DRIVERS AND SOFTWARE FOR MicroTCA.4*

M. Killenberg†, L. Petrosyan, C. Schmidt,
Deutsches Elektronen-Synchrotron DESY, 22607 Hamburg, Germany
S. Marsching, aquenos GmbH, 76532 Baden-Baden, Germany
M. Mehle, T. Sušnik, K. Žagar, Cosylab d.d., SI-1000 Ljubljana, Slovenia
A. Piotrowski, FastLogic Sp. z o.o., 90-441 Łódź, Poland
T. Kozak, P. Prędki, J. Wychowaniak, Łódź University of Technology, 90-924 Łódź, Poland

Abstract

The MicroTCA.4 crate standard provides a powerful electronic platform for digital and analogue signal processing. Besides excellent hardware modularity, it is the software reliability and flexibility as well as the easy integration into existing software infrastructures that will drive the widespread adoption of the new standard. The DESY MicroTCA.4 User Tool Kit (MTCA4U) comprises three main components: A Linux device driver, a C++ API for accessing the MicroTCA.4 devices and a control system interface layer. The main focus of the tool kit is flexibility to enable fast development. The universal, expandable PCIexpress driver and a register mapping library allow out of the box operation of all MicroTCA.4 devices which carry firmware developed with the DESY FPGA board support package. The control system adapter provides callback functions to decouple the application code from the middleware layer. Like this the same business logic can be used at different facilities without further modification.

INTRODUCTION

The MicroTCA.4 crate standard [1, 2] provides a platform for digital and analogue data processing in one crate. It is geared towards data acquisition and control applications, providing a backplane with high-speed point to point serial links, a common high-speed data bus (PCIexpress in this case) as well as clock and trigger lines. In typical control applications large amounts of data have to be digitised and processed in real-time on the front end CPU of the MicroTCA.4 crate.

MTCA4U—The DESY MicroTCA.4 User Tool Kit

The main goal of the DESY MicroTCA.4 User Tool Kit (MTCA4U) [3] is to provide a library which allows efficient, yet easy to use access to the MicroTCA.4 hardware in C++. In addition it features an adapter layer to facilitate interfacing to control system and middleware software. The design layout of the tool kit is depicted in Fig. 1.

THE LINUX KERNEL MODULE

The Linux kernel module (driver) provides access to the MicroTCA.4 devices via the PCIexpress bus. As the basic access to the PCIexpress address space is not device dependent, we follow the concept of a universal driver for all MicroTCA.4 boards. The kernel module uses the Linux Device Driver Model which allows module stacking, so that the driver can be split into two layers: A generic part provides all common structures and implements access to the PCIexpress I/O address space. The device specific part implements only firmware-dependent features like Direct Memory Access (DMA), and uses all basic functionality of the generic part. For all devices developed at DESY the firmware will provide a standard register set and the same DMA mechanism, which permits to use a common driver for all boards. For devices from other vendors the generic part enables out-of-the-box access to the basic features, which can be complemented by writing a driver module based on the generic driver part. Like this the interface in MTCA4U does not change and the new device is easy to integrate into existing software.

Improved DMA Performance

Latest developments have focused on the improvement of the DMA performance. For a universal driver it is important to keep the interfaces to the user space and the firmware simple. In addition, the implementation in firmware should not be too complicated. The available firmware uses a simple DMA core which allows to transfer one contiguous block of memory from the MicroTCA.4 board into one contiguous block of the memory of the CPU. After the transfer is finished, the CPU is notified via a PCIexpress interrupt. Memory allocated in user space is not contiguous in Linux, but memory allocated in the Kernel address space can be contiguous.

Our original DMA implementation was to allocate one big kernel buffer, perform the DMA transfer into this buffer and afterwards perform a copy to the user space (see Fig. 2 A). Performance measurements showed that the copy to user space took about 50 % of the time of the DMA transfer (PCIexpress Gen. 1 on an Intel Core i7 CPU), which was a significant performance penalty.

The improved version tries to minimise the time until the copy to the user space can start (copy latency) and already executes part of the copying while the rest of the data is still being transferred via DMA. For this it performs multiple DMA transfers of a smaller block size instead of transferring everything at once (B and C in Fig. 2). As every DMA transfer ends with an interrupt, this causes additional load on the system and an additional latency. For this reasons the number of DMA transfers should be limited. Version B in Fig. 2 for instance finishes later than version C, although the

* This work is supported by the Helmholtz Validation Fund HVF-0016 “MTCA.4 for Industry”.
† martin.killenberg@desy.de

ISBN 978-3-95450-146-5

CONTROLS MIDDLEWARE FOR FAIR

V. Rapp, GSI, Darmstadt, Germany
W. Sliwinski, CERN, Geneva, Switzerland

Abstract

With the FAIR complex, the control systems at GSI will face new scalability challenges due to significant amount of new hardware coming with the new facility. Although, the old systems have proven themselves as sustainable and reliable, they are based on technologies, which have become obsolete years ago. During the FAIR construction time and the associated shutdown GSI will replace multiple components of the control system. The success in the integration of CERNs FESA and LSA frameworks had moved GSI to extend the cooperation with the controls middleware and especially Remote Device Access (RDA) and Java API for Parameter Control (JAPC) frameworks. However, the current version of RDA is based on CORBA technology, which itself, can be considered obsolete. Consequently, it will be replaced by a newer version (RDA3), which will be based on ZeroMQ, and will offer a new improved API based on the experience from previous usage. The collaboration between GSI and CERN shows that new RDA is capable to comply with requirements of both environments. In this paper we present general architecture of the new RDA and depict its integration in the GSI control system.

INTRODUCTION

Officially started in 2010 the FAIR (Facility for Antiproton and Ion research) [1] project will extend the existing GSI accelerator by many new installations. Those will introduce a plenty of new hardware, which need to be managed and monitored by the control system.

To benefit from the existing experience GSI has decided to start collaborating with CERN on the device software framework called FESA. This collaboration was successful, so the future devices, which will be introduced within the scope of FAIR will be developed with FESA framework. The FESA framework itself is designed to operate within the CERNs system context and depends on the underlying CERN middleware and the messaging layer called Remote Device Access (RDA). The adaptation of the FESA framework to the current GSI middleware is difficult. Moreover the current version of the GSI control system depends on CORBA [2] which itself is a rather an old messaging library with lack of support and shrinking community. Additionally this year CERN released a new RDA version based on the ZeroMQ [3] messaging framework, which is light, modern and well suited to operate in accelerator environment [4]. Therefore the further devices developed for the FAIR project will utilize the new middleware solution.

On the other hand GSI itself is running a considerable amount of old hardware. This is done by using of the own developed middleware messaging layer called Device Access. Upgrading of all old hardware will be costly and difficult to accomplish. Hence the middleware architecture of FAIR will consist of two different parts: the CORBA based Device Access and the new ZeroMQ based RDA.

DEVICE ACCESS

Today's control system at GSI can be roughly divided into 3 parts. First the real time processes run on so called Front End Computers (FEC), which control the equipment and collect the information from it. On the other point of the architecture the mostly Java based Graphical User Interface (GUI) applications, which present the gathered information and allow operators in the control center to set hardware parameters of the beam equipment. Additionally the information form FECs is also monitored and collected by different automated services like logging or monitoring. Those two parts are connected by a middleware called Device Access.

Each process running on FEC is modelling a particular device of the accelerator equipment which is physically connected to FEC and managed by it. A typical device model is an object oriented device presentation, which consist of multiple readable and writable properties (e.g. device status, voltage, measured pressure etc.). Each model can also offer particular methods to control the connected device such as "reset" or "initialize", which can be called from other tiers using the middleware. To access the properties from another process like GUI, Device Access provides basic get, set and method call mechanisms, but also allows subscription for particular properties. Device Access also supports a given context for property values, which is used in multiplexed beam operation. To distinguish beams a simple numeric scheme is used. A basic number is associated with particular beam. Those numbers are referred as virtual accelerator.

Device Access allows addressing by device names. Those are rather convenient to users then the network addresses or CORBA IORs. It also provides flexibility to change the network address of particular FEC, without the need of changing the user software. The name resolution is done by the naming server, which stores the CORBA IORs of particular FECs assigned to their device names. The name resolution itself is transparent for the user and is performed by the middleware.

In addition to functions described above Device Access also provides a basis access right control mechanism [5]. This function is provided by the same server as the name resolution.

ISBN 978-3-95450-146-5

INTEGRATION OF NEW POWER SUPPLY CONTROLLERS IN THE EXISTING ELETTRA CONTROL SYSTEM

C. Scafuri, S. Cleva, Elettra - Sincrotrone Trieste S.C.p.A., Trieste, Italy

Abstract

The Elettra control system has been running since 1993. The controllers of the storage ring power supplies, still the original ones, have become obsolete and are no more under service. A renewal to overcome these limitations is foreseen. A prototype of the new controllers based on the BeagleBone embedded board and an in-house designed ADC/DAC carrier board, has been installed and tested in Elettra. A TANGO device server running in the BeagleBone is in charge of controlling the power supply. In order to transparently integrate the new TANGO controlled power supplies with the existing Remote Procedure Call (RPC) based control system, a number of software tools have been developed, mostly in the form of TANGO devices and protocol bridges. This approach allows us to keep using legacy machine physics programs when integrating the new TANGO based controllers and to carry out the upgrade gradually with less impact on the machine operation schedule.

RENEWAL OF POWER SUPPLY CONTROLLERS

The Elettra synchrotron light source has been running since October 1993. Many of its subsystems have been replaced or refurbished since then, but some of the core systems, such as the magnet power supplies controllers [1], are still operating with the original components. Unfortunately their VME boards are no longer serviced by the manufacturer and their electronic components are no more available on the market. Although we have acquired the capability to repair some of the most common failures, we cannot guarantee the full working of this vital part of the control system in future, considering that the failure rate of the components may rise as they grow older. In order to prevent this potential crisis, we have started a project to renew this part of the control system, starting with the big magnets: bending, quadrupoles, sextupoles.

Requirements and Constraints

The new power supply controller (NewPSC) that replaces the old VME-based system must satisfy several requirements:

- extend the power supply (PS) control system lifetime for at least 10 years;
- use the existing PS electrical interface to the controller with no change on the power electronics side;
- use open design components;
- simplify the present two level architecture using a single controller connected to the PS hardware and to the Ethernet control network;
- extend diagnostics capabilities and functionalities;

The new PS controller must also respect some constraints:

- have equal or better performance and stability compared to the existing PS controller;
- easy installation and replacement;
- the integration of the new PS controller with the existing software of the control system must be transparent;
- run the machine with a mix of new and old PS controllers for testing or upgrading in batches;

The "NewPSC" BeagleBone Based Controller

The NewPSC is based on a BeagleBone board [2, 3], powered by an AM3358 or AM3359 system-on-chip running at 720 MHz; it has a 100 Mb/s Ethernet interface, expansion connectors for I/O pins, open source approach for both hardware and software and runs several flavors of Linux. This powerful and inexpensive board is used as "smart node" in several applications at Elettra [4–6].

The BeagleBone provides the computing power, Ethernet connectivity for the NewPSC to the rest of the control systems and interfaces to the power supply local electronics via a dedicated carrier board developed in house [7]. The carrier board provides analog to digital conversion (ADC), digital to analog conversion (DAC) and several digital input and output channels, replacing the functions formerly carried out by several VME boards (see Fig. 1). The electrical performance has been optimized and characterized [8] in order to guarantee the performance of the NewPSC is equal or better than that of the old system.

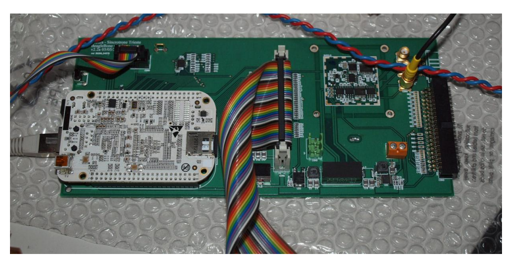

Figure 1: The NewPSC prototype.

INTEGRATION WITH THE EXISTING CONTROL SYSTEM

The control of the NewPSC is implemented as an embedded TANGO device server running on the BeagleBone. TANGO has been adopted at Elettra in 2004 and all new developments since then have been based on TANGO. Many parts of the control system and high level programs are however still running with the old CERN nc/rpc [9] based services, including the controllers of the magnet power supplies. We had to provide some means to integrate the NewPSC TANGO devices with the legacy nc/rpc based programs in

ISBN 978-3-95450-146-5

COMPUTING INFRASTRUCTURE FOR ONLINE MONITORING AND CONTROL OF HIGH-THROUGHPUT DAQ ELECTRONICS

S. Chilingaryan, M. Caselle, T. Dritschler, T. Farago, A. Kopmann, U. Stevanovic, M. Vogelgesang, Karlsruhe Institute of Technology, P.O. Box 3640, 76021 Karlsruhe, Germany

Abstract

New imaging stations with high-resolution pixel detectors and other synchrotron instrumentation have ever increasing sampling rates and put strong demands on the complete signal processing chain. Key to successful systems is high-throughput computing platform consisting of DAQ electronics, PC hardware components, communication layer and system and data processing software components. Based on our experience building a high-throughput platform for real-time control of X-ray imaging experiments, we have designed a generalized architecture enabling rapid deployment of data acquisition system. We have evaluated various technologies and come up with solution which can be easily scaled up to several gigabytes-per-second of aggregated bandwidth while utilizing reasonably priced mass-market products. The core components of our system are an FPGA platform for ultra-fast data acquisition, Infiniband interconnects and GPU computing units. The presentation will give an overview on the hardware, interconnects, and the system level software serving as foundation for this high-throughput DAQ platform. This infrastructure is already successfully used at KIT's synchrotron ANKA.

INTRODUCTION

There are several challenging tasks to be solved while building the computing infrastructure for high-throughput DAQ electronics. The Linux Kernel Driver Interface is volatile and the kernel drivers are hard to develop and maintain. Additional complexity is added by the necessity to synchronize the development of detector hardware and the required readout software. Due to limited bandwidth of system memory, effective streaming of data requires a very efficient realization of the DMA protocol. To reduce the impact of memory subsystem, the vendors of HPC hardware have developed special techniques, like *GPUDirect* [1]. If feedback loops are desired, large computing resources must be available to process dense streams of information. This is often solved using various accelerator cards with heavily parallel architectures. To reduce the latencies of feedback loops also techniques like *GPUDirect* may be applied. Another important aspect is the storage and preservation of the produced data. The storage subsystem should be able to handle several gigabytes of data per second for a long time. Finally, the question of scalability often arises. Systematically solving these questions is a difficult challenge and requires expertise in different areas of hardware, system, and software engineering.

Based on our experience building a high-throughput platform for real-time control of X-ray imaging experiments [2-5], we started to develop a hardware infrastructure and a software middleware with the goal to rapidly deploy data acquisition systems for different types of high-throughput detectors with the data rates up to 8 GB/s. We have evaluated various technologies and designed a scalable architecture based on Infiniband interconnects and GPU computing units. The core components of our system are an FPGA platform for ultra-fast data acquisition, the GPU-based Image Processing Framework "*UFO*", and the fast control system "*Concert*" [6-7]. A lot of work was put in the system services to simplify interaction of the detector, hardware, and our software components. In this work we will focus on the hardware, interconnects, and the system level software serving as foundation for our platform.

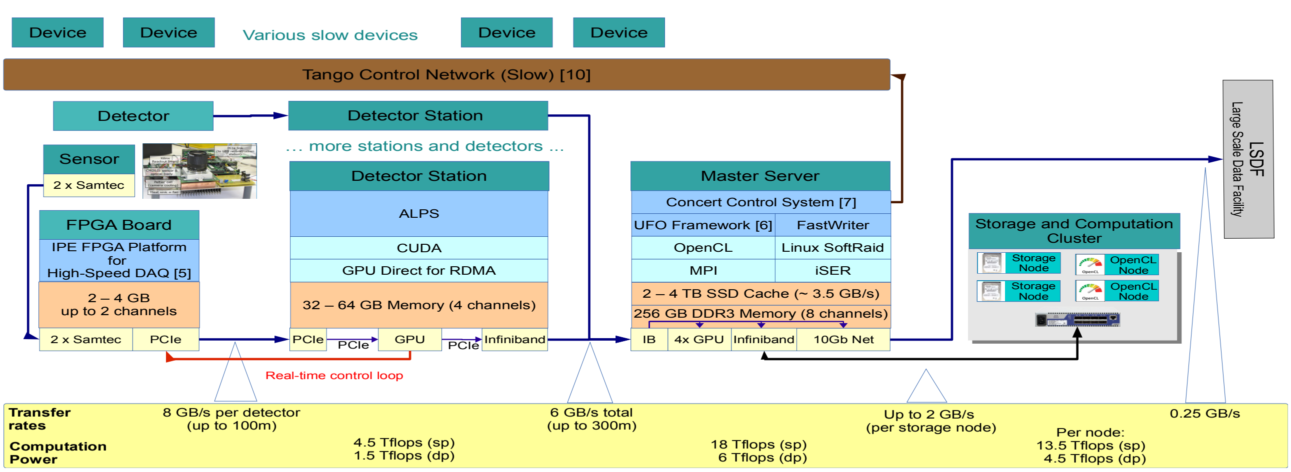

Figure 1: Architecture of control network.

ISBN 978-3-95450-146-5

DATA MANAGEMENT AT THE SYNCHROTRON RADIATION FACILITY ANKA*

D. Ressmann, W. Mexner, A. Vondrous, A. Kopmann, V. Mauch, KIT, Karlsruhe, Germany

Abstract

The complete chain from submitting a proposal, collecting meta data, performing an experiment, towards analysis of these data and finally long term archive will be described. During this process a few obstacles have to be tackled. The workflow should be transparent to the user as well as to the beamline scientists. The final data will be stored in NeXus [1] compatible HDF5 [2] container format. Because the transfer of one large file is more efficient than transferring many small files, container formats enable a faster transfer of experiment data. At the same time HDF5 supports to store meta data together with the experiment data. For large data sets another implication is the performance to download the files. Furthermore the analysis software might not be available at each home institution; as a result it should be an option to access the experiment data on site. The meta data allows to find, analyse, preserve and curate the data in a long term archive, which will become a requirement fairly soon.

