single such application, polling hundreds of devices, may generate ~1000 calls per second to a single server. This load on the server can be reduced if, for example, the intermediate software layers can consolidate such calls into array transfers. TINE now offers various 'second-order' protocol features which go a long way toward not just allowing but 'enforcing' efficient data transfer. We shall describe some of these features in this article.

INTRODUCTION

In this report we concentrate on how the control system protocol can be a limiting factor in scalability regarding large distributed systems. To this end it is necessary to review some popular communication strategies along with application programmer interface (API) paradigms.

DISTRIBUTED DATA FLOW

0th Order: Transaction-based Client-Server

The earliest versions of most popular control system protocols made exclusive use of transaction based client-server polling. This data-flow pattern has the inherent advantage of a ‘keep it simple’ strategy, but can quickly run into scalability issues. These often manifest themselves as server-load problems rather than network-load problems, although both issues are important.

We take the average *load* (per second) on a server due to polling clients to be roughly given by

$$L_S \sim N_c \times N_T \times L_D \times U_T \qquad (1)$$

where L_s is the additional load on the server process due to processing client transactions, N_C is the number of clients, N_T is the average number of transactions per client, L_D is the average dispatch load of a transaction request at the server, and U_T is the average client polling rate. Equation (1) is of course schematic. The *loads* L_S and L_D will be taken to refer to the number of CPU cycles devoted to the client-side transactions.

Note that ‘throwing money and threads’ at the problem does *not* reduce the load as defined above. Faster, multi-core computers are of course able to do more in a given time interval. Using a thread for each transaction can also reduce the impact of sluggish servers on the client side. But in the end, the total number of CPU cycles involved will be the same (if not more, due to extra thread synchronization and context switching).

Similarly, the average load on a server’s network port is

$$L_N \sim N_c \times N_T \times P_T \times U_T \qquad (2)$$

where L_N is the network load (bytes per second), N_C, N_T and U_T are as before, and P_T is the average transaction payload. This does not depend on the number of threads used or the CPU power of the server.

A real reduction in load (server or network) involves reducing either N_C and N_T or both in the above equations. This can either be accomplished artificially (for instance by imposing restrictions on the number of and location of clients allowed to run and the update rates they are allowed to use) or moving to other data flow models.

1st Order: Contract-based Publish-Subscribe

As most control system data is used primarily in display at the client side, moving to an asynchronous publish-subscribe model can work wonders reducing the load on a server. Doing so eschews the ‘keep it simple’ approach, as connection and contract management are needed. A transaction request now results in a *contract* managed by the server, along with a table of attached clients. Nonetheless, the average load on a server due to client requests essentially becomes

$$L_S \sim N_T \times L_D \times U_T \qquad (3)$$

That is, the number of clients no longer plays a role. A transaction request is cached and made once on behalf of all N_C clients.

The outgoing network load (2) essentially remains the same, as the transaction results need to be passed to all interested parties. The incoming contribution to network load is for all practical purposes decimated, as transaction requests are made far less often. In order to further reduce the network load, one can adopt a ‘send-on-change’ policy, or reduce the number of clients by delivering data via multicast (especially effective for those transactions involving large payloads). The TINE control system protocol supports both of these features.

Asynchronous, publish-subscribe based protocols have a much larger domain of applicability, which however is still finite for several reasons. First, the API paradigm

FESA3
THE NEW FRONT-END SOFTWARE FRAMEWORK AT CERN AND THE FAIR FACILITY

Alexander Schwinn, Solveigh Matthies, Dorothea Pfeiffer(GSI, Darmstadt)
Michel Arruat, Leandro Fernandez, Frank Locci, David Gomez Saavedra (CERN, Geneva)

Abstract

Currently the LHC (Large Hadron Collider, located at CERN/Switzerland) is controlled by the use of FESA2.10 (FrontEnd Software Architecture v. 2.10) classes. FESA3 is not only an update of FESA2.10, but a completely new approach. GSI plans to use the FESA system at the complex FAIR facility.

One of the main reasons to introduce FESA3 was to provide a framework which can be shared between different labs. This is accomplished by splitting up the FWK into a common part, which is used by all labs, and a lab-specific part, which allows e.g. a lab dependent implementation of the timing-system.

FESA3 is written in C++, runs a narrow interface (Remote Device Access, a middleware which encapsulates CORBA), supports multiplexing of different accelerator-cycles, is completely event driven and uses thread priorities for scheduling. It provides all FESA2.10 functionalities and additionally introduces several new features.

FESA3 is integrated in the Eclipse IDE as a plugin. Using this plugin, the user can easily create his FESA-class design (xml file), generate the C++ source code, fill the device-specific methods, and deploy the binary on a front end.

As well as the framework the Eclipse plugin has a lab specific implementation.

An operational release for FESA3 is planned end of 2010.

THE PURPOSE OF FESA3

FESA3 is a software framework which provides an easy way for developers to produce device classes by generating most of the code automatically. It supports multiplexing of different accelerator-cycles and many other features which can be used by the class-developer. The main purpose of the framework is to provide an common and unified way to develop device classes. This approach saves a lot of work and simplifies debugging, documentation and code adoption for the class-developer and all involved parties.

THE ROOTS OF THE FESA FRAMEWORK

All early versions of the FESA framework were developed solely by the CERN facility. FESA3 is the first release which is developed as an collaboration between CERN and the GSI. This collaboration was the main reason to restructure some of the Fesa2.10 fundamental internal parts and to finally go for a new major release.

FESA3 continues to provide all services from older versions and as well extends the common approach by additional services which where demanded by the CERN user community.

FESA3 AT THE FAIR FACILITY

For the FAIR facility several new accelerator installations will be built at GSI.

Central aspect is an increased number of research programs resulting in up to five beams in parallel. The FAIR facility will be controlled by a new control system which will be able to support all aspects of the complex GSI/FAIR operations on a common technical basis. The control system for the FAIR facility currently is in the design phase.

One part of this new control system will be the device software which runs on the front ends. FESA was choosen as software framework since it already proved itself at the LHC at the CERN facility and allows to pass the device-specific implementation directly to the device expert.

CLASS DEVELOPMENT WORKFLOW

The FESA3 Eclipse-plugin guides the class developer on his way to develop a FESA3 class. The following steps have to be performed to do so:

1. Design

   In the first step the developer needs to design his class according to his needs. This process involves the specification of *Properties*, *Fields*, *Server-* and *RealTime-Actions* and their dependencies on each other. The design itself is done via a comfortable XML editor, which is integrated in the FESA3 Eclipse-plugin and coupled to an XSD schema for validation.
   (see figure 1)

2. Code Generation

   Code generation may be started in the plugin if the class design is valid. An XSLT engine generates C++ code using the class design as input.

EMPLOYING RTEMS AND FPGAS FOR BEAMLINE APPLICATIONS AT THE APS*

D.M. Kline, S.K. Ross#, ANL, Argonne, IL 60439, U.S.A.

Abstract

At the Advanced Photon Source (APS), the power and flexibility of an Altera Cyclone-II FPGA combined with the Arcturus uC5282 embedded microprocessor running RTEMS, provides a low cost solution for implementing beamline applications.

In this paper, we discuss the approach of coupling an Altera FPGA and the Arcturus uC5282 to implement a time-resolved 32-channel scaler, development using the Altera Quartus-II design environment and the RTEMS tools, as well as an ASYN based EPICS device driver and its integration to the standard scaler record support. Furthermore, we discuss how this approach has been applied to other control system applications, such as for photon counting and flexible CCD shutter timing control.

By employing this approach, a variety of applications can be quickly developed on one hardware platform which realizes real-time performance within the FPGA and provide a cost effective EPICS IOC for exporting data to scientists and users.

INTRODUCTION

There exist many approaches for developing beamline and detector applications at various levels of complexity. The "Generic Digital" approach we employ abstracts the application behaviours and compartmentalizes them, serving as our "design pattern."

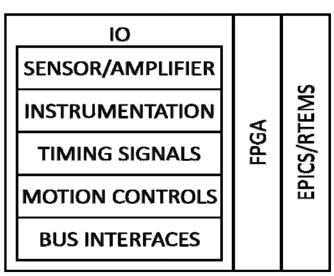

Figure 1: "Generic Digital" conceptual model.

The Input-Output (IO) component represents the specific hardware designed for the particular needs of the application. This may include inputs and outputs at different signal levels, such NIM, TTL, LVDS, or ECL. Bus adapters, such as PC/104, can be developed to take advantage of IO modules developed in-house or commercial, such as motion controllers and input sensors. Typically, the IO component connects to an in-house developed or purchased transition board, which then connects to existing infrastructure.

* Use of the Advanced Photon Source at Argonne National Laboratory was supported by the U. S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.
#skross@aps.anl.gov

The FPGA component includes in-house developed base boards as well as commercially available boards, such as Altera's Cyclone or Stratix development kits. It handles the real-time aspects of the application and serves as a mediator between the IO and EPICS components.

The EPICS [1] component uses the Arcturus uC5282 [3] microcontroller as the IOC and runs the RTEMS [4] real-time operating system. It connects to the FPGA using carrier boards developed at the APS. ASYN [6] drivers are written to interface with the IO hardware. Furthermore, EPICS databases and MEDM screens have been included to implement the application behaviour and user interface.

HARDWARE

Generally, both in-house developed and commercially available hardware is employed within the model. The FPGA component uses two generations of hardware that was developed in-house. More recent versions use development kits offered by Altera.

Generation I

The generation I (GEN-I) base board consists of Altera's FLEX10K FPGA. This implementation is targeted for less complicated applications that don't require many logic elements and requires frequencies lower than 50MHz, such as "divide-by" circuits, combinational logic consisting of a few logic gates, or simple scalers.

Communication between the IOC and FPGA is through a in-house developed Serial Peripheral Interface (SPI). Typically, this is implemented on a Linux based system, such as an EPICS brick (EBRICK) [6]. The EBRICK is a Poseidon Single Board Computer offered by the Diamond Systems Corporation [5]. The FPGA is housed in a 1U rack mountable chassis and uses an external wall mount or desktop power supply.

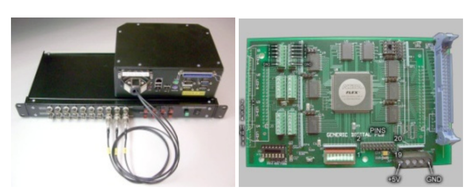

Figure 2: GEN-I/w EBRICK (left) and base board (right).

Generation II

The generation II (GEN-II) base board uses an Altera Cyclone-II FPGA [2] in a 3U VME footprint. It has IO for the uC5282, 24bits TTL IO, and 40bits LVTTL expansion IO for additional daughter boards. This implementation is targeted at higher-end applications that require more logic elements and frequencies greater than

QT EPICS DEVELOPMENT FRAMEWORK*

A. Rhyder#, A. Owen, G Jackson. Australian Synchrotron

Abstract

QCa is a layered software framework based on Qt for accessing EPICS data using Channel Access on a range of platforms. It is used on several beamlines at the Australian Synchrotron. The QCa framework provides object oriented C++ access to control systems using EPICS (Experimental Physics and Industrial Control System). It is based on Qt, a widely used cross-platform application development framework. GUI or console based applications can be written that use QCa at several levels. QCa includes Qt plugin libraries, EPICS aware widgets, data formatting classes, and classes for accessing raw EPICS data in a Qt friendly way. QCa also includes an application for displaying forms produced by the Qt development tool 'Designer'. Using this application a complete EPICS GUI system can be generated without writing any code. A GUI system produced in this way can interact with existing EPICS display tools such as EDM. QCa handles much of the complexities of Channel Access including initiating and managing a channel. Applications using QCa can interact with Channel Access using Qt based classes and data types. Channel Access updates are delivered using Qt's signals and slots mechanism.

INTRODUCTION

Channel Access is described as 'one of the core components of an EPICS system. It is the software component that that allows a Channel Access client application to access control-system data which may be located on different hosts throughout a network' [1]

While CA is the default means to access EPICS data, its use is not trivial. A significant understanding of how CA works is required to execute the steps required to read or write data. The complexity of setting up and terminating CA requests leaves room for error. Also, CA uses a C programming interface and so does not make use of object oriented programming techniques.

Qt is a cross-platform application and UI framework. It includes a C++ class library and a cross-platform IDE.

The QCa framework provides a Qt based C++ framework for easy CA access to EPICS data.

It provides access to EPICS data at several levels from programmatic reading and writing of data, EPICS aware widgets for developing GUI based applications through to EPICS aware Qt plugins such as push buttons, sliders, and text widgets. When these plugins are used within the Qt form development tool 'designer' EPICS GUIs can be developed without the need for any code development.

QCA FRAMEWORK HIERARCHY OVERVIEW

The QCa framework is designed to allow access to CA data in the most appropriate form. The framework is based on a hierarchy of classes as shown in Table 1. This hierarchy is open at all levels to the developer. Appropriate use of the hierarchy is shown in Table 1.

Table 1: QCa framework hierarchy

Type of access to CA data.	**Functionality**	**Main classes**
C++ access to the CA library.	Provides convenient C++ access to the CA library.	CaObject
Qt based access to CA.	Hides CA specific functionality. Adds Qt functionality such as signals and slots.	QCaObject
Data type independent access.	Hides EPICS data types, providing read and write conversions where required.	QCaInteger QCaString QCaFloating
EPICS aware graphical widgets.	Implements graphical Qt based widgets that provide access to EPICS data.	QCaLabel QCaLineEdit QCaPushButton QCaShape QCaSlider QCaSpinBox QCaComboBox QCaPlot
EPICS aware graphical Qt plugins.	Adds Qt plugin interfaces to EPICS aware widgets.	QCaLabelPlugin QCaLineEditPlugin QCaPushButtonPlugin QCaShapePlugin QCaSliderPlugin QCaSpinBoxPlugin QCaComboBoxPlugin QCaPlotPlugin
GUI support widgets	Implements Qt based widgets that support control system GUIs. These widgets do not access the CA library.	AsGuiForm GuiPushButton CmdPushButton Link

*Work supported by the Australian Synchrotron
#andrew.rhyder@synchrotron.org.au

THE BEAMLINE EXPERIMENTS SCHEDULING SOFTWARE*

Yuhong Yan#, Ludeng Zhao, Zhiguo Wang, Yongxin Zhu, and Chun Wang,
ENCS, Concordia University, Montreal, QC H3G1M8, Canada

Abstract

Scheduling the experiments to the beamlines of the synchrotron at the Canadian Light Source (CLS) is a manual procedure so far. Once every six months, the beamline scientists discuss before a whiteboard to schedule as many approved experiments as possible. There are so many constraints on resource capabilities, availabilities, user preferences, and experiment priorities to consider that none has ever been able to check if the manual scheduling results are optimal or not. In the Canarie funded project Science Studio, we are building an automatic scheduling module as a part of the User Office. After the synchrotron users submit their proposals via the User Office, the automatic scheduling module can find an optimal scheduling solution that satisfies all the constraints modelled, if such a solution exists, and display the results on a Web calendar. In this paper, we present our contributions on design and implementation of the scheduling module and our study on automatic scheduling of synchrotron experiments.

THE BACKGROUND

The automation of the scheduling activities at the CLS is part of the Canarie funded project Science Studio. The Science Studio project develops a complete experiment management system [1] that allows the researchers to control the experiment devices, observe the experiment processes, and collect data from their own home bases, instead of travelling to the CLS site.

There are about 30 plus the CLS like facilitates [2] around the world. All the facilities have similar proposal approval procedures, regardless the different frequencies of calls-for-proposals and the length of minimal time slot. Scheduling the approved proposals is done manually. In the CLS, the beamline scientists who are in charge of scheduling experiments on the beamlines use e-mail and documents like spreadsheet and pdf as their primary tools to communicate with the users and manually scratch the schedules on a calendar. In order to make their lives easier, the beamline scientists tend to limit the possible combinations they should consider. Furthermore scheduling under conflicting constraints can easily become intractable as the number of users and proposals increase.

In this paper, we present our solution to automate the scheduling function. The User Office in the Science Studio platform has a proposal management module to manage the proposal submission and review procedure. In the CLS, the call-for-proposals occurs every six months (*aka*. a cycle). The approved proposals can be scheduled into the next four cycles. Our schedule module reads in the proposal information, invokes the scheduling tool, and displays the scheduling results a Web calendar. The beamline scientists can review the schedule and manually fine-tune the schedule over the Web calendar. The design and the model of experiment scheduling can be reused for different facilitates as well. The parameters of the scheduling model and the Web UI of the implementation are tuned for the CLS.

THE SYSTEM ARCHITECTURE

The Science Studio platform is a large J2EE enabled Web application. Figure 1 shows its system architecture. The core of the system architecture is the application tier composed by the User Interface (UI) services, the User Office, and the beamline services.

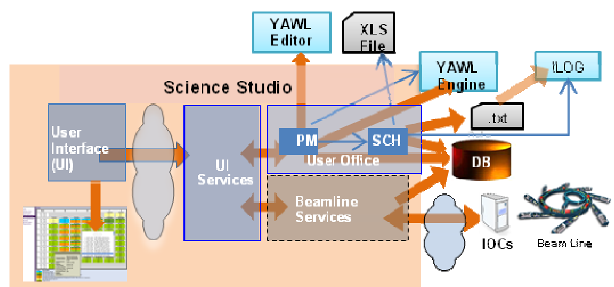

Figure 1: System architecture for Science Studio.

The major functions of the User Office are proposal management (PM in Fig. 1) and scheduling (SCH in Fig. 1). The proposal management module accepts user inputs and manages the proposal review process. The proposal review process can be executed by a workflow engine YAWL. YAWL Editor is used offline to design the proposal review process. When the approved proposals are decided, the scheduling function is invoked.

The scheduling module gets the approved proposals from the database or from the excel sheets currently used, and converts the information about the defined experiments into a text format that can be accepted by the ILOG - the automatic scheduler. A .txt file containing the converted data is the media of transferring the data between Science Studio and the ILOG. The ILOG is invoked by the scheduling module. The ILOG writes the scheduling results into a text file. The scheduling module reads the results and shows them on the Web calendar, meanwhile stores them into the database.

*Work supported by the Science Studio project which is funded by Canarie Network Enabled Platform Program contract number NEP-01
#yuhong@encs.concordia.ca

ACCURATE MEASUREMENT OF THE BEAM ENERGY IN THE CLS STORAGE RING*

J.M. Vogt#, J.C. Bergstrom, S. Hu, CLS, Saskatoon, Canada

Abstract

Resonant spin depolarization was used at the Canadian Light Source (CLS) to measure the energy of the beam in the storage ring with high accuracy. This method has been employed successfully at several other synchrotrons in the past. At the Canadian Light Source, however, resonant spin depolarization is an intrinsic capability of the transverse feedback system, which is based on a Libera Bunch-by-Bunch unit. The Bunch-by-Bunch system used at the CLS was customized to include a bunch cleaning feature based on a frequency-modulated oscillator. By setting the frequency of this oscillator to the spin tune, the beam can be depolarized and the effect can be observed by watching the life time of the beam. No changes have to be made to the permanent setup of the transverse feedback system, and no special instrumentation is required to make the energy measurement.

RESONANT SPIN DEPOLARIZATION

The theory of resonant spin depolarization as a means of measuring the beam energy in a storage ring has been described in detail in Ref. [1]. After injection, the beam polarization builds up with a machine-dependent time constant, usually in the range of a few tens of minutes. Depolarization is then accomplished by applying an RF-signal at the resonant frequency of the spin. The effect of the resonant depolarization is observed either as an increase in the amount of Touschek scattering, or as a decrease of the beam life time. Several facilities have used this method in the past [1-7].

The frequency at which resonant depolarization occurs is a direct measure of the beam energy. Equation (49) in Ref. [1] gives the spin tune ν as:

$$\nu = a\gamma = a\frac{E}{m_e c^2}\,, \tag{1}$$

where

$$a = \frac{g-2}{2} = 0.00115965$$

is the anomalous magnetic moment of the electron, E is the beam energy, and m_e is the electron mass. At the nominal beam energy of the CLS storage ring, which is 2900 MeV, the spin tune is ν = 6.5812.

The expected resonant depolarizing frequency f_{dep} is:

$$f_{dep} = \nu_{frac} \cdot f_o = 1.0197\,\text{MHz}\,, \tag{2}$$

where ν_{frac} is the fractional part of the tune and f_o = 1.7544 MHz is the orbit frequency of the storage ring. Note that there is an ambiguity between $\nu_{frac} = 0.5812$ and $1-\nu_{frac} = 0.4188$, so that another solution for the depolarizing frequency is:

$$f_{dep} = (1-\nu_{frac})\cdot f_o = 0.7347\,\text{MHz}\,. \tag{3}$$

INSTRUMENTATION AT THE CLS

The Transverse Feedback System

The transverse feedback system is based on a Libera Bunch-by-Bunch unit, which was customized to include a frequency modulated oscillator for bunch cleaning [8]. The frequency of this oscillator was set to the spin tune and the amplifiers and the vertical kicker of the transverse feedback system were used to depolarize the beam.

Detection of Depolarization

Because of signal-to-noise considerations, the preferred method of detecting depolarization is by measuring Touschek electrons. However, the arrangement of the magnets in the storage ring and the shape of the vacuum chambers make it impossible to set up Touschek detectors at the CLS. Depolarization therefore had to be detected by observing its effect on the life time of the beam.

MEASUREMENTS

Machine Setup

The machine setup was determined by the following considerations:

- In order to maximize the Touschek effect on the life time, the bunch current had to be as high as possible,
- The bunch current was limited by the head-tail instability,
- In order to minimize the vacuum effect on the life time, the total current had to be as low as possible,
- The total current had to be high enough for a sufficiently accurate measurement of the storage ring current and the life time.

As a compromise, three bunches in the storage ring were filled with a current of about 10 mA/bunch.

*Work supported by NSERC, NRC, CIHR, WEDC.
#johannes.vogt@lightsource.ca

STATUS OF THE FUTURE SPIRAL2 CONTROL SYSTEM

D. Touchard*, P. Gillette, C. Haquin, E. Lemaître, L. Philippe, E. Lécorché# (Ganil / Caen, France)
J.F. Denis, F. Gougnaud, J.F. Gournay‡, Y. Lussignol, P. Mattei (CEA-IRFU / Saclay, France)
P. Graehling, J. Hosselet, C. Maazouzi (CNRS-IPHC / Strasbourg, France)

Abstract

For the study of fundamental nuclear physics, the SPIRAL2 facility, a driver accelerator followed by a rare ion production process, will be coupled with the existing GANIL machine to provide light and heavy exotic nuclei at extremely high intensities. To ease the collaboration with several institutes on the control system design, EPICS has been chosen as the basic framework and a specific care has been taken concerning the software organization and management. While first operational interfaces for power supplies, faraday cups or beam slits are already operational, a triggered fast acquisition system for beam diagnostics, a radiofrequency control system, and an admittance measurement system are going to be achieved. First EDM supervision screens and high level tuning applications based on EPICS/XAL framework have been designed. The use of relational databases, on the one hand for the design of an environment to generate the EPICS databases, on the other hand to manage, set and archive meaningful values of the new facility, is under investigation. From the beginning of last year, two sources followed by their first beam line sections have been tested. Promising results are presented.

THE SPIRAL2 PROJECT

Overview

Following the recommendations of international committees and to fulfil the growing demand of the international physiscists community, in May 2005 the French Research Minister decided to build the new SPIRAL2 facility at GANIL laboratory (CNRS-CEA) in Caen (France) [1]. The project aims to enlarge multi-beam production using Isotope Separation On Line (ISOL) method. A superconducting LINear ACcelerator (LINAC) for light and heavy ions preceded by a radio frequency quadrupole (RFQ) will deliver up to 40MeV/A for 5mA deuteron, respectively 14.5MeV/A for 1mA heavy ion continuous wave (CW) beams [2]. These beams can be used for the production of intense Radioactive Ion Beams (RIB) involving the fusion, fission, transfer reaction mechanisms. More specifically, production of RIB with intense neutron-rich nuclei will be based on the fission of uranium targets bombarded either by neutrons produced by a first impact of the deuteron beam on a carbon converter, or by the direct deuteron or heavy ion beam impact. The RIB post-acceleration will be performed by the existing CIME cyclotron, which is perfectly suited to the separation and acceleration in the energy range up to 10MeV/A for the atomic masses between 100 and 150. SPIRAL2 beams after CIME can be reused in present experimental areas of GANIL (see fig 1).

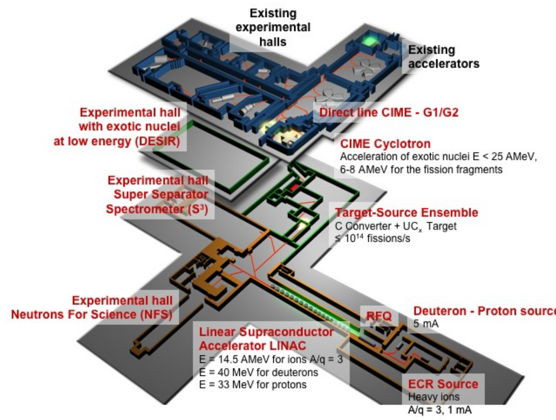

Figure 1: The SPIRAL2 and GANIL facilities.

Milestones

The first primary beams are expected in spring 2012 (phase 1) while the production process is planned more or less one year later (phase 2).

Some parts of the accelerator have been tested: the ion source and its low energy beam line have been in test at CNRS-LPSC (Grenoble) since 2008 and the deuteron source and its coupled beam line are progressively tested at CEA-IRFU (Saclay) since the beginning of this year[3].

Collaboration organisation

In order to build a large machine as SPIRAL2, an international collaborative effort has been made to establish the grounds for the design. A large team composed of people from CNRS, CEA, and international institutes is involved in the scheme. This is also the case as far as the command control system is concerned. The following three French laboratories, GANIL (Caen), IPHC (Strasbourg) and IRFU (Saclay) are currently designing and developing respectively the whole hardware and software command control system phase 1 architecture. For the phase 2, collaboration with the following three laboratories, LPSC (Grenoble), CENBG (Bordeaux), and LPC (Caen) is presently under consideration.

CONTROL SYSTEM

Main choices

The main choice of EPICS [4] as a common framework was early decided to ease pieces of software development and integration efficiency. A set of drivers already

*touchard@ganil.fr
#Spiral2 command control coordinator
‡Injector command control coordinator

SETTINGS MANAGEMENT WITHIN THE FAIR CONTROL SYSTEM BASED ON THE CERN LSA FRAMEWORK

J. Fitzek, R. Mueller, D. Ondreka, GSI, Darmstadt, Germany

Abstract

A control system for operating the future FAIR (Facility for Antiproton and Ion Research) accelerator complex is being developed at GSI. One of its core components is the settings management system.

At CERN, settings management and data supply for large parts of the CERN accelerator complex is done using the LSA (LHC Software Architecture) framework. Several concepts of the LSA framework already fit the FAIR requirements: Generic structures for keeping accelerator data; modular design; separation between data model, business logic and applications; standardized interfaces for implementing the physical machine model. An LSA test installation was set up at GSI and first tests were performed controlling the existing GSI synchrotron SIS18 already with the new system.

These successes notwithstanding, there are issues resulting from conceptual differences between CERN and FAIR operations. CERN and GSI have established a collaboration to make LSA fit for both institutes, thereby developing LSA into a generic framework for accelerator settings management. While focussing on the enhancements that are necessary for FAIR, this paper also presents key concepts of the LSA system.

FAIR

The international FAIR facility with its nine new accelerator installations will be built at GSI, using the existing linac and synchrotron SIS18 as injectors (see Fig. 1).

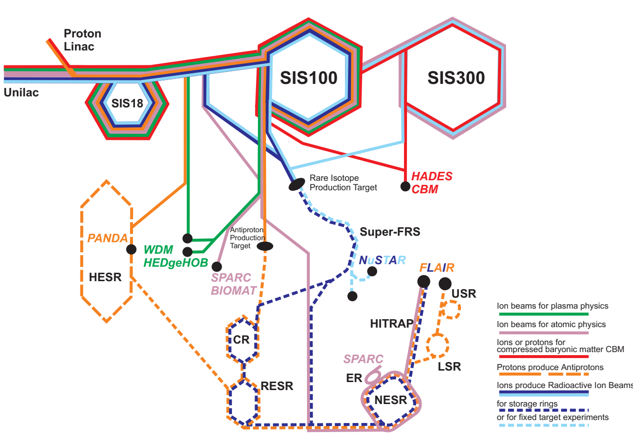

Figure 1: GSI/FAIR beamlines, P. Schuett, GSI 2010.

Central aspect is an increased number of research programs resulting in up to five beams in parallel with pulse-to-pulse switching between different particle types. The future facility will be controlled by a new control system which will be able to support all aspects of the complex GSI/FAIR operations on a common technical basis [4]. The future control system is designed at the moment, keeping well working and proven principles while adopting new methodologies where beneficial.

Important aspects of the control system are generation of settings and data supply. It was evaluated and decided to use the existing LSA framework from CERN for settings management and data supply within the FAIR control system. A collaboration with CERN was set up with joint development effort put into future LSA development [1].

LSA - THE LHC SOFTWARE ARCHITECTURE

LSA was developed at CERN starting in 2001 and is now the core controls software component for settings management and data supply within the CERN control system. For a detailed description of LSA see [2].

LSA - Functional Overview

The LSA system was designed in a generic way and provides clear separation between data model, business logic and applications. Its modular structure allows institute specific implementation to be easily plugged in.

The system covers all important settings management aspects: optics (twiss, machine layout), parameter space, settings generation and management, settings modification (trim), propagation from physics to hardware parameters, operational and hardware exploitation (equipment control, measurements), and beam based measurements.

An accelerator within LSA is modeled by defining its parameter hierarchy – from top level physics down to hardware parameters. Using the optics, the LSA system can already calculate good initial settings. Corrections can be applied to any level of the hierarchy, resulting in a consistent change of many devices at the same time. As an example for a part of such a hierarchy at GSI, see Fig. 3. The LSA system consists of different functional building blocks, which among other benefits entitle physicists to implement the machine model themselves in a structured and simple way.

INTEGRATION OF PROGRAMMABLE LOGIC CONTROLLERS INTO THE FAIR CONTROL SYSTEM USING FESA

R. Haseitl, C. Andre, H. Bräuning, T. Hoffmann, R. Lonsing, GSI, Darmstadt, Germany

Abstract

For the upcoming 'Facility for Antiproton and Ion Research' (FAIR) at GSI, the Front End Software Architecture (FESA) framework built by CERN has been chosen to serve as front-end level of the future FAIR control system [1]. All beam diagnostic devices of FAIR will be controlled by FESA classes that are addressable by the new control system. The connectivity to the old control system is retained, since both control systems will be in operation contemporaneously for several years. Commercially available Programmable Logic Controllers (PLCs) have been installed as part of Beam Induced Fluorescence (BIF) monitors to replace outdated network attached devices and to improve the reliability of the BIF systems. The new PLC devices are controlled by FESA classes which are addressed from the existing C++ software via Remote Data Access (RDA) calls. This contribution describes the system setup and the involved software components to access the PLC hardware.

THE BIF SYSTEM

Beam Induced Fluorescence monitors determine the transverse beam profiles with minimum beam disturbance [2]. The measurement principle is based on the excitation of gas molecules by the passing ion beam in the beam pipe. The emitted photons are measured by digital CCD cameras with image intensifiers to ensure single photon detection. Using two cameras installed above and sideways of the beam pipe, the horizontal and vertical beam profiles are measured simultaneously. Currently, there are four BIF monitors installed in GSI accelerators and transfer lines. For the next years, a final number of seven monitors is anticipated.

Hardware

Each camera has a remote controllable iris to adjust the light intensity illuminating the image intensifier. A smaller aperture of the iris also increases the depth of field. This results in a larger properly focused area in the obtained image. The amplification of the image intensifiers can be adjusted by setting two voltages for the different amplification stages.

The aperture of each iris as well as the amplification of the image intensifiers has formerly been controlled with a self-built, Ethernet connected module, containing several digital-to-analogue converters (DACs). During long term runs of the system, these modules crashed non-deterministically after some hours or days of operation. For the FAIR project, a more reliable solution was desired. The setting of voltages is a common task for PLCs, so this commercially available and field-tested solution was selected.

Software

The software controlling all BIF devices, including irises and image intensifiers, is called ProfileView [3]. The communication with the old hardware is performed via a standard TCP connection. New settings are sent to the device, which replies with an acknowledge message. The communication channel is kept open continuously, to detect failures as soon as possible.