INTRODUCTION

The synchrotron radiation facility ANKA [3] is located at the Karlsruhe Institute of Technology, providing light from hard X-rays to the far-infrared for research and technology. It serves as a user facility for the national and international scientific community. The traditional data management for users of ANKA was based on transportable media. A user took its data home once the beam time was over. This approach is not feasible any more, for several reasons:

- Modern detectors produce more and more data
- No analysis resources at the home location
- Only manual or no bit preservation
- Poor search ability
- Difficult to share results with the global research community

In addition new regularities of the European Union research grants, request long term archiving for research data. Under discussion are periods of ten years. Therefore to preconfigure the analysis tools in a site local cloud environment would ease the usability. As a result we create a new workflow within the ASTOR project. This project is based on an Analysis-as-a-Service (AaaS) approach, based on the on-demand allocation of VMs with dedicated GPU cores and corresponding analysis environment to provide a cloud-like analysis service for scientific users [4]. In order to allow remote access to analyse and download the data, we had to implement an Authentication and Authorization Infrastructure (AAI). In this paper we discuss the workflow from a submission of a proposal to authentication and authorization towards the beamline data management, the analysis and archival of the large raw datasets.

Submission of a Proposal

For all beam time proposals a peer review process is carried out. A user has to submit an experiment proposal in the ANKA proposal submission system ANNA. This is a web interface where details about the experiment are submitted and evaluated. For accepted proposals the users have the option to supply further meta data about each sample. The beam time will be scheduled and the user can prepare the travel to ANKA.

AUTHENTICATION AND AUTHORIZATION INFRASTRUCTURE

Usually a group of people is participating in one proposal and not all of these co-proposers are traveling to ANKA, however usually all members need access to the created data and the meta data. For the workflow at ANKA several web portals and services are required. To ease the usability a Single Sign On (SSO) approach will be provided. As a result each portal becomes a Service Provider (SP) and we introduce our own Identity Provider (IDP) for ANKA. This way we will also be able to include other IDPs, like Umbrella [5] or the DFN [6] (German National Research and Education Network), to authorize people. This has the big advantage, that user do not have to remember several accounts including their passwords.

During the experiment time at ANKA, the user can log on to the beam line data management (BLDM) frontend with the same credentials already used for the submission system.

THE BEAMLINE DATA MANAGEMENT

The beam line data management (see Figure 1) system collects all relevant data from the proposal system to make them available for the experiment. As such no duplicated data is inserted into the system, and proposal meta data can directly be connected to the measurements. This includes all logging information available via Tango [7], e.g. experimental data or vacuum level, sample conditions from the SCADA system WinCC Open Architecture [8]. Each sample gets an ANKA wide distinct identifier. Depending on the sample it can be identified via QR Code or Barcode. Once the user decides that this experiment is finished, one can close this newly created dataset, which triggers the beamline data management workflow.

ISBN 978-3-95450-146-5

PROFIBUS IN PROCESS CONTROLS

M. Clausen, T. Boeckmann, J. Hatje, O. Korth, J. Penning,
H. Rickens, B. Schoeneburg, DESY, Hamburg, Germany

Abstract

The cryogenic installations on the DESY campus are widely distributed. The liquid Helium (LHE) is produced in a central building. Three cryogenic plants are installed. One is in operation for FLASH the other two are currently in the commissioning phase and will be used for the European XFEL. Thousands of I/O channels are spread over the campus this way. The majority of the I/O devices are standard devices used in process control. The de facto standard for distributed I/O in process controls in Germany is Profibus. So it is obvious to use this bus also for cryogenic controls. Subsequently we developed also special electronics to attach temperature and level readouts to this field bus. Special diagnostic tools are available and permanently attached to the bus. Condition monitoring tools provide diagnostics which enable preventative maintenance planning. Specific tools were developed in Control System Studio (CSS) which is -the- standard tool for configuration, diagnostic and controls for all cryogenic plants. We will describe our experience over the last years with this infrastructure.

THE USAGE OF FIELD BUSES

Cryogenic installations for liquid Helium are typically widely distributed. Starting from one or more central helium liquefiers the helium is typically distributed on side by long transfer lines. All of the process I/O is installed along these lines.

It is obvious that the signals may not be connected directly to the process computer. The cable length would exceed technical limits for noise reduction or may cause grounding problems.

Starting with the HERA helium plant we introduced field buses for distributed I/O. Two decades ago we used SEDAC. SEDAC is an in-house standard at DESY. It may be used for distances as long as one kilometre. The disadvantage of an in-house solution is that there's no commercial support and any kind of signal conditioning must be developed and built in house.

In the 1990th it was decided that DESY cryogenic control should go more commercial. As a basis the CAN field bus was chosen. This provided access to various I/O modules with several different signal conditioning types including all of the basic types for process controls like: 4-20mA(in/out) and relays(in/out).

PROFIBUS IN PROCESS CONTROLS

The next level of signal conditioning came into play for the European XFEL. One basic decision was to make use of the state of the art intelligent I/O controllers like pressure transducers and valve positioners. New controllers provide intelligent local diagnostics which is useful e.g. for valve controllers. In addition they can be configured 'on the fly' through the filed-bus which us useful e.g. for pressure transducers – to change the operation range.

Two families of devices came into play: Devices for ProfiNet and those for Profibus. At that time it was not clear whether ProfiNet components would actually make their way into the I/O business or whether ProfiNet would more likely form the connection between PLCs and supervisory controls.

The decision in favour of Profibus proved to be the right one. Up to now no significant number of intelligent I/O devices found their way into the ProfiNet market.

CUSTOM PROFIBUS DEVICES

In cryogenic controls it is not sufficient to rely on the basic I/O signal conditioning. An additional family of cryogenic specific signals must be supported:

Low Temperature Readout

Temperatures below 10K are typically read out with special sensors. Small measurement currents help to avoid heating of the sensor itself. This introduces a high noise to signal ratio. Special custom readout electronics are necessary to achieve high accuracy and a good reproducibility.

Helium Level and Heater Control

Reading helium levels with a thin resistive wire is not an easy task. The measurement current must be high enough to run the wire into normal conductivity. On the other hand the current must be supervised to avoid damage when the vessel is empty or even under vacuum.

Fig. 1 shows the I/O modules which are connected by an internal CAN bus to the Profibus coupler.

Figure 1: Custom I/O: Low Temperature and Level- / Heater Control.

ISBN 978-3-95450-146-5

A PROTOTYPE DATA ACQUISITION SYSTEM OF ABNORMAL RF WAVEFORM AT SACLA

M. Ishii#, M. Kago, JASRI/SPring-8, Hyogo 679-5198, Japan
T. Fukui, T. Maruyama, T. Ohshima, RIKEN/SPring-8, Hyogo 679-5148, Japan
M. Yoshioka, SPring-8 Service Co., Ltd., Hyogo 679-5165, Japan

Abstract

We developed a data acquisition (DAQ) system for abnormal RF waveforms at the X-ray Free Electron Laser facility, the SPring-8 Angstrom Compact Free Electron Laser (SACLA). When a problem occurs, we must diagnose the source quickly. For this purpose, we developed a system that captures an abnormal RF waveform when a problem occurs in an RF system, and stores the waveform data in a database. The system consists of the VME systems, a waveform server, and a NoSQL database system, Apache Cassandra. When the VME system detects an abnormal RF waveform, it collects all related waveforms of the same shot. The waveforms are stored in Cassandra through the waveform server with shared memory as cache to complement Cassandra's eventual consistency model. We constructed a prototype DAQ system with a minimum configuration and checked its performance. In this paper, we report the scheme of the waveform DAQ system and the test results.

INTRODUCTION

SACLA has been operating for user experiments since March 2012. To maintain the scheduled user time as much as possible, we must reduce down time due to failure. To diagnose a failure source, it is quite helpful to collect the data of many accelerator components.

We have been installing two types of database systems since the beginning of commissioning. One is a data logging system with a cycle of several seconds, which provides diagnosis for slow fluctuation, such as environmental temperatures, the flow of cooling water, and the receiving voltage from the electric cubicles. The data are stored in a Sybase relational database management system [1]. Another database system is an event-synchronized data acquisition (sync-DAQ) system that collects beam currents, beam positions, and the phase and amplitude of the RF cavity pickup signals in synchronization with the beam operation cycle at the current maximum of 60 Hz. Shot-by-shot data are tagged with a master trigger number to identify the the beam shot to which the data belong. The master trigger number is generated by counting a master trigger signal distributed from the master oscillator system. Shot-by-shot data with trigger numbers are stored in the MySQL relational database management system [2]. In addition, the DAQ system collects RF waveform data every 10 minutes. At the collection, however, it is difficult to catch an abnormal RF waveform from a rare failure event that may occur only a few times a day. If we can catch the abnormal RF waveform, it is very helpful to analyze the phenomenon, because it has more information than point data, which is sampled from a waveform. Therefore, we developed a DAQ system to capture abnormal RF waveforms, an abnormal WFM-DAQ system. As a first step, we constructed a prototype system at a test stand.

ishii@spring8.or.jp

LOW-LEVEL RF CONTROL SYSTEM

The linear accelerator of the SACLA comprises an electric gun, a 238-, 476-, 1428-MHz multi-sub harmonic bunching and accelerating system as an injector, eight S-band accelerating structures, and 128 C-band accelerating structures and in-vacuum undulators beamlines [3]. The low-level RF (LLRF) system controls the 74 RF units in the SACLA accelerator. A VME system, used at each RF unit, consists of a CPU, a trigger delay unit (TDU), a DAC and three ADC boards. The TDU board delivers the delayed trigger signal to the ADC and DAC boards, and generates the master trigger number by counting the master triggers. The ADC and DAC boards running 238-MHz clock detect the phase and amplitude signals generated by klystron, and generate signals of the IQ modulator for klystron input.

In the VME system, many processes are in operation including an equipment management process (EM), a data logging process, a Sync-DAQ process (SYNQDAQ-EMA) in synchronization with the beam operation cycle, and a feedback process (PID-EMA) for the stabilization of the phase and amplitude of the RF cavity with 100-ms sampling intervals [4].

SCHEME OF ABNORMAL WFM-DAQ SYSTEM

The abnormal WFM-DAQ system consists of a VME system, a waveform server, and Apache Cassandra, which is a key-value database system. Figure 1 shows a diagram of the abnormal WFM-DAQ system.

Detection of an Abnormal Waveform and Transfer of Data

The ADC board has four channels and four-memory banks that can store 512 waveform data in synchronization with a trigger signal. In addition, the ADC board generates an interrupt signal when it detects an abnormal waveform by comparing it with a reference waveform. When the sampled waveform exceeds a defined tolerance

ISBN 978-3-95450-146-5

UPGRADE OF SACLA DAQ SYSTEM ADAPTS TO MULTI-BEAMLINE OPERATION*

Toshinori Abe, Yukito Furukawa, Takaki Hatsui, Yasumasa Joti, Takashi Kameshima, Takahiro Matsumoto, Kensuke Okada*, Takashi Sugimoto, Ryotaro Tanaka, Mitsuhiro Yamaga, JASRI/SPring-8, Hyogo, Japan
Makina Yabashi, RIKEN Spring-8 Center, Hyogo, Japan

Abstract

This paper presents details of a major overhaul of the DAQ (data acquisition) system for user experiments at SACLA (SPring-8 Angstrom Compact Free Electron Laser [1]). The DAQ system has been providing a common experimental framework to various SACLA users since March 2012.

In 2014, SACLA introduced a third beamline to increase the capacity of experiments that can be performed. With respect to the DAQ system, it is a challenge to operate multiple experiments simultaneously. To ensure that the increased capacity of experiments can be handled, the network architecture was redesigned so that controls and data streams are made independent. A new tag supply system, which guarantees reliable synchronization, was implemented. In addition, a 90 TFLOPS supercomputer was installed to meet the growing demand for offline analyzing power.

INTRODUCTION

At SACLA, the DAQ system is required to store shot-by-shot information synchronized with an X-FEL (X-ray Free Electron Laser) beam of 60 Hz at maximum repetition rate. The data throughput goes up to 6 Gbps with images (e.g., X-ray diffraction images) obtained from twelve sensors of an MPCCD (multiport charge-coupled device [2]). The data are stored in a hierarchical storage system capable of more than 7 PB at the last stage. The DAQ system incorporates prompt data processing performed by a 14 TFLOPS PC cluster.

For multi-beamline operation, the control and data streams are duplicated and separated for the beamlines, i.e., they are made independent. In addition, the architecture of the control line is reformed to reduce risk of mishandling. The new tag supply system implemented for synchronization has been operating stably. In the offline part, a 90 TFLOPS supercomputer was installed to boost data analysis capacity.

In the following section, first, we provide an overview of the system; then, we describe recent major upgrades. Further, we briefly discuss features to be implemented in the near future, and finally provide a summary.

DAQ OVERVIEW

The SACLA DAQ system has been described in detail elsewhere [3]. The DAQ system consists of online and offline parts, which have been described in the following sub-sections.

Online Part

Figure 1 shows a schematic view of the online part of the SACLA DAQ system. Image data are transferred over the 10-gigabit Ethernet to a cache storage system, which has multiple writing servers to handle the high throughput. In the data stream are the data-handling servers, which can analyze image data on time. One of important functions on the data handling server is sending live view images to user terminals for the monitoring purpose. Many other small-sized data such as photo diode amplitudes of beam monitors and pulse motors' position of various instruments are stored in the database. The database is also used to store experimental conditions. Because at SACLA, the X-FEL target specimen is likely destroyed by a single shot and certain experiments are sensitive to a slight variation of the beam characteristics, the DAQ should be able to recall and provide the information related to each shot. To this end, all components of the DAQ are synchronized with X-FEL beam shot and data are stored with a unified tag number assigned to each shot.

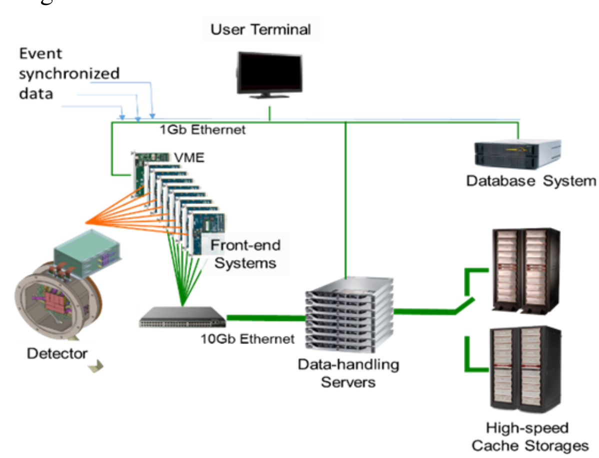

Figure 1: Online part of the SACLA DAQ.

From September 2013 to July 2014, 300 TB of image data were stored through the cache storage, and two cache storages were switched just once.

Offline Part

Figure 2 shows a schematic view of the offline part of the SACLA DAQ system. Through the offline part, SACLA provides a platform for analysis. Users can access the data stored in the cache or in the archive system along with various parameters stored in the

ISBN 978-3-95450-146-5

SARDANA – A PYTHON BASED SOFTWARE PACKAGE FOR BUILDING SCIENTIFIC SCADA APPLICATIONS

Zbigniew Reszela, Guifre Cuni, David Fernandez-Carreiras, Jorg Klora [on leave], Carlos Pascual-Izarra, CELLS-ALBA Synchrotron, Cerdanyola del Vallès, Spain
Tiago Coutinho, ESRF, Grenoble, France

Abstract

Sardana is a software suite for Supervision, Control and Data Acquisition in scientific installations. It aims to reduce cost and time of design, development and support of the control and data acquisition systems[1]. Sardana, thanks to the Taurus library[2], allows the user to build modern and generic interfaces to the laboratory instruments. It also delivers a flexible python based macro environment, via its *MacroServer*, which allows custom procedures to be plug in and provides a turnkey set of standard macros e.g. generic scans. Thanks to the *Device Pool* the heterogeneous hardware could be easily plug in based on common and dynamic interfaces. The Sardana development started at Alba, where it is extensively used to operate all beamlines, the accelerators and auxiliary laboratories. In the meantime, Sardana attracted interest of other laboratories where it is used with success in various configurations. An international community of users and developers[3] was formed and it now maintains the package. Modern data acquisition approaches guides and stimulates current developments in Sardana. This article describes how the Sardana community approaches some of its challenging projects.

SARDANA AND ITS COMPONENTS

Sardana is a distributed control system based on the client-server model. The communication protocol is Tango[4]. Different Sardana configurations are possible depending on the scale of the installation. As an example, a small laboratory could have a single Sardana server exporting one Device Pool and one MacroServer. If needed, multiple Device Pool servers could be distributed over different hosts. Configurations with many MacroServer servers are also possible. Multiple Graphical User Interface (GUI) and Command Line Interface (CLI) clients can communicate with the Sardana system, at the same time.

Device Pool

Scientific installations are characterized by multiple and heterogeneous hardware. The particle accelerators comprises power supplies, vacuum equipment, radio frequency stations, insertion devices and many diagnostics and actuators among others. The laboratories, e.g. a synchrotron beamline, usually consists of even more diverse instruments like, diffractometers, monochromators and sophisticated detectors full of moveable axes and experimental channels. These laboratories frequently require to modify their configuration depending on the experiment being performed. Sardana interfaces all the equipments via the Device Pool and its plugin *controller* classes. Sardana elements could be classified by their types in three main groups:

- moveables: physical motors e.g. stepper or servo motors, piezo actuators and logical pseudo motors e.g. the energy, composed translations
- experimental channels: counters and pseudo counters; 0D – scalar values based on mathematical operations e.g. averaging or integration of samples; 1D – one dimensional detectors e.g. Multi Channel Analyzers, Position Sensitive Detectors; 2D – two dimensional detectors e.g. CCD cameras
- other elements: communication channels, enumerated scalars called I/O registers

The controllers group the Sardana elements, which are organized and identified by the axis number. All these elements are represented by TANGO devices, and could be spread in many Device Pool servers or grouped in a single one. The Device Pool optimizes common control processes. The grouped acquisitions are handled by the M*easurement Group*, which configures and synchronizes the experimental channels. The grouped motions are implemented inside the Pool and if the hardware allows that, motion of all the axes can be started simultaneously by one single command. The API of the controllers and the programmed algorithms take into account all these particularities so access to the hardware is optimized.

MacroServer

A basic requirement for the scientific SCADA system is to provide a sequencer capable to plan and execute experiment procedures. MacroServer, together with its *Doors*, provide these and other functionalities (via Door different client applications access the MacroServer engine). The MacroServer allows the execution of multiple procedures (called *macros*) sequentially or even simultaneously if different Doors are used. A macro can accept input parameters and return a result or produce data, which might be interchanged between chained or nested macro sequences. It is possible to interrupt the sequence execution at any moment as well as temporarily pause and resume it. Features for adding, editing and browsing the available macros are accessible via the MacroServer at runtime. Sardana provides a miscellaneous set of standard macros. Their naming and parameters often follow conventions adopted by SPEC[5] what optimizes the users learning curve. Probably the most useful and sophisticated macros are the generic, n-dimensional scans. The scan process consist of data acquisition of the involved experimental channels while varying the scanning variable (typically a moveable

ISBN 978-3-95450-146-5

A NEW DATA ACQUISITION SOFTWARE AND ANALYSIS FOR ACCURATE MAGNETIC FIELD INTEGRAL MEASUREMENT AT BNL INSERTION DEVICES LABORATORY*

M. Musardo#, D. Harder, P. He, C. A. Kitegi and T. Tanabe
National Synchrotron Light Source II, Brookhaven National Laboratory
Upton, New York 11973-5000, USA

Abstract

A new data acquisition software has been developed in LabVIEW to measure the first and second magnetic field integral distributions of Insertion Devices (IDs).

The main characteristics of the control system and the control interface program are presented. The new system has the advantage to make automatic and synchronized measurements as a function of gap and/or phase of an ID. The automatic gap and phase control is a real-time communication based on EPICS system and the eight servomotors of the measurement system are controlled using a Delta Tau GeoBrick PMAC-2.

The methods and the measurement techniques are described and the performance of the system together with the recent results will be discussed.

INTRODUCTION

The National Synchrotron Light Source II (NSLS-II) at Brookhaven National Laboratory is a new 3 GeV electron storage ring of third generation designed to provide synchrotron radiation with a very broad energy range and an ultra high brightness and intense flux using advanced insertion devices [1]. In order to validate that IDs delivered to NLSL-II meet the tight specification, they are accurately surveyed before installation into the storage ring. Two benches, a Hall probe bench for local field measurement and the Integrated Field Measurement System (IFMS), have been purchased to magnetically survey the IDs [2]. The IFMS supplied by ADC USA Inc, is a set of three field integral measurement systems, a stretch wire bench, a moving wire and a flip coils. An upgrade program has been started; this article reports the first stage of the IFMS upgrade, focused on the flip coil system.

FLIP COIL MEASUREMENT SYSTEM

The measuring bench consists of a long coil of 10 turns of 38 AWG beryllium copper wire stretched between two granite blocks located on both sides of an ID. Each support includes three linear motorized stages to position and move the coil in the horizontal and vertical axes and a rotary stage employed to rotate the coil of 360° with an angular accuracy of 40 arcsec and encoder resolution of about 0.005 deg. It is equipped with rotary encoders mounted to the vertical axis linear stage in horizontal orientation. The flip coil mounts to a special spool on the rotary axis located on each pedestal, it is looped between these spools to form one continuous loop. The width of the coil is 4 mm and the length is about 5 m. In order to reduce the coil sagging the longitudinal axis stage can be tensioned and adjusted manually by varying the zero position. A tensiometer placed in the bobbin give an indirect value of the gravitational sagging of the coil. The tension sensor is made by Omega Engineering and has a range of 100 pounds and a resolution of ± 0.2% of full scale. All stages are assembled on the pedestals, which have an extremely flat tolerance on the top surface of about ±10 µm. Each pedestal has three leveling feet that also provide 10 mm adjustment in the horizontal X and longitudinal Z directions and 50 mm in the vertical Y direction. All linear axes have a Renishaw RELM linear encoder feedback with 0.1 µm resolution and ±1 µm accuracy. The Y and Z axes have 150 mm travel and the X axis have 300 mm travel, which is sufficient to ensure a measuring range appropriate for each magnetic device. Each axis has a brushless DC motor operated in closed loop mode with limit switches and a home limit switch that is used to tell the motion controller that it is near the home pulse located on the encoder. The linear encoders on the motors provide velocity, acceleration and position feedback. The motion control system is based on a Delta Tau GeoBrick PMAC-2 controller, which is responsible for the coordinated motion of four pairs of motors in a master-slave configuration. A serial communication is used to control and synchronize the various hardware components of Delta Tau. Geo-Brick generates the spatial and temporal triggers and convert the analog tension signal to digital. It provides eight axis of servo control and has a position synchronized triggering on four axes, X, Y, Z and rotary. Delta Tau GeoBrick PMAC-2 is a fully programmable motion controller and several software parameters must be configured to properly control each motor.

A digital multimeter Keithley DMM 2701 is used to record sensitive induced voltage signals from the coil. Model 2701 is a $6^{1/2}$-digit high-performance multimeter data acquisition system. It is equipped with a custom switching card to select the measurement source. Results of the DMM are transmitted to the host computer using a standard communication serial RS232 port. Keithley

*Work supported by DOE contract DE-AC02-98CH10886
#musardo@bnl.gov

ISBN 978-3-95450-146-5

INTEGRATING SIEMENS PLCS AND EPICS OVER ETHERNET AT THE CANADIAN LIGHT SOURCE

R. Tanner, S. Hu, G. Wright, CLSI, Saskatoon, Canada
E. Matias, Mighty Oaks, Victoria, Canada

Abstract

The Canadian Light Source (CLS) is a 3rd-generation synchrotron light source on the University of Saskatchewan Campus in Saskatoon, SK, Canada. The control system is based on the Experimental Physics and Industrial Controls System (EPICS) toolkit [1]. A number of systems delivered to the CLS arrived with Siemens, PLC-based automation. EPICS integration was initially accomplished circa 2003 using application-specific hardware; communicating over Profibus. The EPICS driver and IOC application software were developed at the CLS. The hardware has since been discontinued. To minimize reliance on specialized components, the CLS moved to a more generic solution, using readily-available Siemens Ethernet modules, CLS-generated PLC code, and an IOC using the Swiss Light Source (SLS) Siemens/EPICS driver [2]. This paper will provide details on the implementation of that interface. It will cover detailed functionality of the PLC programming, custom tools used to streamline configuration, deployment and maintenance of the interface. It will also describe handshaking between the devices and lessons learned. It will conclude by identifying where further development and improvement may be realized.

SYSTEM OVERVIEW

The general system overview is shown in Fig. 1. Siemens PLCs from the S7-400 and S7-300 families have been fitted with Industrial Ethernet modules.

These PLCs have been delivered by vendors. There are two S7-300s for the storage ring SRF (Superconducting Radio Frequency) cavity, one in the BR (Booster Ring) RF controls, and S7-400s for the Linde Kryotechnik-supplied cryogenics plant and SR (Storage Ring) Thales RF amplifier.

At the time of this writing, EPICS integration of the TUV-certified, failsafe PLCs used in the CLS Access Control and Interlock System (ACIS) and O2 monitoring is still under review.

Because the PLCs perform many critical functions, a separate Virtual Local Area Network (VLAN) had been dedicated for them (referred to as a “plantbus” by the manufacturer), the EPICS IOC (Input-Output Controller) and Engineering Stations (“ES”). Access to the PLCs is meant to be accomplished only via the development stations or the IOC. The ESs have an industrial Ethernet adapter for connecting to the plantbus and commonly – available adapter for connecting to the “office” network. The IOC connects using the default VMWare Ethernet adapter and is configured to be on the VLAN, as well.

Any other services are configured to be available to the VLAN rather than accessible via a gateway or router (the IOC also provides NTP, for example). Traffic is restricted on the VLAN for performance and security considerations.

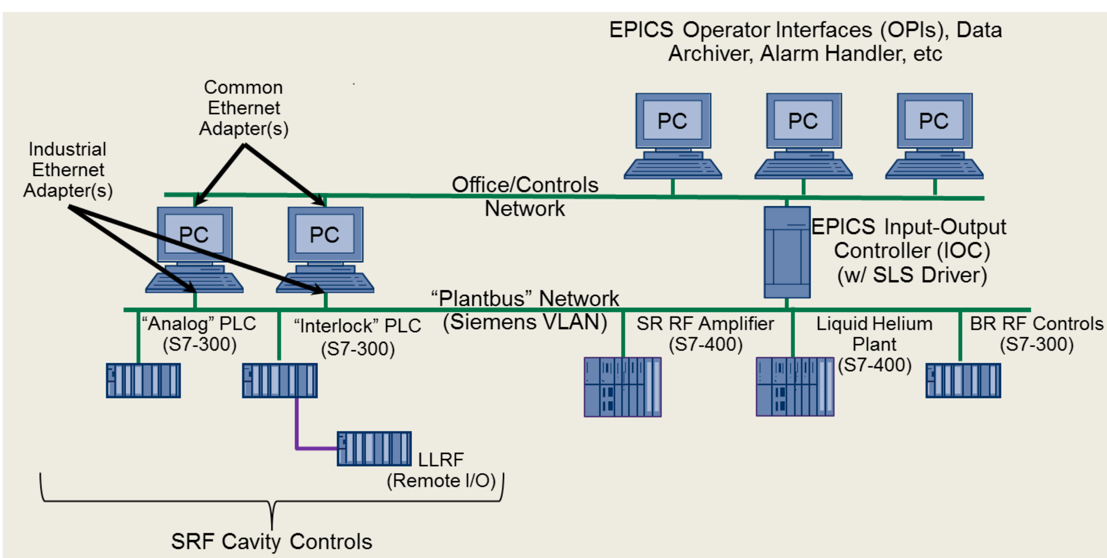

Figure 1: CLS Siemens PLC Network Topology

ISBN 978-3-95450-146-5

SETUP OF A HISTORY STORAGE ENGINE BASED ON A NON-RELATIONAL DATABASE AT ELSA

D. Proft*, F. Frommberger, W. Hillert, ELSA, Bonn, Germany

Abstract

The electron stretcher facility ELSA provides a beam of unpolarized and polarized electrons of up to 3.2 GeV energy to external hadron physics experiments. Its in house developed distributed computer control system is able to provide real time beam diagnostics as well as steering tasks in one homogeneous environment. Recently it was ported from HP-UX running on three HP workstations to a single Linux personal computer.

This upgrade to powerful PC hardware opened up the way for the development of a new archive engine with a noSQL database backend based on Hyptertable. The system is capable of recording every parameter change at any given time. Beside the visualization in a newly developed graphical history data browser, the data can be exported to several programs - for example a diff-like tool to compare and recall settings of the accelerator.

This contribution will give details on recent improvements of the control system and the setup of the history storage engine.

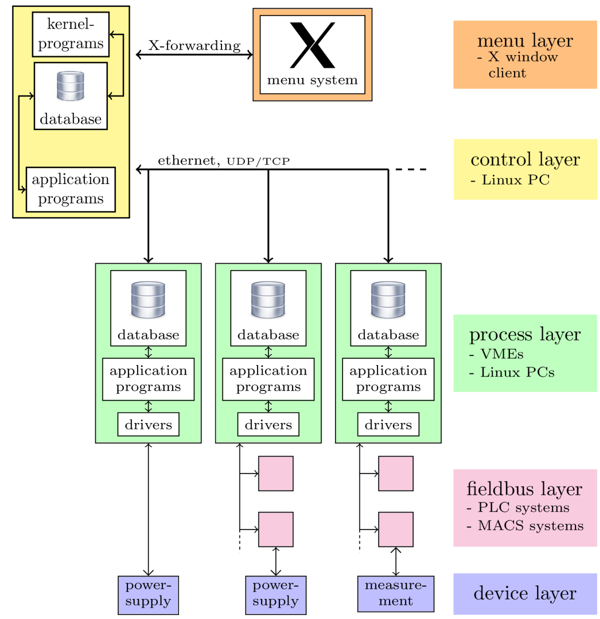

Figure 1: Hard- and software layers of the control system.

INTRODUCTION

The main features of the ELSA accelerator control system [1, 2] include a completely event based data handling model and a separation of the core functionality (database and event handling by the *kernel*) from userspace applications. It combines steering tasks and real time beam diagnostics in one homogeneous environment. A transparent design allows access to the X windows-based graphical user interface from any computer. An overview of the hard- and software layers of the whole system is given in Figure 1, [3].

A key component of the control system is a kernel managing a central shared memory database. The database is separated into several parts, i.e. the *resource base* containing structural information about parameters like limits, max. number of vector elements and the quantity's physical unit. The structural information is complemented by the online database filled with actual parameter values, which are updated continuously at runtime.

There are currently 14 827 parameters defined in the control system. These are grouped into *controlled* ($\approx$ 4000), *measured* ($\approx$ 9000) and other parameters. Each group consists of four different data types: analog values (represented by floating point numbers), digital values (mostly switching values or integers), strings (character sequences) and arbitrary byte sequences.

The update of controlled parameters occurs rather rarely, and is mainly invoked by user interaction or automatic measurement processes. On the other hand most measured parameters are updated on a regular basis, either cycle-synchronous (typically every 5 s) or with arbitrary rates up to 10 Hz. Parameters with higher rates are accumulated in vectors and transfered (also on a regular basis) with a slower rate.

The data rate produced by 675 updates/s (on average) is roughly 50 kB/s to 100 kB/s[1]. This results in a total volume of $\approx$ 6.1 GB per day.

Primary goal of the newly developed archive engine is, of course, to archive all these changes together with a timestamp, regardless of the type or source of the values. Second goal is to keep the investment cost as low as possible. Therefore the archive database should run on a regular desktop computer with no special hardware needs. Here, a bottleneck could be the access time, in which the data can be returned back from the database. For best user experience access times in the magnitude of few seconds are required.

DATABASE BACKEND

Hypertable is a non relational database with Google's *Bigtable* design which was chosen as the database backend. It runs on top of several file systems, including distributed ones (e.g. *HDFS*) and storage in the local file system. The instances of the main server, called *RangeServer*, can be distributed among different machines with one *Master* process for administration.

* proft@physik.uni-bonn.de

[1] 50 kB/s during maintenance, 100 kB/s during usual operation

ISBN 978-3-95450-146-5

NEWS FROM THE FAIR CONTROL SYSTEM UNDER DEVELOPMENT

R. Bär, C. Betz, D. Beck, J. Fitzek, U. Krause, S. Jülicher, M. Thieme, R. Vincelli
GSI Helmholtz Center for Heavy Ion Research, Darmstadt, Germany

Abstract

The control system for the FAIR (Facility for Antiproton and Ion Research) accelerator facility is presently under development and implementation. The FAIR accelerators will extend the present GSI accelerator chain, then being used as injector, and provide anti-proton, ion, and rare isotope beams with unprecedented intensity and quality for a variety of research programs.

This paper summarizes the general status of the FAIR project and focusses on the progress of the control system design and its implementation. This paper presents the general system architecture and updates on the status of major building blocks of the control system. We highlight the control system implementation efforts for CRYRING, a new accelerator presently under re-commissioning at GSI, which will serve as a test-ground for the complete control system stack and evaluation of the new controls concepts.

FAIR

FAIR is a unique new international accelerator facility for the research with antiprotons and heavy ions. When finished (planned for 2019), FAIR will be a host laboratory for basic research for about 3000 scientists from approximately 50 countries.

The FAIR accelerators are a major extension of the present GSI accelerators then being used as injectors. Figure 1 gives an overview of the full FAIR accelerator complex. FAIR will be built in a modular approach, starting with the synchrotron SIS100, two consecutive storage rings CR and HESR and the p-linac proton injector.

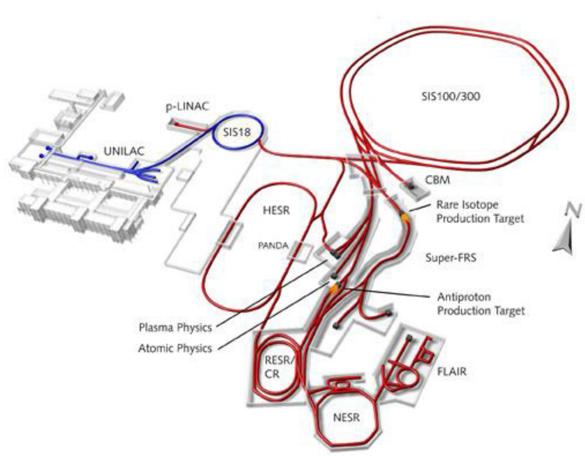

Figure 1: Schematic overview of the GSI (blue, existing) and FAIR (red, to be built) accelerator complex.

Civil Construction

The FAIR complex will cover an area of 20 hectares and require 600,000 cubic metres of concrete as well as 35,000 tons of steel [1]. Construction teams will be building a tunnel to house the heart of the complex, the SIS-100 ring accelerator with a circumference of 1.1 kilometres. The 24 buildings and several tunnels provide sufficient room for a total of 3.5 kilometers of beam-lines as well as huge detectors and a complex technical infrastructure.

Figure 2 gives an impression of the construction site. After clearing and preparing the construction area, approximately 1350 bore piles have already been built to stabilize the subsoil, minimize building settlement and ensure that buildings will settle evenly.

Moreover, several kilometers of access roads have already been built across the site for the construction site traffic.

Figure 2: Aerial photo of the construction site taken on May 25, 2014 (photo: J. Schäfer, FAIR).

CONTROL SYSTEM OVERVIEW

The FAIR accelerator control system comprises the full electronics, hardware, and software to control, commission, and operate the GSI/FAIR accelerator chain with multiplexed parallel beams. The development of the control system takes advantage of several collaborations with CERN by using, adapting and improving framework solutions like the settings management framework LSA, the front-end software framework FESA and the White Rabbit (WR) based timing system as core components.

The general structure of the FAIR accelerator control system is organized in three layers. The equipment layer consists of equipment interfaces, embedded system controllers, and software representations of the equipment. A dedicated real-time network based on White Rabbit is used to synchronize and trigger actions on equipment level. The middle (business) layer provides

ISBN 978-3-95450-146-5

PROGRESS AND CHALLENGES DURING THE DEVELOPMENT OF THE SETTINGS MANAGEMENT SYSTEM FOR FAIR

H. Hüther, J. Fitzek, R. Müller, D. Ondreka, GSI, Darmstadt, Germany

Abstract

A few years into development of the new control system for FAIR (Facility for Antiproton and Ion Research), a first version of the new settings management system is available.

As a basis, the CERN LSA framework (LHC Software Architecture) is being used and enhanced in collaboration between GSI and CERN. New aspects, like flexible cycle lengths, have already been introduced while concepts for other requirements, like parallel beam operation at FAIR, are being developed.

At SIS18, LSA settings management is currently being utilized for testing new machine models and operation modes relevant for FAIR. Based upon experience with SIS18, a generic model for ring accelerators has been created that will be used throughout the new facility. It will also be deployed for commissioning and operation of CRYRING by the end of 2014.

During development, new challenges came up. To ease collaboration, the LSA code base has been split into common and institute specific modules. An equivalent solution for the database level is still to be found. Besides technical issues, a data-driven system like LSA requires high-quality data. To ensure this, organizational processes need to be put in place at GSI.

FAIR CONTROL SYSTEM REQUIREMENTS

Construction work for FAIR has been started in 2011 next to the existing GSI complex. In May 2014, the first key construction phase has been completed with the conclusion of pile-drilling work for the foundations of the facility. Once fully operational, FAIR will provide nine new accelerator installations, using the existing linac and synchrotron SIS18 as injectors [1]. See Fig. 1 for an overview of the FAIR accelerator complex.