After extensive testing of the system, it was decided to replace the faulty devices by PLCs. To control the new hardware, the ProfileView software was adapted to support both hardware variants.

Figure 1: One of the 'satellites' of the BIF installation. It controls two BIF monitors and features two sets of control devices (from left to right): Power Supply, ET 200M controller, SM322 relay element with eight outlets, two SM332 12-bit DACs.

FESA BASED DATA ACQUISITION FOR BEAM DIAGNOSTICS AT GSI

T. Hoffmann, H. Bräuning, R. Haseitl, GSI, Darmstadt, Germany

Abstract

In view of the upcoming Facility for Antiproton and Ion Research (FAIR) at GSI with its increased complexity in beam control and diagnostics, the decision was taken to use the well-tested CERN made Front-End Software Architecture (FESA) as the lowest level of the new control system [1,2]. In the past years, the current stable FESA framework (Version 2.10) has been adapted and installed at GSI, with the major part of the adaptation being the different machine timing models of GSI and CERN. With a stable environment at hand, all current and new beam diagnostic related data acquisition systems will be implemented with FESA. To demonstrate the suitability of FESA for demanding data acquisition problems with high data rates or large amounts of data, two different projects such as the Tune Orbit and POSition measurement (TOPOS) and the Large Analog Signal Scaling Information Environment (LASSIE) are presented. Experiences with implementing standard interfaces such as CAN, GigE and PLCs in FESA applications as well as a move towards low cost Intel based controllers like the Men A20 VME controller or industry PCs running a real time Linux will be discussed.

THE FESA ENVIRONMENT AT GSI

Besides the development of the next generation FESA 3.0 environment by staff of the GSI controls department in collaboration with CERN CO/FE, the GSI beam diagnostic department (BD), which is responsible for the layout of the FAIR beam diagnostics DAQ system, is developing FESA 2.10 classes for dedicated BD systems. These efforts are made to show the feasibility of all expected data acquisition requirements and to train programming of the front-end part of the new control system for FAIR. At the beginning of 2010 the FESA 2.10 installation and integration at GSI was fully accomplished. The environment resides on a powerful blade system, which is the new mainframe of the GSI control system providing NFS based access to all front-end controllers (FEC) and to all branches of code development. Basic information on FESA is given in [1,2]. The main parts of the FESA systems are:

Operating System

At present the operating system (OS) of the GSI control system is a Red Hat Enterprise Linux Server release 5.5 with kernel 2.6.18-92 - x86_64. The OS for the FECs is Scientific Linux CERN 5.4 with kernel 2.6.24.7-rt27, which contains patches for real-time support.

Supported FEC Hardware

FESA 2.10 provides cross compilers for Intel and PowerPC based CPUs. For maintainability reasons the following FEC systems are supported by GSI:

- Standard Industry PC
- Kontron KISS PCI760 with PXEBoot, diskless, Intel AMT remote management system
- MEN A 20 VME CPU with PXEBoot, diskless.

For applications requiring real time behaviour the CES RIO3 CPUs with Lynx OS can be used as an exception.

For the time being the upcoming xTCA for Physics standard [3] as a new form factor is under evaluation for the usage at FAIR. For the tests an Adlink AMC-1000 CPU in an ELMA xTCA-6 frame were chosen. After integration of the diskless system into the control system, the installation will be tested with high bandwidth applications such as GigE video imaging and analog data sampling.

Timing

FESA 2.10 is strongly dependent on the CERN timing system and its timing receiver hardware, which is different to the existing GSI timing. To gain efficient use of the FESA RT action feature a dedicated FPGA based GSI-CERN timing converter was developed. It allows to use the CERN timing receiver hardware with the GSI timing. Although some purely CERN specific features are not available, this converter allows to trigger RT actions by accelerator timing events in a multiplexed beam operation for all three GSI accelerators (UNILAC, SIS, ESR).

JAVA Graphical user interface

To provide GUIs for the developer as well as for the users such as machine operators or system experts the JAVA based concept of CERN was chosen. It consists of the Java API for Parameter Control (JAPC, [4]) and CERN libraries such as the JDataViewer and the CERN middleware (cmw-rda). Due to the JAVA web-start functionality and the general JAVA platform independence, the GUI may be used at any office at GSI.

TUNE, ORBIT, AND POSITION MEASUREMENT (TOPOS)

The first test project for FESA and its related middleware and GUI solutions at GSI was a development for the tune, orbit and position measurement (TOPOS) at the heavy ion synchrotron SIS in collaboration with Cosylab and Instrumentation Technology, Slovenia. The development of the modular extendible TOPOS was performed also in preparation for the FAIR project and the usage at the FAIR synchrotrons. The data acquisition concept is well described in [5].

This very demanding system, with data rates up to

FAIR TIMING MASTER

Mathias Kreider, Tibor Fleck, GSI Darmstadt, Germany

Abstract

In the scope of building the new FAIR facility, GSI will implement a new timing distribution system based on WhiteRabbit. The FAIR system will resemble a tree topology, with a single master unit on top, followed by several layers of WR switches, down to about two thousand timing receivers throughout the facility. The Timing Master will be a mixed FPGA/CPU solution, which translates physical requirements into timing events and feeds them into the WR network. Macros in the FPGA resemble a 32x multicore with a strongly reduced instruction-set, each event processor responsible for a specific part of the facility. These processors interact in real time, reacting to interlocks and conditions and ensuring determinism by parallel processing. A powerful CPU prepares the timing event sequences and provides an interface to the control system. These tables are loaded into the RAMs of each participating processor, controlling their behaviour and event output. GSI is currently working on the WR timing system in close collaboration with CERN, making this system the future of GSI/FAIR. This contribution covers technical details on the expected timing scenario, macro internals and discussion on possible future development.

INTRODUCTION

Purpose

Future GSI/FAIR facility will use timing events to control machine actions. The FAIR Timing Master will centrally generate all necessary events for the whole accelerator facility. These will be used to trigger all beam guiding components as well as all beam diagnostic measurement devices where individual event filters apply for each single front end controller. The timing receiver is integrated into the standard FAIR frontend controller used mainly for power supplies. For all other use cases, especially all beam diagnostic devices, special timing receiver interface cards will be supplied in different form factors. Typical event reaction will be direct trigger output or IRQ. Furthermore a separate high precision clock distribution system called BuTiS for RF components where highest requirements to accuracy and synchronization apply will be closely coupled to the FAIR timing system.

The WhiteRabbit Transport Layer

The future Timing System of GSI/FAIR and CERN will be based on the WhiteRabbit architecture. WR is a deterministic field bus [2], the physical system consists of a non-meshed GbE network topology, running timing services on OSI layer II. Custom switches and endpoints are used for timing measurements and the WR protocol.
WR provides phase compensation and absolute time distribution with an accuracy down to a nanosecond. Forward error correction algorithms are employed to get highest system reliability. Deterministic lag times are made possible by using Quality of Service (QoS). This makes preferring marked high priority packets possible. Since the lag time to destination is reliably known in advance, this allows machine control packets to always arrive on time.

The FAIR Timing Master

To provide an interface to the general control system of the facility, a powerful CPU handles the abstract beam production down to the creation of sequence programs for control of Event Processing Units (EPU).

Every abstract physical part of the accelerator facility like the linear accelerators, synchrotron rings and storage rings, will be represented by a dedicated timing event generator unit.

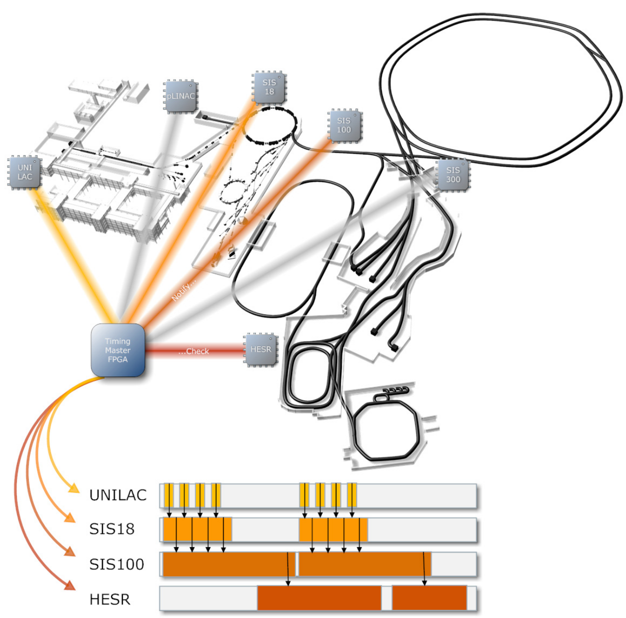

Figure 1: Mapping components to EPU programs

Interaction between these machine parts requires fast synchronisation between their timing schedules. For example, a synchrotron ring needs to time its ejections precisely with the receiving collector ring. The design therefore includes a fast mechanism for exchanges between generators.

FROM AN EMPTY PC TO A RUNNING CONTROL SYSTEM: A KNOPPIX LIVE-CD FOR DOOCS

G. Grygiel, DESY Hamburg, Germany

Abstract

Software deployment of operating and control systems is a hard task for beginners and can be an error prone one for experts. As an evaluation of a potential, fast deployment technique, a Linux/Knoppix Live-CD [1] for the DOOCS [2] control system software has been developed. This CD contains a DOOCS core system, some example and middle layer server programs and basic client applications. Optionally, one can install a Knoppix and DOOCS system directly from the CD. All DOOCS and operating system software are provided as Debian [3] packages. This paper will describe the Live system CD in more detail and discuss the interaction of Java Web-start based applications, other control system client applications, DOOCS name service and device servers.

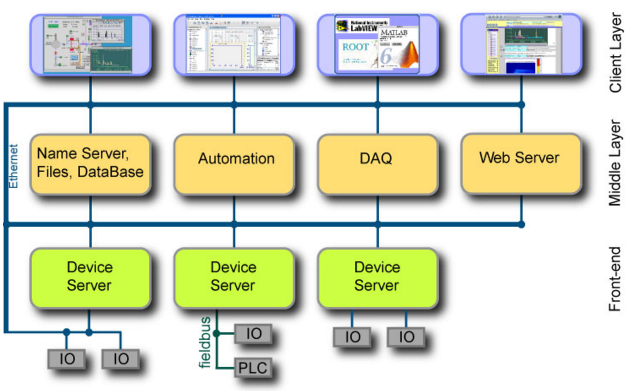

Figure 1: DOOCS Architecture

MOTIVATION

The idea is to run the DOOCS control systems with all major programs directly from a CD. The 'experts' have then an always available and workable system and this e.g. is an USB flash drive on the keychain. It's ment to provide an overview of the entire system, without complex installation and configuration. A beginner receives a fully equipped and functional system. It is possible to start immediately with the development of control system servers and having all tools at hand. The Live-CD also demonstrates the integration of the various controls system architectures, like DOOCS, EPICS [4] or TINE [5], used at modern accelerator facilities.

Features of the CD are:

- Any time, every where available.
- Quick start for beginners.
- Debug tool for experts.
- Demonstrates the whole chain, from the name service, device servers, up to the display.
- Demonstrates the interaction of the various control systems (DOOCS, TINE, EPICS).

CHOICE OF DISTRIBUTION

For almost all components, DOOCS Debian packages have been developed, therefore it should be a Debian based distribution. Currently at DESY the Ubuntu [6] distribution is used. Various tests have shown that the Live-CD made by Klaus Knopper is significantly faster than the Live-CD of Ubuntu or Debian. The KNOPPIX distribution has a very good driver support; it is fast and designed to be run directly from a CD / DVD (Live-CD) or USB stick. The first attempt to remaster a KNOPPIX live CD was immediately successful.

RECIPE TO BUILD THE LIVE-CD

Start with booting from the KNOPPIX CD. A minimum of 3 GB free disk space should be available. Then copy the complete disc to the free space. Then again one can boot the usual Linux system and start changing the content of the KNOPPIX CD. Use 'chroot' to install and configure all control system and other software. With chroot one is able to run a command or interactive shell in a special root directory. Also Internet access is possible out of the chroot environment. Change the look and feel to give the CD a personal note e.g. titles, graphics, menus. All it takes to remaster a KNOPPIX CD is described in the KNOPPIX_Remastering_Howto [7]. There are many UNIX commands to execute; therefore a good UNIX/Linux knowledge is required. It took a view interactions until everything worked and looked as expected.

To speed up the development process:

- Create the CD image.
- Start this image under qemu [8] (processor emulator) with KVM [9] support.

KVM (Kernel-based Virtual Machine) with native virtualization support helped a lot to speedup the development process. The boot up process takes less than a minute. If KVM with native virtualization support is present, it will be used by qemu automatically.

CONTENT

The CD contains a DOOCS example server (SINGENERATOR) which talks also the TINE protocol. Furthermore DOOCS, EPICS and TINE command line tools (CLI) and some graphical java programs.

In detail:

- DOOCS
  - Server programs
    - ENS (equipment name server).

CONSOLIDATING THE FLASH LLRF SYSTEM USING DOOCS STANDARD SERVER AND THE FLASH DAQ

O. Hensler, W. Koprek, H. Schlarb, V. Ayvazyan, C. Schmidt, DESY, Hamburg, Germany
Q. Geng, SLAC, Menlo Park, CA, U.S.A.

Abstract

Over the last years the LLRF group developed many different flavors of hardware to control the RF systems at the Free Electron Laser in Hamburg (FLASH). This led to a variety of firmware versions as well as control system programs and display panels.

A joined attempt of the LLRF and the controls group was made over the last year to consolidate hardware, improve the firmware and develop one DOOCS front-end server for all 6 RF stations. Furthermore, DOOCS standard server are used for automation, like simple state machines, and the FLASH DAQ for bunch-to-bunch monitoring tasks, e.g. quench-detection.

An outlook of new developments for the upcoming European XFEL, using xTCA technologies, will be given.

INTRODUCTION

Over the last 15 years FLASH has evolved from a small test facility with a gun and one 8 cavity-accelerator module, running at about 100 MeV, to a photon science user facility. After the last shutdown in 2009/10 FLASH has been upgraded to 7 accelerator modules with eight 1.3 GHz cavities each, plus a 3rd harmonic module with four 3.9 GHz cavities. This set-up allows FLASH to run at a maximum beam energy of about 1.2 GeV. Presently, six RF stations are required to supply the gun, the 3rd harmonic- and the seven 1.3 GHz modules with RF.

Over this long period, the controls for the Low-Level RF (LLRF) evolved alongside the modifications of the accelerator. Many different flavours of LLRF controller hardware, starting from a pure analogue-based system for the first gun, a successfully used DSP[1] system for the modules and different versions of Simcon and SimconDSP[2] systems were developed. All these systems came with dedicated firmware, device server software and operator display panels, leading to a very inhomogeneous, global control system. Such a system was hard to maintain and applying global automation procedures was very difficult, because of the different structure and naming convention of every device server.

The effort to consolidate the LLRF system during the last shutdown will be described.

DOOCS

The Distributed Object Oriented Control System DOOCS[2] is the leading system for the FLASH accelerator. DOOCS is a standard client/server control system and based on an object-oriented approach at the front-end/server and client/display side. It is mainly implemented in C++, but there is now a Java client-side implementation called jDOOCS, on which the new display tool jDDD[3] is based. An interface for MATLAB clients is provided. The communication protocol is based on ONC Remote Procedure Calls (RPC), but a strong effort is on the way to replace them by the TINE[2] protocol.

HARDWARE

In order to achieve a homogeneous LLRF system, it is very important to start at the hardware level already. It was decided to use only two types of SimconDSP VME boards, which are equipped with ten 14 bit ADCs. One type has a Virtex V50 FPGA from Xilinx installed, which is suitable to run all control algorithms needed and is used as master card. If only additional analogue I/O is required, a SimconDSP board, equipped with a Virtex V40 is used as a slave card. The two boards are interconnected via 1 Gb fibre link to exchange the real-time data.[4]

FIRMWARE

After coming up with a common hardware platform, only a few different version of the FPGA firmware are needed, which have many parts in common, like the VME interface structure. The VME part has been optimized to allow the new 10 Hz operation of FLASH. A mapping file is provided for all VME register and tables allowing to change the firmware independent from the device server. The following firmware versions are needed :

- RF gun: This version is special, because the RF gun has no hardware probe signal. This has to be calculated from the forward and reflected power signals. In addition, the gun is a normal conducting cavity, which requires different control algorithms.
- Master board: This version includes all LLRF control and regulation algorithm as well as beam based feedbacks.
- Slave board: A simplified version to readout the ADCs and calculate the partial vector-sum is needed.

AN ORBIT FEEDBACK FOR THE FREE ELECTRON LASER IN HAMBURG (FLASH)

R. Kammering DESY (Hamburg, Germany), John Carwardine ANL (Argonne, IL, USA)

Abstract

The lack of knowledge of the exact energy profile of the Free Electron Laser in Hamburg (FLASH) and thereby of the orbit response matrix, made the implementation of a conventional orbit feedback in the past very difficult.

The new run period started this spring after extensive modifications of the facility, showed that the responses matrixes seam now to be in good agreement with the theory, thereby allowing the application of standard orbit feedback techniques.

The physics concepts and the chosen architecture to implement such software on the middle layer and interplay with other high-level software components will be discussed. The development and implementation of this software using the DOOCS servers in combination with the dynamic components of the Java DOOCS data display (jddd) allowed a flexible and scalable implementation, which could also serve as a prototype for future implementations at e.g. the European XFEL.

MOTIVATION

The task of stabilizing beam jitter, as it is the case at most synchrotron radiation facilities, is not feasible for the FLASH linac, because the orbit can only be sampled at the maximum of pulse repetition rate of 10 Hz.

So the task of compensating fast-varying errors, for example in magnetic fields of corrector magnets or vibrations due to ground movements is here not the main focus of this orbit feedback implementation.

Instead of this the main objectives for an orbit feedback at a linear accelerator are to:

- restore saved orbits
- compensate long-term drifts
- stabilize the orbit downstream while tuning the machine further upstream
- making localized orbit changes

These are only the most important objectives an orbit feedback could attack. For FLASH it is even further envisioned to change the today practice of using individual steerers (dipole magnets) to tweak the orbit at a certain position along the machine (we will call this the *longitudinal position* in what follows), but instead of this modify beam positions using the orbit feedbacks target values at this longitudinal position.

BASIC SCHEMA OF A BEAM BASED ORBIT FEEDBACK

The basic principle of the FLASH beam based orbit feedback follows the standard techniques as e.g. described in [1]. A linear response matrix (**R**) describes the action of small changes ($\Delta\mathbf{I} = [\Delta\mathbf{h},\Delta\mathbf{v}]$) in the corrector magnet fields (dipoles) on the beam position ($\Delta\mathbf{X} = [\Delta\mathbf{x},\Delta\mathbf{y}]$) measured at the beam position monitors (BPMs).

$$\mathbf{R}\,\Delta\mathbf{I} = \Delta\mathbf{X}$$

Inverting the response matrix allows to derive the needed values to be applied to the correctors to yield a certain change in the beam position. In cases of unequal numbers of BPMs and correctors, the response matrix is non-square which can be inverted using the pseudo inverse or singular value decomposition.

$$\mathbf{I}_j = g\,\mathbf{R}^{-1}(\mathbf{X}_{ref} - \mathbf{X}_{meas}) + \mathbf{I}_{j-1}$$

With the gain factor g = 1 this would lead to a full correction of a given difference between the desired $\mathbf{X}_{ref}$ and actual beam position $\mathbf{X}_{meas}$, if the new current $\mathbf{I}_j$ will be written to the correctors in step j. One will usually work with a gain factor << 1 and also apply some filtering to the $\mathbf{X}_{meas}$ data to avoid ringing and overcorrection.

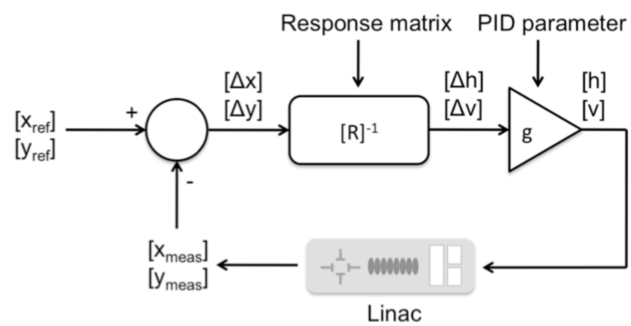

Figure 1: Basic structure of the beam based orbit feedback loop

ARCHITECTURE

The main objectives of the FLASH orbit feedback are not to damp high frequency position jitter, but more to assist operation and decouple actions within the machine.

Therefore it was planned from the beginning to implement this as a pure software feedback with moderate operation frequency (0.5-2 Hz).

The basic architecture for such a software-based feedback therefore follows the classical design of a middle layer server as described in the following section.

DOOCS as basic software infrastructure

The dominant control system at the FLASH facility is the Distributed Object Oriented Control System (DOOCS) [2]. Therefore a logical choice for the orbit feedback is to implement this software using C++ and the existing DOOCS application-programming interface (API). DOOCS offers a natural mapping of the monitors (BPMs) and correctors (steerers) to C++ objects, which significantly eases working with many devices, and thus understandablity of the code.

STATUS, APPLICABILITY AND PERSPECTIVE OF TINE-POWERED VIDEO SYSTEM, RELEASE 3

Stefan Weisse, David Melkumyan (DESY, Zeuthen)
Philip Duval (DESY, Hamburg)

Abstract

Experience has shown that imaging software and hardware installations at accelerator facilities needs to be changed, adapted and updated on a semi-permanent basis. On this premise, the component-based core architecture of Video System 3 was founded. In design and implementation, emphasis was, is, and will be put on flexibility, performance, low latency, modularity, interoperability, use of open source, ease of use as well as reuse, good documentation and multi-platform capability. Special effort was spent on shaping the components so that they can easily fit into small-scale but also into area-wide installations.

Here, we describe the current status of the redesigned, almost feature-complete Video System, Release 3. Individual production-level use-cases at Hasylab [1], PITZ [2] and Petra III [3] diagnostic beamline will be outlined, demonstrating the applicability at real world installations. Finally, the near and far future expectations will be presented.

Last but not least it must be mentioned that although the implementation of Release 3 is integrated into the TINE control system [4], it is modular enough so that integration into other control systems can be considered.

OVERVIEW

The origin of the featured Video System 3 (VSv3) is the Photo Injector Test Facility Zeuthen (PITZ). It is a test facility at DESY for research and development on laser driven electron sources for Free Electron Lasers (FEL) and linear colliders [5, 6].

Currently, VSv3 is almost feature-complete. Since 2008, it has emerged out of its predecessor [7], now known as Video System 2 (VSv2). The current software is a result of more than 10 years experience on video controls at particle accelerators.

As the lifetime of an accelerator facility can be a few years or decades, in contrast to the fast-pace IT world, a few design criteria should be kept in mind. Some API or operating systems can be potentially obsolete just a few years after commissioning. Both environmental considerations (radiation level) and customer demands can require frequent exchange of components and/or software evolution and upgrades. Thus there is a strong motivation to incorporate flexibility, modularity and interoperability in the design.

VSv3 was designed and implemented to meet all of these requirements, as well as those general requirements any video system must meet. These include high performance and low latency.

Selection of key characteristics/capabilities:

- raw greyscale images up to 16 bits per pixel
- raw colour images (24 bit RGB)
- integrated JPEG compression/decompression (grey and colour)
- production-level interfaces and experience in operation of: Prosilica GigE cameras, analogue cameras, JAI GigE cameras, JAI/Pulnix GigE cameras and equipment possible to attach using MS Directshow interface (Webcams etc.)
- high-bandwidth possible [8]
- low latency possible (what you steer is what you get)
- production-level 1.4 megapixel transfer, 16 bit grey, at 10 Hz update rate
- up to 30 frames per second can easily be reached
- Area of Interest (AOI)-only transfer
- shared memory interconnection of server-side components
- multicasting of video images

COMPONENTS

The video system comprises of several different components, selected ones are described in details below (see Figure 1).

The **VSv3 Transport Layer** (VSv3 TL) specifies the layout of a well-defined flexible image data type (header and bits) plus ways of transport which is integrated but not limited to TINE control system. Structure, header fields and pixel data formats are well documented.

Small Grabber Part (SGP) is the central front-end server-side component to acquire video images. To keep the C++ code simple, one SGP process will deal with only one camera at a given time. Various editions of SGP exist. Edition means it supports exactly one API to interface image sources / hardware. Most important editions at the moment are Prosilica, JAI and MS Directshow SDK, all on Windows platform. The C++ source code is kept platform independent as much as possible and references only widely available open source libraries. Thus, migration to other operating systems is expected to be on the order of hours or days. This of course depends on the availability of SDK for the chosen platform.

The connection from image source to SGP can be switched from one image source to another remotely. For example, if only two video streams are wanted in parallel, 20 cameras can be supported with just two SGP server processes. SGP provides one TINE control system output interface with VSv3 TL and one interface to shared memory (SHM).

THE FERMI@ELETTRA CCD IMAGE ACQUISITION SYSTEM

G. Gaio, F. Asnicar, L. Pivetta, G. Scalamera, Sincrotrone Trieste S.C.p.A. ELETTRA

Abstract

FERMI@Elettra is a new 4th generation light source based on a linac-driven Free Electron Laser (FEL) which is currently being built in Trieste, Italy. The CCD image acquisition system is a fundamental diagnostic tool for the commissioning of the new accelerator. It is used for the characterization and tuning of the laser, electron and photon beams. The Tango based software architecture, the soft real-time performance and the embedded image processing algorithms are described.

ACQUISITION SYSTEM

CCD

Three Basler CCD cameras (model scA780-54, scA1390-17 and scA1400-17) are currently integrated in the image acquisition system. All of them provide a Gigabit Ethernet connection and a hardware trigger input for the synchronization, and mainly differ for the number of pixels.

A total of 84 CCD cameras are installed:

- 16 are dedicated to the diagnostics of the photo-injector and seed lasers; their purpose is the measurement of the laser beam trajectory along the optical path and the characterization of the laser beam profile;
- 52 are integrated in the fluorescent screen system, which allows the analysis of the electron and photon beams along the linac and the FEL undulators;
- 16 are installed in the photon beam transport system and will be used for the measurement of the parameters of the photon beam provided to the experimental stations.

Up to 18 among the above mentioned CCD cameras have to be concurrently and continuously acquired.

Image servers

In the final configuration five server computers will take care of the acquisition of all the CCD cameras.

Each of them consists of a one-unit 19-inch rack mount server configured with two Xeon QuadCore 3.0GHz processors, 4Gb of DDR3 RAM and up to six Gigabit Ethernet links. One of them is connected to the control system network, three are dedicated to the acquisition of the CCDs and one is used for the real-time communication through the Network Reflective Memory (NRM) [1].

The servers run a GNU/Linux 2.6 kernel patched by the Xenomai real-time extension [2], which provides them with deterministic capabilities. This is used in particular by the Ethernet driver to share time-critical data among the control system computers using the NRM.

IMAGE PROCESSING

For each CCD, a Tango [3] device server is dedicated to the control of the main parameters like exposure and gain, performs the image processing and makes the results available to client applications running in the control room.

Performance and flexibility to adapt to the beam changes are the requirements that the processing software have to fulfil. The performance must guarantee to meet the deadlines because the acquisition and analysis of the image have to be done shot-by-shot. The maximum repetition rate of the linac is 50Hz, which means that a maximum of 20 ms is available to process each image. For this reason, it is convenient to analyze only the portion of image containing the beam profile, conventionally called Region Of Interest (ROI).

The image processing is divided into three steps: automatic ROI detection, calculation of the beam profile moments and data storing with a precise timestamp.

Automatic ROI Detection

Searching the beam spot inside an image could be a complicated task. Sometimes it is easier to find the parts of the image where there is no beam instead, i.e. to define the background.

In order to perform the ROI detection efficiently, the full scale image is under-sampled. The resulting samples size should be at least twice the minimum size of the beam spot in both planes in order to have at least a few points of the beam in the under-sampled image.

If necessary, the image is smoothed by a low pass filter to mitigate the presence of artifacts. A thorough design of the low pass filter parameters can dramatically enhance the magnitude of the beam profile with respect to the noise due to reflections on the vacuum pipe surface (Fig. 1).

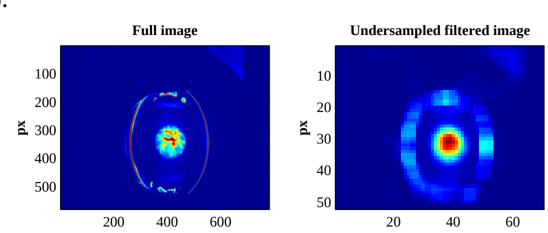

Figure 1: image of the electron beam measured at the exit of the photo-cathod gun, before (left) and after (right) the under-sampling/filtering process.

The background level is estimated through the analysis of the complementary cumulative distribution function of P(*X*<=*x*), which represents the probability that a pixel value *X* is lower then *x*. This task is performed in three steps:

EPICS APPLICATIONS IN THE CONTROL OF SPES TARGET LABORATORY

M. Giacchini, A. Andrighetto, G. Bassato, N. Conforto, L. Giovannini,
INFN, Laboratori Nazionali di Legnaro, Legnaro (Padova), Italy

Abstract

The project of a new facility for the Selective Production of Exotic Species (SPES) has started at LNL. Radioactive ions will be produced by impinging an UCx target by a 70MeV, 200μA proton beam delivered by a commercial cyclotron. Then, the unstable ions will be accelerated by injecting them into the LNL superconducting LINAC. The construction of Target and Ion source prototype (Fig. 1) is at an advanced stage and, after more than two years spent in its construction, preliminary extraction tests were carried out with non-radioactive beams. The control of Target instrumentation is based on EPICS; we describe here the basic choices on hardware and software tools on both IOC and client side and give a brief description of last developments.

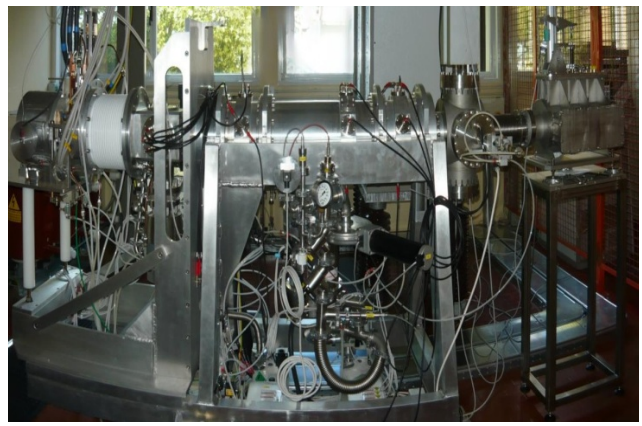

Figure 1: The target chamber and ion source

THE TARGET LABORATORY INSTRUMENTATION

The target instrumentation controls the beam extraction and transport up to a diagnostic station where the physical characteristics of the beam are measured. The beam production is obtained by heating the target to a temperature of about 2000 C, necessary for the optimal extraction of ionized fragments.

The power required for heating is delivered by an array of high current power supplies (LAMBDA GENESYS series) configured in a master-slave chain and providing a current in excess of 1300 A. Other heating methods are foreseen for the future (i.e. using a laser or a microwave source) but currently only the ohmic dissipation has been used. Once extracted, the beam is focused by an electrostatic lens of three quadrupoles f ed by a set of HV bipolar power supplies (a special assembly of rack mount units manufactured by Ultravolt).

The target and the power supplies are placed on an insulated platform that is brought at about 60KV from ground by a FUG (HCP series) power supply. The electrical power required by GENESIS modules is transferred to the platform through a 20KW insulation transformer. An Ethernet transceiver from copper (100Base-T) to fiber optic is used to link the control network to the instrumentation placed over the HV platform; a multi-port Ethernet to serial converter (Comtrol Device Master) is then used to connect the devices equipped with a serial interface.

CONTROL DEVICES

The devices used to control the beam production and extraction are Linux-based IOCs. The LAMBDA-GENESYS master unit has a serial RS232 link to the host controller, which is a standard PC running on CentOS Linux. This OS distribution has been chosen because it is open-source, stable and completely compatible with RedHat. The device support is derived, with minimal modifications, from the driver developed at PSI, based on StreamDevice[1]. The HV power supplies (both Ultravolt and FUG) have an analog interface and are controlled by means of three microIOCs manufactured by Cosylab (SI).

These devices are embedded controllers based on a PC104 board and running under Debian Linux (preloaded on a flash disk). Each unit has three I/O boards, providing an adequate number of analog and digital I/O channels.
EPICS drivers and debugging utilities come built-in with the controller software.

BEAM DIAGNOSTICS

A diagnostic station has been placed at the output of the electrostatic triplet to measure the beam current and profile. The beam current is measured by means of a faraday cup, while the profile is reconstructed by sampling the currents acquired by a set of horizontal and vertical grids. Stepper motors are used to insert/extract the devices along the beam line. The data acquisition system is implemented in a VME crate and runs under Vxworks.

SOFT REAL TIME CONTROL WITH CLIENT/SERVER CONTROL SYSTEM

Y. Furukawa, Spring-8/JASRI, 1-1-1 Kouto, Sayo-cho, Hyogo, JAPAN.

Abstract

Real-time properties have studied for client/server control system on single CPU system with Linux and Solaris operating system (OS) with real-time scheduler. Time jitters were within one msec for Linux OS and for Solaris OS on the MADOCA control system[1] that is the SPring-8 standard control system (CPU was 1.6GHz Intel Atom processor). These results are small enough for many synchrotron radiation experiments such as x-ray diffraction experiments with continuous scanning method. The client application can be described using scripting language, so real-time applications are developed and modified easily. The system has been used in the diffuse scattering beamline at the SPring-8.

INRODUCTION

There are many request on real time controls with msec order time resolution on synchrotron radiation experiments, such as scanning micro probe XRF, continuous scanning x-ray diffraction experiments, etc. In these applications, exact timing is not required because the counting results can be normalized by each step time or integrated intensity of incident x-ray. So the sub-msec order soft real time controls are suitable for these appllications.

To realize real-time application, real time operating system (OSs) has been used, it is, however, difficult to develop the real time applications on theses OSs because it required low-level (device driver or kernel level) software development and there are poor development support tools.

Modern OSs, like Linux or Solaris, have been improved its real time properties and became to be used for real time applications. Under these OSs, soft real time can be realized only set the framework software and these applications to use real time schedulers, such as RT-class on Solaris or FIFO and round robin scheduler on Linux.

There are many single program implementations to realize the real time properties. It requires the detailed knowledge for device control libraries and frame work, it is hard task for x-ray beamline scientist because most of them are not specialist of the control software.

If real time applications can be described using simple scripting languages, many non control specialist can develop the real time applications. It is possible if the client/server type system provides real time properties. In this paper, results of the real time property measurements in the case of the MADOCA control system on the single CPU system and it has enough for the synchrotron radiation experiments.

MEASUREMENTS OF THE REAL TIME PROPERTIES

Real time property measurements were made on Solaris 10 and Linux (vanilla kernel 2.6.34 and real time patch[2] applied kernel 2.6.33.7-rt29). In the Solaris case, parameter hires_tick=1 was set in /etc/sysconfig for 1 msec tick. For the Linux case, tickless kernel and 100Hz tick were set in kernel parameters. All the software were installed on the Atom Z530 (1.6GHz) processor based control sysmte called "Blanc-4" developed at the SPring-8[3]. The blanc-4 has 512MByte main memory and 16Gbye flash memory based storage. All the softwares were set RT-class in the Solaris case (using priocntl command) or FIFO scheduling for the both Linux case (using chrt command).

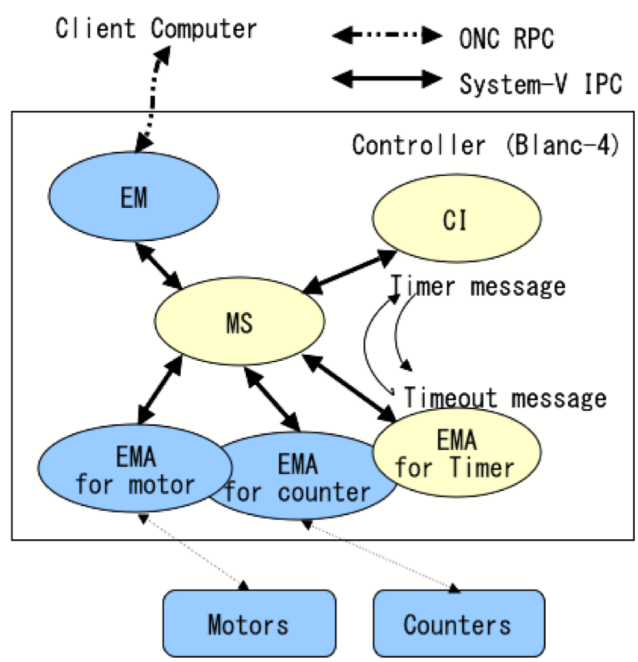

Figure 1: Software scheme of the measurements.

Software scheme based on the MADOCA control framework is shown in Fig.1. Each program communicate using system-V IPC (message queue). Command Interpreter (CI)[4], used as a client software, issued messages to the Message Server (MS). The MS transfers the control message to the Equipment Manager Agent (EMA) which controls actual devices and send back a result message to the CI via the MS. In the measurement, the EM was set as a timer, which returns a result message to the client (CI) after sleeping a given time by the message from the CI as shown in Fig. 2. The time

STARS ON PLC

T. Kosuge, K. Nigorikawa, KEK, Japan

Abstract

The Simple Transmission and Retrieval System (STARS) [1][2] is a message transfer software for small-scale control systems having TCP/IP sockets; STARS can work on various types of operating systems. In this study, we have successfully run the STARS server and client on the F3RP61 (Yokogawa Electric Corporation).

At present, PLCs are used for beamline interlock systems (BLISs) and PCs are used for monitoring and permission control system (CCS) of BLISs at the Photon Factory. Running STARS on a PLC makes the integration of BLIS and CCS possible. This paper provides a detailed description of the process of running STARS on a PLC.

BLIS AND CCS

Over 20 beamlines are in use at the Photon Factory and each beamline has a beamline interlock system (BLIS) for ensuring radiation safety and maintaining a vacuum environment in the beamline (Fig. 1). A PLC is used as a controller for the BLIS; it controls the beamline components (beam shutters, experimental hatches, gate valves, vacuum gases, etc.).

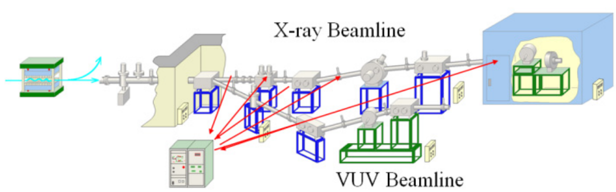

Figure 1: Beamline and BLIS.

The CCS monitors the status of BLIS and controls the permission signal, which permits beamline usage, through the PLC interface installed in each beamline (Fig. 2).

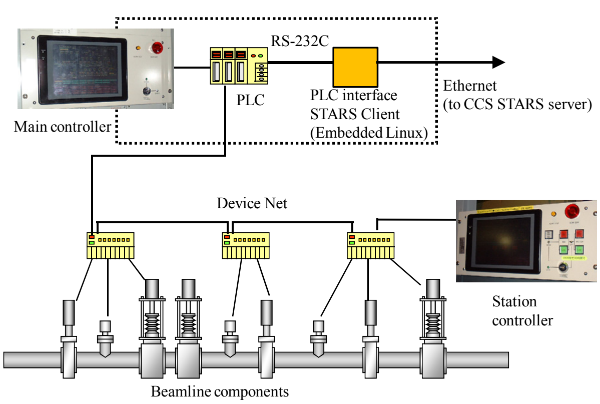

Figure 2: BLIS and PLC interface of CCS.

At present, the RS-232C is used for communication between the BLIS and PLC interfaces. The number of monitoring points that the CCS can support is limited because of the low speed of communication. Integration of the BLIS and PLC interfaces is one of the solutions to this problem.

F3RP61

F3RP61 (e-RT3 2.0/Linux) is a CPU module that can be installed on the Yokogawa FA-M3, which also has EPICS running on it [3]. In this study, we used F3RP61-2L as a test bench (Fig. 3).

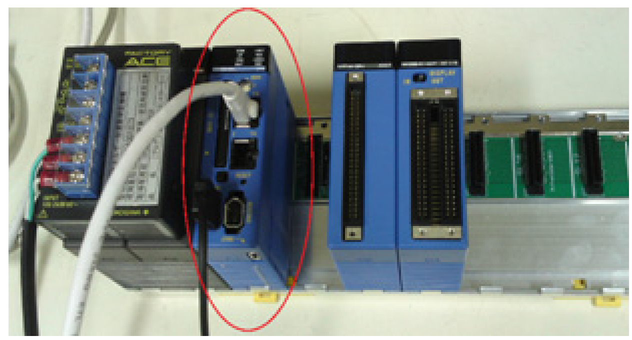

Figure 3: F3RP61 on FA-M3.

STARS

STARS is an extremely simple software for small-scale control systems having TCP/IP sockets as well as the provision for text-based message transfers (Fig. 4). A STARS server can work on various types of operating systems (the STARS server is written in Perl).

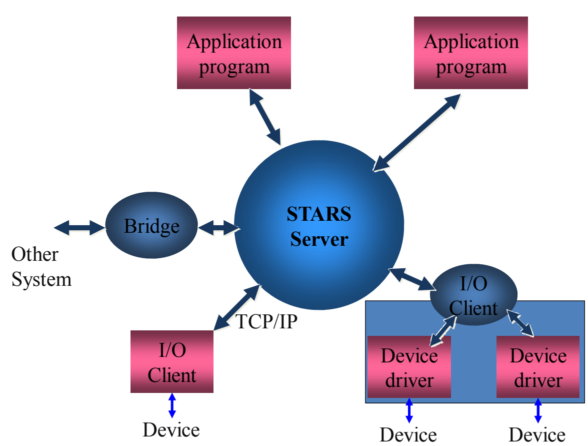

Figure 4: STARS server and clients.

STARS consists of client programs (STARS clients) and a server program (STARS server). Each client is

IMPROVEMENTS FOR SIMPLE OPERATION AT SAGA-LS ACCELERATOR

Y. Iwasaki#, T. Kaneyasu, Y. Takabayashi, S. Koda, SAGA Light Source, Saga, Japan

Abstract

The SAGA Light Source is a medium-size synchrotron light research facility located at Kyusyu Island, Japan. The control system of the SAGA Light Source has been developed in the early phase of the machine commissioning. The application programs were developed using PC-LabVIEW. Commercial off-the-shelf input/output devices, such as PLC with a MS-Windows PC server, compose the input output controller with a high cost-performance ratio. ActiveX CA is used for the communication protocol between the server PCs and the client PCs. All of the components of the accelerator except the timing system are now controlled using PCs. Although the control system is stable, having many client PCs complicated the daily operation. Thus, we developed a multi-purpose client program, which is running on MS-Window 7 with a touch panel display. Furthermore, we constructed communication interface between the accelerator control system and the radiation interlock system to set the interlock mode from the accelerator control system. By using the developed multi-purpose client program and the interface to the radiation interlock system, the numbers of procedures necessary for daily accelerator operation have been significantly reduced, making the daily operation simple.

SAGA-LS CONTROL SYSTEM

The SAGA Light Source (SAGA-LS) is a medium-size synchrotron light research facility located at Kyusyu Island, Japan, and the accelerator consists of a 255 MeV injector linac and 1.4 GeV electron storage ring [1], [2]. At this time, all of the accelerator components are controlled by a digital system except for the timing system. For connectivity to the accelerator hardware, we selected commercial off-the-shelf distributed input/output (I/O) devices, such as a programmable logic controller (PLC) (Yokogawa: FA-M3) and distributed I/O controller devices (National Instruments: Fieldpoint). A difficulty at the SAGA-LS facility is its tightly restricted budget, which limits the number of staff in the facility. Thus, the control system for SAGA-LS should be simple and robust, yet inexpensive, easy to develop, and easy to maintain. One of the solutions to this problem is the use of off-the-shelf products, including PCs. The off-the-shelf I/O device and server PC works as the PC Input Output Controller (PC-IOC). Figure 1 shows a schematic view of the control layer of the SAGA-LS control system. For clarity, many of the accelerator components are omitted. We developed applications in the PC-LabVIEW environment because accelerator staffs are familiar with the PC-LabVIEW.

The PC-based control system is widely used in many facilities because of the high cost-performance ratio of using PCs. Especially recent improvements in the performance and the cost effectiveness of PCs have made them attractive for use in the accelerator control system. There are sophisticated and well-established control systems based on workstations or PC-UNIX, such as the Experimental Physics and Industrial Control System (EPICS). However, it is difficult to modify and expand the EPICS system with limited accelerator staff. Fortunately, the number of control items of the SAGA-LS is now approximately 600 and there are very few demands for real-time control. The only exception is the synchronous operation of power supplies for the four minutes of the energy ramping in the storage ring. In this case, a PLC with a preloaded ramping pattern is suitable. Hence, we designed a MS-Windows PC-based control system with off-the-shelf I/O devices [3], [4]. For the communication protocol between the server PCs and the client PCs, we used ActiveX channel access (CA) [5], which emulates the EPICS CA protocol. MySQL was adopted as the database system. Recent progress on the control system for both the linac and the insertion devices are summarized in reference [6]. The feedback control system for the magnet power supplies using external DC current transformer, feed-forward orbit, tune and coupling correction systems have been developed in past years.

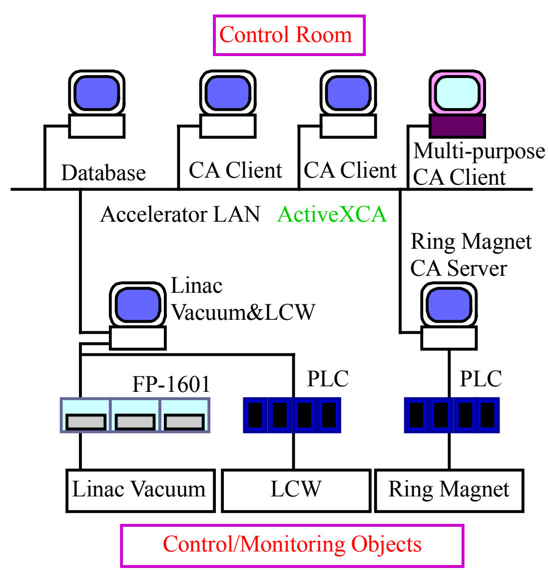

Figure 1: Schematic view of the control layer of the SAGA-LS.

#iwasaki@saga-ls.jp

CONTROL AND TIMING SYSTEM DESIGN OF CPHS *

Qiang Du#, Hui Gong, Xialing Guan, Jie Wei, Jianmin Li, Beibei Shao
Department of engineering physics, Tsinghua University, Beijing 100084, China

Abstract

The Compact Pulsed Hadron Source (CPHS) in Tsinghua University is designed as a university based comprehensive hadron research and application platform. This paper describes the control and timing system of CPHS.

INTRODUCTION

The project of CPHS in Tsinghua University consists of an accelerator front-end—a high-intensity ion source, a 3 MeV radiofrequency quadrupole linac (RFQ), and a 13 MeV drift-tube linac (DTL), a neutron target station—a beryllium target with solid methane and room-temperature water moderators/reflector, and experimental stations for neutron imaging/radiography, small-angle scattering, and proton irradiation. [1,2]

The control system of CPHS consists of an EPICS (Experimental Physics and Industrial Control System) based distributed run-time database and control system, a timing and event distribution system, and a digital low level RF control system.

The timing and event distribution system defines the global system time frame as well as specific events that trigger local devices by an event generator and receiver framework, so that the time delay of each event could be controlled in 10ns resolution, and the timing jitter of trigger signal is below 0.1ns. The hard-real-time machine protection system is also integrated in the event system so that a fault event could be responded within 50 micro-seconds. Field control signals such as water temperature, vacuum level, magnetic current, beam diagnostics, and low level RF (LLRF) phase and amplitude are monitored and controlled via the EPICS database through Ethernet.

EPICS BASED CONTROL SYSTEM

Control System General Layout

As shown in Fig 1, the EPICS control system uses several input/output controllers (IOC) to manage local process variables and establish a distributed database. The IOCs are running Linux/RTEMS kernels with device support of different local bus interfaces (serial, GPIB, stepper motor, DAQ modules, etc), communicating with local instruments monitoring and controlling water temperature, power supply, vacuum status, and LLRF status. All EPICS records are accessible from control room via Ethernet by Channel Access protocol, and are managed through Operator Interfaces (OPI) for monitoring, data logging, alarm handling, and some interlocking control. The application server and development server are responsible of providing dhcpd/bootp/nfs services for net-booting IOCs and maintaining IOC kernels, IOC applications, bootup scripts and EPICS records.

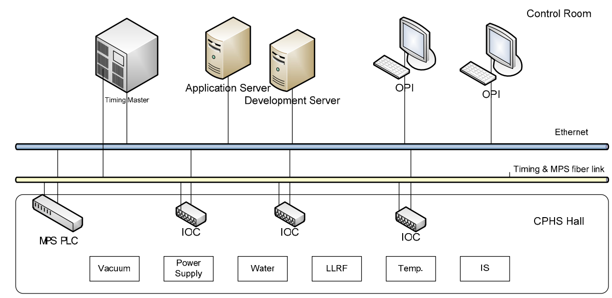

Figure 1: EPICS control system for CPHS

TIMING AND EVENT DISTRIBUTION SYSTEM

Timing System General Layout

CPHS timing events are generated, encoded and distributed through optic fiber at 108.3MHz rate (325MHz divided by 3), and then decoded by different local receivers. (Fig 3.)

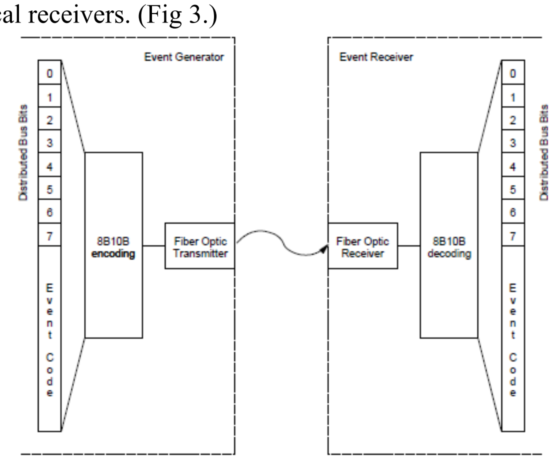

Figure 2: Event frame [3]

The event generator (EVG) is responsible of creating and sending out timing events to an array of event receivers through a fanout module. The event transfer rate is derived from the linac RF master frequency at 325MHz. The EVG is also capable of synchronizing to the AC line at 50Hz and phase delay to adjust the triggering position relative to the main voltage phase.

*Work supported by “985 Project” of the Ministry of Education of China, CAS Sciences Hundred People Initiative (KJCX2-YW-N22)
#duqiang@tsinghua.edu.cn

TINE/ACOP STATE-OF-THE-ART VIDEO CONTROLS AT PETRA III

J. Bobnar, I. Križnar, T. Kusterle, Cosylab, Ljubljana, Slovenia
D. Melkumyan, S. Weisse, DESY Zeuthen, Zeuthen, Germany
P. Duval, G. Kube, J. Wilgen, DESY, Hamburg, Germany

Abstract

The TINE/ACOP video system is a complete state-of-the-art solution for streaming beam video, featuring live analysis and live beam image display inside ACOP video component, which can be placed in any Java Swing panel. After a number of iterative improvements and embellishments, the system has matured to stable production quality in the beginning of year 2010. The system consists of the following components: a TINE device server captures a video image [1] and encodes it to the standard TINE IMAGE format. The TINE transport layer streams the IMAGE objects to clients as it would any other data chunk [2]. The Java TINE client passes the IMAGE object through the analysis Java bean, which then performs fast statistical analysis of beam position and size. The streamed image plus analysis data are displayed in the Java video component, which is part of the ACOP components. Additional capabilities are background subtraction, automatic or manual threshold subtraction, enhanced coloring and saving snapshot as PNG file. Optionally, the analysis bean can be used standalone as a common service and results are further distributed via an intermediate TINE server written in Java.

INTRODUCTION

The origin of the TINE Video System goes back to the design of the Photo Injector Test Facility Zeuthen (PITZ), which is a test facility for research and development on laser driven electron sources for Free Electron Lasers and linear colliders [1]. The optimization of an electron gun is only possible with the help of an extensive diagnostic system, including the video system.

The whole video system includes a rich set of components, covering the low level hardware integration and image grabbing, to the transport protocol and data visualization tools.

In this article we will focus on the upper level of components, which have recently been upgraded and put to use also at DESY Hamburg.

DATA ACQUISITION AND TRANSPORT

The image acquisition is implemented in a grabber server written in C++. The main purpose of this server is to acquire grayscale images from the image source and pre-process the data (e.g. compression).

The transfer of the high resolution image (up to 2 megapixels) is done using the TINE transport protocol. TINE allows various choices of data transport including multicasting, unicast UDP and TCP. Combining this with compression algorithms the TINE video system easily achieves updates at 10 frames per second.

The image transported by TINE is packed into a dedicated IMAGE data type, which is composed of an image header providing meta information about the image (frame size, bit depth etc.) and the actual image data of variable size – TINE is not limited to the transport of a fixed size image, but can be used to transfer any size one desires (within the limits of the network traffic). The IMAGE data type can also be embedded within TINE structures and is used as a standard method of exchanging image data between video system components.

IMAGE VISUALIZATION AND ANALYSIS

Java has been selected as the target platform/technology for the video system clients. The client side is responsible for visualization of the image as well as performing the data analysis and processing of the image data. In some respects we might expect Java to reduce the execution speed of the software, which would be a trade off for platform independence. This does in fact play a role regarding for instance graphics or low-level networking functionality. However, due to the high processing power of today's desktop computers, this is no longer a serious drawback and Java has proven to be very powerful and easy to use for writing the video clients.

A dedicated AcopVideo bean has been implemented, following the conventions and standards of the ACOP framework [3]. This automagically provides some common functions and tools (e.g. connection selection, drag and drop), as well as makes it easy for other developers to provide rich-clients that deal with the video.

The AcopVideo bean was implemented in pure Java, which means that it doesn't use any native resources (besides the standard ones provided by JVM) and is completely platform independent. The AcopVideo bean was designed with performance in mind, which drove the architecture and implementation of the drawing algorithm. The performance of the video bean today easily satisfies the requirements of the operation control.

In addition to high performance, the video bean provides much functionality, which is not available in the older native or commercial video clients. The AcopVideo can display any TINE video channel or still image, which can be either loaded from several standard image files (JPEG, PNG, etc.) and quality (8 to 24 bits per pixel), or provided through the TINE channel (using the event notification system in order to minimize the necessary network traffic). The AcopVideo also offers several other options for image visualization and enhancements, such as different color modes for luminosity data, histogram equalization, aspect ratio changes and zooming, display of meta information etc.

APPLICABILITY OF XAL FOR ESS

Jaka Bobnar, Cosylab, Ljubljana, Slovenia
Steve Peggs and Charles Garrett Trahern, ESS, Lund, Sweden
Todd Satogata, Jefferson Lab, Newport News, Virginia, U.S.A.
Thomas Pelaia II and Christopher K. Allen, ORNL, Oak Ridge, Tennessee, U.S.A.

Abstract

XAL is a Java-based application framework, developed at the Spallation Neutron Source (SNS). The framework is designed to provide an accelerator physics programming interface to the accelerator, and it allows creation of general-purpose applications dedicated to various parts of the accelerator.

The backbone of the XAL framework is an XML-based description of the accelerator. The XML file provides the list of all devices, their properties, and relationships between devices within the system. Since the accelerator structure is defined in the relational database, XML can be generated directly from the database using appropriate adapters. This allows the framework to be more generic and enables it to run on different sites using various configurations.

The generality of XAL and the rich set of applications and tools provided by SNS make the framework very appealing for use at other accelerator sites. The European Spallation Source (ESS) is being built in Sweden, and is similar in complexity to the SNS. XAL has therefore been considered for use at ESS for high-level applications. The applicability of XAL and prototyping for ESS are discussed in this article.

INTRODUCTION

The XAL framework was developed by SNS as a part of their accelerator physics activities. It was designed to provide a common set of tools and applications used in machine physics and accelerator control. Today the framework includes a vast set of applications such as Orbit Correction, Wire Scanner Analysis, Scanning Application etc. These applications are all used in day-to-day activities in the SNS control room.

XAL was designed from the start to be as independent from machine details as possible. Therefore a specific model was defined which provides a detailed description of the accelerator. At start-up the model is parsed and used by the framework to gain access to various parts of the accelerator. The model allows XAL use at different accelerator sites without changing the code, since the model is supplied as a set of configuration files and is the only part of the framework that needs to be adapted.

Recently XAL went under major restructuring in order to make the code even more transparent and to allow easier development of site specific applications and components. ESS, being a similar machine to SNS, appeared as a potential heavy user of this framework (now named Open XAL).

ACCELERATOR MODEL

The backbone of the XAL framework is the accelerator model. The model describes the layout of the accelerator and its parameters as they are used by the applications.

The XAL model is defined in a hierarchical structure within an XML file. This XML file is composed of several different accelerator sequences, which consist of other sequences or components each describing a particular segment of the accelerator. Combined together, they form a hierarchy of the complete accelerator down to every particular device that can affect the beam path. In addition, the XML file also provides all the necessary pieces of information required for the control of a particular physical device. For example, the magnet description includes the strength of the magnetic field, its position within the accelerator, the power supply associated with it etc. [1].

XAL uses EPICS as the underlying control system to communicate with the accelerator hardware. EPICS communication uses a single “Process Variable” (PV) as the fundamental unit for communication with high-level software via a protocol called Channel Access. Therefore, in addition to the physical description of the devices, the XML model also carries information about associations between EPICS PVs and accelerator devices. A single device can have several different PVs, each assigned to one particular device attribute.

Based on the information in the XML file, a Java model is constructed by the XAL upon start-up of an application. Each component within the XML structure is mapped to a Java device object and can be treated as such in XAL applications. Users can set or read the attributes associated with any of those devices simply by changing the value of a particular field in that object, and changes are immediately reflected in the real system through the PV registered for that particular attribute.

Though XAL currently supports only EPICS control system, the underlying mechanism is abstracted so Channel Access can be replaced by other communication protocols. This permits some aspects of XAL to be truly portable between accelerator sites.

Taking into account all the aforementioned pieces of information, one can end up with an enormous XML model, which might be very difficult to maintain. Therefore, XAL works together with the central database, which stores all the required information. A dedicated XAL application gathers the information from the database and generates the XML file. This ensures that the model is always consistent with the accelerator and

CCCP - COSYLAB COMMON CONTROL PLATFORM

Miha Rescic, Cosylab, Ljubljana, Slovenia
Ziga Kroflic, University of Ljubljana, Ljubljana, Slovenia

Abstract

Cosylab common control platform (CCCP) is a lightweight hardware control platform designed to provide a simple interface to various types of hardware components and fast and simple integration of such hardware into control systems. The core of the platform is the scripting language lua. This lightweight and flexible scripting language provides software real-time control of hardware modules over all provided connections (RS232, Ethernet, USB, SPI, CAN, I2C, GPIO) as well as fast and simple ways of implementing modules for more complex structures (FPGA). The platform provides various levels of control with an embedded GUI or full remote control over an embedded web server, archiving capabilities with a database back-end and different device simulator modes. The platform's small footprint, high degree of flexibility and high level of hardware abstraction make the CCCP an ideal control platform for more complicated hardware instruments and at the same time a perfect main control board for devices that incorporate various complex hardware elements. The design and possible implementations of this platform will be discussed in this article.

INTRODUCTION

Development of a control system is never an easy nor a straightforward task. With the complexity of today's technologies, if we're speaking of technologies in general or of technologies applied in specific fields, the number of different components or building blocks of the control systems and the complexity overall grow rapidly.

Within this rapidly expanding field it is very difficult to find a common ground and usually much effort is spent on developing highly specific solutions capable of tackling only a limited array of problems. Thinking of common grounds in control systems field brings to mind a reusable, as generic as possible platform that would represent the base of the control system. This was the motivation behind CCCP: minimize the efforts needed for base platform development and allow emphasis on more specific and complex components development, integration, testing and QA.

ARCHITECTURE

The crucial element of the platform is the architecture. CCCP tries to keep logical entities separated from each other as much as possible. This way, reusability and efficient design are possible.

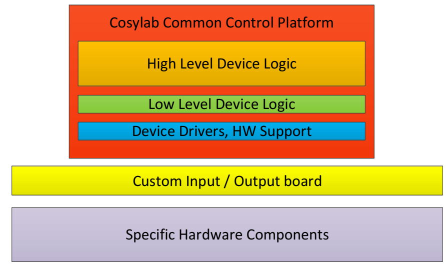

Figure 1: CCCP Architecture

Custom input / output board

On the lowest level of the CCCP architecture is the customized input / output board. Although the board itself is not a part of the CCCP platform it provides problem or component specific solutions regarding hardware connections, specific protocol implementations or more advanced logic (see Fig. 2). The custom board development is bundled together with the CCCP platform development in order to provide the optimum solution for the specific problem.

Some of the IO board's main purposes are described below.

- Target hardware development away from the platform core and towards specific implementation needs.
- Provide advanced logic and (hard) real-time support with FPGA.
- Allow connectivity with existing CCCP IOs or implementation of any custom IO required.
- Minimize the complexity of custom HW development.
- Minimize the amount of redundant development efforts regarding non-reusable hardware.

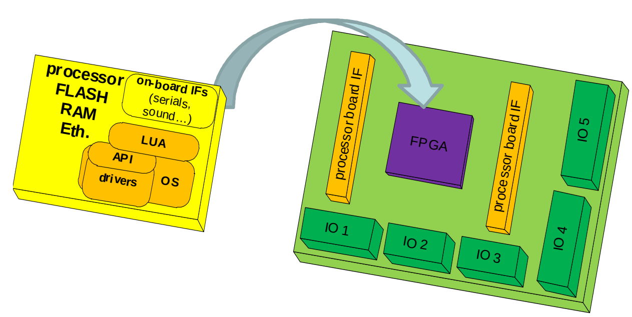

Figure 2: Custom IO board

Device drivers and hardware support

The layer residing directly over the custom IO board, the lowest layer of the CCCP core architecture, provides

PROGRAMMING INTERFACES FOR RECONFIGURABLE INSTRUMENTS

Matej Kenda, Hinko Kočevar, Tomaž Beltram, Aleš Bardorfer, Instrumentation Technologies d.d, Solkan, Slovenia

Abstract

Application Programming Interfaces (APIs) provided by the manufacturers of the instruments for the accelerators are a very important part of the functionality. There are many interface standards (EPICS, TINE, Tango,...) and even same standard can be used in various ways.

Important features of modern instruments are reconfigurability and embedded computing.

The developers of instruments that need to be connected to a control system are facing different requirements: adherence to standard protocols and support of reconfigurable instruments with diverse capabilities with a consistent interface.

Instrumentation Technologies has implemented a well accepted solution with its proprietary Control System Programming Interface (CSPI) layer and adapters for each standard protocol.

There are new challenges like reconfigurability, quality of service, discovery and maintainability that are being addressed with improved Measurement and Control Interface (MCI).

CONTROL SYSTEM AND SOFTWARE INTERFACES

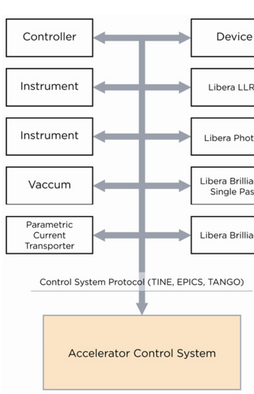

Figure: 1

There are quite some parameters that define environment in which the Control System operates. We can find heterogeneous instruments with different levels of complexity. Beside that the equipment is distributed over large remote regions and needs to provide reliable access regardless of the distance from the control room (see Fig. 1). Another characteristic of such operating environment is that the control is centralized, but the data acquisitions is distributed and to some extent also the data processing.