The designated operation modes of FAIR put demanding requirements on the new control system currently in development. To optimize the number of concurrent research programs, the facility will provide up to five beams in parallel with pulse-to-pulse switching between different particle types. Additionally, great flexibility shall be provided, allowing to change the parallel operation schemes on a daily basis.

Tight resource restrictions make meeting these requirements even more challenging. After thoroughly evaluating possible options for most effectively implementing a new settings management component for the FAIR control system, enhancing CERN's existing LSA framework was identified as the most suitable approach.

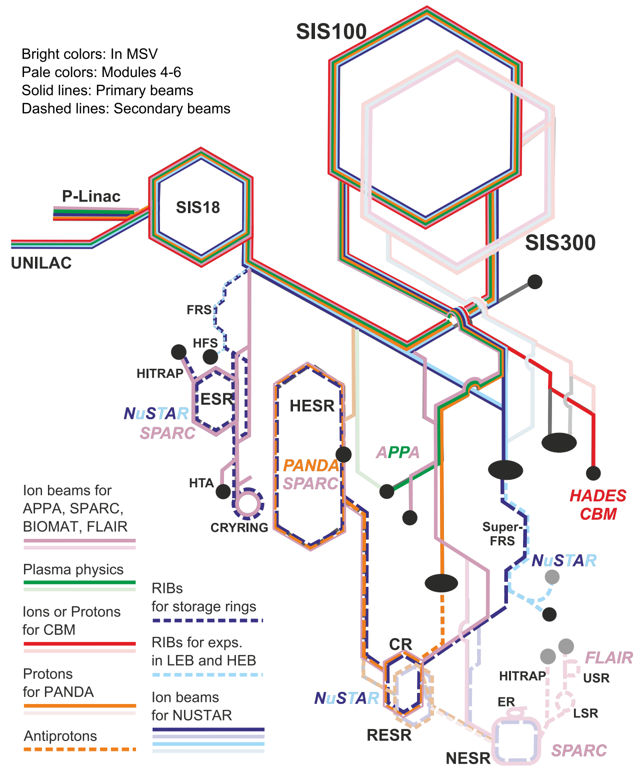

Figure 1: Overview of the FAIR accelerator complex, P. Schütt, GSI.

COLLABORATION WITH CERN

LSA is being developed at CERN since 2001. In the following years, it has constantly evolved and matured [2]. Today, CERN uses LSA to control the majority of its accelerators.

Since 2007, GSI and CERN collaborate to enhance LSA for mutual benefit and to use it as the core component for settings management within the FAIR control system.

LSA has been chosen because of its completeness regarding all important settings management aspects and also certain characteristics that are especially important for implementing enhancements towards FAIR. The framework has been designed with extensibility in mind, reflected by modular structure and separation of concerns in its design. It provides generic means of modelling different accelerators and plug-in mechanisms for adding functionality while a modern architecture ensures scalability.

As powerful as LSA already was when the collaboration started, it still requires major enhancements to support the specific requirements of FAIR. Significant steps towards this goal have been made since progress has last been reported on publicly.

FESA3 INTEGRATION IN GSI FOR FAIR

S. Matthies, H. Bräuning, A. Schwinn, GSI, Darmstadt, Germany
S. Deghaye, CERN, Geneva, Switzerland

Abstract

GSI decided to use FESA (Front-End Software Architecture) as the front-end software tool kit for the FAIR accelerator complex. FESA was originally developed at CERN. Since 2010 FESA3, a revised version of FESA, is developed in the frame of an international collaboration between CERN and GSI. During development of FESA3 emphasis was placed on the possibility of flexible customization for different environments and to provide site-specific extensions to allow adaptation for the contributors. GSI is the first institute different than CERN to integrate FESA3 into its control system environment. Some of the necessary preparations have already been performed to establish FESA3 at GSI. Examples are RPM packaging for multiple installations, support for site-specific properties and data types, first integration of the White Rabbit based timing system, etc.. Further developments such as e.g. integration of a site-specific database or the full integration of GSI's beam process concept for FAIR will follow.

INTRODUCTION

GSI's FAIR [1] project is a challenge and a chance to establish a revised control system solution. A couple of years ago it was decided to develop the main parts of the future control system for FAIR (such as FESA, LSA [2] and the middleware CMW [3]) in the frame of an international collaboration with CERN.

This paper gives an overview of how the FESA3 framework is extended to suit into the future FAIR control system environment.

MODULARITY OF FESA3

To establish the FESA framework on sites different than CERN the main focus during development of FESA3 was modularity and extensibility. Modularity is achieved by clear separation of its components and involved technologies into core- and site-specific packages. Extensibility of the FESA3 framework is ensured by the possibility to provide site-specific extensions to the core packages. This involves the FESA3 framework packages as well as the components of the FESA3 plug-in for the integrated development environment Eclipse.

In general the core packages contain the common code base that is used by both participating sites. The common part provides the interfaces, (abstract) base classes as well as the functionality that does not have to be extended.

The FESA3 framework combines the usage of different technologies and programming languages such as XML, XSLT, Python, C++ and JAVA.

The site-specific components extend the common part by providing the functionality required only by the implementing site. This is realized by using software design concepts such as inversion of control and inheritance, depending on the technology used. Figure 1 gives an overview of the main FESA3 framework components. The extension packages are marked by "-EXT" which stands for either CERN or GSI. Accordingly a similar structure is realized for the parts that constitute the JAVA based FESA3 Eclipse plug-in.

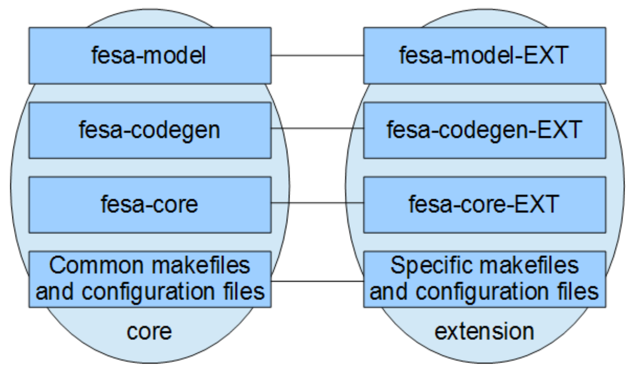

Figure 1: FESA3 Framework Components.

SITE-SPECIFIC FESA FEATURES

The FESA3 installation at GSI has several site-specific features and extensions.

Standard Properties

To provide a common interface of FESA3 based equipment software to the application layer, an elaborated set of standard properties is defined at GSI. Standard properties are common properties that each accelerator device should provide to the application layer. Typical examples are properties such as Status, Power, Init or Version. Site-specific properties may be coupled to site-specific data field types. The properties as well as their data fields are pre-defined in the site-specific template for new FESA3 equipment software. For operational FESA based equipment software to be used within the FAIR control system environment these properties must be implemented by the FESA equipment software developer. GSI's FESA development guideline outlines this and the other issues to be considered when developing productive FESA3 based equipment software for operation of the FAIR accelerator complex.

ISBN 978-3-95450-146-5

THE FAIR R³B PROTOTYPE CRYOGENICS CONTROL SYSTEM

H. Simon, T. Hackler, C. Betz, E. Momper, M. Zaera, C. S. Schweizer, M. Stern,
D. Sánchez-Valdepeñas, GSI Helmholtz Centre for Heavy Ion Research, Darmstadt, Germany

Abstract

The superconducting GLAD (GSI Large Acceptance Dipole) [1] magnet is one of the major parts of the R3B (Reactions with Relativistic Radioactive Beams) experiment. The cryogenic operation will be ensured by a fully refurbished TCF 50 cold box and oil removal system. One of the major design goals for its control system is to operate as independent as possible from the magnet control system. The cold box control system is seen as a first prototype for the later cryogenic installations in the Facility for Antiproton and Ion Research (FAIR). The operation of the compressor, oil removal system, and the gas management was successfully tested in January 2014. Within late winter 2014 a first cool-down of the refurbished cold box is planned. Once the magnet will be delivered, the magnet and the cryogenics controls will be commissioned together. To do all these learning and realization steps can be seen as preparatory work for novel industrial control systems to be established at the FAIR facility [2].

INTRODUCTION

The FAIR R³B Prototype Cryogenics Control System comprises a fully refurbished TCF 50 cold box and an oil removal system from DESY (Deutsches Elektronen-Synchrotron).

The cold box tubing has been modified in order to meet the requirements of the cryogenic process being adapted to the GLAD magnet operation. In the refurbishing and upgrading process, an outlet was added to the shield cooling of the superconducting coil. Also all sensors and actors have been especially chosen in order to test and select possible equipment for the later FAIR installation. This included two versions of actors for the valves and passive pressure sensors which are apparently more radiation resistant. The full instrumentation of the cryo plant has been renewed and a new control cabinet replacing the previous controls has been installed.

The control system development process is following a staged implementation. First step was to understand, to design and to implement all process functionality for compressor and oil removal system inside a S7-319F with PROFIBUS and PROFINET I/O modules using WinCC OA as SCADA platform. In the second step the program logic designed in step 1 has been successfully migrated to a new version based on the CERN Unified Industrial Control System (UNICOS) framework [3]. This was the first time at GSI, that UNICOS has been used. As next steps there is the design and implementation of all algorithms and control parameters needed for the cold box processes (cool-down, shield/magnet supply and warm up) foreseen.

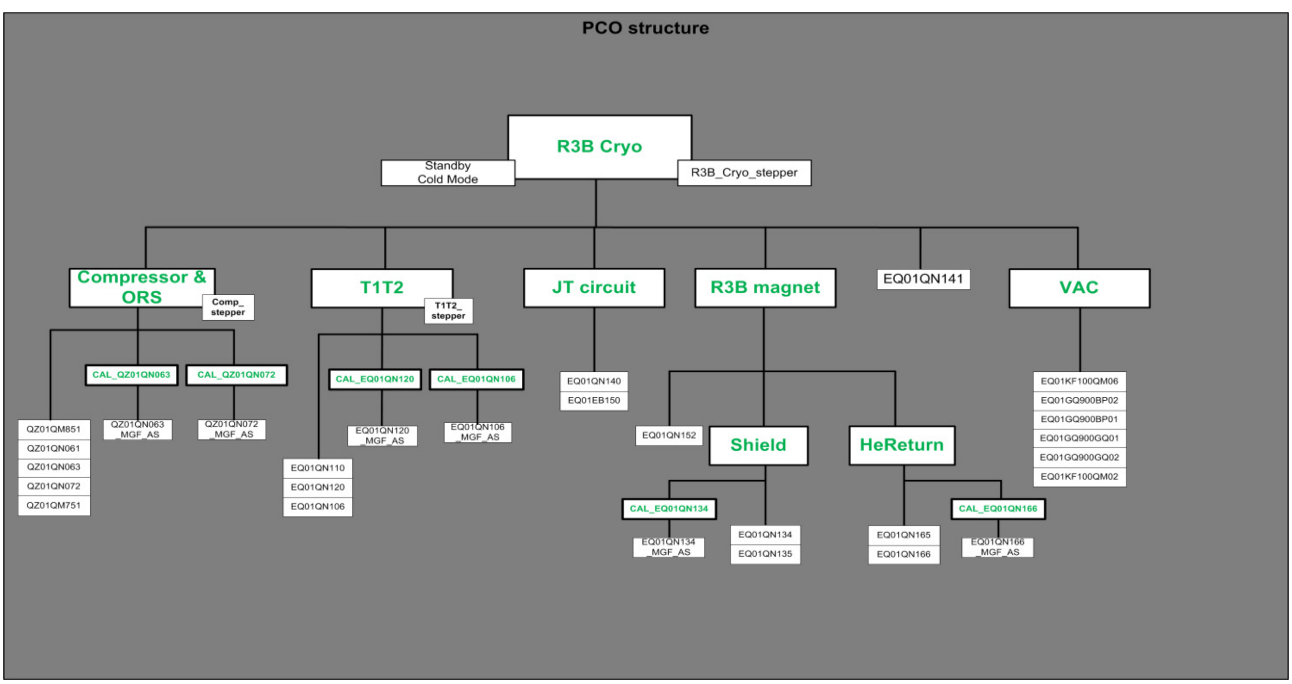

Figure 1: UNICOS unit (PCO) structure of the R³B facility.

ISBN 978-3-95450-146-5

AN EXTENSIBLE EQUIPMENT CONTROL LIBRARY FOR HARDWARE INTERFACING IN THE FAIR CONTROL SYSTEM

M. Wiebel, GSI, Darmstadt, Germany

Abstract

In the FAIR control system the SCU (Scalable Control Unit, an industry PC with a bus system for interfacing electronics) is the standard front-end controller for power supplies. The FESA-framework is used to implement front-end software in a standardized way, to give the user a unified look on the installed equipment. As we were dealing with different power converters and thus with different SCU slave card configurations, we had two main things in mind: First, we wanted to be able to use common FESA classes for different types of power supplies, regardless of how they are operated or which interfacing hardware they use. Second, code dealing with the equipment specifics should not be buried in the FESA-classes but instead be reusable for the implementation of other programs. To achieve this we built up a set of libraries which interface the whole SCU functionality as well as the different types of power supplies in the field. Thus it is now possible to easily integrate new power converters and the SCU slave cards controlling them in the existing equipment software and to build up test programs quickly.

INTRODUCTION

As GSI is building up the FAIR [1] project and thus doing renovations all over the facility, it was decided to build up a new control system for the accelerator. This is done as a collaboration project with CERN. As part of the new system, the FESA (Frontend Software Architecture [2]) framework deals with all frontend related tasks.

To integrate our numerous hardware designs with the framework, we decided to build up the FESL (Front-End Support Library) as a lightweight approach to implement flexible equipment interfacing.

This paper sketches the workflow developing a FESA class and describes the challenges resulting from it. As a consequence the requirements to the FESL are depicted, followed by the description of the resulting structure of our library. As a last part we discuss the usage of FESL in the context of the FESA framework und give a short outlook to the future development of the library.

USING THE FRONTEND CONTROLLER

As described in [2] the development of a FESA class follows a specific workflow leading to a ready to use equipment software. After designing the class in an XML based document, one can automatically generate a set of C++ source code frames. These frames are filled with specific implementation and are compiled to a ready to use FESA class. In the deploy unit one or more FESA classes are linked with the run-time core to build an x86-Linux executable. This executable can then be delivered to a front-end computer of choice.

The FESA framework is designed to be flexibly tailored to the broad range of equipment in the accelerator, but due to its rather long development cycles, it lacks the flexibility needed during an early development phase. Especially in our case, as we often have several variants of the devices. Writing test software for different power supplies forced us to walk through the aforementioned development process over and over again. Several only slight differences in the equipment behavior, led to an unwanted overhead of work and time.

REQUIREMENTS TO FESL

To cope with the above described limitations, we tried to decouple the equipment specifics from the implementation of the control system specific parts. Thus we came up with several requirements we had for our Front-end Support Library.

First of all it should unify and simplify the usage of the

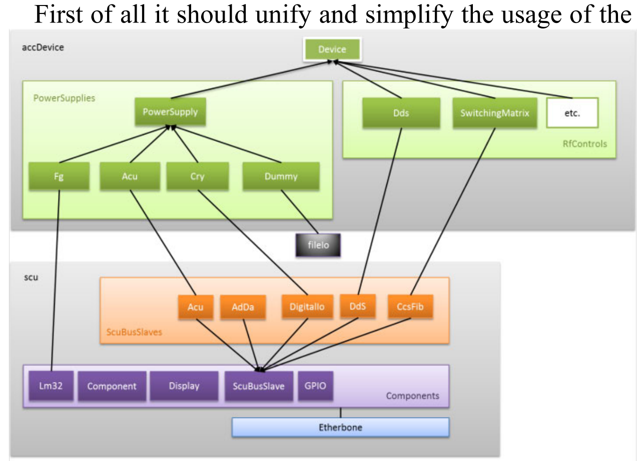

Figure 1: Overall structure of the FESL.

different power converters we are addressing through our front-end computers, namely the SCU (Scalable Control Unit [3]). Being an industry PC with a bus system for interfacing electronics, the SCU is used as the standard equipment interfacing in the FAIR project. Power supplies and other equipment are connected to it using slaves of an internal bus.

ISBN 978-3-95450-146-5

AN OPTICS-SUITE AND -SERVER FOR THE EUROPEAN XFEL

S. Meykopff, DESY, Hamburg, Germany

Abstract

For the European XFEL and the upgraded FLASH facility we require a tool for beam optics calculations. A newly developed software library manages accelerator parameters and performs beam dynamics calculations. In addition a server offers an interface between the library and the control system. A MATLAB interface allows convenient access to the optics server. This framework provides an online model which is integrated in the control system. The online model is used for a simulated European XFEL environment with realistic control interfaces. We use this environment for extensive software developments and tests.

At the FLASH facility a MATLAB toolbox is in use for the calculation of optics parameters [1, 2]. The FLASH2 update adds a second undulator beam line to the facility. The second beam line is not supported in the MATLAB toolbox and must be considered in a successor. Currently the European XFEL is in construction which has up to 5 beam lines [3]. A beam optics tool will be required for commissioning and routine operation. We decided to develop a code for both facilities. This code will provide the accelerator layout in conjunction with the different beam lines. The code calculates transport and response matrices and allows fitting of optics parameters. The matching of beam parameters will be an important feature of the new code. To provide clearly arranged software the new code is developed as a library.

OPTICS LIBRARY

The description of the accelerator is stored in the optics library and will be initialised from a SQlite database [4]. In the case of the European XFEL the SQlite database is generated from an Excel sheet which is distributed by the machine layout coordinators [5]. The sources of the FLASH description are some MAD8 output files [6]. In both cases the generation of the database will be done with MATLAB. The optics library provides these descriptions as static data. The static data includes element names, type names, positions, and covers all columns of the Excel sheet.

The design of the optics library provides multiple setups in the same time. These setups cover variable parameters like steerer magnet angles or k values of quadrupole magnets. The setups are kept separated with a unique setup name. Each setup is independent.

An external code is called for the beam dynamics calculation. Currently the code ELEGANT is in use [7]. To start an ELEGANT run a lattice file and a command is required. The static information and the current setup information are used to generate both files. The output format of elegant is SDDS. The SDDS files were read from the library and stored internally. A dispatch queue collects all ELEGANT runs and executes them parallel. The code of the optics library is written in C. The design of the interface allows one to use it as a shared library.

OPTICS SERVER

The optics server offers a connection to the control system. The TINE interface was chosen as control system because it's in use at both facilities. The optics server provides all functions of the optics library. Every optics server call includes the setup name. This constraint allows a multi-user run with independent setup parameters.

The layer between the control system and the optics library is thin (see figure 1 for details). Mostly parameter checks and conversion is implemented in this layer. A major addition is the observe module. The observe module checks for updates of the RF system or changes of power supply currents. On demand a defined setup will be updated with these values.

We provide a virtual beam position monitor server for tests with simulated orbits. The orbit will be set by a call from the control system. In the optics server is a small push orbit module. This module delivers the simulated orbit to the virtual beam position monitor.

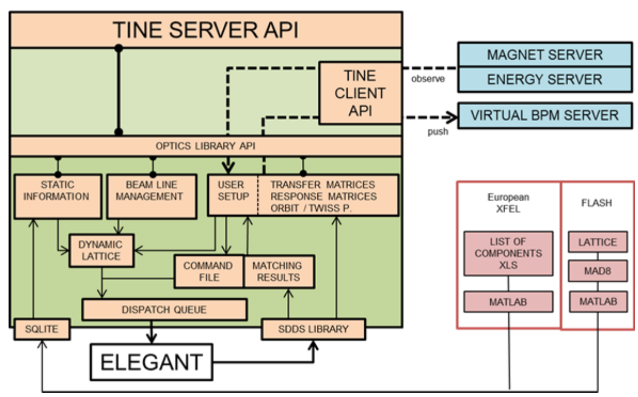

Figure 1: Optics server internals: TINE interface to clients and connections to other servers. Optics library internals with information databases, file generator, job dispatcher, access to SDDS output, and SQLite database as source.

INTERFACE TO MATLAB

In the DESY control room MATLAB is an important tool. The XOptics MATLAB object is a convenient interface to the optics server. The lower interface will be provided by XCOMM. XCOMM is a unified interface for TINE and DOOCS control systems (overview in figure 3) [8]. All functions of the optics server are covered by the XOptics object. MATLAB offers a tabulator expansion for object methods and properties. This feature is a notable simplification for the user of XOptics. Figure 2 shows a code example.

ISBN 978-3-95450-146-5

A UNIFIED MATLAB API FOR TINE AND DOOCS CONTROL SYSTEMS AT DESY

J. Wilgen, S. Meykopff, DESY, Hamburg, Germany

Abstract

At the European XFEL, MATLAB will play an important role as a programming language for high level controls. We present xcomm, a standard MATLAB API which provides a unified interface for TINE and DOOCS control systems. It supports a wide variety of data types as well as synchronous and asynchronous communication modes.

DOOCS AND TINE

The two main control systems at DESY, DOOCS [1] and TINE [2], have been in operation for decades at different accelerators at DESY. The TINE control system, originally created for the HERA collider, is in operation at PETRA, DESY2/LINAC2, and REGAE [3]. The DOOCS control system operates the FLASH accelerator [4]. For the European XFEL, ongoing efforts are being made to integrate both control systems [5].

MATLAB IN CONTROLS

During the past years, MATLAB has become increasingly popular for application development in DESY Control Systems. MATLAB programs are in operation at FLASH, PETRA, and REGAE. At the European XFEL, MATLAB will be used as a major programming language for high level control applications.

Several MATLAB APIs for DOOCS and TINE already exist at DESY, with different scopes and interfaces. Each of them supports only subsets of the data types and the features of the control systems. Some are platform dependent. For the high level controls at European XFEL and FLASH, it was required to have a common, complete and well-tested API with an intuitive and robust interface. Therefore we decided to build a new, unified MATLAB API.

XCOMM FEATURES

Communication Protocol

Xcomm uses the TINE protocol to communicate with both TINE and DOOCS servers. DOOCS servers have already integrated TINE by means of a built-in adaptor [6]. When the adaptor is enabled, a DOOCS server automatically becomes visible in the TINE name space and can be accessed by any TINE client as shown in Fig. 1. Relying on TINE, xcomm can therefore serve as a common MATLAB API for accelerator control systems at DESY. An exception is the Karabo control system [7] of the European XFEL undulator beamlines, which is currently not supported, but is planned to be accessed through a TINE gateway, as far as needed.

Simple Interface

Xcomm is implemented as a MEX (MATLAB Executable) function in the C programming language. It is called as a function which can be controlled by a variable list of parameters, as common in MATLAB. Using a single function interface, programs can send data to or receive data from any control system address. The complexity of the underlying client API is hidden from the user.

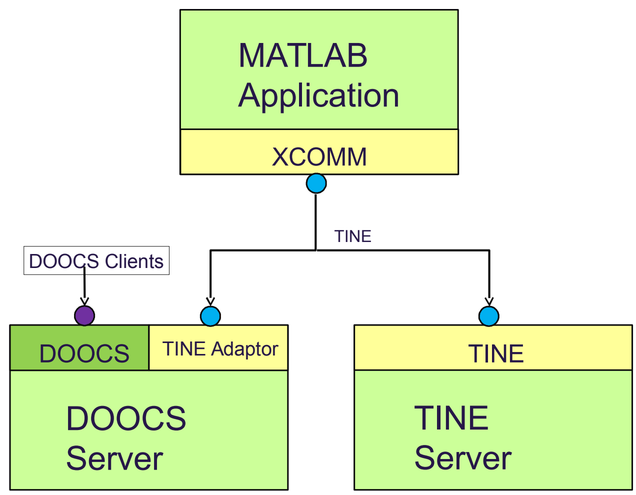

Figure 1: Communication with different servers.

```
result = xcomm('<target address>'
         [,<outputData>]
         [,'optionName',optionValue,...])
```

Both the DOOCS and TINE address styles are supported. The programmer does not have to know if the connected server is a DOOCS or TINE server.

The result of an xcomm call is a structure array which includes an error string, a time stamp in both MATLAB and text formats, and the received data. Since the actual content of the data depends on the request, the data may contain a scalar value, an array, a structure array or a cell array of any of these, dependent on the data type and the number of data units returned. Data sent to the server must be formatted in the same way.

Multiple Data Types

TINE and DOOCS have many pre-defined structure types which xcomm supports. A structure type is mapped to a corresponding structure array in MATLAB.

TINE servers can also have properties with user-defined data types. Xcomm can handle these data types

ISBN 978-3-95450-146-5

VACUUM INTERLOCK CONTROL SYSTEM FOR EMBL BEAMLINES AT PETRA III

A. Kolozhvari, U. Ristau and S. Fiedler
European Molecular Biology Laboratory EMBL, Hamburg Unit, Germany

Abstract

A vacuum interlock control system has been developed for EMBL structural biology beamlines at PETRA III synchrotron. It runs on Beckhoff PLCs and protects instruments by closing corresponding vacuum valves and beam shutters when pressure exceeds a safety threshold. Communication with PETRA III interlock system is implemented via digital I/O connections. The system is integrated in the EMBL beamlines control via TINE and supplies data to archive- and alarm subsystems. A LabVIEW client, operating in TINE environment, provides a graphical user interface for the vacuum interlock system control and data representation.

INTRODUCTION

Three EMBL beamlines for structural biology are in user operation at PETRA III synchrotron on the DESY site in Hamburg. Most of the beamline instruments operate in ultra-high vacuum, and therefore they need an efficient vacuum control system for protection against vacuum incidents. In this paper we describe the second version of the system that is fully integrated into the beamline control environment [1].

CONCEPTS

Controlled Object and Elements

The Vacuum Interlock Control System of EMBL at PETRA III, as any control system, evaluates and changes states of elements constituting the controlled object. The object is a vacuum system of a beamline divided into vacuum sectors, and the elements are vacuum gauges, ion getter pumps, vacuum valves, a front-end and a secondary beam shutter and vacuum sectors.

A sector usually includes X-ray optical instruments operating in ultra-high vacuum, pumps and pressure gauges. It is separated by vacuum valves from the adjacent sectors. A value of pressure in the sector may be obtained from the pressure gauges. Also, if the ion getter pumps are used, the pressure may be calculated from the pump current.

The vacuum valves are the main elements which can be actuated. The control system "by itself" is allowed only to close the valves; they may be opened only by an operator command.

Besides the valves, the control system supplies two permission signals to the Interlock Control System (ICS) of PETRA III, that commands the front-end and the secondary beam shutters of a beamline. In case of withdrawal of the permission signal the ICS must immediately close the corresponding shutter.

The system does not control the ion getter pumps. It only uses data from their controllers for evaluation of states of the sectors.

States of the Elements

A state of an element of the system is a combination of its "physical state" and its control mode. The control mode defines how the system should interact with the element. Three control modes are defined for the system elements: “automatic”, “disabled” and “intervention”.

The physical states defined for the vacuum gauges and ion getter pumps are: “OK”, “Error” and “HV off”.

The physical states of valves and beam shutters are: “open”, “closed”, “undefined” and “error”.

The defined physical states of a sector are: “OK”, “Warning”, “Bad vacuum” - the pressure is above lower threshold, “Very bad vacuum” - the pressure is above ion getter pump shutdown threshold, and “Unknown”.

ALGORITHM

The main task of the vacuum interlock control system is protection of the equipment in case of vacuum incidents. The system continuously monitors the pressure in all sectors of a beamline. If its value exceeds a predefined threshold, the system closes the corresponding valves and isolates the sector.

At each PLC cycle the following actions are performed:

- Reading of the vacuum gauge values and currents and high voltage states of the ion getter pumps. Update of their states and calculation of absolute pressure values.
- Permissions to open beam shutters are set to true. Their values will go to the ICS at the end of the PLC cycle.
- For each sector the system updates its state taking the worst pressure value reading from the gauges, and if they are not available then the one from the ion getter pumps. The system changes a beam shutter permission according to the sector state.
- For each valve, depending on its state and on the states of the connected sectors, the system decides whether the valve must be closed or not and closes the valve if necessary. After that, it checks for an operator command to open/close the valve. If it is possible, the system executes the command and acknowledges it. Otherwise, it sends a negative acknowledgement. If the valve participates in permission to open a beam shutter and if it is either open or disabled, the system keeps the previous value of a shutter permission. Otherwise, it sets the permission to false.

ISBN 978-3-95450-146-5

THE EMBL BEAMLINE CONTROL FRAMEWORK BICFROCK

U. Ristau, A. Kolozhvari and S. Fiedler,
European Molecular Biology Laboratory, EMBL, Hamburg Unit, Germany

Abstract

The EMBL hosts three beamlines at the Petra Synchrotron at DESY. The control of the beamlines is based on a LabVIEW TINE Framework. Working examples of the layered structure of the control software and the signal transport with the Fieldbus based control electronic using EtherCAT will be presented as well as the layout of the synchronization implementation of all beamline elements.

INTRODUCTION

Three EMBL beamlines for structural biology are in user operation at PETRA III synchrotron on the DESY site in Hamburg, Germany. A layered control software architecture and industrial control electronics [1] is implemented and operational on all three beamlines. Key elements are the TINE control system [2], EtherCAT fieldbus electronics [3] and the LabVIEW software package [4]. The control of all backbone instruments of the beamlines is being integrated in one framework called BICFROK (BeamlIneControlFRamewOrK). The framework should provide a graphical representation of the entire beamline to the beamline operator. It should give him access to all instrument control applications in a well arranged manner and should display monitor values of the principal parameters relevant for the synchrotron beamline operation.

CONTROL ELECTRONIC

The control of nearly all instruments is achieved by standardized electronic units. An example of such a unit, the device control for an X-ray focusing mirror set at EMBL beamline P13 for macromolecular crystallography, is shown in Figure 1. The center piece of the units is a Beckhoff CX1030 embedded PC which communicates via an EtherCAT protocol with the Beckhoff EtherCAT hardware modules that are connected to the single devices of the complex X-ray optical instrument. For the sake of standardization, the selection of EtherCAT modules in different control units varies only a little. At present, there are 30 control units distributed in vicinity of the instruments. The local installation reduces the length of the cables from the instrument to the control unit and the problems associated with this. As the units are installed in radiation controlled areas they have to be protected against background X-ray scattering. Radiation

Control Computer and Ethercat Master

All control units are equipped with a Beckhoff CX1030 PC where all control software components are installed. The TINE control system is installed embedded on all of the control electronic masters. Part of the TINE Toolkit installation are the TINE common device interface CDI and the TINE Motor and Scan server. CDI acts as the low level device server which acts on one side as common interface to devices using bus plugs Figure 4 and on the other hand export the devie functionality as TINE properties.

The low level CDI TINE server communicated then with the generic TINE motor and scan server which exports motor control and scan features. It is a component of the TINE toolkit. The PLC runs on the same computer and offers a ADS communications.

Ethercat Installation

The soft real time fieldbus system EtherCat is Ethernet based. EtherCAT allows clock synchronisation of 1µs (+/-20ns). The cycle frequency for PLC's varies between 1 kHz and 10 kHz. Every installation is logically represented as a master installation. The Beckhoff EL-6692 acts for time synchronization between master units. With this concept the whole beamline signals like motor and encoder positions, intensities, vacuum pressure and temperatures monitors are synchronized with respect to each other.

Motor Controller

Currently 150 axis Beckhoff EtherCAT Modules EL7041 for stepper motor control are in operation at the beamline. 150 EL-5101counter acts as motor encoder readout. In operation are 2 servo motor axis controlled by Ax5125 synchron asynchron motor drivers, 24 Attocube ANC350 piezo translations and 12 PI axis piezo motor controller are controlled via Ethercat.

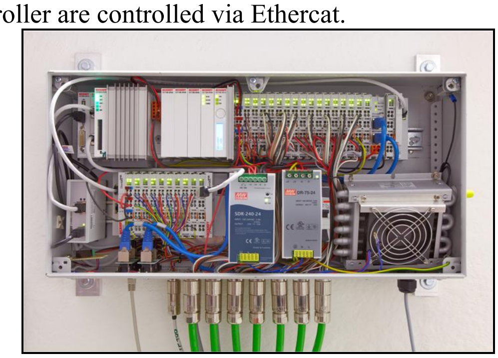

Figure 1: KB mirror control unit of the EMBL Beamline P13.

ISBN 978-3-95450-146-5

STATUS OF THE FLUTE CONTROL SYSTEM

S. Marsching*, N. Hiller, V. Judin, A.-S. Müller, M.J. Nasse. R. Ruprecht, M. Schuh
Karlsruhe Institute of Technology, Karlsruhe, Germany

Abstract

The accelerator test facility FLUTE (Ferninfrarot, Linac-Und Test-Experiment) is being under construction nearby ANKA at the Karlsruhe Institute of Technology (KIT). FLUTE is a linac-based accelerator facility for generating coherent THz radiation. One of the goals of the FLUTE project is the development and fundamental examination of new concepts and technologies for the generation of intensive and ultra-broad-band THz pulses fed by femtosecond electron-bunches. In order to study the various mechanisms influencing the final THz pulses, data-acquisition and storage systems are required that allow for the correlation of beam parameters on a per-pulse basis. In parallel to the construction of the accelerator and the THz beam-line, a modern, EPICS-based control system is being developed. This control system combines well-established techniques (like S7 PLCs, Ethernet, and EPICS) with rather new components (like MicroTCA, Control System Studio, and NoSQL databases) in order to provide a robust, stable system, that meets the performance requirements. We present the design concept behind the FLUTE control system and report on the status of the commissioning process.

INTRODUCTION

FLUTE (Ferninfrarot, Linac- Und Test-Experiment) is a new accelerator facility [1,2] being under construction at the Karlsruhe Institute of Technology (KIT). FLUTE is designed as an accelerator test facility, aiming at studying and improving techniques for producing very short electron bunches and studying the mechanisms involved in the generation of THz radiation from these electron bunches.

In the first stage, the accelerator will consist of a 7 MeV photo injector followed by a 40 MeV linac and a magnetic chicane used as a bunch compressor, as depicted in Fig. 1. Finally, synchrotron radiation generated in the final magnet of the bunch compressor or by an optionally inserted foil after the magnet is coupled out into a THz beamline for use by experiments. This THz radiation can then be analyzed in order to investigate properties of the electron bunch. However, it can also be used for independent experiments.

In order to make systematic studies of the parameters and mechanisms affecting the bunch compression and the generation of THz radiation, a pulse-synchronous data-acquisition system is needed. Such a system has to record key parameters (e.g. accelerator settings, RF pulses, laser pulses, beam charge and profile) for every single electron bunch, so that they can be correlated. At the same time, the control system has to provide live information, that is needed in order to optimize the accelerator operation.

* sebastian<dot>marsching<at>partner.kit.edu

CONTROL SYSTEM DESIGN

The design of the FLUTE control system was driven by the aforementioned demands as well as by the experience with the control system used for the ANKA accelerators.

We chose EPICS [3] as the core control-system framework for several reasons: First, EPICS is already used at many accelerators and thus there is a large number of already existing device drivers, reducing the development costs. Second, the underlying concepts are simple and thus easy to understand. This allows for a short training of new team members, so that they can quickly start working on control-system related tasks. Finally, EPICS is already in use at ANKA, so that we can benefit from our experience.

We use standard off-the-shelf components whereever feasible. This means that most computers are x86_64 systems running a Linux operating system. All systems are connected to a private IP/Ethernet network. The individual network connections use Gigabit Ethernet, however the backbone is already designed for 10 Gigabit Ethernet, allowing for higher data rates in the future.

We use the MicroTCA [4] platform for fast data-acquisition and feedback systems (e.g. the low-level RF (LLRF) system and beam-position monitor readout). This allows us to use the x86_64 platform while having the input/output (I/O) capabilities required for those applications.

For slow control tasks, we use devices with embedded IP/Ethernet controllers or serial interfaces where possible. Other devices are integrated using SIEMENS S7 programmable-logic controllers (PLCs) [5]. We use GigE Vision [6] cameras, allowing for a direct connection to the control system network.

We use Control System Studio (CSS) [7] as the main operator's interface to the control system. CSS is already in use at ANKA and provides an integrated user interface with tools for designing operator's panels, plotting archived data and displaying the alarm status.

DATA ACQUISITION

While the accelerator will initially operate with a repetion rate not exceeding 10 Hz, the data acquisition system is designed for repetition rates of up to 50 Hz to allow for upgrades in the future. This does not mean, that all diagnostics components can operate at this frequency, however the data-acquisition framework is supposed to handle this rate.

In EPICS, each process-variable (PV) sample has a time stamp with nanosecond precision. However, it is hard to synchronize the clocks of different components so accurately, that the time stamp can be used to correlate samples. There are systems like White Rabbit [8] that can provide a sufficiently synchronized clock across different computer

ISBN 978-3-95450-146-5

MAGNET POWER SUPPLY CONTROL MOCKUP FOR THE SPES PROJECT

M. Giacchini, M. Montis, M. Contran, INFN – Legnaro, Padova, Italy
M.A. Bellato, INFN- Padova, Padova, Italy

Abstract

The Legnaro National Laboratories employs about 100 Magnet Power Supplies (MPSs). The existing control infrastructure is a star architecture with a central coordinator and ethernet/serial multiplexers. In the context of the ongoing SPES project, a new magnet control system is being designed with EPICS[1][2] based software and low cost embedded hardware. A mockup has been setup as a test stand for validation. The paper reports a description of the prototype, together with first results.