Based on that we can define interface requirements from the Control System's point that must cover following areas:

- device discovery, identification and capabilities
- operation mode control and configuration parameters
- events, alarms and health state monitoring
- data acquisition and attributes (data type, size, offset, time-stamp)
- error handling

INSTRUMENT MANUFACTURER'S VIEW

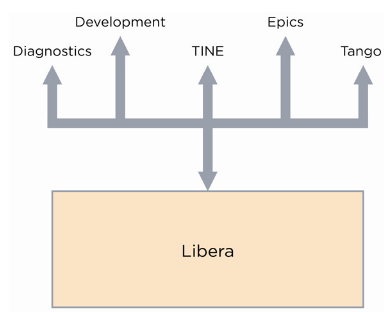

Figure: 2

From the reverse point of view, an instrument can be used in different environments (see Fig. 2). Requests for data can come from different sources for different purposes.

- **Control System**: Different types of control system protocols
- **Other instruments**: Instrument interoperability, multiple instruments working together, clustering, shared processing,
- **Development Lab**: Development, testing of new, updated instruments
- **Maintenance**: Diagnostics, repair

Not all of the access paths are active concurrently.

A great deal of the information access has a common denominator, defined by the type of the information requested.

EMBEDDED COMPUTING

Using embedded computers in the instruments enables instruments to behave as network attached devices with built-in control system interfaces.

Embedded computer can be used to

- **control** the instrument's operation
- perform a part of **digital signal processing**
- provide **remote access** to the instrument

The embedded computer is one of the important components of an instrument, because it provides convenient way to bring all of the parts (hardware modules, FPGA, software) of an instrument together into a working application and perform certain digital signal processing.

Software running on the embedded computer can seen as one of the variable parts of a reconfigurable instrument.

RECONFIGURABLE INSTRUMENTS

Physical setup and behaviour of the instrument is not completely defined during manufacturing.

EPICS IOCCORE REAL TIME PERFORMANCE MEASUREMENTS ON COLDFIRE MODULE*

Shifu Xu#, Hairong Shang, Robert Laird, and Frank Lenkszus
Argonne National Laboratory, Argonne, IL 60439, U.S.A.

Abstract

Since Experimental Physics and Industrial Control System (EPICS) is becoming more widely used in accelerator control systems and the EPICS Input/Output Controller (IOC) has ported to different operating systems, the performance of EPICS IOCcore on different hardware and software platforms is crucial. This paper will provide real-time performance measurements of EPICS IOCcore on a Coldfire module uC5282 and on two different OS platforms: RTEMS 4.9.2 and uClinux 2.6.21. The most recent EPICS base and extensions are used to build the test application.

INTRODUCTION

As more and more Coldfire uC5282 modules are being used at the Advanced Photon Source (APS) and other sites, it is of interest to know the EPICS IOCcore real-time performance on this platform. Similar performance measurements were done on the MVME2100 [1]. Based on the measurement software [2], a few changes have been made to measure on the Coldfire uC5282 module. These real-time parameters are measured on both RTEMS 4.9.2 and uClinux 2.6.21 platforms: interrupt latency, context switch latency, and total response latency. Two more parameters are measured on the uClinux 2.6.21: interrupt top half to bottom half, and interrupt bottom half to user space interrupt service routine (ISR).

MEASUREMENT PLATFORM

All measurements were performed on a Coldfire uC5282 module from Arcturus Networks [3]. The module has a MCF5282 Freescale Coldfire microprocessor with a 64-MHz Coldfire RISC core. It has a 16-Megabyte SDRAM, 4-Megabyte flash memory, and 512-k byte on-chip flash. In order to generate an external interrupt for the module to measure the latency, an APS custom-made Coldfire bridge board and Altera Stratix II development board were used. Figure 1 shows the hardware platform.

The development host machine is an x86-based Linux PC running Fedora Core 10, with a tftp client and an NFS server running on it. The target module's bootloader has a tftp server to receive the OS image.

Two OSs are evaluated on the Coldfire module target: RTEMS 4.9.2 and uClinux 2.6.21. uClinux 2.6.21was downloaded from Arcturus Networks with the non-preemptive kernel. This version includes built-in board support packages (BSPs) for the Coldfire modules. The cross-compiler tools for the uClinux 2.6.21 and applications were also provided by Arcturus Networks. Because of the resource limitations of the Coldfire uC5282 module, efforts were made to optimize the uClinux kernel in order to get better performance.

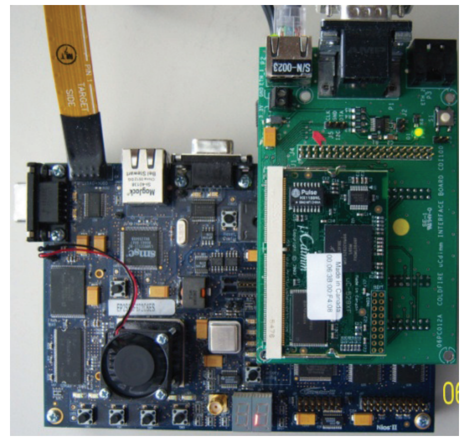

Figure 1: The hardware platform.

The most recent EPICS base 3.14.11 was used for the test. A few new EPICS base configuration files were created for the Coldfire uC5282 module on the uClinux platform.

MEASUREMENT SOFTWARE

The software from [2] is generic EPICS IOCcore performance measurement software for target OSs such as vxWorks, Linux, and RTEMS. Figure 2 shows the software structure.

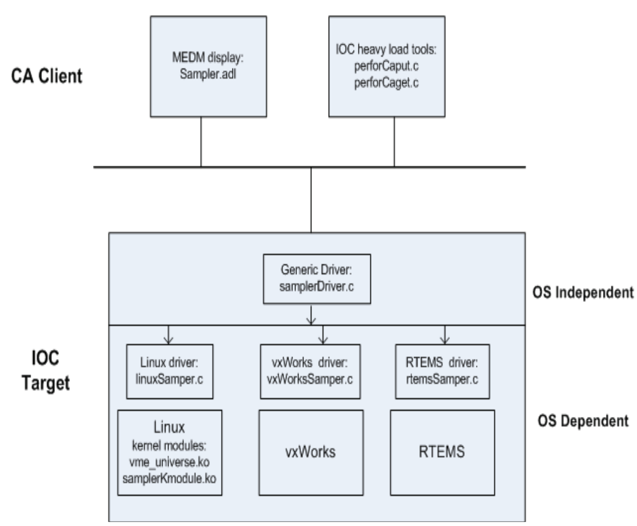

Figure 2: The measurement software structure.

* Work supported by U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.
# xusf@aps.anl.gov

HIGH LEVEL MATLAB APPLICATION PROGRAMS FOR SPEAR3§

J. Corbett *et al*, SLAC, Stanford, CA 94309

Abstract

The SPEAR3 control system nominally operates with the EPICS toolbox on top of VMS hardware. The simultaneous use of Matlab Middlelayer (MML) and Accelerator Toolbox (AT) allow for parallel, high-level machine control and accelerator physics applications that communicate with the control system via EPICS Channel Access (LabCA). While the majority of the MML and AT software is machine independent, site-specific high-level applications are also required to control the accelerator. This paper describes several such high-level application programs that have been developed for control and diagnostics at SPEAR3. Examples include a time-dependent waveform display gui, beam steering applications, transport line optics correction, SR beam diagnostics and add-ons to the main MML routines.

INTRODUCTION

The SPEAR3 light source came as the result of a Basic Energy Sciences committee recommendation following a review of U.S. synchrotron radiation facilities in 1997 [1]. Before formal DOE/NIH funding arrived in 1999, preliminary lattice design and system engineering specifications were developed on project seed money. During this time, it became clear that the historical, yet dated, SPEAR control system would need to be largely replaced [2], in particular the high-level application programs. The new system would utilize EPICS operating on a VMS platform which opened up the possibility for Channel Access communication with external programs. In order to consider options for modern application development platforms, a satellite meeting was arranged at the 1998 International Computational Accelerator Physics Conference in Monterey, CA. Presentations included options for SDDS, TCL/TK and X-Windows software.

At the time of the Monterey conference, Matlab was already in use at SSRL for data processing and off-line accelerator physics calculations. Matlab had also been used extensively at the SLC for data acquisition, data reduction and to some degree machine control. At the ALS in Berkeley, Matlab was in use for command-line driven machine control and data processing [3], and had the interesting feature that the top-level language closely mimicked accelerator simulation programs such as TRACY [4]. At the same time the first versions of the Matlab Accelerator Toolbox [5] utilizing TRACY transport physics were available for simulation studies at SSRL.

During the Monterey meeting, a proponent of IDL made an interesting observation – since recent versions of Matlab contained graphical interface commands why not use it to develop high-level application programs [6]? With Channel Access connectivity embedded in Matlab (LabCA) [7], a complete solution was available with control system communication, gui capability, user-friendly data reduction software and accelerator simulation tools that could be integrated into a single, all-in-one software package. The gavel fell and a new project was born – high level application programs at SPEAR3 would be developed and written in Matlab†.

In a stroke of luck, the main author of Matlab Middle Layer (MML) [8] was finishing work on an SBIR grant at SLAC and was available to consult with SSRL on application development for SPEAR3. The first project was to convert the FORTRAN version of the Linear-Optics-Closed-Orbit (LOCO) program to Matlab [9]. It was then recognized that SPEAR3 needed a 'middle layer' to provide easy connectivity between the accelerator physicist and storage ring. By introducing Matlab code utilizing accelerator modeling syntax developed at the ALS, a straight-forward database-drive system was devised for simulation and control.

As more of the ALS software was integrated into the system, the functionality of higher-level programs such as, orbit, tune, dispersion and chromaticity measurement expanded. In order to retain the ability to pass the new software back to the ALS, programs were written in a 'machine independent' format driven by simple MML initialization files to associate accelerator elements and their indices with girder locations, database channel names, hardware limits, conversion factors and specific locations within the AT lattice file.

First tests of machine independence were made in trials at the Canadian Light Source and then again at the ALS. Interestingly, machine-independence also created a structural rigor within the software that ultimately simplified high-level program development and streamlined switching between on-line and simulation control modes. Hardware-to-physics conversion factors also enabled the user to 'switch' between hardware (e.g. amps) and physics (e.g. m^{-2}) units with a single command. Similarly, the AT lattice pointers automate switching between on-line and simulation modes with a single command. File directory specifications were then incorporated to automate data file look-up and data storage needs for machine control and simulation.

In the sections to follow we describe high-level application program developments at SPEAR3 in the areas of waveform variable display, main ring and transport line machine tuning and optical diagnostics.

†the philosophy was, and still is, 'anything that can be written in EPICS will be written in EPICS'.

§Work supported by US DOE Contract DE-AC03-76SF00515 and Office of Basic Energy Sciences, Division of Chemical Sciences.

A NOVEL APPROACH FOR BEAM COMMISSIONING SOFTWARE USING SERVICE ORIENTED ARCHITECTURE*

G. Shen, BNL, Upton, NY 11973, U.S.A.
P. Chu, J. Wu, SLAC, Menlo Park, CA 94025, U.S.A.

Abstract

A novel software framework is under development, which is for accelerator beam commissioning and operation. It adopts a client/server based architecture to replace the more traditional monolithic high level application approach. A minimum set of commissioning and operational services has been defined such as simulation server service, directory service, magnet service, and bpm service, etc. Most of them have been prototyped. Services can use EPICS pvData as its data container and pvAccess as communication protocol. This paper describes conceptual design and latest progress for some services.

INTRODUCTION

Traditionally, an accelerator application needs to deal with many functions such as connection to various signals, data from physics modelling, data plotting, complicated program flow and error handling. If all such computation is built in a single standalone program, the complexity level of the program may result poor performance, unreliability and code maintenance difficulty. Also, if any application needs a new feature which is not provided by an easy interface, it is hard to implement the feature without major restructure of the existing program.

On the other hand, if heavy computation functions can be distributed as running modules residing on various servers and serving up data via proper service protocol, the Graphical User Interface (GUI) application itself can be a simple thin client receiving the data from the servers. This service oriented architecture (SOA) approach can in general improve both performance and reliability of applications.

In this paper, some preliminary result for simulation or model service, Linac energy management (LEM) service and possible communication protocols such as EPICS pvAccess are reported. Work plan for the SOA is also described.

SERVICE ORIENTED ARCHITECTURE

One can identify some essential services for accelerator operation by surveying the functionalities of existing applications. The granularity of services depends on functionality shared by clients, performance, robustness coding complexity, and maintenance. On one hand, too narrow of a service means many more services in total and could cause maintenance trouble. On the other hand, a single service providing too many functions could reduce its performance and reliability. Figure 1 shows a typical top level SOA diagram with a few services.

Furthermore, services can be distributed to multiple servers with virtual machines technology. A distributed system can avoid one service bringing down others. One can also add a redundant server for any critical services.

Advantages for SOA approach are described in detail below.

Easy Application Development

Coding an application with many functions can be tedious. On the other hand, some functions can be shared by several applications. A well-designed SOA approach can greatly reduce the burden on end developers. Applications can then become “thin” clients without much inline computation. Only simple “get/set” data communication with the service providers will be needed. Coding up a complicated application such as controlling an experiment will require much less time and effort. Yet, all the high quality of supporting functionality is fulfilled because the complication is maintained on the server side. This means that even a program written in scripting language such as Matlab script can still have the same high quality of error handling and message logging without additional coding efforts.

Data Control

Because the services are centralized control, i.e. typically only one particular service instance running at a time. This approach can avoid conflict among multiple clients accessing the same device; for instance, feedback and Linac Energy Management (LEM) program might change the same corrector at the same time but magnet server can shedule the two requests properly.

Better Application Memory Management

For individual applications, SOA can avoid large memory and CPU consumption due to heavy computation and data process. Therefore, it can also reduce the chance of client application program crashing.

Service Swappable

It is not necessary to replace all traditional functions with services overnight. One can implement a service at a time. If an old service is replaced by a new one, the application programming interface (API) should remain the same so the client application can pick up the service seeminglessly. This also means the SOA work is highly scalable depending on the available resources. Furthermore, a new service should go through rigorus test before any client application in production can actually use it.

*Work supported under auspices of the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 with Brookhaven Science Associates, LLC, and in part by the DOE Contract DE-AC02-76SF00515

PC –BASED TECHNOLOGIES FOR DIAGNOSTICS, MEASUREMENT AND CONTROL

J. T. Truchard, K. Schultz, National Instruments, Austin

Abstract

Over the years, the use of COTS technology has enabled scientists and researchers to focus on their experiments rather than the instrumentation. COTS systems that leverage the rapid advancement in the computer industry for measurement and control needs provides high performance and flexibility while keeping the costs low. This presentation will focus on the various models of computation that allows for rapid application development, multicore and FPGA technologies for instrumentation and interoperability between software and hardware platforms. Adoption of COTS technology at different labs such as CERN, ESO and Los Alamos National Labs will also be covered.

PRESENTATION ONLY

MATLAB WORKSHOP REPORT

W. J. Corbett, SLAC, Menlo Park, California

Abstract

This is the MatLab Workshop report as Chaired by Jeff Corbett

PRESENTATION ONLY

PROGRESS STATUS FOR THE PETRA3 EMBL BEAMLINES

U. R. Ristau, M. DiCastro, EMBL, Hamburg;
P. Duval, DESY, Hamburg;
S. Fiedler, A. Pazos, EMBL, Hamburg

Abstract

The EMBL-Hamburg currently commissions three new Beamlines at the DESY Synchrotron Petra3. A small angle solution scattering (SAXS) Beamline and two X-ray crystallography (PX) Beamlines. Beamline Control system is TINE. New TINE features for Beamline control have been developed in close collaboration with DESY. The standard TINE tools like the Alarm Handler the Archive Reader and CDI [Common Device Interface] as well as the high level TINE Motor Server are integrated in the EMBL Beamline control architecture. The Beamline control software consists of a layer concept which will be presented. The control electronics are Programmable Logic Controllers (PLCs) which run the realtime EtherCAT fieldbus protocol. The PLCs perform analog and digital input/output, motor control, signal synchronization and local feedback between devices. The concept will be presented based on the first experience during Beamline commissioning.

PRESENTATION ONLY

synApps: EPICS APPLICATION SOFTWARE FOR SYNCHROTRON BEAMLINES AND LABORATORIES*

T. M. Mooney#, ANL, Argonne, IL 60439, U.S.A.

Abstract

synApps[1] is a collection of EPICS [2] application software originally intended to support the needs of scientists performing experiments at synchrotron-radiation beamlines. The collection contains general-purpose software that extends or exploits capabilities of EPICS base, and a large amount of instrument-specific software that uses EPICS to control and provide a user interface for off-the-shelf electronics.

This paper will provide an overview of synApps, describe how the software is deployed at the Advanced Photon Source, and highlight recent additions.

OVERVIEW

synApps is a collection of EPICS modules that supplement the record types, device support, and other software infrastructure included in EPICS Base. Because it was written to support scientists conducting a wide variety of experiments, most of the software in synApps is general in purpose, and was engineered to serve many needs at once, by abstracting from specific sets of requirements general solutions for classes of problems.

But this focus on general solutions does not distinguish synApps from other EPICS-application software. Most EPICS software is general purpose, in part because EPICS is a collaborative effort. synApps differs from mainstream EPICS-application software in three ways: it contains a small amount of synchrotron-specific software, it provides infrastructure to support run-time programming, and it provides infrastructure to support data acquisition.

synApps consists of the following modules, grouped according to the kinds of applications they support.

General-Purpose Modules

- **autosave** – Saves the values of EPICS process variables, and restores them after a reboot.
- **busy** – Extends EPICS' execution tracing to include client software.
- **calc** – Provides variations of the EPICS *calcout* record for systems of expressions (*transform* record), string expressions (*sCalcout* record), and arrays (*aCalcout* record).
- **sscan** – Supports *scans* (systematically set conditions; acquire and store data).
- **std** – Supports scalers, sequences of operations, and PID loops.

Hardware Specific Modules

- **areaDetector** – Supports multidimensional detectors.
- **camac** – Supports CAMAC hardware.
- **dac128V** – Supports an IndustryPack digital-to-analog converter.
- **delayGen** – Supports delay generators.
- **dxp** – Supports DXP digital-signal processing spectroscopy systems.
- **ebrick** – Supports the EPICS Brick, a PC104-based computer running Linux, as an EPICS *IOC* (Input/Output Controller).
- **ip** – Supports various message-based (e.g., serial, GPIB) devices.
- **ip330** – Supports an IndustryPack analog-to-digital converter.
- **ipUnidig** – Supports an IndustryPack digital I/O module.
- **love** – Supports Love controllers.
- **mca** – Supports multichannel analyzers and multichannel scalers.
- **modbus** – Supports Modbus devices.
- **motor** – Supports stepper and servo motors.
- **quadEM** – Supports a four-channel electrometer.
- **softGlue** – Provides user-programmed digital logic and I/O.
- **vac** – Supports vacuum-related devices.
- **vme** – Supports VME hardware.

Synchrotron-Radiation Specific Modules

- **optics** – Supports X-ray monochromators, slits, optical tables, and other synchrotron-radiation equipment.

Other Software in synApps

- **xxx** – Provides a template for an EPICS IOC directory using synApps.
- **utils** – Provides miscellaneous software related to synApps, including support for migrating from one version of synApps to another, support for a data-file format used by synApps scan software, and support for rapid EPICS-database programming.

Software Distributed with synApps

synApps makes use of the following EPICS modules that are not part of synApps, but are distributed with it: **allenBradley**, **asyn**, **ipac, seq**, **stream**, and **vxStats**.

*The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

#mooney@aps.anl.gov

USING EZCAIDL TO CONNECT TO EPICS CHANNEL ACCESS FROM SHADOWVUI FOR DYNAMIC X-RAY TRACING*

Alan Duffy#, Canadian Light Source Inc., Saskatoon, Saskatchewan, Canada

Abstract

Using the ezcaIDL library, for IDL [1], to provide an interface to EPICS [2] Channel Access through the EZCA [3] library, a simple XOP [4] extension was written that initializes ezcaIDL and thus allows access to a set of simplified IDL interface commands to connect to Channel Access from within XOP and hence from SHADOWVUI (an XOP extension) [5]. The XOP widget-based driver program is a commonly used front-end interface for computer codes of interest to the synchrotron radiation community. It models x-ray sources and characterizes optics. Extensions, such as SHADOWVUI, are optionally loaded to easily expand its functionality. SHADOWVUI is a complete Visual User Interface for SHADOW [6], which is an essential tool for x-ray optics calculations and ray-tracing. SHADOWVUI is an interactive tool for designing an optical system and visualizing results as graphs and histograms. The working scheme is to define the source and the optical elements by entering their parameters. The author has taken the usual SHADOWVUI simulation of an x-ray system a step further by using ezcaIDL to interface with the EPICS control system to access the positions of optical components in real life and then run a corresponding simulation based upon these.

INTRODUCTION

In order to predict the performance of an optical system in general and in particular a synchrotron radiation beamline, ray tracing methods are used. An essential tool for x-ray optics calculations is the ray-tracing program SHADOW, developed at Nanotech Wisconsin (University of Wisconsin), and has been used in the synchrotron community during the last 20 years. A complete Visual User Interface for SHADOW aptly named SHADOWVUI may be used as a higher level interface with graphics and menus to prepare the SHADOW inputs. It is available as an extension to another commonly used software package called XOP, a commonly used front-end interface for computer codes that model x-ray sources and optics. Essentially, the SHADOW inputs define the optical system as a collection of optical elements (mirrors, slits, screens, etc.) placed in sequential order. SHADOW generates and traces a beam from the source (e.g. bending magnet, wiggler, or indulator) sequentially through the system. The important point is that the SHADOW inputs define the optical system which usually serves to model a real synchrotron beamline. However, the parameters are static and do not change until the user enters new ones.

CONCEPT

The concept of running a dynamic x-ray tracing simulation of a beamline is straightforward (take the live positions and put them in the simulation engine), but requires some preliminary work creating the model in SHADOWVUI and determining the corresponding inputs to use from the actual beamline. This involves defining the source by supplying its parameters (e.g. energy, etc.) and defining the various optical elements with their parameters (e.g. mirror types, source plane distances, image plane distances, etc.), and how they relate to beamline parameters. The ezcaIDL library provides the tool necessary to read the beamline parameters that are maintained by the EPICS control system. The only catch is that one must define how the variables in the model are related to the parameters of the beamline. The newly developed XOP extension is used in conjunction with SHADOWVUI and requires as input a user created IDL structure defining the relationship between beamline parameters (*i.e.* process variables) and SHADOWVUI variables to make connections between the live position of the beamline optics and the variables in the simulation model.

Positioning Optical Elements

The position of each optical element in SHADOW is defined relative to the previous element (or source), not the laboratory reference frame. The user inputs the incidence and reflection angles of the central ray at each optical element as well as source and image distances to define the system. In an aligned system the central ray coincides with the optical axis, however the user has complete freedom of specifying incidence angles that are zero, positive, or larger than 90 degrees, as long as the user understands how to interpret the results. It is also not necessary for the image and source distances to correlate to the location of an actual image or object in the optical sense either. The sum of the image distance (from the previous element) and source distance simply defines the separation between optical elements in the SHADOW model. In fact, it is advantageous to think of these distances not as defining the optical element positions *per se*, but as defining the origins of their coordinate systems. Then use the mirror movement option available to place the optical components in their proper locations. Using this option to place an optical element prevents unnecessarily moving subsequent components with their positions defined relative to previous components and avoids having to recalculate distances and angles.

*Work supported by CLS
#alan.duffy@lightsource.ca

A SIMPLE DAQ SYSTEM BASED ON LABVIEW, PHP AND MYSQL

M. Tanigaki*, K. Takamiya, R. Okumura,
Research Reactor Institute, Kyoto University, Kumatori, Osaka 590-0494 Japan

Abstract

A tiny and simple DAQ system has been designed and developed for the application to the control system in our institute. This DAQ system is based on on LabView, MySQL and apache, and shows good compatibility with LabVIEW-based system like the control system for the FFAG complex in our institute. The current status for the development, as well as the recent accelerator-related status in our institute, will be introduced.

INTRODUCTION

An FFAG accelerator complex[1, 2, 3] has been developed as a proton driver for the feasibility study on ADS performed in the research reactor institute, Kyoto University. The control system for this FFAG accelerator complex has some requirements on the flexibility, simplicity and reliability. The control system is required to have a sufficient flexibility towards major and minor modifications in the design and equipments of accelerator complex during the construction, and to achieve a certain level of easiness on its use and development for the people in our institute, who are little familiar to accelerator itself. Additionally, high reliability and stability from the points of the nuclear safety and the radiation protection are required since the combined operation with a nuclear fuel assembly is planned in the feasibility study on ADS.

To meet such requirements for the present control system, we have developed a control system [4] based on LabVIEW, known as its user-friendly GUI environment, and PLC known as one of the most reliable control devices in the field of factory automation. This control system for the FFAG complex has proven itself to have sufficient performance and to satisfy the requirements on the design through the construction and operation of the FFAG accelerator complex, in its operation for years. Based on this success, this control system has been applied to other equipments and facilities. One of such typical examples is that the application to the pneumatic transportation facility in KUR[5].

On contrary to the control system itself, little efforts have been made for the data acquisition system up to now. In most of the application cases, a simple data logging feature is included in VIs by using the functions of LabVIEW such as the chart VI. As the increasing demand on the systematic management of the data for the multiple devices and on the simplified method of DAQ for the users, we have started the development of a DAQ system for our control system.

* tanigaki@rri.kyoto-u.ac.jp

In this paper, the outline and current status of our DAQ scheme are introduced.

DAQ SYSTEM WITH ODBC DRIVER

At present, the FFAG accelerator complex in our institute is under modification to the injection scheme using H^- beam. The FFAG injector will be replaced to an 11 MeV H^- proton linac by the end of the fiscal year 2010. Additionally, the inclusion of the present control system to a new control system based on EPICS, intending to the inclusion of this FFAG accelerator to a larger accelerator complex for the pulsed neutron source. Therefore, the main application of the control & DAQ system for now is the devices and instruments equipped to the 5 MW reactor, especially the pneumatic transportation facility for the neutron irradiation [5].

The outline of the pneumatic transportation apparatus and the control system is shown in Fig. 1. The control system for this pneumatic transport system is the same architecture as that for the FFAG complex [4]. The low level sequences of PLCs for controlling the pneumatic transportation system has been implemented in PLCs, and the man-machine interfaces (MMIs) are developed with LabVIEW on conventional PCs. In addition to the controlling system of the pneumatic transportation apparatus and the monitoring system, related external systems such as radiation control systems and measurement systems for experiments are integrated. This integrated system might well be able to realize secure operating and management of the pneumatic transportation apparatus.

In this pneumatic transport system, a DAQ system based on the ODBC driver, LabVIEW and MySQL is developed. So called, a “SQL Command Generator” VI is implemented into every MMI PC as a sub VI of MMI VIs. Since

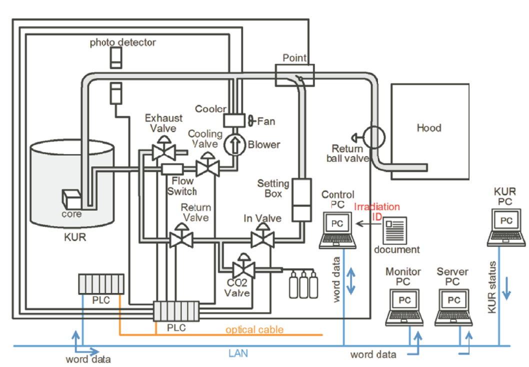

Figure 1: Outline of the pneumatic transportation apparatus and the new control system.

WEB SERVICES CYBER-SECURITY ISSUES*

D. Quock#, ANL, Argonne, IL 60439, U.S.A.

Abstract

The Web's potential for distributed programming has been proven not only in the business realm, but also in the accelerator controls domain. Web services describes clients and servers that communicate over the Internet's Hypertext Transfer Protocol (HTTP) using predefined Internet-based Application Programming Interfaces (APIs). It is the uniqueness of Web services transactions such as cloud computing, data sharing, and data archiving that give rise to the security concerns of Web services (authentication, data integrity, non-repudiation, and privacy). At Argonne National Laboratory's Advanced Photon Source, Simple Object Access Protocol (SOAP)-based Web services were implemented into the Integrated Relational Model of Installed Systems (IRMIS) as the application interface to Oracle's Content Server document management software. This report reviews the basics of Web services, cyber-security issues that are inherent for Web services, current Web services security implementation practices, and future directions of Web service security development efforts where the overriding goal of Web services security is to focus on managing risk and protecting data.

BASICS OF WEB SERVICES

In simplest terms, Web services are distributed Internet applications that have standard-based interfaces. Web services are typically thought of as being divided into two main technologies:

1. Big Web Services: This technology uses Extensible Markup Language (XML) messages that follow the Simple Object Access Protocol (SOAP) standard.

2. RESTful Web Services: The representational state transfer (REST) software architecture uses PUT, GET, DELETE and POST HTTP methods to integrate Web browsers with underlying client/server software applications.

Service-Oriented Architecture (SOA) is model-based software that is typically constructed from loosely coupled Web services. SOA can be broken down into the three layers: business workflow, Web services, and communication [1]. Table 1 demonstrates that SOA adds three layers on top of the standard client-server architecture and shows the associated Web services standards that are used at each layer.

*Work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.

#quock@aps.anl.gov

Table 1: SOA Architectural Layers

Architectural Layer	Web Services Standards
Business Workflow	BPEL WSCI
Web Services	WSDL UDDI
Communications SOAP	 XML
Client-Server Transports HTTP SSL TCP/IP	

Web Services Standards

At the highest level of SOA, Business Process Execution Language (BPEL) is used to describe and execute the business processes. An alternative to BPEL is the World Wide Web Consortium (W3C)'s standard Web Service Choreography Interface (WSCI). These two standards are currently diverging as industry is divided in its support of either business workflow standard. The role that BPEL (or WSCI) plays in SOA is orchestrating the overall business workflow by providing mapping between the services and business processes through documents.

At the next level of SOA, the standard Web Service Description Language (WSDL) provides static interface definitions for the software components that are accessible to clients. The Universal Description, Discovery and Integration (UDDI) is a specification for repositories where organizations can publish services that they provide and describe the interfaces to their services via WSDLs.

At the communications layer of SOA, messages are transmitted through SOAP, which is an envelope containing a header and body. The services that are communicating with each other can be identified through their unique name contained in SOAP messages.

ADVANCED PHOTON SOURCE WEB SERVICES

The benefits of SOA to organizations is the flexibility of implementing business processes on top of Web services and the ability to compose and re-compose systems frequently. SOA provides a peer-to-peer style of architecture with a general statelessness of services. One example of how Web services technology was implemented at Argonne National Laboratory's Advanced Photon Source is in the interaction between the in-house built IRMIS accelerator controls relational database

REMOTE ACCESS TO THE VESPERS BEAMLINE USING SCIENCE STUDIO*

D. Maxwell#, D. Liu, E. Matias, D. Medrano, CLS, Saskatoon, Canada
M. Bauer, M. Fuller, S. McIntryre, J. Qin, UWO, London, Canada

Abstract

Science Studio is a web portal, and framework, that provides scientists with a platform to collaborate in distributed teams on research projects, and to remotely access the resources of research facilities located across Canada. The primary application for Science Studio is to provide scientists with remote access to the VESPERS beamline at the Canadian Light Source synchrotron in Saskatoon Saskatchewan, and to readily process data from this beamline at the SHARCNET high performance computing facility in London Ontario. The VESPERS beamline is a complex instrument that is composed of many devices, such as valves, motors and detectors, which are all controlled through the low-level EPICS control system. Science Studio implements a simple, intuitive and functional web-based interface to the beamline for device control and data acquisition. The Science Studio experiment management system allows the acquired data to be easily organized and shared with the research team. This paper will provide an overview of the design, implementation and capabilities of the Science Studio system, with a focus on remote control of the VESPERS beamline.

SCIENCE STUDIO OVERVIEW

The Science Studio web portal is mostly implemented in Java, and uses server-side web technology common to enterprise applications such as Java Servlets, Java Messaging Service (JMS), Java Database Connectivity (JDBC) and Java Server Pages (JSPs). In addition, many high quality open-source frameworks and libraries have been leveraged to build a highly functional web portal. The Spring [1] framework is used extensively throughout to build very robust and highly configurable servlets using the Model-View-Controller (MVC) architectural pattern. The iBATIS [2] Object-Relational Mapper (ORM) library is used to easily persist objects to a MySQL [3] relational database. The XStream [4] library provides fast object marshalling capabilities in both XML and JSON formats. Security functionality is provided by the JSecurity [5] framework using some custom extensions. Other Java libraries and tools include Apache Log4J [6], Apache Commons [7], Apache Tomcat [8], Apache ActiveMQ [9] and Jetty [10].