INTRODUCTION

To improve the reproducibility and stability of the magnet control system used at LNL to the accelerators, the previous control system should be upgraded to a new one. The control system chosen for this task and to all site was EPICS. The previous system was based on terminal servers. The power supplies were connected to the terminal server which exports the serial interfaces to the central control workstation server. On this one a C program runs to manage the communications, and makes available the Graphical User Interface (GUI) to the operators.

This start architecture force the user to use only one GUI operator interface per time. The communication interface itself realize sub-groups concentrates them on terminal servers which relay on a serial interface converters; this makes, suddenly, communication problems in case of faults of only one power supply on the chain.

Basing on this experience and the strong EPICS architecture, the new system should have a great granularity, where, the communication shouldn't be concentrate, either the GUIs could be duplicated and moved without effort.

HARDWARE CONFIGURATION

The Input Output Controller (IOC) selected to this application is a low cost embedded PC (Figure 1). The power requirements necessary to this object is low enough to use the Power Over Ethernet technology. This makes extremely cheap the cabling, but requires the necessary power on the Ethernet switch. We are evaluating various kind of embedded PC which could play this role, the final decision has not been done, but the tests already done seems we have made a good technology selection.

The power supply of the magnets can use two kind of serial communications: RS232, RS422. The previous system has used the second one. The pros of that choice were the reduction of the amount of the lines, the reduction of the cabling which means a final reduction of the total cost; but the cons was quite bad, if someone of the serial controller of the power supplies on the line goes on communication fault state all the controllers on that lines are not controllable until the fault was fixed.

The new system makes all the communication independently using the RS232 connection, which realize a point to point connection between the controller and the embedded PC. Later, if this should be the case, we move back to the RS422, which is more strong in front of noises, but in any case we would use the point to point connection, to make the system robust as much as we can, minimizing communication delay.

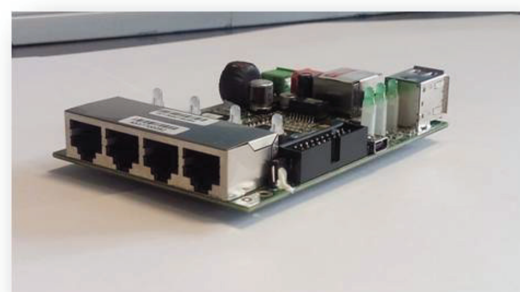

Figure 1: Example of embedded used as IOC for the magnet system.

SOFTWARE

The core of the software, the device communication was accomplished by using ASYN[3]. A basic version used on other site has been modified and customized to fit the requirements of our site. More works has requested the development of the GUI operator interface, based on Control system Studio (CSS), as visible in Figure 2.

After 20 years of operation on the old magnet system the operator itself, for a first time in the history of our laboratories, has been trained to design and realize the skeleton of that GUI. Themselves feels comfortable using a GUI designed by them, that is going to make the change of the control system more smooth.

ISBN 978-3-95450-146-5

IFMIF EVEDA RFQ LOCAL CONTROL SYSTEM TO POWER TESTS

M. Giacchini, M. Montis, L. Antoniazzi, INFN-LNL, Legnaro, Italy

Abstract

In the IFMIF EVEDA project, normal conducting Radio Frequency Quadrupole (RFQ) is used to bunch and accelerate a 130 mA steady beam to 5 MeV. RFQ cavity is divided into three structures, named super-modules. Each super-module is divided into 6 modules for a total of 18 modules for the overall structure. The final three modules have to be tested at high power to test and validate the most critical RF components of RFQ cavity and, on the other hand, to test performances of the main ancillaries that will be used for IFMIF EVEDA project (vacuum manifold system, tuning system and control system). The choice of the last three modules is due to the fact that they will operate in the most demanding conditions in terms of power density (100 kW/m) and surface electric field (1.8*Ekp). The Experimental Physics and Industrial Control System (EPICS) environment [1] provides the framework for monitoring any equipment connected to it. This paper reports the usage of this framework to the RFQ power tests at Legnaro National Laboratories [2,3].

INTRODUCTION

The RFQ Local Control System (LCS) Architecture approved by the IFMIF-EVEDA Collaboration is designed to optimize reliability, robustness, availability, safety and performance minimizing all the costs related to it (purchase and maintenance). Following this philosophy and the IFMIF-EVEDA Guidelines, we realized a control system network composed by two different kinds of hosts:

- Physical machines for critical control system tasks;
- Virtual hosts in machines where no particular functional task or hardware is required.

The architecture realizes the 3-layer structure described in the Guidelines and each layer defines a proper hosts group (equipment directly connected to the apparatus, control devices, Human-Machine Interface) while the EPICS framework provides the interface between them.

LCS CORE SYSTEM

The core workstation server provides capabilities that enable controls engineers to deploy customized environments for their application perfectly aligned with the "Common Software Guidelines". All the regular EPICS services, are backed up regularly and can be moved and cloned easily. Key enabling technology for this is the virtualization and the provisioning; this approach allows the installation saves floor space, power and cooling.

In the IFMIF-EVEDA RFQ LCS, Logical Volumes are used to define main server's partition table and the virtual hard disks used to realize virtual hosts. In this way, it is possible to manage any resourceful saturation in according to the free resources provided by the main server. Because of the role covered by the server, all the unnecessary services and ports are switched off following an hardening policies to keep safe as much as possible the control system.

Archiver

The Channel Archiver is an archiving tool-set for EPICS based control systems. It can archive any kind of record available through the EPICS Channel Access. The deployable Archiver prepared into the manager server is ready to use. The retrieving interface is based on Control System Studio (CSS) framework. After the commissioning, this archiver will be switched off and replaced from the central one available from the Central Control System (CCS).

Figure 1: RFQ LCS Racks for Power Tests.

Deploy and Backup

In any control network, cases of hardware failure or breakdown can be very dangerous, especially in the case of infrastructure similar to the IFMIF facility. The need to restore the controls functionality in the shortest time is therefore fundamental. The manager machine is designed to realize an automated management for new machinery's configuration inside the RFQ LCS. In particular, it is possible to connect a new device to the network, indicate which type of play rule it must realize and the network itself will auto-configure it. The manager host uses dedicated open source software for provisioning the entire control system network, while backup service saves

ISBN 978-3-95450-146-5

UPGRADE OF BEAM DIAGNOSTICS SYSTEM OF ALPI-PIAVE ACCELERATOR'S COMPLEX AT LNL

M. Giacchini, G. Bassato, M. Poggi, M. Montis, INFN/LNL, Legnaro, Italy
B. Liu, CIAE, Beijing, China

Abstract

The beam diagnostics system of ALPI-PIAVE accelerators has been recently upgraded by migrating the control software to EPICS[1]. The system is based on 40 modules each one including a Faraday cup and a beam profiler made of a couple of wire grids. The device's insertion is controlled by stepper motors in ALPI and by pneumatic valves in PIAVE. To reduce the upgrade costs the existing VME hardware used for data acquisition has been left unchanged, while the motor controllers only have been replaced by new units developed in house. The control software has been rebuilt from scratch using EPICS tools. The operator interface is based on CSS; a Channel Archiver based on PostgreSQL[2] has been installed to support the analysis of transport setup during tests of new beams. The ALPI-PIAVE control system is also a bench test for the new beam diagnostics under development for the SPES facility, whose installation is foreseen in middle 2015.

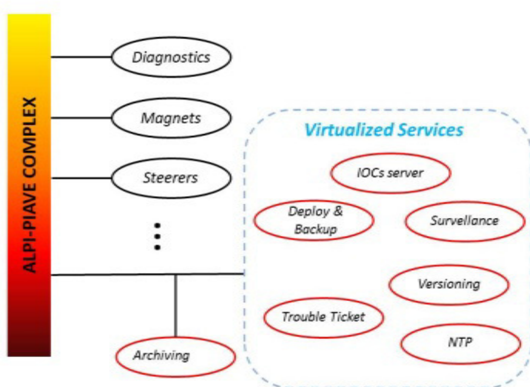

Figure 1: PIAVE-ALPI complex.

INTRODUCTION

Beam diagnostic system is a critical task for managing the accelerator complex, because it is the first stage where scientists and operators understand machine status and behaviour and execute the operations required to provide the desired beam.

Diagnostics system will be a part of the control system architecture under implementation at INFN/LNL Laboratory (Figure 1). One of the greatest improvement coming from the assumption of EPICS as main control system framework is the creation of a distributed environment where every sub-system is integrated and interconnected with others. As consequence, operators and scientists can have an easier and more direct control to the entire apparatus. The upgrade related to the diagnostics system is realized focusing in two aspects: the reuse, as much as possible, of the hardware already installed in ALPI (to contain costs) and the integration into the EPICS environment. First development stage was realized in 2011[3].

DIAGNOSTICS SYSTEM ARCHITECTURE

In principle the ALPI-PIAVE diagnostics environment is composed by 48 diagnostics boxes installed along the apparatus containing, each one, a faraday cup and a couple of grids (horizontal and vertical) where every single grid is made of 40 wires. To avoid white noise inducted by cables, current signals are converted to voltage next to the diagnostics box. After the conversion, grid signals are multiplexed and serialized before the transport to the acquisition system composed by an ADC card installed in VME crates. The multiplexer is driven by a counter whose clock is generated by the ADC itself, to have the signal transmission synchronized with the conversion. At the beginning, the software layer used for this sub-system was based on C programs developed on VxWorks OS for the VME systems and custom java application developed inside the INFN for the Human-Machine Inteface (HMI).

With the focus of renovate and upgrade the control system software, EPICS was chosen as control system framework for the new environment, bringing all the advantages coming from this solution (performances, data distribution, etc.). As consequence, the actual set of programs and tools available to control the ALPI-PIAVE complex has been substituted piece by piece with the new EPICS application.

According to EPICS documentation, VxWorks is supported by EPICS since its origin; therefore VME systems are not completely changed: the VME processor implements an Input/Output Controller (IOC) providing the acquisition of grid signals and faraday cups. Databases loaded in the IOC are not very complex, due to simplify the maintenance, and provide minimal processing on raw data. Signal from beam profilers and faraday cups are acquired by XVME566 board produced by XYCOM: this kind of hardware provides 12 bit resolution and support a conversion rate of 100 KHz in streaming mode. As indicated in Figure 2, a dedicated device driver realized by M. Davidsaver realizes the software interface between the field and the EPICS server host.

ISBN 978-3-95450-146-5

STARS: CURRENT DEVELOPMENT STATUS

T. Kosuge, Y. Nagatani, KEK, Tsukuba, Japan

Abstract

Simple Transmission and Retrieval System (STARS) is a simple and useful software for small-scale control systems running on various operating systems.

STARS is used by various systems at the KEK Photon Factory (e.g., the beamline control, experimental hall access control, and key handling systems), and the development of STARS (e.g., development for many kinds of STARS clients and for interconnection of Web2c and STARS) is ongoing. We describe the current development status of STARS.

STARS OVERVIEW

Simple Transmission and Retrieval System (STARS) [1] consists of client programs (STARS clients) and a server (STARS server) program. Figure 1 shows some STARS configuration examples. Each STARS client is connected to the STARS server via a TCP/IP socket and communicates using text-based messages. Every STARS client has its own node name, which is unique in the system; a message with a destination node name sent to the STARS server by a client is delivered to the corresponding STARS client by the STARS server. STARS uses this simple message transfer mechanism to provide control system functionality.

Figure 1: Layout example of STARS server and clients.

Multi-platform

The STARS server-side application is written in Perl and can run on various operating systems (e.g., Windows, Macintosh, and Linux). Development of a control system using STARS consists primarily in writing STARS client programs. The developer can choose his or her favorite operating system or programing language, if they support TCP/IP Socket and the handling of text string functions.

Simple Certification

The STARS server contains a simple certification procedure performed at STARS client connection time. STARS avoids client misconnection through three function steps, as follows:

- Host name certification
- Node name and keyword certification
- Node name and host name certification (optional)

Interface Libraries

STARS interface libraries (for Perl, .NET, Java, C, and ActiveX) assist in STARS client development. Developers need not be skilled in TCP/IP Socket or in the Node name and keyword certification procedure to use a STARS interface library.

INSTALLATION STATUS

STARS is used as a common beamline control system at the Photon Factory for introducing various systems. Table 1 lists the recent installation status of STARS at KEK.

Table 1: STARS Installation Status (as of September 2014).

Category	Beamline or Control System
PF-2.5GeV Ring X-ray	BL-1A, BL-3A, BL-4B2, BL-5A, BL-6A, BL-6C, BL-7C, BL-8A, BL-8B, BL-9A, BL-9C, BL-10A, BL-10C, BL-12C, BL-14A, BL-15A, BL-17A, BL-18B
PF-2.5GeV Ring VUV and Soft X-ray	BL-2A, BL-2B, BL-11A, BL-11B, BL-13A, BL-16A, BL-19B, BL-20A
PF-AR	NW-1A, NE-3A, NW-2A, NW-10A, NW-12A, NW-14A
Slow Positron Facility	SPF-A, SPF-B
Other Systems	Beamline Interlock Central Control System, Experimental Hall Access Control System, Key Handling System, Front-End Monitoring System, Radiation Monitoring System

CLIENT DEVELOPMENT

STARS contains three types of client programs (User, I/O, and Others). Users can add control system functions by adding STARS client programs without system stoppage.

ISBN 978-3-95450-146-5

DEVELOPMENT AND APPLICATION OF THE STARS-BASED BEAMLINE CONTROL SYSTEM AND SOFTWARE AT THE KEK PHOTON FACTORY

Y. Nagatani, T. Kosuge, KEK, Ibaraki, Japan

Abstract

The Simple Transmission and Retrieval System (STARS) [1] is a message transferring software for small-scale control systems, originally developed at the Photon Factory (PF). It has a server-client architecture using TCP/IP sockets and can work on various types of operating systems. Since the PF adopted the STARS as a common beamline control software, we have developed a beamline control system that controls optical devices such as mirrors and monochrometers. We also developed various systems and software, such as the information delivery system of the PF ring status based on the STARS and Three-fold Integrated Networking Environment (TINE) [2], and measurement software based on the STARS, for the PF beamlines. Currently, many useful STARS applications are available, such as device clients, simple data acquisition, and user interfaces. We will describe the development and installation status of the STARS-based beamline system and software.

OVERVIEW OF STARS

The Simple Transmission and Retrieval System (STARS) consists of a server program, the *STARS server*, and client programs, the *STARS clients*. Each client is connected to the STARS server through a TCP/IP socket, and it communicates via text-based messages. Each client program has its own unique node name, and it sends text-based messages using the destination node name to the server, which then delivers the messages to the destination client. A STARS server, which is written in Perl, can run on various operating systems. STARS users can upgrade the system by writing client programs, and STARS clients can participate in the system at any time without system stoppage.

STARS MESSAGE STANDARDIZATION

Originally, the STARS had the following simple rules for messages:

- A message that starts with "@" (e.g., @message) is a reply to a command.
- A message that starts with "_" (e.g., _message) is an event.
- Any other type of message is a command.

For adopting the STARS as a common beamline control software, moreover, we standardize the STARS messages that control the device components including such as mirrors and monochrometers. Table 1 shows commonly used standardized STARS messages.

Table 1: Commonly used Standardized STARS Messages

"GetValue"	STARS command to get the target value.
"SetValue"	STARS command to change the target value.
"IsBusy"	STARS command to get the moving status of the target.
"_ChangedIsBusy"	STARS event to notify the change in moving status of the target.
"_ChangedValue"	STARS event to notify the change in the target value.

The standardization of STARS messages enables reusable beamline control programs and software to be developed. In addition, this standardization allows anyone using the same message format to control these baseline programs and software.

Further, the standardization of STARS messages reduces developmental costs associated with the creation of control programs; moreover, it enables beamline users to develop their own control programs.

BEAMLINE CONTROL USING STARS

Installation Status

Since the Photon Factory (PF) adopted the STARS, a common beamline control software, the development and installation of STARS-based beamline control systems have been increasing every year. Recently, more than 30 beamlines have introduced STARS-based beamline control systems (see Table 2).

Table 2: Installation Status of STARS-based Beamline Control Systems (October 2014)

Category	Installed beamline
2.5 GeV Ring X-ray	BL-1A, BL-3A, BL-4B2, BL-4C, BL-5A, BL-6A, BL-6C, BL-7C, BL-8A, BL-8B, BL-9A, BL-9C, BL-10C, BL-12C, BL-14A, BL-15, BL-17A, BL-18B, BL-18C
PF-2.5 GeV Ring VUV and Soft X-ray	BL-2, BL-11A, BL-11B, BL-11D, BL-13A/B, BL-16A, BL-20A, BL-28
PF-AR	NE1A, NE3A, NE7A, NW2A,

ISBN 978-3-95450-146-5

RENOVATION OF PC-BASED CONSOLE SYSTEM FOR J-PARC MAIN RING

S. Yamada*, KEK, Sokendai, J-PARC, Ibaraki, Japan

Abstract

Console system for J-PARC Main Ring (MR) was designed in 2007 and had been used for accelerator commissioning and operation since then. It was composed of 20 diskless thin clients and 10 blade servers. Both of them are PC-based computers running Scientific Linux (SL) 4 as their operating system. In 2013, migration to ordinary fat clients was triggered by update from SL4 to SL6. Intel NUC was selected based on result of preliminary investigation and use experiences of those thin clients. Its evaluation was carried successfully out during commissioning of MR. Thin clients were replaced by 22 NUCs that work as fat clients. Migration scenario and technique of managing console system are discussed.

INTRODUCTION

Performance improvement of computers and its price reduction are quite remarkable in recent years. The performance of a PC as small as palm-size surpass that of a server computer 5 years ago. In this paper, renovation of console computers for J-PARC MR which would withstand next 5 years is described.

The accelerator control system for MR was constructed in 2007 [1] using EPICS [2] and SL4 [3] as its control software framework and operating system, respectively. It has been operational since beam operation of MR started [4] in May 2008. GUI applications of the control system were designed and implemented using EDM and MEDM, which require X Window System. In order to achieve load balancing, the control system consists of thin clients which work as X terminals and blade serves to run those applications. HP Compaq t5720 and t5730 were introduced as thin client. They are diskless machines which boot from network using an image file common to all of them so that manageability would be advantageous. IBM BladeCenter HS20 and HS21 were introduced as blade server.

RENOVATION OF CONSOLE SYSTEM

Performance of MR has been steadily improving during 6 years of beam operation. At the same time, number of blade servers was increased [5] as functionality of applications for accelerator control became rich. However, the operation method was unchanged since 2007, which uses thin clients to display the GUI and blade server to the CPU.

Model Selection and Evaluation

In 2012 support for SL4 was ended. This triggered selection for new console computers that would withstand next 5 years. An Intel NUC DC54327HYE running 64-bit SL6 was evaluated in commissioning of MR from December 2013 to January 2014.

* shuei@post.kek.jp

Figure 1: Photograph of HP Compaq t5730 (left), Intel NUC (right), and a D Battery for Size Comparison (center). T5720 is not shown but its appearence is almost same as t5730.

Following points are taken into account in the selection based on accelerator operational experience:

- Sufficient CPU power and amount of memory.
  Increase of demands to use heavy applications such as web browsers, PDF viewers, and office suites led to high load on blade servers. It is also foreseen that even more CPU power and memory are required in near future as GUI environment for EPICS will transfer from EDM and MEDM to Control System Studio (CSS) [6] which is becoming mainstream. Hence, it was considered that run applications locally on a console computer rather than remotely.
- Compatibility with SL6.
  Operating systems for blade servers have been migrated from SL4 to SL6 and EPICS I/O Controllers (IOCs) [7] underway. Using identical operating system for console will simplify its management and enables smooth transition.
- GPU support by the operating system.
  Matrox EpicA TC2 / TC4, which are graphics chips of previous thin clients, are not supported by SL. Proprietary device driver from the manufacture is available for SL4 and SL5, but update was stopped in 2009. No driver is available for SL6. Next model will be used for accelerator operation over next few years. Therefore a model was selected such that its GPU is supported by the operating system out of the box.

ISBN 978-3-95450-146-5

CONTROL SYSTEM OF TWO SUPERCONDUCTING WIGGLERS AND CONPENSATION MAGNETS IN THE SAGA LIGHT SOURCE

Y. Iwasaki#, SAGA Light Source, Tosu, Japan

Abstract

The SAGA Light Source (SAGA-LS) is a synchrotron radiation facility. Three insertion devices: a 4 T superconducting wiggler, an APPLE-II undulator, and a planar undulator, are used for synchrotron radiation experiments. For the further demand of hard x-ray experiments, we are investigating to install a second superconducting wiggler. To compensate strong tune shift caused by the superconducting wiggler, a part of the quadrupole magnets families will be excited by independent power supplies. We developed the control system of the storage ring power supplies for the superconducting wiggler. The PLC used for control of the new quadruple magnets power supplies are linked by optical fiber cable to synchronize each main power supplies in the storage ring. We tested the synchronicity of the PLC sub unit for the new quadrupole magnets in the test bench.

SAGA LIGHT SOURCE

SAGA-LS is a synchrotron radiation facility consisting of a 255 MeV injector and a 1.4 GeV storage ring [1]. The electrons from the linac are injected into the storage ring, and ramped up to 1.4 GeV in 4 minutes. The user mode operation for synchrotron radiation experiments with a stored current of 100 mA has been performed since 2006 [2], and the further developments of the accelerator have been made [3]. To meet the needs of the hard x-ray experiments a 4 T superconducting wiggler was installed in 2010 [4,5]. We perform the accelerator operation at the maximum stored current of 300 mA at this stage. Fig. 1 shows the layout of the accelerator complex of the SAGA-LS and beam lines. For the further requirements of the hard x-ray experiments, we began to investigate a second 4 T superconducting wiggler. The control system of the second superconducting wiggler and compensation magnets power supplies system are developed.

CONFIGURATION OF THE POWER SUPPLIES

The configuration of the quadrupole magnet family (QF1 Family) is shown in Fig.2. To correct of the strong tune shift (Horizontal: -0.031, Vertical: 0.068 measurement result [5]) caused by the superconducting wiggler, a part of the quadrupole magnets are excited by the independent power supplies. The lattice of the SAGA-LS storage ring is 8 symmetry Double Bend type, and there are 3 quadrupole families (QF1, QD1, and QF2 families). A pair of quadrupole doublet (QF1 and QD1) is used for the tune correction. The new quadruple magnets power supplies for the tune correction must work simultaneously to the main power supplies of the storage ring magnets during the ramp up period, because the new quadrupole power supplies constitute a part of the main quadrupole magnets families of the storage ring.

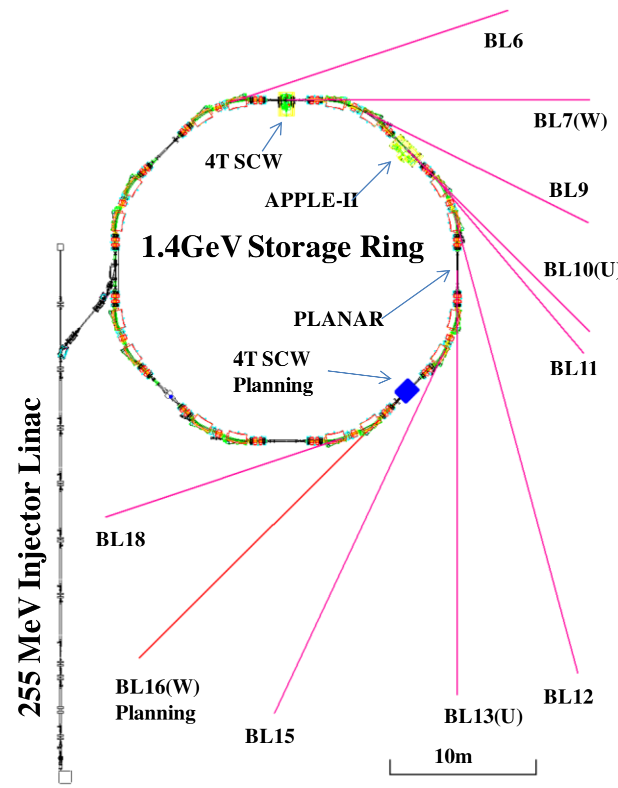

Figure 1: Layout of the SAGA Light Source.

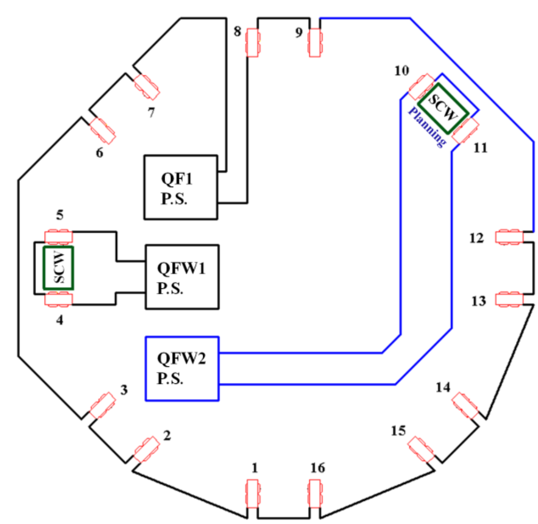

Figure 2: Configuration of QF1 Power Supplies Family. The blue lines indicate modifying section.

#iwasaki@saga-ls.jp

ISBN 978-3-95450-146-5

PERSONNEL SAFETY SYSTEM IN SESAME

M. Mansouri Sharifabad, I. Saleh, A. Ismail, SESAME, Allan 19252, Jordan

Abstract

SESAME (Synchrotron-light for Experimental Science and Applications in the Middle East) is a "third-generation" synchrotron light source under construction in Allan, Jordan. Personnel Safety System (PSS) in SESAME restricts and controls the access to forbidden areas of radiation. The PSS is an independent system which is built on Safety PLCs. In order to achieve the desired Safety Integrity Level which is SIL-3, as defined in IEC 61508, several interlocks and access procedures have been implemented in the system fulfilling characteristics such as fail-safe, redundancy and diversity. Also a system meant for monitoring and diagnostics of PSS is built based on EPICS and HMI. PSS PLCs which implement interlock logic send all the input and output bits and PLC status information to EPICS IOC which is not an integral function of PSS operation. This IOC will be connected to other control system's IOCs to send informative signals describing the status of PSS to the main control system in SESAME. In addition, 5 combined Gamma-Neutron radiation monitors which are distributed around and over the booster area send interlock signals to personnel safety system.

INTRODUCTION

Personnel Safety System is an independent access control and interlock system which interlocks accelerator operation and controls access to shielded enclosure to prevent personnel from exposure to high level of radiation forbidden by the law. PSS is designed based on the implementation of IEC61508 standard for programmable safety systems. In SESAME the safety Integrity Level of PSS interlocks is SIL-3. In addition, redundancy and diversity techniques have been applied to increase the reliability of these safety interlocks. For example to inhibit the operation of Microtorn, the safety permissions will be removed from two different parts of the system; Microtron Trigger and Microtron High Voltage Power Supply. The first phase of SESAME PSS controls the Booster tunnel which contains Microtron and Booster ring. Following the adoption of IEC 61508 Standard, SESAME decided to use Allen Bradley GuardLogix® controller which is a dual processor solution that uses a primary controller and a safety partner to achieve SIL-3 [1,2].

PSS DESIGN PRINCIPLES

Based on the experience from other synchrotrons the following design principles have been taken into account:

• PSS must be failsafe and PSS safety interlocks must dump the beam in case of any emergency status so that the safety integrity level equals SIL-3.

• Emergency stop and search buttons must be installed in the area covered by PSS, they need to be easily accessible, clearly labelled and distinguishable.

• Audible and visual warnings should be provided prior to accelerator operation and status indicators should be present to reflect the actual conditions of the machine.

• PSS needs to be a totally independent system and all PSS cables run in dedicated conduits and cable trays which are not shared with any other system [1,3].

PSS SAFETY FUNCTIONS

The PSS has four main states, OPEN; which means there is free access to Booster tunnel, INTERLOCKED; after the search patrol has been completed, RESTRICTED and SECURED.

Booster tunnel is considered cleared of personnel when the search patrol is completed. The search process starts upon generation of Start Search command in PSS cabinet in control room. Search buttons distributed all around the Booster tunnel should be pressed in sequence within certain time limit based on the search path designed by the safety officer. During the search patrol audio warning is broadcasted by the sound system, asking the personnel to leave the tunnel [1,3].

The restricted access function is foreseen to provide short visits of personnel to the tunnel without the need for a new search. Like the Search patrol, the restricted access needs to be permitted by control room. After taking one of the personal keys the person is authorized to enter the Booster tunnel during RESTRICTED access mode (up to two people). In order to return back to INTERLOCKED state all personal keys must be in place on PSS cabinet door, tunnel door locked and the finish RESTRICTED access permitted by control room. The SECURED state exists when PSS is in INTERLOCKED state and the keys on the PSS cabinet in control room are in the position that allows the trigger of Microtron [1,3].

The tunnel door is locked by an electromagnetic lock and its status is monitored by a magnetic switch (SIL-3) and the feedback from electromagnetic lock [1,3].

In emergency cases (e.g. an individual is trapped in the Booster tunnel) they should press one of the red Emergency Stop buttons, distributed around the tunnel and on Booster PSS cabinets. This will immediately trip the interlock system, unlocks the Booster tunnel and stops the injection process [1,3].

To monitor the radiation level in service area, Booster roof and the inner ring of the Booster (pool), five combined neutron-gamma radiation trolleys (by Thermo Fisher) are distributed in the mentioned zones. Three digital signals are continuously provided by each trolley to PSS, monitoring the status of radiation level and error signals. PSS will react accordingly to each of these signals [1,3,4].

ISBN 978-3-95450-146-5

CLIENTS DEVELOPMENT OF SESAME'S CONTROL SYSTEM BASED ON CSS

A. Ismail, I. Saleh, M. Mansouri, Y. Dabain, SESAME, Allan, Jordan

Abstract

SESAME is a third generation synchrotron light source under construction near Amman (Jordan). It is expected to begin operation in 2016. SESAME's injector (Microtron) and pre-injector (Booster Ring) have been commissioned. Commissioning of the storage ring is expected in 2016. The control system at SESAME is based on EPICS. EPICS IOC's are used for the servers. Control System Studio (CSS) is used for the clients. CSS BEAST alarm handler is used to identify all the critical alarms of the machine including configuration and visualization. This paper presents the architecture and design of the CSS BOY graphical user interfaces (GUIs) and CSS BEAST alarm handler for the different subsystems. It presents the standards followed in the development of SESAME's clients. SESAME will use an archiving tool based on CSS to access process variable history.

INTRODUCTION

SESAME consists of a 22 MeV Microtron, an 800 MeV Booster Synchrotron and a 2.5 GeV Storage Ring.

Control System Implementation uses (EPICS) base R3.14.12. Servers are implemented as EPICS Input/output Controllers (IOCs). Clients are implemented using a custom build of Control System Studio (CSS) based on V.3.16. Siemens S7 PLC controllers are used for the machine interlocks. An Allen Bradley PLC controller is used for the Personal Safety System (PSS). VME hardware is used for the timing system. Development and administration platforms use Scientific Linux 6.4. A Git version control is used to track development and documentation. All clients, servers, and controllers are connected to an isolated machine network. There are twelve virtual servers are reserved to run the IOCs, archive system, alarm system and Git repositories.

The control systems have been implemented for the Microtron, Transfer Line 1 (TL1) and Booster. The Booster's control system is divided into seven subsystems: vacuum, power, RF, diagnostics, cooling, timing and Personal Safety System (PSS). Each control subsystem consists of one or more clients, servers, and controllers. This paper will focus on the design and implementation of SESAME's clients based on CSS.

CUSTOMISED CSS

Control System Studio (CSS) is a combined effort of several parties, including DESY (Hamburg, Germany) and SNS (Oak Ridge, TN). It provides a collection of control system tools in a common environment, based on Eclipse [1]. Control System Studio (CSS) is designed to serve as an integration platform for engineering and operation of today's process controls as well as machine controls systems. CSS is a complete environment for the control room covering alarm management, archived data displays, diagnostic tools and last but not least operator interfaces [2].

A custom build of Control System Studio (CSS) based on V.3.16 has been implemented in SESAME (Fig. 1). Archive system and alarm handler plugins have been integrated to the CSS custom build as they were not included in the basic build. A thumbwheel function has been added to the input box widget, which provides a compact fine-tunable control to the set values and it replaces the old thumbwheel widget (Fig. 2). A digit of a set value is marked and changed by using the arrow keys of the keyboard. The custom build also includes a new feature for showing the CSS screens name on the title bar of each screen. Another feature has been added when multiple CSS windows are opened, only one instance of identical windows is allowed at a time.

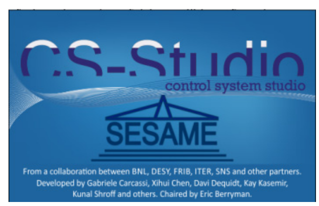

Figure 1: CSS Start Screen.

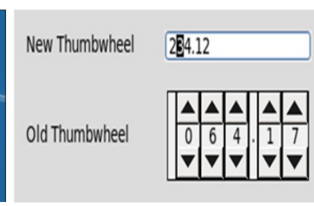

Figure 2: New Thumbwheel.

OPERATOR INTERFACES

SESAME uses CSS BOY (Best OPI, Yet). It is an OPerator Interface (OPI) development and runtime environment which enables monitoring and controlling of an EPICS system. It has a modern graphical editor and a modern web browser style runtime. It is dynamic via rules or scripts and it has comprehensive types of widgets. SESAME has two environments of CSS; one is used for development in which OPIs can be edited and modified, and the other is used for deployment which disables modification and hides all the development windows of eclipse. SESAME's client development is maintained and tracked using a version control system called Git. Git is a distributed revision control and source code management system [3]. Clients' repository contains CSS core build and CSS workspaces that contains all the developed OPIs held in main folders. SESAME OPIs are arranged and classified depending on the machine main parts; there are three main folders which contain all the required OPIs for the Microtron, TL1and Booster control systems. The booster control system is divided into seven subsystems; each one has its own sub-folder and OPIs for the control

ISBN 978-3-95450-146-5

THE APPLICATIONS OF OPC UA TECHNOLOGY IN MOTION CONTROL SYSTEM

Min Wang#, Hong Luo, Min Li, Jinmei Dong, Ruishi Mao, Tiecheng Zhao,
Institute of Modern Physics, Chinese Academy of Sciences,
Lanzhou 730000, China

Abstract

The establishment of data model is more abundant based on OPC UA (Unified Architecture) technology, which has good platform independence and high reliability. Thus it becomes a new direction in the field of data exchange of industrial control. In the paper, the motion control model based on redundant ring network is built by using NI 3110 industrial controller and servo motors. And the data structures used in parallel communication between the upper computer and multi-terminal motors are designed by using OPC UA technology. So the problem of inconvenient data exchange between the RT system of lower controller and the Windows system of upper computer may be solved.

INTRODUCTION

Motion control is widely used in modern industrial automation. The design is based on the application of the heavy ion accelerator in Lanzhou, China, which is firstly used in motion control of four beam diagnostic detectors. OPC UA (OPC Unified Architecture) technology with the advantages of good platform independent, security and so on, has become the main means to the data interaction in present industrial control, and is widely used in distributed control system.

THE ADVANTAGES OF OPC UA TECHNOLOGY

The traditional OPC specification is based on the COM/DCOM technology of Microsoft, which cannot meet the requirements of modern industrial automation in the aspect of interoperability, security, reliability and so on [1]. So OPC foundation released the newest unified method of data communication, namely OPC UA, which not only covers OPC DA, OPC HDA, OPC A&E, OPC security, but also expands many new functions [2].

Compared with the traditional OPC technology, OPC UA has the following advantages:

- Unified access approach, which means it can provide a unified address space and service model, but also has the function of semantic recognition, solves the problem that is the same information cannot be accessed in a unified way.
- Reliability, which means adjustable timeout can make the mistakes found and corrected, all this makes us deal with errors and failures of communication more easily.
- Security, which means the technology of underlying communication used for message transfer between the applications based on OPC UA provides the functions of encryption and marking.
- Independency of platform, which means OPC UA technology based on the SOA (Service-Oriented Architecture) of Web Service makes the application development no longer dependent on any particular operating system.
- Relevance of data, which means OPC UA provides the correlation functions of data nodes, rather than puts the node as a single data point [3].

#wangmin@impcas.ac.cn

THE ARCHITECTURE OF MOTION CONTROL SYSTEM

The motion control system is used for the motion control of beam diagnostic detectors of the cyclotron. The beam diagnostic detectors contain one SS (scintillation screens), one FC (faraday cup) in LEBT (low energy beam line), and another FC, one scraper in MEBT (medium energy beam line), so five servo motors are needed for the motion control (see Fig. 1).