Data Model

Science Studio defines and implements a data model to capture the metadata associated with scientific research. Figure 1 is a data object relation diagram for this data model. The objects belonging to the experiment model have been indicated. A primary objective of Science Studio is to allow scientists, and other people, to collaborate; therefore an important part this data model is the *person* object. A *person* represents a user of the system and contains information such as their name, affiliation, email address and mailing address.

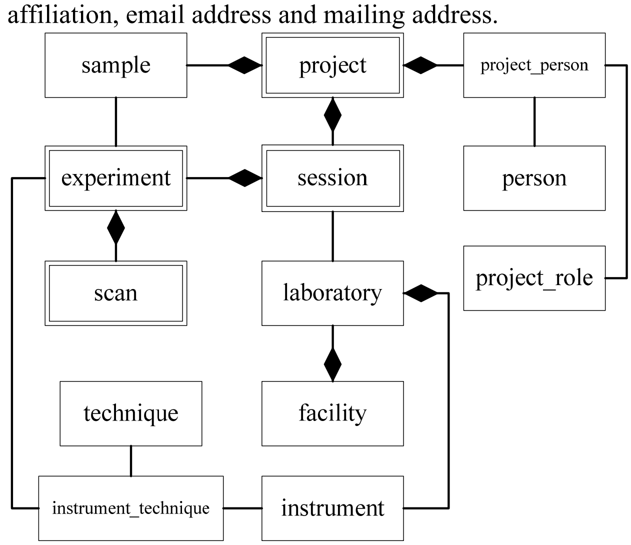

Figure 1: Data object relation diagram for the Science Studio data model, with the experiment model indicated.

Research projects are the foundation of experiment management in Science Studio. For that reason, the *project* object is the top-level organizational element for the hierarchical experiment model. A *project* is composed of *person*, *sample* and *session* objects. The collection of *persons* represents the people collaborating on a project, or simply a project team. A *sample* represents the physical specimen that is the subject of investigation for a project team. A *session* is composed of *experiment* objects and represents the reservation or allocation of resources to the project team for a specified time period. An *experiment* is composed of *scan* objects and references a *sample*, *instrument* and *technique* object. A *laboratory* is composed of *instruments* that are associated by location or function. An *instrument* references *technique* objects and represents a device or resource used to conduct an experiment. A *technique* represents the method or process used by an instrument to produce data. A *scan* represents the actual experimental data produced by an instrument, and contains information about its storage location and file format.

Standard Data Format

The manipulation of experimental data is a requirement for most scientific applications. Science Studio specifies a standard format for experimental data files to facilitate

*This work was funded by CANAIRE under R&D project NEP-01; and performed at the Canadian Light Source which is supported by NSERC, NRC, CIHR and the University of Saskatchewan.
#dylan.maxwell@lightsource.ca

RESEARCH METADATA MANAGEMENT AT THE AUSTRALIAN SYNCHROTRON

Richard. Farnsworth, Alistair Grant, Andrew Rhyder, Australian Synchrotron, Melbourne Australia
Nick Hauser, Bragg Institute ANSTO, Sydney Australia.

Abstract

This paper details the approach the Australian Synchrotron [1] is using, in collaboration with the Australian Neutron Source, run by the Bragg institute, part of ANSTO [2] (Australian Nuclear Science Technology Organisation) called OPAL (Open Pool Australian Light-water Reactor) for some of the data and metadata management issues. It explores the data and user policies, describes the quantity and quality of data and demonstrates the way forward based on both existing and future directions in e-research, network communications, user proposal and material databases, portal technologies and integration techniques. The role of standards for access and metadata creation is also explored. This work is funded by an educational infrastructure grant administered by Australian National Data Services.

DATA POLICY

In order to progress with publicly funded research facilities data and metadata publishing the data policies must be clear. The answer to the questions who owns the data, when can you make it public, what can you do with it should be clear. At the Australian Synchrotron twenty-four months is allowed to the principle investigator to publish publicly. The period is thirty-six for Bragg institute instruments (ANSTO, OPAL). Either facility may choose to process the raw data in order to make is accessible or publishable. There is a growing trend worldwide towards open technical data. There is also a growing trend towards publishing not only a scientific paper, but also the raw data that was used to produce it.. As both the Australian Synchrotron and the Bragg institute are publicly funded, technical data created at either location should be at some point made available to the public. Currently there are some Australian mechanisms for achieving this. One is called the ANDS portal. [3] ANDS stands for the Australian National Data Service.

DATA QUANTITY

The Australian Synchrotron operates nine beamlines producing around two to three Terabytes of experimental data per day across a wide variety of disciplines from protein crystallography, medical, through to the conservation and restoration of cultural objects and works of art. In 2009 over five hundred groups conducted research at the Australian Synchrotron. More are expected this year and the next. The Australian Synchrotron expects to be producing at least eight terabytes per day when the next round of ten beamlines are installed in the coming two to five years. Even if the Australian Synchrotron just keep operating the existing beamlines, there will be a significant increase in data collection because of both the continual improvement in detectors and the overall efficiency or "duty cycle" of the beamlines. The objective of this project to make that data available publicly. The data will be stored in the curated archives immediately; however authorisation for access will be allowed or otherwise depending on when the data can be made available. Much smaller volumes of data are created at the Bragg institute.

This project is seeking to provide services so that researchers and institutions can manage their data. To give them the power of something like "Google" over their data – that is the ability to search, catalogue and access. This promotes the use and re-use of data and so adds to the efficiency of the data generating ability of each facility.

MECAT

The chosen a name for this project is "MeCAT" [4], as a nod to a similar project/product called ICAT. It was a requirement to name the project, rather than the technology being used. The project is to enhance the technology to enable those things aforementioned efficiencies.

COLLABORATION

It is worthwhile noting the collaboration details. The two facilities have decided that if they collaborate and pooled resources between the two similar facilities in Australia, we could effectively get twice the efficiency of the software development dollar in terms of software resources.

It also a major step towards the creating an Australian culture of same software in similar institutions. This leads to the same experiences for researchers. This is becoming increasingly strategically important to both facilities. It would be ideal if every institute used exactly the same software everywhere such that experimenters trained in the use of software one area or instrument could use the same skills in another. This is probably never going to be completely possible, but this project assists by moving toward that philosophical direction. It also helps with the data management, because the software automatically moves the data for those researchers that come from known institutions to their home institutions.

OBJECTIVE OF PROJECT

The objective of this project is to provide services to researchers to manage their experimental data and to provide data search and access to the broader research

DIAMOND'S TRANSITION FROM VME TO FIELDBUS BASED DISTRIBUTED CONTROL

I.J.Gillingham, S.C.Lay, R.Mercado, P.Hamadyk, M.R.Pearson, T.M.Cobb, M.T.Heron, N.P.Rees
Diamond Light Source, Oxfordshire, UK

Abstract

The interface layers of Diamond's accelerator and photon beamline control systems have predominantly been implemented as VME-based systems. Forthcoming control systems, for new photon beamlines, have requirements necessitating a divergence from Diamond's adopted design patterns, including a reduction in available rack space, and we also need to consider the management of hardware obsolescence. To address these issues, a new standard based on PCs and Ethernet field buses to the instrumentation has been defined. This paper will present the new design, how the design transition is being effected and the key benefits to Diamond.

INTRODUCTION

Diamond Light Source is a 3 GeV third-generation light source with a 561 m storage ring (SR), a full-energy booster (BR) and a 100 MeV pre-injector Linac[1]. The photon output is optimised for high brightness from undulators and high flux from multi-pole wigglers. The current operational state includes 19 photon beamlines, with a further three beamlines in an advanced stages of design and construction. A further phase of photon beamlines is now proposed, and subject to funding, detailed design and construction of these 10 beamlines will commence from 2011.

In planning for the next phase of photon beamlines, it was timely to consider the control system architecture applied to future beamlines, associated front ends and experimental stations.

EXISTING CONTROL SYSTEM ARCHITECTURE

Accelerator and beamline control systems use a consistent approach to interface to the hardware, with most equipment interfaced through embedded VME systems. To support the interface requirements of the equipment, a range of I/O modules based on Industrial Pack (IP) modules (ADC, DAC, Serial, DIO) and VME modules (IP carrier, motion, scalar and timing) is used. The field signals are interfaced via either transition modules or front-panel connections. A VME microprocessor (MVME5500) runs VxWorks and EPICS to serve up the control information to client applications. There are in excess of 250 VME-based systems running as part of Diamond's control system[2]. In addition, the electron BPMs run EPICS IOCs directly on the Libera beam processing hardware, and soft IOCs running under Linux on PC hardware concentrate and process data or interface to network attached devices over manufacturer-specific protocols. One anomaly to this approach has been video cameras which have been interfaced to the VME IOC using Firewire and a PMC Firewire adapter located on the VME processor board.

REASON FOR CHANGE

The existing control system architecture has served well for the existing accelerators and beamlines; however it was defined nearly ten years ago, so in the context of the next phase of beamlines the opportunity to reconsider the standards is being taken. In doing so, it is clear that not all the hardware capability of VME is required for beamline control; neither is the use of a hard real-time operating system such as VxWorks. It is also apparent that most I/O functionality required for control of beamline equipment can now be realised through Ethernet-attached I/O. There is also now good infrastructure for developing and managing Linux based EPICS IOCs on a PC architecture.

OUTLINE REQUIREMENTS FOR PHOTON BEAMLINES

In considering the requirements for photon beamline control the following technical systems are identified:

- Motion control
- Vacuum instrumentation and other serial devices
- Video cameras
- Analogue and digital signals
- Programmable logic controllers
- Timing signals

The interface from the IOC to the equipment should make use of the installed network cabling, thereby reducing I/O-specific cabling and giving flexibility in reconfiguration and addition of equipment without the need to pull new cables.

There should be greater partitioning of the IOC functionality by technical area, e.g. motion, camera and vacuum, by running a greater number of EPICS IOC instances, either as separate processes on one Linux system or as single processes, each on a virtualised Linux system. This would minimise the disturbance to beamline operation when making changes that necessitate restarting an IOC.

The I/O associated with the control system should be located close to the equipment being interfaced; i.e. for signals located in experimental and optics hutches, the I/O modules should be co-located in these areas. However, this is constrained by the possibility of radiation-induced damage to I/O in the optics hutches of high energy

A DISCRETE HYSTERESIS MODEL FOR PIEZOELECTRIC ACTUATOR AND ITS PARAMETER IDENTIFICATION

Y. Cao and X. B. Chen#
Department of Mechanical Engineering
University of Saskatchewan, Canada

Abstract

Hysteresis is an important nonlinear effect exhibited by piezoelectric actuators (PEA) and its modelling has been drawing considerable attention. This paper presents the development of a novel discrete model based on the concept of auto-regressive moving average (ARMA) for the piezoelectric-actuator hysteresis, and its parameter identification method as well. Experiments were carried out to verify the effectiveness of the developed model. The result obtained shows that the developed model can well represent the hysteresis of the PEA.

INTRODUCTION

Piezoelectric actuators (PEA) have been widely used in nanopositioning applications, such as AFM , STM , DVD disc reading and writing [1], diamond lathe machine [2], lithography, X-ray imaging [3]. However, the performance of a PEA can be significantly degraded by its hysteresis. Hysteresis is a memory effect of piezoelectric actuators and, as a result, the hysteresis exhibited at an given time instant depends on not only the input at the present time but also the operational history of the system considered. In order to develop control schemes on PEA, modelling of PEA has been drawing considerable attention and several models have been resulted to describe the hysteresis effect, such as Preisach model [4], the ferromagnetic material model [5] and the nonlinear auto-regressive moving average model with exogenous input (NARMAX model) [6]. However, most of the models developed in literatures are continuous and the model-based controller design is proceeded in the continuous time domain. With the advance of computer technology nowadays, controllers are mostly implemented digitally. Note that not all the continuous controllers can work on the sampled digital system as desired since the discrete sampling can sometimes make the continuous system unstable. Therefore, it is advantage to develop a discrete hysteresis model of PEA for its digital controller design. Unfortunately, little work about the discrete hysteresis model or the digital controller design for PEA has been found yet. In this paper, the ferromagnetic material hysteresis model is discreted and, by combining it with the concept of auto-regressive moving average (ARMA), a novel model is developed to represent the hysteresis of PEA. Specifically, the next section of this paper is the introduction to the discrete ARMA-based hysteresis model, which is followed by the experimental identification and verification results by using the discrete ARMA-based hysteresis model as compared to the general discrete form of hysteresis model [7]. The last section gives the conclusions of the paper and future work.

xbc719@mail.usask.ca

DISCRETE ARMA-BASED HYSTERESIS MODEL

The ferromagnetic material hysteresis model introduced by Adriaens and Koning [5] is illustrated in the following:

$$\dot{y} = \alpha |\dot{x}| [f(x) - y] + \dot{x} g(x) \tag{1}$$

where x is the input of the hysteresis and y is the output, $f(x)$ and $g(x)$ are functions of x with which you can “shape” the hysteresis loop. It has been experimentally verified that this differential equation is also suitable for describing electric hysteresis such as PEA. In theory, PEA shows the length saturation. In practise, however, the displacement of the PEA stays far away from saturation. Therefore, chose $f(x) = ax/\alpha$ and $g(x) = b$ as the shape function, Equation (1) can be rewritten as:

$$\dot{y} = |\dot{x}|(ax + cy) + b\dot{x} \tag{2}$$

where $c = -1/\alpha$. [7] applied the difference equation to discrete Equation (1) as follows:

$$\begin{aligned} y(k+1) - y(k) = &|x(k+1) - x(k)|[ax(k) + cy(k)] \\ &+ b[x(k+1) - x(k)] \end{aligned} \tag{3}$$

This paper discrete Equation (1) by integral.

Discrete form of the ferromagnetic material hysteresis model

When the input signal is monotonically increasing, $\dot{x} > 0$, take integral on both side of equation (2) in one sampling interval, one can derive:

$$\int_{kT}^{(k+1)T} \dot{y} dt = a \int_{kT}^{(k+1)T} \dot{x} x dt + c \int_{kT}^{(k+1)T} \dot{x} y dt + b \int_{kT}^{(k+1)T} \dot{x} dt \tag{4}$$

where T is the sampling interval.

Equation (4) leads to:

$$\begin{aligned} y(k+1) - y(k) = &\frac{1}{2} a [x^2(k+1) - x^2(k)] + c \int_{x(k)}^{x(k+1)} y dx \\ &+ b[x(k+1) - x(k)] \end{aligned} \tag{5}$$

which is the discrete form of the first order hysteresis differential equation (2).

Using trapezoid equation to estimate the integral term, Equation (5) yields:

$$y(k+1) = a \frac{\alpha(k+1)}{2 - c\beta(k+1)} + b \frac{2\beta(k+1)}{2 - c\beta(k+1)}$$

AUTOMATION OF THE MACROMOLECULAR CRYSTALLOGRAPHY BEAMLINES AT THE CANADIAN LIGHT SOURCE

M.N. Fodje*, R. Berg, G. Black, P. Grochulski, K. Janzen, Canadian Light Source, 101 Perimeter Road, Saskatoon, SK, Canada S7N 0XN.

Abstract

The Canadian Macromolecular Crystallography Facility (CMCF) is a suite of two beamlines 08ID-1 and 08B1-1. Beamline 08ID-1, is an undulator beamline for studying small crystals and crystals with large unit cells, while beamline 08B1-1 is a bending-magnet beamline for high-throughput macromolecular crystallography with a high level of automation. The primary method of access to CMCF 08B1-1 will be remote, in what is commonly referred to in the field as ”Mail-in” crystallography. We are developing a software system for automating both beamlines, with modules for beamline control, experiment control, data analysis, information management, and graphical user interaction. The system is developed using the Python programming language and makes use of popular open-source frameworks such as Twisted, Django and GTK+. Once completed, the system will allow automation of the macromolecular crystallography experiment from experiment setup to data analysis, thereby increasing the efficiency of the CMCF beamlines and reducing the need for user travel to the synchrotron.

BACKGROUND

The growing impact of macromolecular structural analysis to pharmaceutical, academic and industrial research has resulted in a growing demand for access to protein crystallography beamlines. This demand is reflected not only in the number of samples available for analysis, but also in the increased number of scientists from different fields now using structural information in their research. As a result, many more users with less crystallographic training are demanding access to macromolecular crystallography (MX) beamlines at synchrotron facilities. Fortunately, the MX experiment is highly amenable to automation [1]. It is not surprising therefore that there are many on-going efforts by various synchrotron facilities to provide highly automated MX beamlines to the community of users [1-3].

The synchrotron MX experiment can be broken down into distinct steps (see Table. 1). These include sample preparation, beamline setup, sample mounting, sample alignment, sample characterisation, data acquisition and data processing. The details of each step may vary based on the specific sample being examined and type of experiment desired With the exception of the first step, which is usually carried out by experimenters at their home laboratories, the remaining steps can be automated to a very high degree. It is therefore possible in principle to build a fully automated beamline where experimenters simply prepare and send their samples to the beamline, data is automatically acquired, and experimenters are never needed on-site.

Table 1: Steps involved in an MX experiment

Step	Description
Sample Preparation	Samples are frozen in cryogen at the home laboratory and couriered to the synchrotron by experimenters
Beamline Setup	The beamline is configured and optimized
Sample Mounting	The sample is mounted on a Goniometer
Sample Alignment	The sample on the Goniometer is positioned such that the sample rotates within the X-ray beam, for data acquisition
Sample Characterisation	Initial data frames are collected and processed to obtain improved parameters to be used for data acquisition
Data Acquisition	Data frames are collected
Data processing	Data frames are integrated and reduced to reflection files for further analysis and structure determination by the users

Automation of an MX beamline requires tight integration of various hardware and software components. In addition to the beamline hardware required for delivery of a high-quality and stable beam at the sample position, robotic sample mounting devices and computer hardware for data processing are also required. The software system is a central component of every automated beamline and great care has to be taken to ensure that it is reliable and enables the acquisition of the best possible data. Here, we describe the architecture and implementation of the software for automation of the 08B1-1 and eventually the 08ID-1 beamlines at the Canadian Light Source (CLS).

SOFTWARE ARCHITECTURE

The software system being developed for automation of the CMCF Beamlines is a modular system, layered above the low-level beamline instrument control system which is based on the Experimental Physics and Industrial Control System (EPICS). The main system modules are the Experiment Management Module (EMM), the Beamline Control Module (BCM), the Data Processing Module (DPM), and the Information Management Module (LIMS) (see Fig. 1).

* Email: Michel.fodje@lightsource.ca

MECHANICAL VIBRATION MEASUREMENT SYSTEM AT THE CANADIAN LIGHT SOURCE

J.W. Li, E. Matias, Canadian Light Source, Saskatoon, SK, Canada
X.B. Chen, W.J. Zhang, University of Saskatchewan, Saskatoon, Canada.

Abstract

In recent decades, synchrotron radiation has developed into a valuable scientific tool around the world. At synchrotron radiation facilities, the mechanical vibrations in the optics hutch and experimental hutch, especially in the vertical direction, enlarges the beam size and changes intensity of the monochromatic X-ray beam. To investigate mechanical vibrations at the Canadian Light Source (CLS), a vibration measurement system was developed. This paper presents our investigations on mechanical vibrations at four beamlines and endstations at the CLS.

INTRODUCTION

At synchrotron radiation facilities, the vibration of the electron and/or photon beam, especially in the vertical direction, enlarges the size and changes its intensity. This degrades the performance of the beamline. It is reported that the amplitude of floor vibrations at the ATF2 project is approximately 50 μm, which is even larger than the vertical beam spot size expected at ATF2 [1]. In another report related to synchrotron radiation lithography, the quality of micro structures fabricated by the lithography beamline is greatly affected when the amplitude of the vibration is bigger than a quarter of the minimum feature size [2].

Many other factors that are responsible for vibrations at synchrotron radiation facilities were reported in the literature, such as traffic, human activities, strong wind and/or ocean waves, water pipes, and moving mechanical components. Thus, careful investigations of vibrations at synchrotron radiation facilities are crucial, especially if the photon beam size is within a few micrometers.

Studies of vibrations have been conducted at synchrotron radiation facilities worldwide and a brief review can be found in [3]. Although the CLS floor was carefully designed, we found that beamline developments still necessitate carrying out vibration studies. In this study, we investigated vibrations in the experimental and optics hutches at four beamlines and endstations at the CLS: CMCF 08ID-1 beamline, HXMA 06ID-1 beamline, REIXS 10ID-2 beamline, and the STXM endstation at SM 10ID-1 beamline. This work identified key vibration sources.

INSTRUMENTATIONS

The Canadian Light Source Vibration Data Acquisition system includes a Vector Signal Analyzer (VSA) (Model: Hp Agilent 89410A; Manufacturer: HP) and accelerometers (Model: 393B31; Manufacturer: PCB PIEZOTRONICS). Accelerometers produce a voltage proportional to the acceleration of their connected object. The VSA converts the output voltage of the accelerometers into a voltage power spectral density (S_v). Acceleration power spectral density (S_a, unit: $(m/s^2)^2/Hz$) is obtained from S_v by the following equation [4]:

$$S_a = \frac{S_v}{a^2} \tag{1}$$

where a is the sensitivity of the accelerometers. Displacement PSD (S_d, unit: $\mu m^2/Hz$) is calculated using S_a by the following equation [4]:

$$S_d = \frac{S_a \times 10^{12}}{(2\pi f)^4} \tag{2}$$

The RMS displacement over a given frequency band (f_1, f_2) can be calculated using the following equation [4]:

$$Z = \sqrt{\int_{f_1}^{f_2} S_d(f)df} \tag{3}$$

The sensitivity of the accelerometer a=1.02 v/(m/s^2). The frequency range of the measurement is 0.1 Hz to 300 Hz. The frequency resolution of the accelerometer is better than 0.1 Hz. In this study, we used two indexes for vibration evaluation--the displacement power spectral density (PSD) and the root mean square (RMS) displacement. The displacement PSD shows the strength of the displacement variation as a function of frequency. The RMS displacement represents the amplitude of displacement variations within a specific frequency range.

IDENTIFICATION OF VIBRATION SOURCES

The experimental set-up is discussed in [3].

Fan coil unit

The fan coil unit is hung on the ceiling in the CMCF 08ID-1 experimental hutch (SOE). Figure 1 shows that the fan coil unit induced vibrations have frequencies of 25.5 Hz (RMS displacement: 2.0×10^{-4} μm), 26.5 Hz (RMS displacement: 3.1×10^{-4} μm), and 53 Hz (RMS displacement: 4.0×10^{-5} μm) which is the harmonics of 26.5 Hz.

Detector cooling system

The equipment is used for cooling the detector of the MicroProbe endstation and it is approximately 0.5 m away from the microprobe endstation in the HXMA 06ID-1 experimental hutch. The microprobe endstation is very sensitive to vibrations since a very small beam spot

REMOTE ACCESS TO A SCANNING ELECTRON MICROSCOPE USING SCIENCE STUDIO*

D. Maxwell#, E. Matias, CLS, Saskatoon, Canada
M. Bauer, M. Fuller, S. McIntryre, T. Simpson, UWO, London, Canada

Abstract

Science Studio is a web portal, and framework, that provides scientists with a platform to collaborate in distributed teams on research projects, and to remotely access the resources of research facilities located across Canada. The Western Nanofabrication Facility is located at the University of Western Ontario and houses a variety of instruments for lithography, deposition and characterization. One of these instruments is an Oxford Instruments X-ray System fitted to a Scanning Electron Microscope. This x-ray system has been integrated into Science Studio. This allows users to remotely access the system and to upload experimental data into Science Studio. Remote control of the instrument is provided using a remote desktop, so users have access to the full capabilities of the instrument. Through Science Studio, access control and session management are also provided for this instrument.

SCIENCE STUDIO

The Science Studio web portal is an extensible platform that allows scientists to collaborate on research projects, and provides remote access to scientific resources. One resource that is integrated into this system is the VESPERS beamline located at the Canadian Light Source synchrotron [1]. Science Studio provides beamline users with remote access to this powerful scientific tool, and allows experimental data to be easily shared among the project team.

Science Studio is also a framework that can be used to more easily enable remote access to other devices. This framework provides session and experiment management features. Session management allows for remote access to be allocated or scheduled for a specific project team. Experiment management allows the project team to organize and share experimental data. Within the framework is a customizable web portal that provides users a single consistent entry-point for remote access and other services. This web application allows users to manage experiment information and experimental data using in a rich web interface. Security features, such as single sign-on and access control, are also included in the Science Studio framework.

X-RAY MICROANALYSIS SYSTEM

The Western Nanofabrication Facility (WNF) is an open user facility at the University of Western Ontario (UWO) for the fabrication of micro- and nano-structures. This facility has an assortment of equipment and instrumentation that provides its users with a wide range of capabilities; including lithography, deposition, etching and characterization [2]. An instrument of particular interest to users is the LEO (Zeiss) 1540XB Scanning Electron Microscope (SEM) with an integrated Oxford Instruments X-Ray Microanalysis (XRMA) system.

The SEM is a stand-alone instrument with specialized hardware and software for device control and data acquisition. The SEM control software is used for

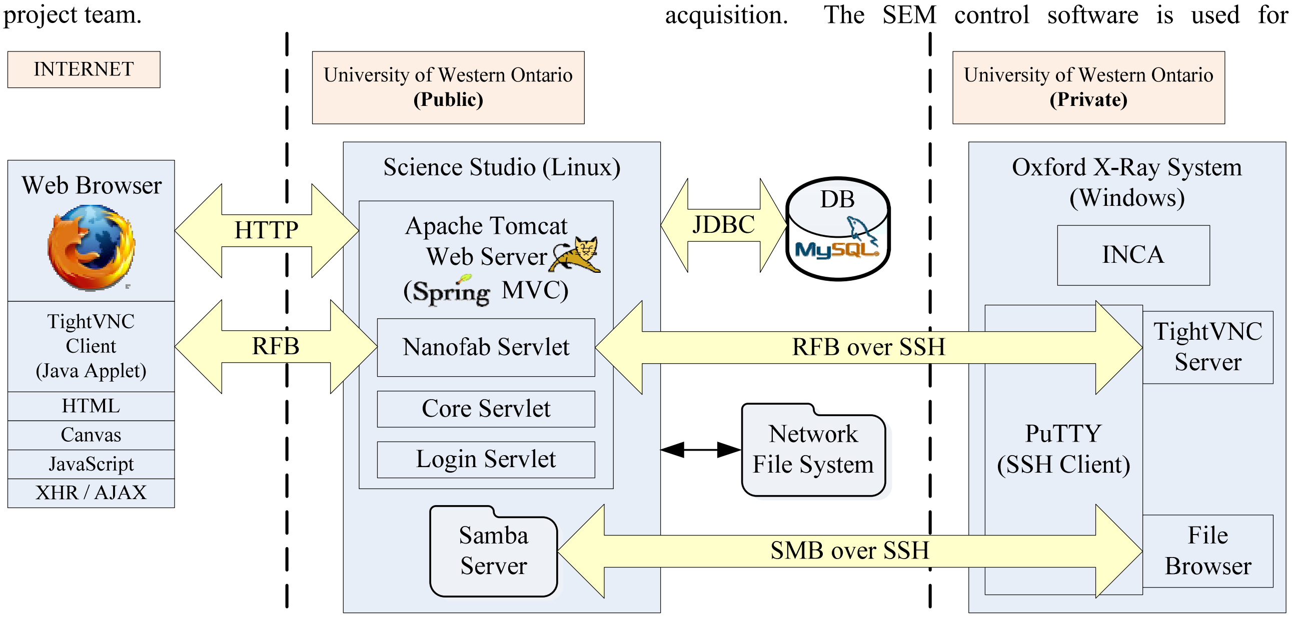

Figure 1: Science Studio architecture for remote access to the x-ray microanalysis system.

*This work was funded by CANAIRE under R&D project NEP-01; and performed at the Canadian Light Source which is supported by NSERC, NRC, CIHR and the University of Saskatchewan.
#dylan.maxwell@lightsource.ca

CLS USER SERVICES WEB PORTAL*

D. Medrano#, L. Carter, D. Maxwell, CLS, Saskatoon, SK S7N 0X4, Canada

Abstract

The Canadian Light Source (CLS) User Services Web Portal is a collection of web applications that allows users and staff to manage experiment proposals, complete safety training and submit end-of-run surveys. Each user wanting beam time must submit a proposal describing their experiment. Once submitted, the proposal goes through a peer-review process where it is either approved or rejected. All on-site personnel are required to complete safety training. Staff and users are provided with training modules which are completed online. Most training modules consist of two parts: the presentation and the exam. The exams are graded automatically and the results are stored. At the end of each run, users are encouraged to complete an online survey. The survey gives users the opportunity to provide feedback on what was good about their CLS experience and what can be improved to provide them with better service. This paper will give an overview of the design, implementation and capabilities of web portal.

INTRODUCTION

The CLS is an international research facility with a large number and variety of visitors, including students, scientists and contractors. Access to the facility is based on three different roles: users, staff and contractors. Users are scientists that come from all over the world to run scientific experiments on the beamlines, collect data and publish results. Staff operate and maintain the facility and support users in their research. Contractors are personnel hired by the staff for a certain amount of time to complete a task. Contractors can range from labourers to project managers. The goal of the portal is to provide a user friendly and maintainable system to store information associating people with these roles.

PORTAL ARCHITECTURE

Figure 1 shows the architecture of the web portal. It consists of six parts. The first part is the Apache Tomcat servlet container. It is in charge of hosting the four web applications which are the main focus of this paper. These web applications include: training, proposal submission, end-of-run survery and proposal information Application Programming Interface (API). The second part is the Microsoft Active Directory (AD) server. Its purpose is to store all user, staff and contractor login information. Usernames and passwords are authenticated against AD when someone logs into the portal. All communication to AD is done through the Lightweight Directory Access Protocol (LDAP). The third part is the MySQL database server. It contains multiple databases for storing training, proposal, and end-of-run information. Java Database Connectivity (JDBC) is used to communicate with the server. The fourth part is the workflow engine. The workflow engine is used to create the peer-review process a proposal must go through once it is submitted. Workflows are created using the Yet Another Workflow Language (YAWL) [1] and communication is done through the Simple Object Access Protocol (SOAP) [2]. The fifth part is other web systems that make use of the proposal information API. Communication is done through HTTP using Representational State Transfer (REST) [3]. The last part is the web browser which an end-user uses to access the web portal.

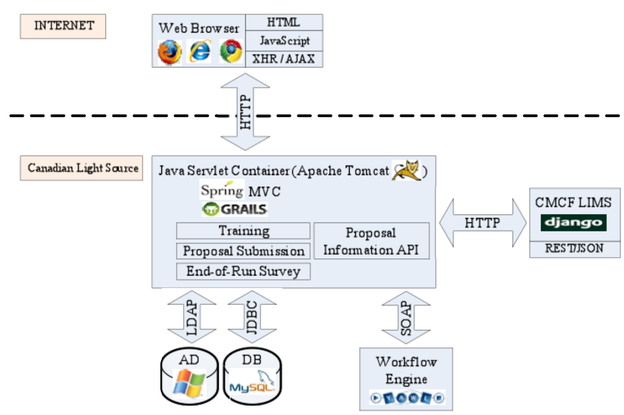

Figure 1: Architecture of the web portal.

TRAINING WEB APPLICATION

Any person coming to or working at the CLS that requires unescorted access must successfully complete and maintain their training. There is a standard set of training modules everyone must complete depending on their role. Users, staff and contractors use the web application to complete training and review training history. When someone takes an exam, a test is generated from a bank of questions in the database. Once a test is submitted, it is marked automatically by the system and the results are recorded. If a person fails the test they must retake it. Staff with the role of administrator use the application to create, edit or delete training modules, generate training reports, create new logins, manually enter training scores and manage training groups and roles.

The training application is written in Java using the Spring Model-View-Controller (MVC) [4] web framework. MVC is an architecture pattern that isolates domain logic from input and presentation. Figure 2 shows the breakdown of the application.

*Work supported by CANARIE under R&D project NEP-01 and performed at the CLS which is supported by NSERC, NRC, CIHR and the University of Saskatchewan.
#Dionisio.Medrano@lightsource.ca

EPICS DATA ACQUISITION SOFTWARE AT THE CLS

G. Wright, R. Igarashi, Canadian Light Source, Saskatoon, Saskatchewan, Canada

Abstract

The Canadian Light Source (CLS) Data Acquisition library provides a simple scan and store interface for CLS beamlines. Originally intended as a tool for testing and commissioning, it has been used in *QT* and *GTK+* user applications at the beamlines. The current version supports dynamic loading of custom output modules to allow re-definable data transport methods and multiple simultaneous output formats.

INTRODUCTION

During the construction phase of the initial beamlines at the CLS, the need for a simple graphical user interface to perform simple motion control and record results became apparent. An application was written using the GTK+ toolkit. As time progressed, the scanning code was moved into its own library, and this library provides the building block for a number of applications at CLS beamlines. These applications now provide a signification portion of the data acquisition toolkit available at a number of the CLS beamlines.

KEY FEATURES

Configuration

A simple configuration file defines the scan and data collection (**events**).

All data acquisition structures can be accessed and manipulated by the calling program. Standard configuration files can be used to define the initial scan, and with simple data structure manipulation the acquisition can be modified without requiring a custom configuration file.

All Process Variables and range values for acquisition can be specified by macro strings. Again, a standard configuration file can be updated for different data ranges without dealing directly with the configuration.

The data output stream is passed to **Output Handlers** to determine where and how the data is dealt with. This provides the opportunity for customized handling for new data formats or visualization without rebuilding the library.

All scans and events run in independent threads. This allows simultaneous collection of data or recording of large data sets simultaneously with a positioner update.

Scans and Events

The acquisition library has two main components. The first component, the scan, is the definition of the control for the experiment – typically moving a device, and then requesting a detector to detect. A single scan typically only controls a single device. Scans can be nested, allowing multi-dimensional scans.

The second component is the event. An event is triggered from a scan, typically when a detector has finished reading. The event collects data from a list of process variables and requests an Output Handler to deal with the data.

Output Handlers

The section of code that generates the most controversy is the part that defines the output data format. In the data acquisition library, this part of the code has been generalized to a set of function calls that can be set at run time. An object-oriented "factory" for creating links to different handlers interfaces different data formats and different data transports to the acquisition library, allowing a great deal in flexibility on the calling applications' part to deal with new display and storage requirements, and even allow multiple simultaneous data files to be written in different formats.

The second benefit of using output handlers is that any viewer becomes generalized, and can either be integrated directly in the application (as was done for the *Motor Scan* screen in IDA (Interactive Data Acquisition, described later) or streamed to an independent application (such as BLGraph or Grace).

The output handlers have two components: the data format, and the data stream. The data format defines the appearance of the data, whether it's a simple comma separated value text output, or a compact binary output. The data stream is the destination, such as a data file, a named pipe, or a TCP/IP port.

Each instance of an output handler has a set of properties that can be updated through a standard application interface. An application can obtain information on properties and allow these to be controlled directly by the user, so new handlers can be configured without needing to update the user interface software.

Data Visualization

The primary tool used for data visualization with the acquisition library is **BLGraph**. This ROOT-based display tool is highly configurable. Dynamic selection of data fields for display, and manipulation of multiple fields with functions, as well as customizing configurations for quick set up of standard acquisition runs makes this a good match for acquisition library. As well, previously recorded data can be retrieved and viewed.

APPLICATIONS

QT Widget Support

The Qt Widget library from Nokia offers a powerful Rapid Application Development tool (**designer**) and the

CLS LINAC SAFETY SYSTEM UPGRADE

Hao Zhang, Elder Matias, Grant Cubbon, Carmen Britton, Robby Tanner, Carl Finlay
Canadian Light Source Inc., Saskatoon, Canada

Abstract

The Canadian Light Source (CLS) upgraded the safety system for Linear Accelerator (Linac) in October 2009. IEC 61508 SIL 3 certified components and methods were adopted in the development of the new system. This paper outlines major aspects of the upgrade.

INTRODUCTION

In the CLS, Access Control and Interlock Systems (ACIS) are used in restricted areas to protect personnel from radiation hazards. In the Linac area, a legacy ACIS was used since 1980's until October 2009. The system was based on early Micro84 Programmable Logic Controller (PLC). Given the age of the system, difficulty in procurement of spares as the vendor had discontinued support for the platform; a decision was made to upgrade. Another reason is the old AICS used 120 VAC whereas CLS has adopted 24 VDC for all other control systems. The upgrade ensures the Linac ACIS is consistent with other systems in the facility. All the old sensors, wirings, components, and PLC units were removed. The new ACIS was redesigned and built from scratch.

The new ACIS adopts a two-level, redundant protection mechanism which consists of two independent chains, one governed by a safety-rated PLC system providing SIL-2 as defined by IEC 61508 [1], and a relay-based hardware logic to provide diversity for safety functions.

The system controls access to an area divided into 6 lockup zones [2]. The zone layout was also changed in the upgrade. The zones contain the electron gun, accelerator sections, switchyard, LINAC-to-Booster Transfer Line (LTB), the LTB/Booster Ring (BR1) interface and some adjacent areas including the BR1 RF cavities.

Fundamentally, all lockup zones operate in the same principle, each having its own Emergency Off Stations (EOS), Door Interlock Switches (SWDI), Lockup Stations (LUS), zone lockup lights (ZLL) and horns (HRN).

BACKGROUND

Regulatory Context

CLS holds a Particle Accelerator Operating Licence (PA1OL-02.00/2012) issued by the Canadian Nuclear Safety Commission (CNSC) to operate as a Class 1B facility; as a result the definition of internal process is left to the CLS with the CNSC providing review, oversight, and audition.

Project Plan/Management

The upgrade was carefully planned and documented. The plan identifies project objectives and goals, specifies the upgrade scope, lists standards and guidelines for the development, and defines roles and responsibilities of team members. The plan also includes work structure breakdown, budget, timelines, and a list of documents need to be generated or modified. The plan served as the guiding document during the development process.

Safety System Development Process

The upgrade followed a V-model variant for safety system development.

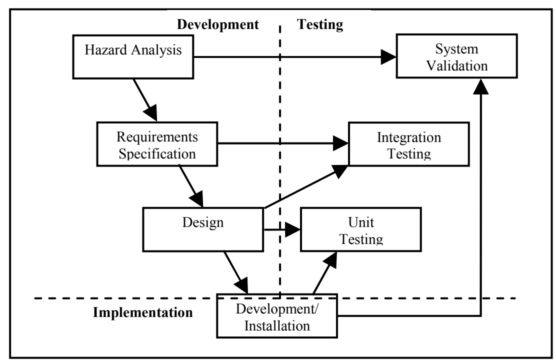

Figure 1: Safety System Development Process

The process starts with the hazard analysis, based on which requirements and specifications are generated, and design and implementation naturally followed from there. Testing was performed in all stages. Respectively, integration and unit testing verify the design meets the requirements and the installation is done as the design.

Hazard Analysis

Since the layout of Linac lockup zones was to be changed, a Hazard Analysis (HAZAN) [3] was necessary to identify the hazards and associated mitigations required with regard to the proposed redesign and upgrade. This was performed by the Health, Safety, and Environment (HSE) department of CLS. The document issued was used as input to the following development stage.

Requirements

The hazards which have been identified and allocated to the ACIS for mitigation in the HAZAN were then examined and refined to generate requirements for the ACIS. Other internal or external guidelines, such as human factor guideline [4] and Canadian Electrical Code were also incorporated as requirements in this stage. Operation experience on the old Linac ACIS and other ACISs was also taken into consideration. A design manual was generated to document all requirements. Linac lockup zone layout drawings were generated to

* Research described in this paper was performed at the Canadian Light Source, which is supported by the NSERC, NRC, CIHR, the Province of Saskatchewan, Western Economic Diversification Canada, and the University of Saskatchewan.

FEC IN DETERMINISTIC CONTROL SYSTEMS OVER GIGABIT ETHERERNET

Cesar Prados Boda, Tibor Fleck, GSI, Darmstadt, Germany

Abstract

Forward Error Correction (FEC) is a technique for recovering from bit errors and frame losses in real-time network applications. Classic recovering strategies, like TCP retransmission, are not suitable due to delay, timing and bandwidth constraints. In this paper, we introduce the FEC technique in a novel deterministic fieldbus, White Rabbit [1] (WR). WR is developed over frame-based computer networking technology, Gigabit Ethernet, GbE. WR provides an effective and resilient way to serve as a deterministic data transfer medium and to interconnect large distributed systems, like Control Systems for Particle Accelerators. The reliability of WR falls on the FEC, which provides the means to guarantee that only one control message per year will be lost or irretrievable as a result of the Bit Error Rate of the physical medium (fiber optic or copper). We propose in this paper a FEC base on LDPC [2], and tailored for broadcast communication in switched networks over noisy channels without retransmission.

INTRODUCTION

Control systems have distributed nodes that need to be connected under specific operation constraints: synchronization accuracy, determinism, bandwidth limitation etc... Besides, the medium over which the communication happens, is a noisy channel where the bits of the frame could be erased or modified. Also, the switches used to propagate the information can mislay or dump such information as a result of collisions in the routing process. So as to ensure an adequate performance of a control system, it has to be endowed with a mechanism capable of overcoming the errors in the communication. Such mechanism is called Frame Error Protection (FEP) and among the different alternatives, in this paper the Forward Error Correction will be discussed. We present the groundwork of an underway research to provide high reliability to time-critical control systems based on GbE and switched networks. The paper is organized in three sections. The first section presents the framework where the FEC is being developed, WR Project, and its boundary conditions. The second section presents how these boundary conditions affect the transmission of data over GbE. In the final section, we analyze the whole scenario and present a FEC scheme to ensure the required reliability.

CONTROL SYSTEMS AND WHITE RABBIT PROJECT

WR is a solution to the generic problem of transferring data in a fast, deterministic and safe manner. WR Protocol (WRP) [4] allows the delivery of timing and control data over a Gigabit Ethernet LAN. WR can be seen as an extension of Gigabit Ethernet, which provides synchronous mode, deterministic routing, bi-directional exchange of frames between nodes and precise delay measurement.

The synchronous mode is achieved by using Synchronous Ethernet along with IEEE 1588, PTP protocol. This combination of protocols provide the means to distribute through the physical layer a common clock within the entire network up to e.g. 2000 stations, allowing 1ns synchronization and 20ps jitter. The frame transmission delay between two stations will never exceed the sum of 64 byte clock cycle plus the propagation time in the longest communication path of the network.

To distinguish between WR and other possible Ethernet traffic in the network, two different frames are defined: SP, Standard Priority frame, which is non-deterministic, and HP, High Priority frame, which is deterministic. The latter frame type is specified in the WRP network to transport messages with the highest priority. HP are frames for time-critical control data, as a consequence, they are routed with lowest latency as possible, forcing fragmentation of non-HP traffic if required. These frames have absolute priority over SP frames and non-WR traffic to maintain low and deterministic transmission delay.

Coming along with the protocol, compliant hardware is being developed in order to support the protocol's features. There are three essential devices: White Rabbit Master, which generates the HP frames and is master clock as well, White Rabbit Switch and White Rabbit Receiver. As a consequence of the device's role and application requirements, the number of units needed in a standard network will consist of one WR Master, M WR Receivers, and $N_{WR_{Switch}}$ with P downlink ports each.

WR allows different approaches to organize the topology of the network depending on the specific requirement of the applications. The strategy for data transmission is based on the distribution from the master to all the other nodes of the network, directly or indirectly according to a Star or Tree topology. The HP frames will be broadcasted from the top of the network, where the WR Master dwells, to the bottom of the network reaching all the WR Receivers.

One of the principal features of the protocol is the notion of determinism, used to guarantee the execution of events within a certain period time. On account of the differential

LLRF CONTROL SYSTEM UPGRADE AT FLASH

V. Ayvazyan, K. Czuba, Z. Geng, M. Grecki, O. Hensler, M. Hoffmann, M. Hoffmann, T. Jezynski, W. Koprek, F. Ludwig, K. Rehlich, H. Schlarb, C. Schmidt, S. N. Simrock, H.-C. Weddig
DESY, Hamburg, Germany

Abstract

The Free Electron Laser in Hamburg (FLASH) [1] is a user facility providing high brilliant laser light for experiments. It is also a unique facility for testing the superconducting accelerator technology for the European XFEL and the International Linear Collider. As a test facility, the accelerator undergoes a constant modification and expansion. The last upgrade was started in autumn 2009 and has finished recently [2]. The beam energy is increased to 1.2 GeV by installing a 7th superconducting accelerating module. The new module is a prototype for the European XFEL. In order to increase the free-electron laser (FEL) radiation intensity by linearization of the beam phase space the 3rd harmonic superconducting RF cavities are installed in the injector. The old DSP based LLRF control system [3] has been completely upgraded to latest generation controller boards, down-converters for higher intermediate frequency, algorithms like beam loading compensation, feed-forward waveform generation, etc. are improved. In order to improve the reference frequency signals the master oscillator and frequency distribution system has been upgraded as well.

INTRODUCTION

The FLASH injector consists of a laser-driven photocathode in a 1.5-cell RF cavity operating at 1.3 GHz with a peak accelerating field of 40MV/m on the cathode. The electron injector section is followed by a total of seven TESLA type 12.2 m long accelerating modules each containing eight 9-cell superconducting niobium cavities. The accelerating gradients of the cavities are typically between 20 MV/m and 25 MV/m. Four cavities of sixth module and seventh module are providing gradients above 30 MV/m. The accelerating modules are powered by four RF stations consisting a klystron (tree 5 MW klystrons and one 10 MW multi-beam klystron), a high voltage pulse transformer and a pulsed power supply (modulator). In addition, the RF gun has its own RF station with a 5 MW klystron. The gradient and phase accelerating field (vector sum) of the RF gun and the accelerating modules are controlled by dedicated LLRF regulation system which has been completely upgraded during shutdown period. The FEL radiation is provided by 30 m long undulator section. The undulator consists of periodic structure of permanent magnets which have a fixed gap of 12 mm. The wavelength of the FEL radiation depends on the energy of the accelerated electrons. It can be tuned between 4.3 nm and 120 nm.

After the upgrade a successful operation of FLASH at a wavelength of 4.45 nm has been achieved [2]. For 4.45 nm radiation wavelength the accelerator provides beam energy of 1.207 GeV.

PRINCIPLES FOR LLRF CONTROL

The RF system signal flow is shown in figure 1. The cavity probe signal is converted from the cavity frequency of 1.3 GHz to an intermediate frequency (IF) of 250 kHz for superconducting modules and 54 MHz for 3rd harmonic module. This lower IF holds the original amplitude and phase information of the field inside the cavity.

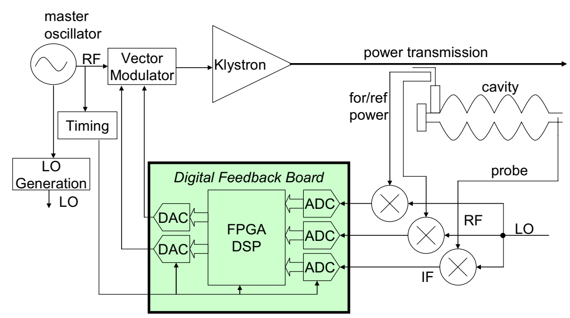

Figure 1: Architecture of the LLRF system.

It is digitized with ADCs (sampling rates of 1 MHz or 81 MHz are used). The digitized signal is going to the digital field detector which extracts the I and Q components out of the input stream. We use two different methods: IQ-sampling and so-called non-IQ-sampling or IF-sampling. The resulting field vector of each cavity is multiplied by a rotation matrix to calibrate amplitude and phases. Finally the field vectors of 8 cavities are summed up for the vector sum of a whole cryogenic module, and those of 2 cryogenic modules are summed up to the vector sum of the RF station which is driven by single klystron. The vector sum of the 16 cavity fields represents the total voltage and phase seen by the beam. This signal is regulated by a feedback control algorithm which calculates corrections to the driving signal of the klystron. The measured vector sum is subtracted from the set-point table and the resulting error signal is amplified and filtered to provide a feedback signal to the vector modulator controlling the incident wave. A feed-forward signal is added to correct the averaged repetitive error components. Beam current information (measured by toroids) is used to scale the feed-forward table to provide fast feed-forward corrections if the beam current varies. The cavity detuning is determined from forward power, reflected power, and probe signal and is used to control the fast piezo tuners to reduce cavity detuning errors to less than a tenth of the cavity bandwidth.

SCRIPTING TOOLS FOR BEAMLINE COMISSIONING AND OPERATION

A. Pazos#, S. Fiedler, EMBL-Hamburg, Hamburg, Germany
P. Duval, DESY, Hamburg, Germany

Abstract

Scripting tool capabilities are a valuable help for beamline commissioning and for advanced user operation. They are the perfect complement to static Graphical User Interfaces allowing one to create different applications in a rapid way. A light middle-layer for scripting support has been foreseen for the EMBL structural biology beamlines at the PETRA III synchrotron in Hamburg, Germany, to provide 'controlled' rather than 'direct' access to the control system devices. This prevents conflicts with the control system and allows control of the supported operations. In order to account for the wish of different scripting languages by the beamline scientists an extension of the scripting capabilities of the TINE control system has been implemented. To the existing shell support, a Python extension (PyTine) has been added and a Perl wrapping has been also prototyped (tine4perl). An explanation of these implementations and the different wrapping possibilities is also described in this paper.

INTRODUCTION

The EMBL-Hamburg outstation is commissioning three beamlines at the new PETRAIII light source at DESY (Hamburg). In addition, two beamlines at the DORIS storage ring are available for testing and prototyping the arriving instruments.

The control software is based on a client/server architecture integrated with the TINE control system [1]. Each device exports a TINE server that allows its remote operation. Flexibility has been a key feature since the design phase. For this reason different kinds of programming languages like C/C++, Python and Labview™ are supported.

The client side is mainly represented by Graphical User Interfaces (GUI) that connect themselves to the existing device servers. Two kinds of GUIs are available depending of the application. On one side there is an advanced control GUI that allows the operation and tuning of the entire beamline. This is mainly used by the beamline operators and experienced personnel. On the other side there is a GUI for visiting scientists with limited functionality with the main purpose of performing the data collection.

Some procedures, not even supported by the advanced GUI, need to be executed during the commissioning. Moreover, advanced users have the requirement of executing different strategies that are not foreseen at the user GUI.

In both situations the availability of a flexible and rapid way of executing this set of actions is very desirable. For this reason a scripting layer has been introduced at the software architecture allowing one to “glue” calls to the device servers. For gluing and system integration a scripting language can be 5-10 times faster than a system language [2] and the strong typing makes the programs easier to manage.

It is not desirable to the overall operation of a beamline that a user, not familiar with the installed hardware, is allowed to freely execute server functions. Of course, there are control system security measures, but overlaying the servers with a light scripting interface makes the system safer. Thanks to this scripting layer, the naming convention of the functions can be freely chosen.

SCRIPTING REQUIREMENTS

On the basis of our experience with beamline operation and after evaluating the specifications given by the beamline scientists, a list of requirements for the desired scripting environment was compiled:

- Easy to learn (for the developers and for the users)
- Easy to maintain
- Flexible (possible to refactor)
- Dynamic (does not need variable declarations)
- Well defined syntax
- Well documented
- Possible to control the accessible functionality
- Separated from the device specific layer
- Command-line support
- Sequencer support
- Reliable
- Secure
- User proof
- Multi-platform
- Open-source

TINE FOR SCRIPTING

The TINE control system originally supported a set-up of shell commands meant to build shell scripts both in Linux and Windows. Examples for these are the ‘tget’ (to receive data from a server) and the ‘tput’ (to send data to a server) commands. These functions are implemented in C and make use of the TINE C API. They receive as an input the necessary information (address, property, data type and data size) to make a call to a server.

#apazos@embl-hamburg.de

THE ANKA B-FIELD TEST FACILITY CONTROL SYSTEM, BASED ON A SPEC MACRO PACKAGE ENHANCED SETUP*

Karlheinz Cerff, Thomas Spangenberg, Wofgang Mexner, Institut for Synchrotron Radiation, (ISS)-ANKA, Karlsruhe Institut of Technology, (KIT)-Campus North, Germany.

Abstract

The ANKA B-field test facility provides users with a flexible tool to investigate magnetic field distributions of different setups of coils or permanent magnets, optimal sensor types, geometrical alignments of probes and the possibility to change the independent physical stimuli to generate and alter magnetic field distributions [1]. From the point of Software development it is taken as an example of a straight-forward device implementation with a recently introduced type of macro based 'building block system' for devices in SPEC, [2]. This macro package provides the C-like SPEC with an object orientated framework with a namespace and class concept to represent the power supplies of different brands, probe positioning devices and measurement amplifiers.

INTRODUCTION

The B-Field Test facility provides measurement data of magnetic field distributions of coils or permanent magnet structures, within the range of um spatial resolution, over positioning ranges up to meters, devices in use are,

- a stepper motor driven, encoder monitored linear positioning probe, equipped with a variable geometrical arrangement of Hall-sensors to measure B-field induced voltage gradients.
- Two power supplies, consisting of a main and a second, multiple power supply, driving individual shaped I-current ramping functions for corrector coils.
- A Digital Multi-Meter (DMM) of Keithley, type 'k2700' to read out, up to n Hall-probes.

The control software package should also generate a raw data fit for a polynomial of variable degree i (i< =9), for up to n Hall-probes. At last the control system monitors the safe operation of the Test facility, for example it shuts down the main power supply when a superconducting coil under test is quenching.

IMPLEMENTATION

In the context of the 'Macro package based Enhancemend of SPEC controlled Experimental Setup'[3], this means that the device properties are stored as elements of data structures (SPEC global associative arrays). The task of the software development is, to

- set up an abstract model of the B-Test Facility hardware devices.
- write the device drivers for B-Test Facility motor, power supplies and digital multi meters.
- linking the resulting SPEC macro functions to the Interface generated by enhanced macro package.

The introduction of a set of interfacing rules minimizes the risk of damage to existing SPEC-structures, furthermore it opens the possibility to port in this way generated SPEC-'classes' to other experimental facilities.

Table.1: B-Test facility, list of realized implementation of functions, devices, SPEC '-instances' and –'classes'.

physical function	device	SPEC-'instance'	SPEC–'class' (macro)
motor controller, one channel	OMS-Maxv	'm0'	Motor.mac
main power supply, 1 channel.	FUG NTV-1000	'fugbig'	Fug.mac
power-supply small, 8 channels	FUG NTV-100	'fug'	Fug.mac
Digital multi-meter Hall-probes	Keithley, K2700/7703	'k2770' 'Hall n'	Anka-Keithley.mac

Setting up the B-Test Facility, the two power supplies are defined as members of the 'class', represented by FUG.mac. They are both instantiated as objects 'fug' and 'fugbig' in the declared global associative array 'FUG', writing a set of device dependent standard-values to it. SPEC-associative Arrays offer as possible arguments arbitrary strings or numbers instead of integers [2]. In the 'class'-macro keithley_anka.mac, the Keithley DMM is instantiated as object "k2700" and the connected Hall-probes as objects "Hall-1"-"Hall-15. The minisetup class' macro contains the 'standardvalues' declarations and a data fit object to fit raw data to a polynomial up to the order of nine.

Benefit

- Two FUG devices, representing nine power supply 'objects' can be accessed by 11 (for the main power supply) and 73 (for the corrector power supply) standard-function calls obeying the naming rules introduced by the macro package.
- Up to fifteen Hall-probes have to be addressed by 255 standard function calls for the Hall-probes plus three functions for the K2770.

The advantages using the object oriented approach is clearly visible, there is no need to write, a set of 84 nearly identical conventional SPEC-functions for power supplies and additional 255 functions to handle the output, in addition existing ANKA-beamline driver modules for motors can be used.

MACRO PACKAGE BASED ENHANCEMENT OF SPEC CONTROLLED EXPERIMENTAL SETUPS

Thomas Spangenberg*), Karlheinz Cerff, Wolfgang Mexner
Institut for Synchrotron radiation, ISS, ANKA, KIT-Campus North, Karlsruhe, Germany

Abstract

Certified Scientific Software's program package spec [1] for X-Ray diffraction and data acquisition provides reliable instrument control to scientists at synchrotrons and other facilities worldwide. It's very flexible C-like macro language provides a large number of degrees of freedom for experiment control as advantage and as big disadvantage at the same time. A large number of programmers with their own ideas and naming conventions are contributing to the growth of functionality. At the same time the risk of collateral damage by accidentally overriding already existing functions and variables grows constantly. To solve this dilemma a new object oriented like software development concept for spec is proposed. A few naming rules plus a macro package in combination with a single client-server-application expand the manageability and options to control experiments considerably. As main goal spec gets an object-like handling and a standardized user interface of newly introduced devices. A generic server-client based interface allows a smooth integration of spec in more complex control environments via TANGO [2].

INTRODUCTION

Most of the physical and logical devices provides the opportunity to operate them in a simplified model as a set of independent properties which are offered by a certain remote interface. Therefore it becomes possible to integrate them rapidly into its own measurement setup either by direct driver support or by some macro integration.

As an example, the software package SPEC with its flexible macro language and various interfaces offers a number of paths to implement additional hardware into an experiment.

It will be shown that the risk of interfering solutions can be avoided for the device integration by introducing a few design rules in combination with a macro package. Additionally the client server based export possibilities of the integrated devices will be increased significantly.

MACRO PACKAGE AND DATA STRUCTURING

The basic idea of that macro package is to organize and handle devices object like although SPEC's pure macro based programming language definition doesn't support objects directly. But the provided data structures permit with a few limitations an object like structuring of data and a macro supported creation of specific functions to manipulate them.

Starting from the abovementioned simplified device model the representation of the device properties is stored into SPEC's associative arrays (see Fig. 1) which yields three advantages.

- First, all objects of one class are stored in only one array variable. It is evident, that a naming conflict can be prevented by using a single identifier per class.
- Second, due to SPEC's data type definition any type of data can be stored into this array.
- Third, the two dimensional index organized by strings is well suited to store data differentiated into 'objects', their properties, and their methods.

The data organization of the macro package is basically funded to associated arrays and is introducing a naming convention to their indices. SPEC defines associative arrays as a string indexed data object which stores any type of information. The first dimension of the two dimensional index is used for the device name. The name is usually chosen as an acronym which describes the device function in the experiment (e.g. vc1 for vacuum controller 1, see fig. 1).

The second part of the index string is primary subjected to the device property. Additionally the first character is used to transport the minimal necessary information about the represented property which is used for the automatically generation of the user interface. The implemented scheme is as follows:

- '$' indicates internal variables. There are no user functions provided.
- '*' indicates read only properties or variables. Read functions are provided.
- '!' indicates a command. A command function will be available.
- no special character indicates a read/write property. Read and write functions are provided.

The formal initialization overhead due to the macro package is very small. There are 2 functions for the whole macro package and only 3 additional steps are needed to implement a new device. There are:

- The formal declaration of the device instance by name and device type, followed by
- The initialization and declaration of start-up values and finished by
- Initializing the device or synchronizing the stored information.

The macro package evaluates the stored data and creates automatically the functions to manipulate them obeying the fixed naming scheme. Thereby the whole user functionality will be generated.

*) thomas.spangenberg@kit.edu

STUDY CASE OF A COLLABORATION PORTAL FOR A INTERNATIONAL SCIENTIFIC PROJECT

Marcin Trycz and Luciano Catani, INFN-Roma Tor Vergata,

Via della Ricerca Scientifica, 1 - Roma, Italy

Abstract

In this paper we present the results of the design, development and preliminary evaluation tests of a web-based collaboration portal aimed at supporting the teamwork of an international scientific collaboration.

In the academic research environment often people use very simple collaboration tools, usually chosen out of habit. In the case of international collaborative projects, in which people don't work physically in the same place for most of the time, these important tools are far from being effective and appropriate. For instance, a collaborative scientific project is made of teams of specialists from different research institutions and countries that need to share files, drawings, pictures, software etc. and document the progress of their work. The different tasks of the project are managed by work groups (WGs) of specialists that organize their work by scheduling meetings, workshops and by setting deadlines. Quite often a single researcher contributes to more than one work group.

The aim of our Portal is to offer a suite of web instruments fulfilling the above requirements without adding extra complexity to the procedures the scientists are familiar with.

INTRODUCTION

In the present-day world of science, research projects are often carried out by large collaborations of different research institutions and universities. The need of specialists for each task of the project requires the contribution of top-quality scientists from different parts of the world working in collaboration across the different phases of the project development: design, operation and the analysis of results.

Sometime the collaboration is based on in-kind contribution from each partner requiring a constant interaction to ensure the perfect matching of the components.

Given these requirements, a continuous and effective communication among members of work groups, and a constant coordination of the latter, is crucial for the successful development of the tasks. At higher level, WGs leaders should continuously check the progress of their own group against each other to ensure a uniform development of the project.

Scientists, compared to many other professional communities, are certainly skilled and well trained in using computers and computer networks because of the important role these instruments have in their daily work.

As consequence of this familiarity, scientists spontaneously tend to profit from computer based communication and collaboration tools, selecting by themselves the solution they consider more appropriated.

This explains the tendency to develop solutions to their collaboration needs that simply implement the tools they are more familiar with: email especially, for communication and documents distribution, file servers, Internet shared agenda, polling services etc.

Experience teaches that, in spite of their familiarity with Internet technologies, or probably as a consequence of it, scientists are somehow reluctant to accept dedicated all-in-one project management solutions that might be selected and suggested by the management. Often, they are convinced that the collaboration instruments they currently use are sufficient or even more effective that the new one.

The above considerations suggested us to start the development of a web portal aimed to providing a "smooth" replacement of basic communication and collaboration tools with a centralized server.

OVERVIEW OF TOOLS AND TECHNOLOGIES

The first task of our analysis process was the identification of the framework suited for our needs.

We ended up with three candidates representing a wider spectrum of technologies: Xoops, Joomla and Liferay. The first two are PHP frameworks; the latter is by now the only open source Java Portal. A deeper analysis showed that only Liferay would have fulfilled our user scheme. On the other hand this portal it's far from being simple, but its complexity can easily be hidden to the final user.

THE COLLABORATION PORTAL

The Portal is meant to be a web-based integrated set of tools supporting a large collaborative project as a communication and documentation service. The two main goals of the Portal are prompt and effective information sharing and well ordered archiving. The Portal also contains additional services that support other aspects of the collaborative work, a calendar for instance, and by taking advantage of the Java portlet [1] "plugability", others can be added if need be.

Users Management

As we already mentioned, the main reason for adopting Liferay was its powerful user management capability. In international, or large national scientific collaboration scientists from many different research organizations

DEVELOPMENT OF IMAGE PROCESSING SYSTEM ON EMBEDDED EPICS FOR BEAM DIAGNOSTICS

J. Odagiri, K. Furukawa, T. Obina, M. Satoh, High Energy Accelerator Research Organization (KEK), 1-1 Oho, Tsukuba, Ibaraki, Japan

Abstract

A new image processing system was developed based on EPICS and the FA-M3 PLC made by Yokogawa Electric Corporation. The hardware of the system comprises an F3RP61 CPU module running Linux and an F3UM02 frame grabber module. The CPU functions as an IOC to analyze the raw image data acquired and transferred by the frame grabber module on the PCI-bus, which connects the two modules. A custom record, graphicsRecord, holds the raw image data and the results of analysis as well as parameters set by the user over the network. GUI panels were created by using EDM in order to display the image and to set relevant control parameters into the fields of the graphicsRecord being stationed on the memory of the F3RP61-based IOC. It was confirmed that the developed system is able to acquire image data, analyze them appropriately, and send them over the network to a host computer to display the results of analysis. The design and results on performance measurement of the system is also reported.

INTRODUCTION

It had been common practice to use a desktop PC with frame grabber boards installed in it for beam profile monitoring. This approach allows us to broaden the range of choice of the frame grabber boards and the PC for the purpose. On the other hand, short lifetime of the products and less reliability of the hardware forces us to replace the system frequently to increase burden in maintaining the system in the long run.

In order to solve the problem, we have adopted embedded technology with Experimental and Industrial Control System (EPICS) running on a Programmable Logic Controller (PLC) made by Yokogawa Electric Corporation [1]. Fig. 1 shows the image processing system under test. The main specifications of the F3UM02 frame grabber module are listed in Table 1.

HARDWARE CONFIGURATION

The system comprises an F3RP61 CPU, which runs Linux as its Operating System (OS), and an F3UM02 frame grabber module. The two modules are connected with each other by using not only the PLC-bus on the backplane but also an additional PCI-bus. Both of the modules have a PCI- connector on the side panel to stack them for faster data transfer. The image data acquired with the frame grabber module is transferred to the F3RP61-based CPU by using DMA. The CPU executes the Input / Output Controller (IOC) core program of EPICS on Linux. The IOC analyzes the raw image data and sends it with analyzed results to a host computer which functions as an Operator Interface (OPI) of EPICS.

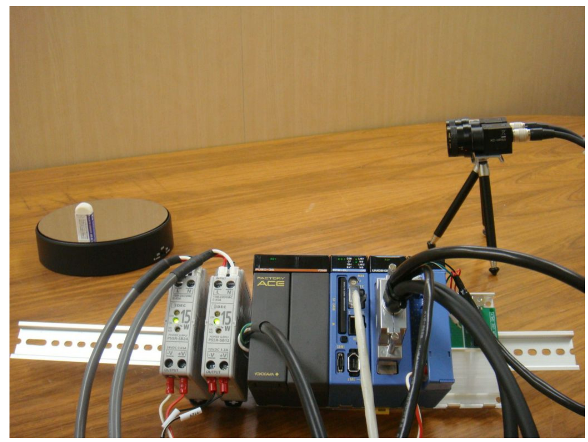

Figure 1: Image processing PLC unit under test. The left most black module (two slots) is the power supply module. The F3RP61 CPU comes to the right of the power supply module. The module just right to the CPU module is the F3UM02 frame grabber module.

Table 1: Main Specifications of F3UM02

Item	Specification
Number of Channels	2ch
Compatible Camera Types	Single Tap (8bits/pixel) Dual Tap (bits/pixel) RGB Colour (24bits/pixel)
Max. Connections	6 monochrome cameras
Resolution of Digitizer/Channel	8 bits
A/D Converter Frequency	100 MHz

SOFTWARE DEVELOPMENT

Record Support

An existing spherical record type, graphicsRecord, which had been created for a seat-gas beam profile monitor was used with some modifications for the analysis of raw image data, such as subtraction of background image, calculation of the projection to both horizontal and vertical directions, searching the peak position in the projection, calculation of the total amount of the light and so forth [2].

CONTROL AND ACQUISITION SOFTWARE COMPLEX FOR TBTS EXPERIMENTS

A. Dubrovskiy, CERN, Geneva, Switzerland

Abstract

The Two-beam Test-stand (TBTS) is a test area in the CLIC Test Facility (CTF3) to demonstrate the high power RF extraction and acceleration at a high accelerating gradient, which are feasibility issues for the Compact Linear Collider (CLIC) project. In order to achieve an efficient data collection, an acquisition and logging software system was developed. All year round these systems store the main parameters such as beam position, beam current, vacuum level, pulse length etc. For predefined events they also gather and store all information about the last several pulses and the machine status. A GUI interface allows from anywhere to plot many logged characteristics at a maximum of 10 minutes delay, to go though all events and to extract any logged data. A control interface configures actions and long-term control procedures for conditioning accelerating structures. The flexible configuration of the logging, the acquisition and the control systems are integrated into the same GUI. After two years operation the critical components have shown highly fault-tolerant. Logging data are used for physic researches.

INTRODUCTION

CTF3 is a test facility which addresses the feasibility demonstration of the Compact Linear Collider (CLIC) [1,2]. The CLIC machine will produce electron-positron collisions at the nominal center of mass energy of 3 TeV at a luminosity of $2\times10^{34}\,cm^{-2}\,s^{-1}$ with a two-beam acceleration scheme. This scheme is studied in the Two-beam Test-stand (TBTS), which is a part of CTF3. An electron beam (the drive beam) of 12 GHz is generated from a 1.5 GHz electron beam in a Delay Loop and Combiner Ring and then sent to the TBTS. The drive beam of an intensity of up to 32 Amps passes through a Power Extraction Structure (PETS). The extracted 12 GHz RF power from the drive beam is used to accelerate the second, low-intensity, beam (the probe beam). In the TBTS set-up the CLIC feasibility, stability and protection issues are studied, such as the beam changes during the deceleration, the RF extraction properties by the PETS, the high-gradient acceleration, as well as the Two-beam scheme performance and the fault-tolerance [3].

A control and acquisition high-level software complex was developed in order to assist all TBTS experiments, measurements and control routines. From the user point of view, the acquisition and logging parts of the system must be extremely reliable and robust; and it must work round-the-clock. The software system is flexible and adaptive to failures of hardware or software components involved in the TBTS set-up. Another issue is that the development of the software continues during several years such that it follows the requirements of R&D experiments and the hardware installation; and it remains light in support and compatible. The automatic control part contains a material protection mechanism and an accelerating structure processing.

FRAMEWORK

The TBTS software design approach is based on a model-driven architecture. The software developing process contains two distinct periods of time. During the first and initial period the developer followed the waterfall model approach. Specifications for different software aspects were completed iteratively during an extended period of time. That is why the first four stages of the software development consecutively alternated: requirements analysis → software design → integration → testing → requirements analysis → and so on. During this period the full range testing is very time consuming and some aspects remain unknown. Hence the testing, the validation and the performance estimations were made for some aspects of typical situations. The model merging is one of the most difficult processes during this phase. In order to simplify this problem, a core model was designed, which covers the static part of the set-up and it remains independent of the software and hardware realisations. The core software model was developed based on the instrumentation, controllers and machine time triggers layouts and general specifications. The acquisition model defines the generalized device interfaces for different data access interfaces and different types of equipments. The control model depends on only the core model and the acquisition model. At the end of the first period most of this was defined and realized in the server part of the software. The remaining part is gradually put in operation during the second period taking into account the importance of the blocks. So the second period of the development relies on the feed-back from results, goals and tasks of experiments and set-up changes. During this period the development becomes lighter and faster, the development tends to be agile.

ACQUISITION

The acquisition part of the software complex is to obtain all necessary information about the CTF3 machine status and the experiment. CTF3 can run in several modes for the TBTS beam lines:

- only the drive beam is on;
- only the probe beam is on; synchronizly;
- probe and drive beams are on, but not synchronized;
- probe and drive beams are on and synchronized.

Moreover all measurement equipments are located on different front-end crates in the network, and

ESTIMATION OF THE RESPONSE TIME AND DATA FLOWS IN THE TOTEM DETECTOR CONTROL SYSTEM

F. Lucas Rodríguez, CERN, Geneva, Switzerland

Abstract

The TOTEM experiment at the LHC is composed by 3 different detectors, (Roman Pots silicon detectors, CSC T1 and GEM T2 telescopes). The Detector Control System (DCS) is generated in a highly automated process from external representations for connectivity and behavior. From these representations (one of them, the Finite State Machine tree) it is also possible to estimate the response of the system. It is possible to assign weights to each one of the nodes and estimate data transfers among subsystems, memory consumptions, reaction times, storage needs,. . . The main purpose of those estimations is not to do a full predictability analysis of the system, but just to provide a help in case of performance problems.

INTRODUCTION

The TOTEM (Total crOss secTion, Elastic scattering and diffraction dissociation Measurements) experiment at CERN [1, 2] will measure the size of the proton and also monitor accurately the LHCs luminosity [3]. To do this TOTEM must be able to detect particles produced very close to the LHC beams.

TOTEM consists of “Roman Pot Stations” (RP), “Cathode Strip Chambers” (CSC) Telescope 1 (T1) and “Gas Electron Multipliers” (GEM) Telescope 2 (T2). The T1 and T2 detectors are located on each side of the CMS interaction point in the very forward region, but still within the CMS cavern. Two Roman Pot stations are located on each side of the interaction point at 220 m and 147 m inside the LHC tunnel. Each Roman Pot station consists of two groups of three Roman Pots separated by a few meters.

Such kind is in the learning phase that will produce elaborated requirements for the Control System [4] [5].

TOOL FOR THE CALCULATIONS

An specific tool has been developed for the calculation of the rate of information exchanged among all the hardware component using the method proposed in next Section. It uses the pinout information of the detector and some heuristics to build a Finite State Machine (FSM) of the detector [6]. Also assigns to each level different values according to the Product Breakdown Structure (PBS) tag for the calculation factors as seen in Table 1.

Table 1: PBS configuration entry for Temp. Sensors

Property	Value	Explanation
Id	E.03.05.03	The identifier in the PBS.
Description	E. M. - Temp.	A text description of the PBS identifier.
Information Chunk	4.00 B	The information size of the value transmitted in the readout.
Variation Prob.	1	The probability of changing the value between two readout intervals.
Archiving Freq.	300 s	This value is multiplied by the probability of change and the information size.
Archiving Node	4.70 GB	Is the information that this node has to archive only by itself.
Archiving Overhead	16.00 B	Represents the overhead in the structure to store the InformationChunk in a database.
Readout Rate Freq.	500 ms	This value is multiplied by the probability of change and the information size.
Readout Rate Node	37.50 Kb/s	Is the information that this node is generating only by itself.
Readout R. Overhead	8.00 B	Represents the overhead of the InformationChunk in the communication protocol.
Time Execute	100 ms	Time needed to execute a request.

This tool is highly modular, and the process of calculation has three steps:

1. Parse the tables and build the FSM tree.
2. Assign and match the PBS entries in the tree.
3. Execute correspondent algorithm for the calculations.

Each algorithm is contained in a independent class that is dynamically loaded. It explores the FSM tree and the PBS items and calculates certain tags. Also, an algorithm can generate new tags in the PBS item during its execution.

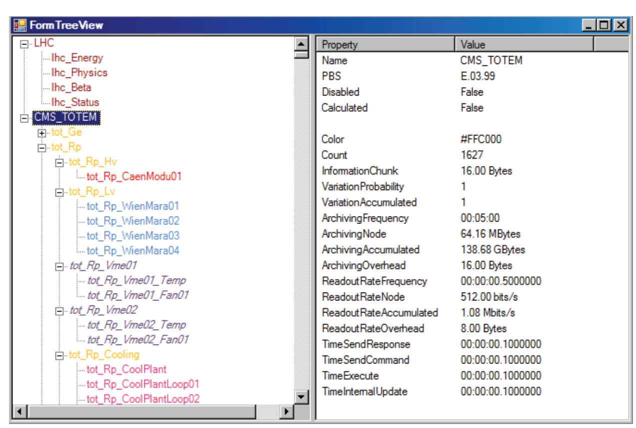

Figure 1: User interface of the tool

PLANS FOR MONITORING TPS CONTROL SYSTEM INFRASTRUCTURE USING SNMP AND EPICS

Y. T. Chang, Y. K. Chen, Y. S. Cheng, C. Y. Wu, C. H. Kuo, Jenny Chen, C. J. Wang, K. H. Hu, K. T. Hsu

National Synchrotron Radiation Research Center, Hsinchu 30076, Taiwan

Abstract

The Taiwan Photon Source (TPS) control system is one of the crucial systems for the accelerators and beamlines. It is necessary to monitor the status of the control system components such as housekeeping parameters of cPCI EPICS IOC crates, network traffic, connections between computers, etc. The equipment room environment including electric power, temperature, fire alarm, and water leak will also need to be watched. Using Simple Network Management Protocol (SNMP), the behaviour of network-attached devices can be monitored for administrative attention. Since the TPS control system is based upon the EPICS framework, the monitoring system is planned to adopt the EPICS support with SNMP. This paper will describe the system architecture of this monitoring system.

INTRODUCTION

Taiwan Photon Source (TPS) [1] will be the new 3 GeV synchrotron radiation facility to be built at National Synchrotron Radiation Research Center, featuring ultra-high photon brightness with extremely low emittance. The construction began in February 2010, and the commissioning is scheduled in 2013.

TPS control system will be implemented by using the Experimental Physics and Industrial Control System (EPICS) [2] framework. The various devices are integrated with EPICS based Input Output Controller (IOC) via control network connection. Figure 1 shows the architecture of TPS control system.

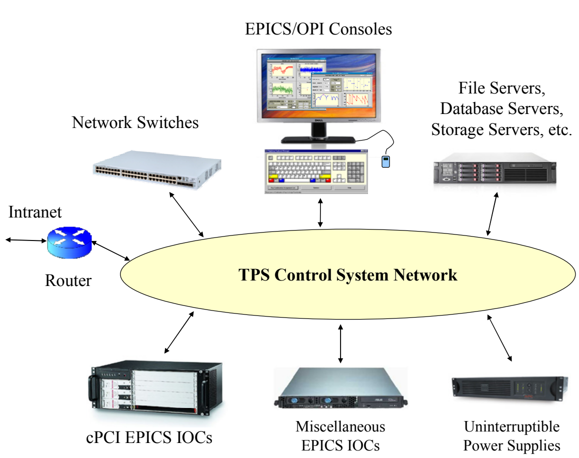

Figure 1: Possible SNMP-compatible devices in the TPS control system network

There are 24 Control Instrumentation Areas (CIA) which distributed along the inner zone just outside of the machine tunnel. Each CIA serves for one cell of the machine control and beamline interface. EPICS IOCs and major control devices connected to the control system are installed inside CIAs.

The TPS control system is designed for high availability. Its infrastructure must be reliable. Due to the long distance between control room and CIAs, an infrastructure monitoring system is planned to be implemented for gathering status of control system components such as CompactPCI (cPCI) IOC crates, network switches, servers, Uninterruptible Power Supplies (UPSs), etc. A dedicated EPICS IOC is planned to be used for housekeeping to monitor the health condition of these devices. When abnormal situations occur, e.g. crate temperature overheat, power supply breakdown, fan failure, or network disconnection, the monitoring system will automatically display the warning messages on the operator interface (OPI) screen and send out the alarm notification by voice call and E-mail. We can receive the early notification before a problem turns into a disaster. In addition to the warning messages, the monitoring system will also generate the warning reports or charts which can indicate the problems at the same time. Software tools such as MATLAB will be used to create these warning reports or charts automatically.

SYSTEM ARCHITECTURE

Simple Network Management Protocol (SNMP) is an industry standard protocol for managing statistical data of the network-attached devices. It is based on the client-server architecture and consists of three components: managed device, agent, and Network Management System (NMS). A managed device is a network node that implements an SNMP interface that allows access to specific information. An agent is a software module that resides on a managed device which reports information via SNMP to the NMS. The NMS is an application which runs on the manager and regularly polls data from agents.

SNMP mechanism associates with the Management Information Bases (MIBs) which describe the structure of the management data of a device. MIB uses a hierarchical namespace containing object identifiers (OID). Each OID identifies a variable that can be read or set via SNMP.

The devSNMP [3] is the EPICS device support with SNMP that allows us to access management data from any network device in the same manner as we are used to for the EPICS PVs. Current devSNMP supports only

DATA ACQUISITION AND STUDIES OF VIBRATION MOTION IN TLS BEAMLINES

P.C. Chiu, C.H. Kuo, K. H. Hu, Jenny Chen, Y. S. Cheng, Y. K. Chen, K.T. Hsu
NSRRC, Hsinchu 30076, Taiwan

Abstract

TPS (Taiwan Photon Source) is being under construction while TLS (Taiwan Light Source) is still on operation at the same NSRRC site. It was observed that the stability of photon beam intensity (Io) of TLS seemed a little deteriorated at daytime, when civil work is busy, compared to the nighttime. The intensity changes at different beamlines, however, aren't consistent with each other in each time, furthermore not so agreeing with the electron beam. Therefore, to correlate how the ground vibration due to civil construction effected on beam behaviour, the vibration measurement system is integrated into the existing TLS control system. The system will support waveform acquisition which could be acquired on demand. Meanwhile, realtime 10 Hz rms detector which could be archived continuously is also considered to be built in the future.

INTRODUCTION

The TPS is a 3 GeV energy electron ring with 512 meter circumference and planned to be delivered to users' end stations in 2014. During the periods of its constructions, the TLS at the same site will continuous be on operations. The quakes caused by excavators or pile drivers as Fig. 1 seem to have deteriorated the stability of beamline intensity (△Io/Io) from 0.1% up to 10% or more. On the other hand, these stability indicators △Io/Io between different beamlines have been not always concordant. Furthermore, it has been confused us over a long period that the indicators sometimes became worsen while the related subsystem remained normal even before TPS construction. It is suspected that different characteristics of vibration of different girders quite would be one of possible causes. Therefore, to clarify these inconsistent and not-yet-explained phenomena, the data acquisition system of vibration is planned to be built and continual expanded. In this report, the infrastructure of vibration data acquisition system will be presented as well as correlations of electron orbit, photon beam and vibrations of several spots will be shown.

INFRASTRUCTURE OF DATA ACQUISITION FOR VIBRATION

The DT8837 manufactured by Data Translation Inc. is employed as data acquisition tools for the accelerometers and photon intensity of beamlines distributed around the rings. The device supports functionality of bias current enable for ICP input. The equipped Ethernet interface is convenient for cabling and UDP trigger packet also provides sufficient synchronization mechanism for the distributed modules. Fig. 2 shows the infrastructure of the related system. All of the data from electron beam, photon beam, and vibrations could be synchronous acquired by software trigger within 100 msec. As Fig. 2 shown, besides the 10 Hz data from IOC/ILC could be acquired in real-time and archived, the fast transient motion could be also observed in adjustable higher time resolution and sampling rate up to 10 kHz.

Figure 1: TPS construction site in Sep 2010.

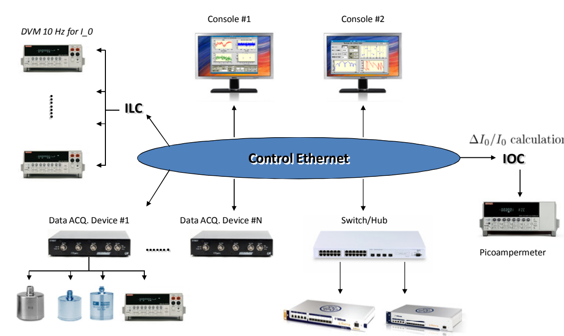

Figure 2: Infrastructure of data acquisition for vibration and the other related subsystem.

STATUS OF NORMAL OPERATION

Fig. 2 shows the normal status of the beam stability in quiet: the stability of beamline intensity (△Io/Io) is usually under 0.1%, the spectrum amplitude of electron beam stability is also less than 0.5 um below 50 Hz. The overall RMS stability of electron beam can achieve submicron level from DC to 50 Hz in normal operation [1][2]. The mechanic design of BL11 looks better than BL10's where the vibrations of three-axis at BL11 are all

COMPUTATIONAL STRATEGIES IN OPTIMIZING A REAL-TIME GRAD-SHAFRANOV PDE SOLVER USING HIGH-LEVEL GRAPHICAL PROGRAMMING AND COTS TECHNOLOGY

L. Giannone*, R. Fischer, K. Lackner and ASDEX Upgrade Team
Max-Planck Institut für Plasmaphysik, EURATOM-IPP Association, D-85748 Garching, Germany
P.J. McCarthy
Department of Physics, University College Cork, Cork, Ireland
Q. Ruan, A. Veeramani, M. Cerna, J. Nagle, M. Ravindran, D. Schmidt, A. Vrancic, L. Wenzel
National Instruments, Austin, TX 78759-3504, Texas, USA

Abstract

Big physics control experiments require enormous computational power to solve large problems with demanding real-time constraints. Sensors are acquired in real-time to feed mathematical routines, which then generate control outputs to real-world processes. For tokamak control, a non-linear PDE needs to be solved in real-time with a cycle time of less than 1 ms.

We report on an alternative approach based on LabVIEW that solves the critical plasma shape and position control problems in tokamaks. Input signals from magnetic probes and flux loops are the constraints for a non-linear Grad-Shafranov PDE solver to calculate the magnetic equilibrium. An architecture based on off-the-shelf multi-core hardware and graphical software is described with an emphasis on seamless deployment from development system to real-time target. A number of mathematical challenges were addressed and several generally applicable numerical and mathematical strategies were developed to achieve the timing goals. Several benchmarks illustrate what can be achieved with such an approach.

INTRODUCTION

The magnetic equilibrium for a tokamak is described by the Grad-Shafranov equation :

$$R\frac{\partial}{\partial R}\left(\frac{1}{R}\frac{\partial\psi}{\partial R}\right)+\frac{\partial^2\psi}{\partial Z^2}=-\mu_0 R\mathbf{j}(R,Z), \quad (1)$$

where ψ is the poloidal flux function, **j** is the current density, R is the radial component and Z is the axial component (see figure 1). This problem is commonly solved by a cyclic reduction algorithm [1, 2, 3]. A magnetic equilibrium for discharges with plasma current is reconstructed on a 33 x 65 grid using 40 magnetic probes and 18 flux loop difference signals. The right hand side current density term is calculated by a weighted least squares fit to the measurements which yields coefficients for the basis current density profiles [2, 3, 4]. Three basis current density profiles were chosen in the first round of development and found to adequately fit the experimental magnetic probe and flux loop measurements [5]. The currents from the poloidal field coils are also needed to compute the value of ψ on the spatial grid.

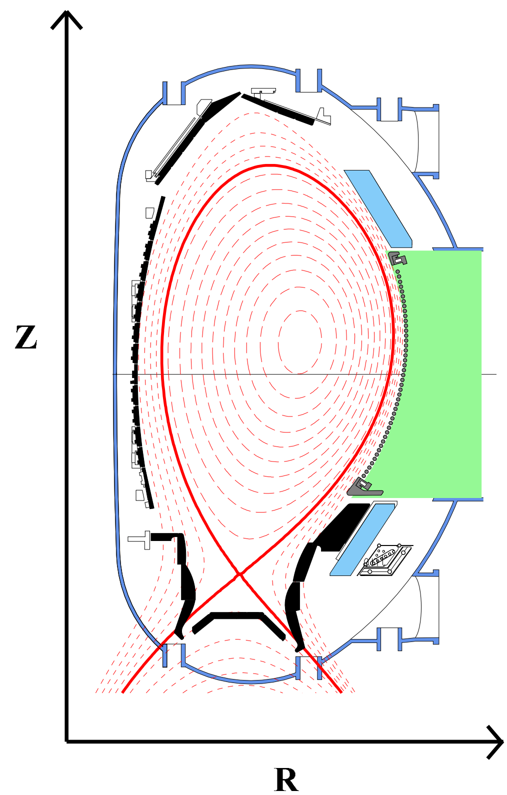

Figure 1: The cross section of the ASDEX Upgrade tokamak showing the flux surfaces of the magnetic equilibrium (red dotted lines) and plasma separatrix (red solid line).

REAL TIME GRAD-SHAFRANOV SOLVER

We report on a new spectral-based algorithm to solve the Grad-Shafranov equation in an unbounded domain. The new algorithm adapts a method commonly used to solve the Poisson equation in cylindrical coordinates. The use of discrete sine transforms (DST) along the Z-axis and a

* Louis.Giannone@ipp.mpg.de

ESS CONTROLS STRATEGY AND THE CONTROL BOX CONCEPT*

T. Satogata#, Jefferson Lab, Newport News, Virginia, U.S.A.
I. Verstovsek, K. Zagar, and J. Bobnar, Cosylab, Ljubljana, Slovenia
S. Peggs and C.G. Trahern, ESS, Lund, Sweden

Abstract

The European Spallation Source (ESS) will be constructed by a number of geographically dispersed partner institutions in an international collaboration [1]. This increases organizational risk, as control system integration will be performed by a large number of quasi-independent teams. Significant effort will be put into standardization of hardware, software, and development procedures early in the project. The ESS will use EPICS, and will build on the positive distributed development experiences of SNS [2] and ITER [3-5]. The basic unit of standardization is called the Control Box. This consists of one or more input/output controller (IOC) computers, zero or more I/O modules, PLC subsystems, and intelligent special-purpose controllers, and includes software and integrated development environment support. We present the challenges faced by Control Box plans for ESS, and expected benefits.

INTRODUCTION

Lund was chosen as the ESS site in May 2009. The Design Update phase (Jan 2011 to Dec 2012) will be completed with delivery of a Technical Design Report (TDR). ESS will deliver proton beam through a ~420m superconducting linac, and is expected to begin delivering beam to users in 2019. ESS will eventually deliver a nominal average proton current of ~50 mA at ~2.5 GeV in ~2 ms long pulses with a repetition rate of ~20 Hz to a single neutron target station, for a nominal average beam power of 5 MW.

There are several base assumptions for ESS control system planning:

- ESS will use the EPICS control system.
- ESS will use the Linux operating system in the controls service tier.
- ESS will use the Oracle relational database system as a project-wide RDBMS.

*Authored by Jefferson Science Associates, LLC under U.S. DOE Contract No. DE-AC05-06OR23177.
#satogata@jlab.org

After approval of the CDR in late 2012, the ESS project will proceed with R&D and construction, installation, and commissioning. ESS partner institutions doing development and R&D work over many geographical locations will be supplied with Control Boxes and given tools to enforce standards for common data management issues such as naming conventions, source code control, and controls development environment.

THE CONTROL BOX CONCEPT

The SNS project faced similar distributed controls and integration development challenges [2]. Several later projects, particularly ESS and ITER, are following the SNS distributed collaborative accelerator construction model and also require early broad controls coordination.

The Control Box concept is similar to the Plant System Host (PSH) concept used in ITER controls development [3]. In ITER terminology, the Control Box philosophy is realized with the concepts PSH, mini-CODAC [4], and Plant System I&C (instrumentation and control). The main purposes of the Control Box are to:

- allow independent and yet standardized subsystem controls development,
- enforce consistency between subsystems (possibly including target and experimental stations),
- facilitate testing of new components (e.g. EPICS drivers),
- allow centralized acceptance testing of subsystems through the control system,
- validate technology decisions,
- reduce risks early to lower projection integration uncertainty and effort,
- force early documentation of standards,
- and minimize throw-away hardware and software development.

An example structure of an ESS Control Box is shown in Fig. 1. The ITER Plant System I&C document [5] discusses the different available approaches to Control Box design.

CONTROL SYSTEMS FOR NEW LARGE EXPERIMENTS

J. Dedic, M. Plesko, R.Sabjan*, I. Verstovsek, K. Zagar, Cosylab, Ljubljana, Slovenia

Abstract

We discuss control systems of accelerators and similar projects that are presently still in design and early construction phases, such as FAIR [1], ESS [2], MedAustron [3], NSLS II [4], ITER [5], etc, and comparing them against the approaches of the last two decades and explain the new trends that are emerging:

- From the organizational perspective, control system architectures are established earlier in the project, allowing them to adapt to the machine physics requirements better as well as allow for modeling and simulations.
- In software, there is much less emphasis on custom codes than there was in the past. Instead, standard and off-the-shelf components and frameworks already used at existing accelerators are becoming the preferred choice, not only reducing risks, but also allowing for reuse and sharing.
- In hardware and networks for real-time control and data acquisition, there is a strong trend from custom electronics development to standard and off-the-shelf solutions. This in particular applies to systems like timing, machine protection, BPMs and LL RF. When custom solutions are needed, flexible hardware technologies (e.g., FPGA) are chosen to allow for future extensibility.

INTRODUCTION

Building a control system for a large experiment has always been a difficult task which required dedicated effort from a big group of people. And we have to thank controls groups in accelerator and the rest of big physics communities for their great achievements.

Control systems evolved in the recent decades, together with information technology, computer science and electrical engineering. In the starting days, little equipment, be it either software or hardware, was available off-the-shelf. A handful of physics labs with difficult requirements, for which solutions have never been implemented, were just not commercially interesting. This led to lots of custom work in the labs. From custom IO board development to advances in computer networking and developing whole software frameworks, nothing was taken for granted. Engineers were also scientists.

During the years, big number of experimental projects and the advance of computing allowed widespread standardization of components. Standard technologies are applied in every aspect of a modern control system, some systems can even be bought completely and some, which are only based on standard technology, but still require a lot of work before installed and commissioned. We shall look at some examples from the current experiments on which we collaborate.

At the end we shall try to summarize and find trends and consequences of progress. The main question is whether the everyday work of controls groups has changed and what does this mean for the main priorities that need to be set at the beginning of every project.

STANDARDIZATION IN LIGHT SOURCES

Plenty of light sources were built in the last decades and they have a lot common with respect to the control system. Control system packages (e.g. EPICS [6] or TANGO [7]) have matured through collaboration and can be easily deployed. They are supported on multiple standard hardware platforms (PC, VME, PXI etc.) and operating systems (Linux, Windows, Unix, Macintosh etc.). They provide solutions for most of your needs. Infrastructure applications like archiving, alarm handling or error logging are provided together with GUI builders and interfaces to many programming languages. Usually, even more than one implementation exists.

Increasing market has attracted industry as well. High performance electronics, made specifically for experiments' requirements is available off-the-shelf. Not only chips, but complete systems like digital BPM electronic [8] or timing systems [9] can be bought. Many equipment or subsystem vendors provide control system drivers with their products and they offer to implement them for the control system package of your choice.

Project leaders and funding agencies know this as well – control system budget has typically fallen from 10% to 5% of the machine's budget (not counting the building and beamlines). The challenge today is to implement a control system with state-of-the-art technology, but with a smaller budget and/or on a shorter time-frame, not sacrificing quality, of course. This prioritizes organizational aspects of the project which will be discussed in later sections.

PUSHING THE LIMITS OF CONTROL SYSTEM COMPONENTS

Other experiments (we have recently worked with ITER, FAIR, ESS and MedAustron) are still hiding more technical challenges and questions. Some examples are explained below.

Machine Protection System

One such example may be a complicated timing system or very flexible, but still safe machine protection system.

*rok.sabjan@cosylab.com

RTEMS WORKSHOP REPORT

G. Wright, CLS, Saskatoon, Saskatchewan

Abstract

This is the Workshop report as presented by Glen Wright (Canadian Light Source) on October 8, 2010 at the 2010 PCaPAC Conference held in Saskatoon, Sk, Canada

PRESENTATION ONLY

WHITERABBIT - A NOVEL, HIGH PRECISION TIMING SYSTEM

M. Kreider, R. Baer, T. Fleck, C. Prados (GSI, Darmstadt)
E. Garcia Cota, J. Serrano, T. Wlostowski (CERN, Geneva)

Abstract

The WhiteRabbit timing network is a deterministic field bus, based on synchronous GBit Ethernet and the Precision Time Protocol (PTP). The WR protocol was designed to provide precise timing and event distribution for high end real-time systems and was therefore chosen as the timing basis for the new GSI FAIR accelerator facility. With precise phase measurement to compensate for signal propagation delay, a timing accuracy down to sub-nanosecond range is feasible. To achieve necessary determinism and robustness (packet loss of 10^{-12}), an OSI layer two Forward Error Correction and Quality of Service protocol have been introduced to the concept. Special switches wield the WR protocol, while being transparent to normal Ethernet traffic. Switch hardware is currently under development at CERN and will be a mixed FPGA/CPU solution. Working prototype cards have been introduced at the 3rd WR Workshop at CERN in 2009, demonstrating phase measurement and PTP capabilities. The presentation will contain detail on technical concepts, current project status, as well as future areas of application will be part of the discussion.

INTRODUCTION

Purpose

WhiteRabbit was designed to provide very accurate clock synchronisation to a facility and control its machines with equal precision. Any event sent to a physical machine causes a certain action to be executed at a given absolute time.

The goal here is to the know the exact link delay to destination in advance, so each outgoing event can be sent out early enough to arrive on time.

In order to achieve that, certain unpredictable factors to the response time have to be addressed. One is packet loss due to data corruption on the physical medium, the other factor is collisions resulting from packet switching in the network.

NETWORK LAYOUT

WR utilises GigaBit Ethernet on fiber or copper links. Fiber links have an advantage here, because copper tranceivers and their channel encoding logic are more complex and often show a non-determinstic behavior. Optical links enable a higher measurement accuracy on link delay.

The topology of WR system may take any non-meshed form, since time synchronisation must be unidirectional. If the network is indeed meshed, a Spanning-Tree algorithm must be used to avoid loops in time distribution.

GSI/FAIR is planning to employ a Tree Topology with a GPS receiver as UTC timing reference at the source. Below come several layers of switches, fanning timing out to endpoints throughout the facility.

WR uses special switches and endpoints to wield its protocol. Current design of the WR switch has one uplink and sixteen downlinks, each has a second physical port for redundancy. GSI/FAIR is planning a system with roughly two thousand timing receivers

Making extensive use of commercially available ethernet basic components lowers costs for WR switches and endpoints. It will be possible to integrate non-White rabbit nodes into the network. WR is compatible with PTP devices and can time sync these nodes. However, PTP nodes can only be synchronized with reduced accuracy, since they lack the special hardware for high precision phase measurement.

General purpose ethernet nodes could also be connected to the network. While being compatible with basic functions, WhiteRabbit design does not support full Ethernet standard at the time.

TECHNOLOGY

Synchronous Ethernet - SyncE

SyncE describes the special case of IEEE 802.3 ethernet standard where the recovered RX clock from its master is used as its own TX clock, making the whole system synchronous. 8b/10b channel encoding is used to make RX clock recovery from the incoming RX data signal possible. This adjustment is done in hardware and is the basis for the PTP fine measurements.

Phase Measurement - Aliasing and DPLL

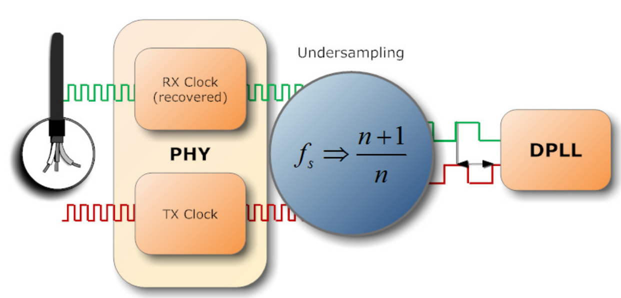

Figure 1: Aliasing and Phasemeasurement

After SyncE has adjusted the PTP clients frequency to the masters, the PTP can now measure the time difference

FLASH DAQ DATA MANAGEMENT AND ACCESS TOOLS

* V.Rybnikov, V.Kocharyan, K.Rehlich, E.Sombrowski, T.Wilksen

Abstract

The Free Electron Laser in Hamburg (FLASH)[1] at DESY is a user facility for the photon science community. It produces laser light of short wavelengths from the extreme ultraviolet down to soft X-rays. To study, monitor and document the machine performance and parameters and also to collect the results of the experiment measurements, a fast data acquisition (DAQ) system is being used. Having above 1000 linear accelerator diagnostics channels collected by the DAQ currently results in a data rate of ~100 Mb/s. The large amount of data requires corresponding data storage and management to enable efficient data retrieval. This paper will focus on the data paths, storage and bookkeeping. A number of tools provided for the users to work with DAQ data will be described. The current status of the achieved performance in the data storage and retrieval will be covered as well.

INTRODUCTION

The FLASH DAQ [2] system was launched in summer 2004. Its main tasks are: collecting LINAC beam relevant data in real time, providing the data to feed-back and monitoring tools as well as storing it for an offline analysis. The DAQ system is also used by FLASH user experiments to store their data together with information coming from LINAC. This allows easy correlations between the experiment measurements and the LINAC state. A set of tools is provided for data visualization and analysis.

DATAFLOW

The dataflow in the FLASH DAQ and all involved components are shown in Fig. 1. There are two types of data collected by the DAQ. Fast data include channels with beam related information (beam position monitors, etc.) and currently collected with the shot repetition rate of 10 Hz. All other channels considered as slow (magnet currents, etc) and collected with the maximum rate of 1 Hz. The data is collected by fast (FC) and slow collectors (SC) correspondingly via Ethernet. The collectors put data to the Buffer Manager (BM) [3] for online access. Distributors (DS) read data from the BM according to the stream descriptions provided by the Run Control during the DAQ configuration procedure. The data streams are pushed to the Event Builder (EVB) and further to the Writers (WR). The latter writes data to files on a local disk. The files from the local disk are copied to a huge RAID array and accessible via NFS for the public. The experiments data is usually copied to tape for the permanent storage.

*Deutsches Elektronen-Synchrotron, DESY, Hamburg, Germany

DATA MANAGEMENT

The DAQ data management components control the data flow and guarantee all required data is written to data files and to the tape if required. The components keep track of the written data in order to assure fast data access. The rest of the paper will be devoted to the description of those components.

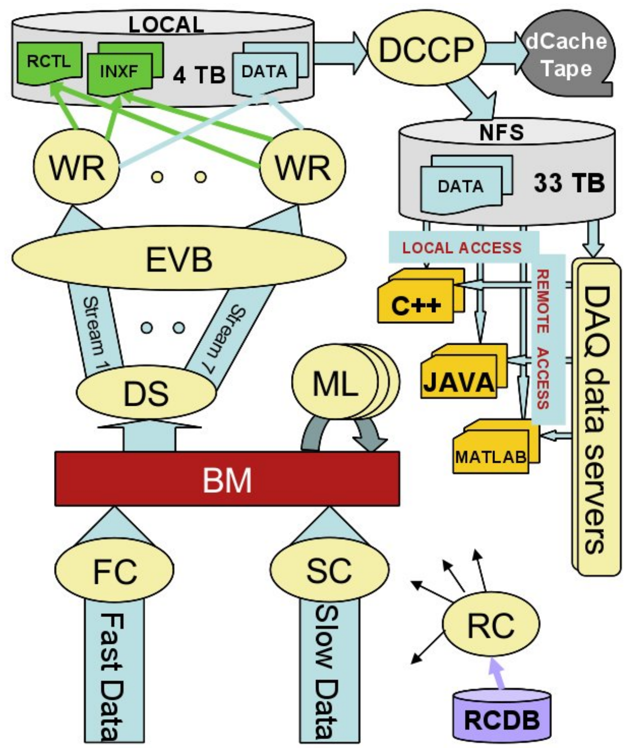

Figure 1: FLASH DAQ dataflow and access tools

Fast Channel Data

The fast channels (~ 700 channels) provide the most part of the data volume (~99.5%). It means that the reduction of the total data amount strongly depends on the configuration of the front-end DAQ senders. The front-end configuration is performed by the Run Control process during the DAQ configuration. The RC is capable to set every parameter for the spectra that are usually sent by the front-end (e.g. start, increment, length). For that the RC has a group of run parameters when changing one of them changes a spectrum parameter in a group of channels (e.g. the same device types). The different sets of run parameters are stored in the Run Modes of the Run Control data base [4]. Every Run Mode corresponds to a

BEAM PROFILE MONITORING SYSTEM FOR XFEL/SPRING-8

T. Matsumoto#, A. Yamashita, JASRI/SPring-8, 1-1-1, Kouto, Sayo, Hyogo 679-5198, Japan
S. Inoue, SPring-8 Service Co, Ltd., 2-23-1, Koto, Kamigori, Ako, Hyogo 678-1205, Japan
Y. Otake, RIKEN/SPring-8, 1-1-1, Kouto, Sayo, Hyogo 679-5148, Japan

Abstract

A beam profile monitoring system was developed for XFEL/SPring-8. In this paper, we focus on an image processing system. The image data can be recorded with the synchronized data acquisition system of XFEL/SPring-8. The system is composed of 46 screen monitors (SCMs) and the transverse size and shape of the electron beam are measured down to a resolution of 10 µm. The SCMs provide a valuable tool for beam commissioning in terms of optimization of beam transport and measurement of beam emittance. The imaging system uses CCD cameras that are connected by Camera Link. An image data is selected using the Camera Link selectors and is then processed by an image server. A diagnostic tool for the beam profile monitoring system requires many functions: real-time image monitoring, image analysis, camera control, screen control, etc. We developed a GUI (Graphical User Interface) using Python as a tool to flexibly implement the functions required for the image data. The system was successfully implemented on the SCSS prototype accelerator and it operated as intended. The system can thus be applied to the beam commissioning of XFEL/SPring-8, which is planned for March 2011.

INTRODUCTION

The Japanese X-ray free electron laser (XFEL/SPring-8) is under construction at the SPring-8 site, and its beam commissioning will begin in March 2011 [1]. XFEL/SPring-8 will generate an X-ray laser with a wavelength that is less than 0.1 nm via the SASE (Self-Amplified Spontaneous Emission) process. To achieve this goal, high-precision beam characteristics (a low emittance electron beam less than 1 π mm mrad, etc.) are required, and various types of beam diagnostic tools must be positioned at each stage of the accelerator [2]. In total, 57 RF cavity beam position monitors (RF-BPMs), 49 screen monitors (SCMs) for beam profile measurement, and 35 current transformer (CTs) for beam charge measurement will be installed.

In this paper, we describe an image processing system equipped with 46 SCMs that is used for transverse beam profile measurement with an accuracy of about 10 µm. The remaining 3 SCMs are used for longitudinal beam profile measurement, which will not be addressed here.

The beam profile monitoring system plays the important role of tuning the beam during the beam commissioning. The system is used for optimization of the beam transport and the measurement of beam parameters (emittance, twiss parameters, etc.). In order for the beam commissioning of XFEL/SPring-8 to proceed smoothly, a prototype of the system has been developed and was implemented in the SCSS prototype accelerator to confirm its performance.

SCREEN MONITOR (SCM)

The configuration of an SCM is shown in Figure 1. The SCM system is composed of a screen, a screen actuator, an optical system, and a data acquisition system with a CCD camera. Since the beam is destructed by the screen, the screen actuator moves the screen outside of the beam orbit when it is not used. The material of the screen components was selected depending on the beam energy. For higher energy (>30 MeV), metal foil was used for the optical transition radiation (OTR) while for lower energy (<300 MeV), Ce:YAG was used for fluorescence. In order to achieve a high position resolution, the optical system is equipped with a custom-made lens. The zoom range can be adjusted through the operation of a motor. The position resolution is about 3 µm at a magnification of four times and satisfies a required resolution (10 µm). For equipment controls such as stepper motor controller of the zoom adjustment, Programmable Logic Controllers (PLCs) are used [3]. Two types of CCD cameras are used: a JAI CV-A10 CL (monochrome, 0.46 M pixel, 60 fps) and a JAI CV- M4+CL (monochrome, 1.45 M pixel, 24 fps). For each SCM, the proper CCD camera was selected according to its needs.

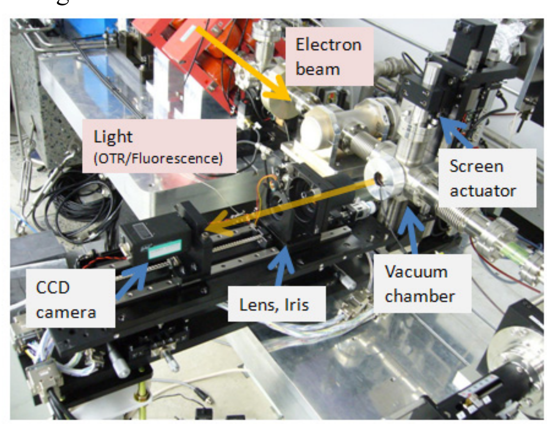

Figure 1: Screen monitor for XFEL/SPring-8.

IMAGE PROCESSING SYSTEM FOR SCM

An overview of the image processing system for SCM is shown in Figure 2. Communication with CCD cameras

#matumot@spring8.or.jp

EMBEDDED CONTROLLER FOR INDUSTRIAL CT TRIGGER MODULE

G. Gong, T.Xue, J.Li, Dept. of Engineering Physics, Tsinghua University, Beijing, China, 100084

Abstract

The industrial CT is used to generate a 3D image of the inside of an object; it consists of an accelerator x-ray source, detector array, readout electronics and control system. A trigger module collects the position information from three decoders installed all the 3 moving axis and generates trigger signal to the x-ray source and readout electronics. The trigger module is remotely accessed by the SCS (system control station) via a fast Ethernet connection. The trigger module utilizes an embedded controller board which consists of a PowerPC controller running the Linux operation system, and a FPGA connected to the PowerPC local bus as a customized peripheral to carry out the trigger logic. With different interface mezzanines and online firmware upgrade, the trigger module has great flexibility to work with different decoders readout electronics.

INTRODUCTION

Originally developed as a medical diagnostic tool, the Computed Tomography (CT) can provide detailed internal information of human body. This technology has also been applied to non-destructive inspect objects that have the indispensable requirement for safety and reliability like high-speed railway train wheels or the air plane turbine engineers. Without the constraints of patient movements or dose restrictions that exist in the medical CT, the industrial CT can achieve better resolution by applying much stronger x-ray source and a much longer exposal time [1].

The industrial CT consists of an x-rays tube, a rotary table, the detector array, the readout electronics, the trigger module and the data analyse and image reconstruction computer.

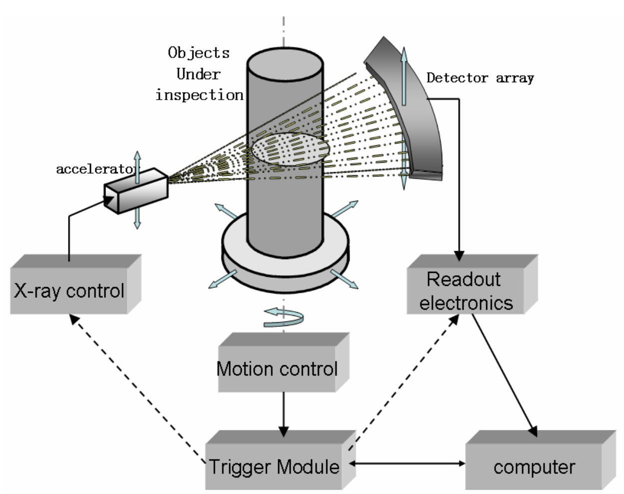

Figure 1: typical schematic block diagram of industrial CT.

A typical schematic block diagram of industrial CT is given in figure 1.

The object to be inspected is located on the rotary table between the x-ray tube and the detector; the X-ray source and the detector are relatively stationary and can move together in the vertical direction along the object; the rotary table can move in two directions and rotate. By setting the relative motion between object and the x-ray/detectors, the industrial CT can be configured to work in direct radiography (DR) mode, second generation CT mode or third-generation CT mode. The x-ray source is working in pulse mode to reduce radiation dose and prolong the life, the readout electronics are also work in gated integration mode to suppress the detector dark noise.

TRIGGER MODULE STRUCTURE

As seen from Figure 1, the trigger module is one of the key components in the industrial CT; it connects with all the other control blocks and manipulates the working flow of the equipment. The functional block structure of the trigger module is shown in figure 2; a picture of the module is given in figure 3.

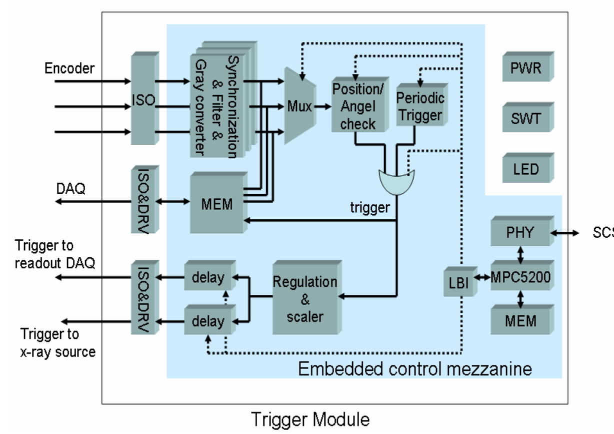

Figure 2: Functional block of trigger module.

Its main function blocks are provided by the embedded control mezzanine which will be described later. A description of all the other elements is given below:

Encoder input

In each axle of the drive motor, there is an absolute rotary encoder installed. They have 16 resolution bits and 8 turns bits to cover the whole scan range. The position and angle the object under inspection can be achieved from the output of these encoders, which are normally in gray code that has only one bit difference between any two consecutive values.

The output of those encoders are converted into the local electrical domain by isolation transistors, a 3 out of 3 low-pass filter removes the noise glitch signals that couple into the cable, then the original Gray code are

ITER CONTROL SYSTEM DEVELOPMENT ENVIRONMENT

K. Zagar, Cosylab, Ljubljana;
F. Di Maio, A. Wallander, ITER, St Paul lez Durance

Abstract

ITER is a large tokamak fusion facility whose construction is organized in such a way that the plant systems it consists of will be delivered in-kind by the seven participating countries. Integration of the plant systems is thus expected to be particularly challenging. To make integration of instrumentation and controls of plant systems as smooth as possible, the ITER Control, Data Access and Communication (CODAC) team prepared a Plant Control Design Handbook, which specifies in detail the interfaces between the plant systems and the central system. To facilitate compliance with the handbook, a development environment has been prepared for plant system controls developers, which standardizes the operating system, control system infrastructure, development tools and development processes. In this paper, we describe the ITER CODAC development environment: what open-source software was selected, how it is packaged for ITER CODAC, what infrastructure for development is in place, what challenges were encountered and what the roadmap is for the foreseeable future.

PRESENTATION ONLY

DATABASE-DRIVEN STATUS ANALYSIS IN BEAM OPERATION AT THE HEIDELBERG ION THERAPY CENTER

K. Höppner*, R. Cee, M. Galonska, T. Haberer, J. M. Mosthaf, A. Peters, S. Scheloske
Heidelberg Ionenstrahl-Therapie Centrum (HIT)
HIT Betriebs GmbH am Universitätsklinikum Heidelberg, Germany

Abstract

The HIT (Heidelberg Ion Therapy) center is the first dedicated European accelerator facility for cancer therapy using both carbon ions and protons, located at the university hospital in Heidelberg. It provides three treatment rooms, two with fixed beam exit (operational since Nov. 2009 and Sept. 2010, respectively), and the first gantry worldwide where the beam exit can be rotated by 360 degrees, currently under commissioning.

HIT uses a PC-based proprietary software system for accelerator controls with an Oracle database for storing device parameters, beam history, error logging etc. Since medical treatment of humans requires a high level of quality assurance, a detailed analysis of beam quality and error logs is needed. We wrote a series of database applications using Python to perform these tasks automatically and create daily reports on beam statistics and parameters, machine status and errors occurred. Additionally, some graphical applications on top of the commercial control system help the scientists and operators in the beam commissioning of the new therapy treatment rooms and the gantry. We will present these applications and show how they are used at HIT.

INTRODUCTION

The HIT accelerator setup as shown in Fig. 1 consists of

- two ion sources, currently used for producing carbon and proton ions (a third ion source is to be installed soon)
- a linac accelerating the ions to 7 MeV/u,
- a synchrotron used to accelerate the ions to their final energy as defined by the patient-specific treatment plan, and
- four high energy beam transport lines providing the beam to the horizontal treatment rooms, the rotatable gantry or the additional station dedicated to quality assurance (QA), research and development.

Cancer treatment with different ion types and of different patients in parallel requires a multiplexed beam operation with the possibility to switch the ion source and beam destination from pulse to pulse, every source/destination combination identified by a virtual accelerator number. Beam parameters can be chosen from a matrix of 255 energy values, up to 6 focus sizes and up to 15 different intensity values, denoted as MEFI parameters. For commissioning and quality assurance, the beam is requested by the Accelerator Control System (ACS) directly, while in therapy mode the Therapy Control System (TCS) requests the beam characteristics determined by the treatment plan via a communication interface to the ACS.

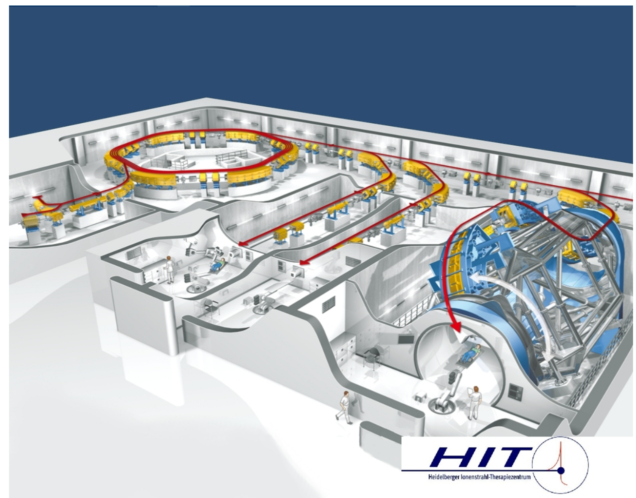

Figure 1: Overview of the HIT accelerator complex. (The QA station is not shown.)

HIT uses an accelerator control system built by a German company for automation and process control hard- and software [1]. It runs on Windows servers, using Oracle 9i as database backend. An upgrade to Oracle 11g is planned for the end of this year. While changes to the GUIs of the control system require invention by the supplier, we can easily access the tablespace used for accelerator controls in the Oracle database. Thus, we were able to develop database applications for various tasks in beam analysis and machine commissioning.

APPLICATION ENVIRONMENT

For a fast development of database applications that may easily be deployed on different systems (Windows for client PCs in the accelerator control room, Linux for a server that is mainly used for the web based electronic logbook [2]), we decided for Python 2.6 [3] as a cross platform OO scripting language. Python provides a variety of builtin and 3rd party modules, including an Oracle module [4] compliant to the Python Database API 2.0. Table 1

* klaus.hoeppner@med.uni-heidelberg.de

QUARK: A DYNAMIC SDLC METHODOLOGY*

V. Vuppala, J. Vincent, NSCL, East Lansing, MI 48824, USA. #

Abstract

No single Software Development Life-cycle (SDLC) methodology works well for all types of software projects. The project may require a methodology that can be very predictive to very adaptive based on characteristics such as requirements volatility, requirements clarity, project criticality, complexity, and size. We describe a new iterative approach that can vary from being more adaptive to being more predictive during its iterations. The project characteristics change with iterations, and the SDLC adjusts accordingly by changing its parameters. We also discuss the results of using this methodology for projects at National Superconducting Cyclotron Laboratory (NSCL).

INTRODUCTION

Last few decades have seen an evolution of SDLC models to address the software-crisis. Some of these are Waterfall, Spiral, V-Process, RUP, and Agile among others. Each model has its advantages and drawbacks, and not all of them work for all types of software projects [1]. Some of them are predictable in terms of cost and schedule but rigid in terms of requirements, whereas others are adaptive to changes but less predictive.

In our organization there was a need to implement processes to instil engineering rigor into software development. The following were the requirements for the process model:

- Provide transparency and predictability
- Work with limited customer availability
- Not overly bureaucratic, low overhead
- Support project management
- Support critical and non-critical systems

We evaluated various models but found them to be inadequate for our needs. Many organizations, especially in the software industry, choose from a set of SDLC models based on the project characteristics. This was not an option for us, as it required the project team to be proficient in multiple software development methodologies. As a result, we developed a set of processes for software development and project management, which resulted in the Quark Model (QM). It is based on CMMI-Dev 1.2, PMBOK 4, and ISO 9000-3 standards.

Iterations

QM uses an iterative approach to software development. QM iterations are parameterized, and governed by the following parameters (QMPs):

- Duration: The duration, in terms of calendar time, of the iteration
- Change Control: Specification of Major and Minor scope changes
- Documentation: The detail and amount of documentation
- Communication: Meeting intervals and duration within project team, and with Customer
- Planning: Level of detail in planning
- Quality Controls: Frequency of Design and Code reviews, and test methodology.

By adjusting the QMPs, for each iteration, the process can be adjusted from being more adaptive to being more predictive, and anywhere in-between.

Projects

Projects are central to the QM model. A software project is a temporary endeavour undertaken to create a unique software product [2]. It is characterized by certain attributed (PCTs). Some of the PCTs that vary during the execution of a project are:

- Project Team Requirement Clarity: Project team's understanding of the requirements
- Customer Requirement Clarity: Customer's understanding of the requirements
- Size: Size of the project in terms of cost, code base, team size, etc
- Estimate Confidence Level: Accuracy of cost and schedule estimates
- Technology Expertise: Familiarity with the solution technology

Some of the PCTs remain relatively constant during the course of the project, such as criticality of the project, safety and security requirements, quality requirements, timeline constraints, customer Availability, bespoke or custom software, contract type, and team location.

QUARK MODEL

Figure 1 illustrates the Quark Process Model. The PCTs

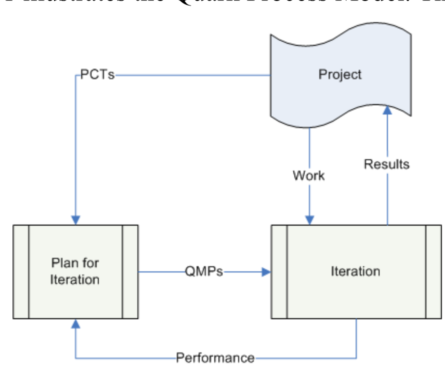

Figure 1: Quark Process Model.

Email Addresses: {vuppala,vincent}@nscl.msu.edu
* This work is funded by National Science Foundation and Michigan State University

EXPERIMENT BASED USER SOFTWARE

D.K. Chevrier, Canadian Light Source Inc., Saskatoon, Canada
M. Boots, Department of Physics and Engineering Physics,
University of Saskatchewan, Saskatoon, Canada

Abstract

The Spherical Grating Monochromator (SGM) and Resonant Elastic-Inelastic Xray Scattering (REIXS) beamlines are located at the Canadian Light Source (CLS). A novel approach to software design has been undertaken to simplify user interactions with these beamlines. While the SGM and REIXS beamlines are structurally different, the techniques available are quite similar. The software is developed to provide seamless acquisition of data, strong data management tools, and easy transition between beamlines for end users. The end result is software focussed on experiments rather than software focussed on beamlines.

INTRODUCTION

One reality of modern science is that 90% of "conducting an experiment" involves sitting at a computer and interacting with software. Traditionally, the CLS has found the resources to develop *beamline software* for each new beamline. In principle, this is a good thing. However, as the facility grows and matures there is a sense that the software used at the beamlines needs to evolve as well. As the vision of the CLS – "[t]o be a global leader and a recognized centre of excellence in synchrotron science and its applications" [1] – makes clear, the purpose of the facility is to support science. As such, evolving our software from *beamline software* to *experiment software* seems like a way to better support science. It is important to note that having *beamline software* is a natural part of the software progression. When a beamline is under development and commissioning, the essential first requirement for software is to provide direct and detailed control over all the separate components that make up the beamline. The importance of this existing software should not be questioned: there would be no way do any science, nor to evolve user software to the next level, had this critical work not been done.

With this background in mind, there are clear ways to address long-standing user issues and improve the experience and efficiency of conducting research at the CLS. The evolution from beamline-centered software to experiment-centered software is accompanied by an evolution from engineering software to designing a *user experience*. That is, there is a shift from the relatively straight-forward task of stating that "software requires the ability to do *functions* A, B, and C using *widgets* X, Y, and Z" to a more holistic need for software that "makes it *quick* and *intuitive* for users to do *tasks* A and B". Because of this change from concrete to descriptive requirements, there are competing types of requirements to keep in mind. In principle the requirements of functionality, appearance, and connectivity will compete with each other as each component is designed and developed. Thus, every component within the software needs to work properly, look appeasing to the user, and be able to connect with other related tasks the user wishes to do.

In addition to a discussion about the concepts and ideas of making an experiment centered software package for users, some time must be devoted to exploring how this can be best achieved from a programming standpoint. While important, the examination of the programming principles will take a backseat to the fundamental vision.

From the inception of this project, we sought to cast as wide a net as possible to determine what users needed out of experiment based software. A summer student was given the task of shadowing users on a number of different beamlines looking for features that were exceptional, tasks that could be simplified, and common irritations that users experienced. Additionally, a workshop was conducted at the CLS Annual Users' Meeting to act as a focus group for new software concepts. A number of outstanding ideas were generated and have been incorporated into the current design.

USER CONCEPTS

Would it not be wonderful if users could sit down and just start doing experiments when they first get to the beamline? Could it be made so software would help users with their experiments – giving them guidance when needed and remaining unobtrusive when not? Would it be so bad if users only needed one software tool from the beginning of their experiment until the end? The vision of experiment based user software is to offer all of these opportunities to users, regardless of their experience level or background, in a way to allows them to concentrate on the science they know. At the same time, the user experience needs to be as pleasant and efficient as possible. The question we must pose is whether it is possible to achieve this and, if it is, how best can that be done? Presuming it is possible, the software evolves from controlling individual acquisitions and beamline actions to managing the acquisition, the data, the beamline, and the experimental process as a whole.

Acquisition Management

Currently, many users experience a steep learning curve when they arrive at the CLS, the steepest part of which is becoming familiar with the unique controls of the beamline they are working on. A common experience might be that of an expert user doing simple x-ray absorption spectroscopy (XAS) at the SGM beamline. Although this user likely knows as much as, if not more than, the beamline staff about the scientific technique itself they are still forced to learn how to conduct XAS on the SGM beamline – which controls to set, how to setup a scan, which detectors to look at, and so forth. Any time a

DATA ACQUISITION FROM HETEROGENEOUS SENSOR NETWORKS: THE CASE OF NEPTUNE CANADA THE WORLD'S LARGEST CABLED OCEAN OBSERVATORY

B. Pirenne, Ocean Networks Canada, University of Victoria, BC, Canada

Abstract

Ocean Sciences is at the crossroads: it is entering the brave new world of "Big Science". The first of a new generation of large facilities, the NEPTUNE Canada cabled ocean observatory (www.neptunecanada.ca) will be presented from the point of view of a sensor network composed of hundreds of diverse instruments. The challenges we faced will be reviewed, together with the selected network design, data management and data distribution approaches. Special emphasis will be placed on the architecture of the system and on the more recent developments and concepts used to help scientists in their exploitation of the data. Finally a number of the early discoveries made with the new facility will be briefly described.

CABLED OCEAN OBSERVATORIES

Cabled ocean observatories are remote observing systems that provide power and communication media to a host of underwater instruments and sensors. Consequently, the instruments are (almost) always on-line and sufficient power is provided to the assets to ensure uninterrupted data flow covering multiple environmental parameters at high resolution in a four dimensional space. Observatory systems considered here also provide a significant ability to remotely manage their assets (ie, provide a real-time command ability for specific instruments). As an example, NEPTUNE Canada is composed of a fully redundant 800-km cable loop and has the ability to provide 9kW of power at up to 10 different locations of scientific interest. Figure 1 shows the layout of the NEPTUNE Canada observatory as well as its currently defined 6 main locations, five of which are instrumented. They reside at depth between 20 and 2700 meters.

Each of the locations is equipped with a "node" that reduces the line voltage of 10 kVDC down to 400 VDC and offer data connection points for up to 4 Gbps. In a area covering up to a few km^2, extension cables can be run from the nodes to sites of interest, where platforms with actual instruments and sensors are installed. The platforms are typically composed of a "junction box" whose role is to be the local "power bar", providing plugs for instrument power and communication, converting the 400 V input to 15, 24 or 48 Volts and translating the instrument serial protocol to IP where necessary.

The instrumentation measures physical and chemical parameters of the ocean (temperature, salinity, oxygen content, CO_2, currents speed and direction at different depths, ...), but also has a number of more specific devices such as underwater video cameras, electro-magnetic experiments, vertical profilers that move through the water column, small vehicles on track (crawler), ... all of which would not be possible without the availability of ample power and the ability to command them in real-time. Figure 2 illustrates the crawler, itself a device equipped with various chemical and physical sensors, cameras, etc.

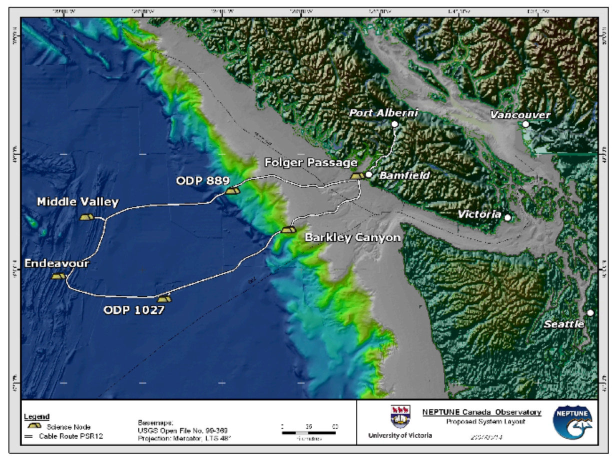

Figure 1: Map of the area covered by NEPTUNE Canada west of Vancouver Island. Please note the 800 km cable loop and the various location of scientific interest, and their "node".

The entire system represent the extension of the Internet under the Ocean, which was the vision put forward by the proponents of such a system many years ago.

Figure 2: A small tethered vehicle on track. It can roam within 50 m from its central position. It is equipped with various physical and chemical sensors and a camera.

FRIDAY CLOSEOUT PRESENTATION

E. Matias, CLS, Saskatoon, Saskatchewan

Abstract

Closeout Presentation by Elder Matias, Chair of the 2010 PCAPAC workshop

PRESENTATION ONLY

PCAPAC 2012 ANNOUNCEMENT

C. H. Wang, IHEP Beijing, Beijing

Abstract

This is the power point presentation introducing the venue for PCaPAC 2012

PRESENTATION ONLY