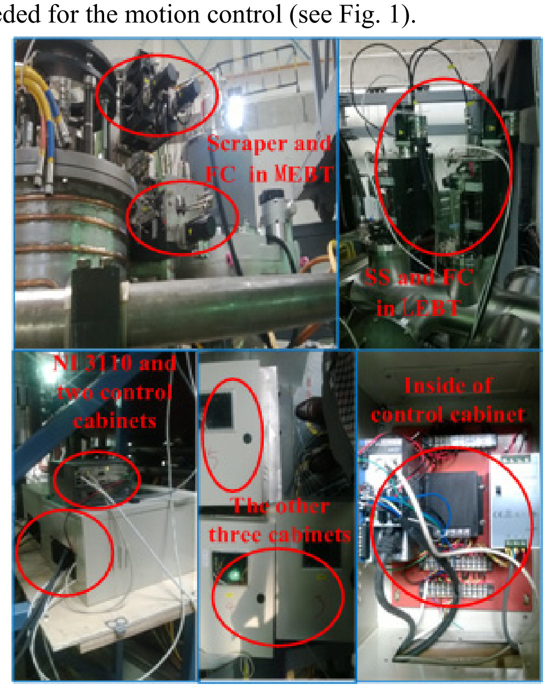

Figure 1: Photos of the field in motion control.

Because Kollmorgen servo drive supports EtherCAT communication protocol [4], which has obvious advantages in topological structure, clock

ISBN 978-3-95450-146-5

THE MEASUREMENT AND MONITORING OF SPECTRUM AND WAVELENGTH OF COHERENT RADIATION AT NOVOSIBIRSK FREE ELECTRON LASER

V. V. Kubarev, S. S. Serednyakov
Budker Institute of Nuclear Physics SB RAS, Novosibirsk, Russia

Abstract

This paper describes in detail the architecture and capabilities of the system for measurement of the free electron laser (FEL) radiation spectrum. The measurements are performed with a monochromator and a step-motor with a radiation power sensor. The measurements result in the transmission of the curve of the radiation spectrum to the control computer. As this subsystem is fully integrated into the common FEL control system, the results of the measurements – the spectrum graph, average wavelength, and radiation power calculated – can be transmitted to any other computer in the FEL control local area network, as well as to computers of the user stations.

INTRODUCTION

A high-power FEL based on a multiturn energy recovery linac (ERL) [1] is under construction now at Budker Institute of Nuclear Physics. The first and second phases of the project have already been commissioned and are currently in operation.

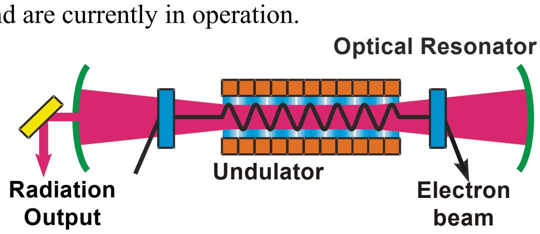

Figure 1: FEL Operation layout.

During its operation, the FEL generates coherent radiation (see Fig.1), which is used for various experiments. The wavelength of this radiation depends on some accelerator parameters and beam energy and is expressed with formula

$$\lambda = \frac{d}{2\gamma^2}\left(1 + \frac{k^2}{2}\right), \quad (1)$$

where λ is the radiation wavelength; d is the undulator period; γ is the relativistic factor of electrons; k is the undulator parameter; $k = k_0 \cdot I$, where I is the current in the coils of the electromagnetic undulator, and k_0 is the constant of proportionality.

As seen from Eq. (1), the radiation wavelength can be tuned via change in the beam energy or adjustment of the undulator current. Besides, the FEL radiation has a rather narrow spectrum, which depends on different FEL parameters. Therefore, the real-time monitoring of the spectrum, power and average wavelength of the FEL radiation is necessary for operators and FEL users throughout FEL operation. For this purpose, we created a separate system with a monochromator.

HARDWARE AND STRUCTURE OF RADIATION MONITORING SYSTEM

The layout of the system is presented in Fig.2. Its main device is the monochromator, which is used for measurement of the FEL radiation spectrum. The FEL radiation enters the entry window of the monochromator, and the radiation with transverse amplitude distribution corresponding to the input radiation spectrum goes out through the exit window. Thus, to obtain the radiation spectrum, one has to read out the intensity distribution at the exit of the monochromator and transmit it to the computer. That is done with the radiation sensor installed on a support, which is moved horizontally by a stepper motor. The stepper motor controller is connected to the IBM-PC via the RS-485 interface.

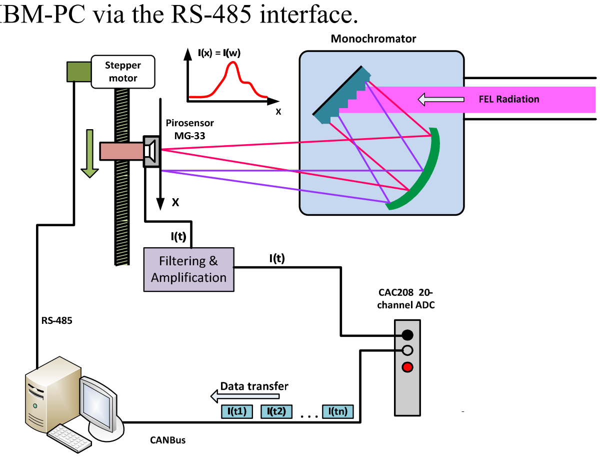

Figure 2: Layout of the system for measurement of the FEL radiation spectrum.

For proper measurement of the radiation power, the flux of measured radiation is modulated with a special rotating shutter. This procedure is necessary because the pyroelectric radiation sensor applied can measure only a variable signal. After some filtering and amplifying, the signal is measured with an ADC. The CAC208 device with CAN interface, developed at BINP [2], is used for

ISBN 978-3-95450-146-5

EPICS BEAST ALARM SYSTEM HAPPILY PURRS AT ANKA SYNCHROTRON LIGHT SOURCE

I. Kriznar, Cosylab, Ljubljana
S. Marsching, Aquenos GmbH, Baden-Baden
E. Hertle, E. Huttel, W. Mexner, N. J. Smale, A.-S. Mueller, KIT, Eggenstein-Leopoldshafen

Abstract

The control system of the ANKA synchrotron radiation source at KIT (Karlsruhe Institute of Technology) is adopting new, and converting old, devices into an EPICS control system [1]. New GUI panels are developed in Control System Studio (CSS). EPICS alarming capabilities in connection with the BEAST alarm server tool-kit from the CSS bundle are used as an alarming solution. To accommodate ANKA future requirements as well as ANKA legacy solutions, we have decided to extend the basic functionality of BEAST with additional features in order to manage the alarming for different machine operation states. Since the database of alarm sources is been populated from scratch, we have been able to take a fresh approach in management and creation of alarm sources to build-up alarm trees. The new alarm system is being continuously used, tested and refined, and has been in the production environment since the end of 2013.

INTRODUCTION

ANKA is an electron synchrotron radiation light source located in Karlsruhe, Germany. The storage ring is generally operated at an energy of 2.5GeV with a typical beam current of 200 mA and a life time of 20 hours. The ANKA machine control system has been gradually moved towards EPICS based solutions. Since EPICS is relatively fresh at ANKA we tried to use the best that is available in the EPICS field.

Overview

An alarm system catches, transports, processes and visualizes error conditions (malfunctions) of hardware and software. An alarm system can report problems early, before serious consequences develop, and can provide data for later analysis. It propagates the alarm states from the producers (IOCs, drivers, services, applications) to the clients.

A complete alarm system includes the following components: an alarm source on IOC or services, an alarm aggregation service and a book-keeping service, an alarm archive, an alarm distribution service, a GUI viewer for new and archived alarms, a GUI configuration manager of the alarm service. All these building blocks are nicely covered by CSS alarm system [2], called “Best Ever Alarm System Toolkit” or BEAST [3].

Status Monitoring vs. Alarms

It is very important to distinguish between a status monitoring and an alarm event. Status monitoring system and associated displays gives an overview of information about certain control system components or areas. Changes of status are potential sources of alarms if certain conditions are met. The Alarm view of a system is therefore a dynamic view of status changes, which might be dangerous to the operation.

A short example for illustration: If a server panel displays control system servers with little LED lights attached; Then, if a light is green, the server is running, if red, the server is down. This is status display. On the contrary when an alarm panel is blank, everything should be OK. When one of the servers goes down, the change of status triggers a notification, this is an alarm message, which appears in a table of active alarms until it is acknowledged by the operator. When an operator sees an alarm he/she can open a related status display for more information and consequently take appropriate action.

ALARM SOURCES

We can distinguish several kinds of EPIC’S records in relevance to alarming, thus several kinds of alarm sources.

Control point alarm sources are low level alarm sources. These are records, which are used to control or monitor devices and connected hardware. They generate alarm notifications directly during EPICS record processing and are connected to record intrinsic conditions: like value outside alarm limit or bus errors.

Context dependant alarm sources. are the higher level alarms; they calculate alarm conditions from values obtained from several sources. For example, the machine operation status provides the context, which then defines if an alarm is raised or not for some condition.

Gateway alarms. These are software records which monitor alarm sources, which are not part of EPICS, and forwards their alarms as EPICS alarms.

Alarms which originate in an **interlock system** or any other safety systems, which are generally independent from the rest of control system.

The Context Dependent Alarms

Some machine or device status conditions present valid alarm source only under specific conditions. For example, a certain magnet (or its power supply) turned ON can present a reason for an alarm when the machine is in INJECTION mode, but the same status could be OK or irrelevant when the machine is in SR RAMPING mode.

ISBN 978-3-95450-146-5

IMPLEMENTATION OF THE DISTRIBUTED ALARM SYSTEM FOR THE PARTICLE ACCELERATOR FAIR USING AN ACTOR CONCURRENT PROGRAMMING MODEL AND THE CONCEPT OF AN AGENT.

D. Kumar, G. Gašperšič, M. Plesko, Cosylab, Ljubljana, Slovenia
R. Huhmann, S. Krepp, GSI, Darmstadt, Germany

Abstract

The Alarm System is a software system that enables operators to identify and locate conditions which indicate hardware and software components malfunctioning or nearby malfunctioning. The FAIR Alarm System is being constructed as a Slovenian in-kind contribution to the FAIR project. The purpose of this paper is to show how to simplify the development of a highly available distributed alarm system for the particle accelerator FAIR using a concurrent programming model based on actors and on the concept of an agent. The agents separate the distribution of the alarm status signals to the clients from the processing of the alarm signals. The logical communication between an alarm client and an agent is between an actor in the alarm client and an actor in the agent. These two remote actors exchange messages through Java MOM. The following will be addressed: the tree-like hierarchy of actors that are used for the fault tolerance communication between an agent and an alarm client; a custom message protocol used by the actors; the message system and corresponding technical implications; and details of software components that were developed using the Akka programming library.

INTRODUCTION

The FAIR Alarm System is composed of three major layers: a generation layer, a processing layer and a client layer. The connecting glue between the layers is the messaging system which allows the layers to communicate by passing messages into each other's queues and topics.

The Generation Layer

The alarm generators are the components that raise/lower alarm signals which are transported to the processing layer through a Java Message Oriented Middleware. The main purpose of the generators is to produce the alarm signals containing an alarm identification and state of the alarm that can be active or inactive. They are also responsible for handling the fast alarm oscillations. The alarm generators produce life-cycle messages notifying the processing layer about their health. The alarm generators must be registered with the processing layer before the alarm signals can be sent. This gives the processing layer a chance to prepare the environment for alarm generator monitoring and alarm signal receiving. During the registration process the processing layer also checks that the alarm identifications are known to the system. If they are not known, the processing layer creates a default configuration for the unknown alarms.

The Processing Layer

The core of the alarm system is the alarm processor which is responsible for alarm signal processing and dispatching of the processed alarm signals to the client layer via an agent. The alarm processor also monitors the alarm sources. The alarm signal processing includes: matching the alarm signal with its configuration, updating the alarm state, alarm masking, and alarm archiving. The alarm processors are stateless and session-less, working in groups to share the load of the alarm processing. Processed alarm signals are not dispatched directly to the client layer but are sent to the agents which have active client sessions. The client layer accesses the alarm system only through an agent by opening a client session. All client requests are handled by the agents. The agent is responsible for handling alarm reduction, subscribing and filtering alarm state changes, acknowledgement of the alarms, sending filtered alarm state changes to the subscribed clients, and searching the alarm state and alarm archive.

The Client Layer

There are many types of alarm clients: the alarm monitoring viewer showing the state of the alarms, the alarm archive browser displaying the alarm history from a selected time range, the alarm configuration editor which issues CRUD operations on the alarm configuration. Common to all alarm clients is that they access the alarm system through an Alarm Client API. The alarm clients open many concurrent and independent sessions through which they issue requests to the alarm system and receive replies and alarm state changes. When a session is opened in the client layer, another session is also created on the agent. A hierarchy of actors [1] is created on both layers establishing a logical communication channel between a client session actor and an agent session actor. The physical communication is done through the actor hierarchy where individual actors take different roles: session management, session supervision, service worker, JMS message producer, and JMS message consumer.

ISBN 978-3-95450-146-5

VACUUM PUMPING GROUP CONTROLS BASED ON PLC

S. Blanchard, F. Antoniotti, F. Bellorini, JP. Boivin, J. Gama,
P. Gomes, H. Pereira, G. Pigny, B. Rio, H. Vestergard
CERN, Geneva, Switzerland
L. Kopylov, S. Merker, M. Mikheev
IHEP, Protvino, Russia

Abstract

In particle accelerators, high vacuum is needed in the beam pipes and for thermal isolation of cryogenic equipment. The first element in the chain of vacuum production is the pumping group. It is composed by: primary pump, turbo-molecular pump, valves, gauges, process and interlocks devices. At CERN accelerators, the pumping groups may be found in several hardware configurations, depending on the environment and on the vacuum system used; the control is always based on Programmable Logical Controllers (PLC) communicating with the field equipment over a field bus; pumping groups are controlled by the same flexible and portable software. They are remotely accessed through a Supervisory Control and Data Acquisition (SCADA) application and can be locally controlled by a portable touch panel. More than 250 pumping groups are permanently installed in the Large Hadron Collider, Linacs and North Area Experiments.

INTRODUCTION

This paper describes the new control software for the turbo molecular pumping group of type “VPG_6A”. It is based on operational modes and phase sequencers. It includes sequential process for pumping, leak detection and venting modes. Optional features for automatic restart and automatic venting are also available.

The mechanical layout of the pumping group is illustrated by the Fig. 1.

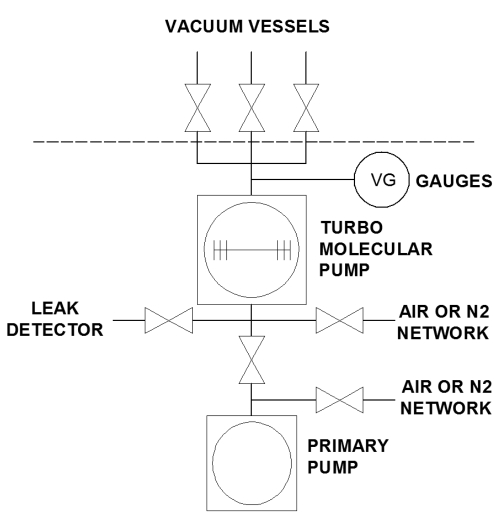

Figure 1: Pumping Group Layout.

CERN Accelerators have more than 120 km of beam and thermal isolation vacuum vessels. Turbo molecular vacuum pumping groups are used for rough pumping, for leak detection and to maintain the vessels under high vacuum. The primary pump performs initial pumping from atmospheric pressure to rough vacuum (between 10 and 0.1 Pascal) then the turbo molecular pump achieves high vacuum (between 1.10^{-4} and 1.10^{-7} Pascal).

HARDWARE

The new software is compatible with any hardware control crates based on Siemens® PLC S7-300 or S7-400 series with a minimum of 64Kb working memory. That includes the recently designed prototype shown in Fig. 2 but also with the 15 years old crates.

Pumping group devices are controlled using direct PLC Input/Output or Profibus® fieldbus.

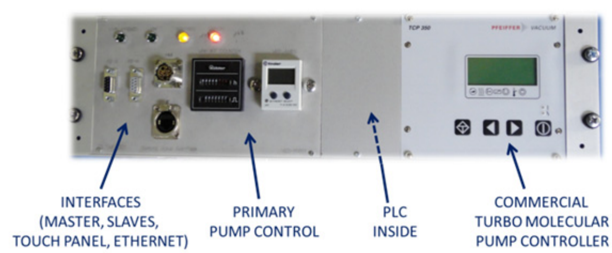

Figure 2: Prototype Pumping Group Controller Crate.

The controller crate commands the pumps and valves; and gets the feedback status from them: opened / closed from the valves, and a minimum of 3 status bits from the pumps.

For the primary pump: “ON” status is the feedback of the pump power supply actuator; “NOMINAL SPEED” status is given by the motor current monitor and “ERROR” status is the feedback of the circuit breaker.

For the turbo molecular pump: “ON” status is the pump rotation detected feedback; “NOMINAL SPEED” status is given when the pump rotor reaches nominal speed threshold (in the most of the cases, 80% of the maximum speed) and “ERROR” status occurs when load is too high, when no rotation is detected, or when the rotor acceleration time is too long.

The standard hardware architecture, as shown in Fig. 3, is composed of a local crate, a control crate, a touch panel (local Human Machine Interface), a gauge controller, a PLC master and a SCADA server.

A Profibus® network connects the touch panel, the control crate, the gauge controller and the PLC master.

ISBN 978-3-95450-146-5

DIAGNOSTICS TEST STAND SETUP AT PSI AND ITS CONTROLS IN LIGHT OF THE FUTURE SwissFEL

P. Chevtsov#, R. Ischebeck, PSI, 5232 Villigen, Switzerland

Abstract

In order to provide high quality electron beams, the future SwissFEL machine needs very precise beam diagnostics tools. At the Paul Scherrer Institute (PSI), the development of such tools required setting up a dedicated automated test stand, which could significantly simplify the development process. The test stand is equipped with not only major SwissFEL beam diagnostics elements (cameras, beam loss monitors, wire scanners, etc.) but also their controls and data processing hardware and software. The paper describes diagnostics test stand controls software components designed in view of the future SwissFEL operational requirements.

INTRODUCTION

The future SwissFEL facility [1] at the Paul Scherrer Institute (PSI) will produce superb photon beams for a variety of cutting-edge user experiments. The project aims to use the lowest electron beam energy compatible with 1 Å facility operation. In these conditions, to ensure successful facility lasing, the machine must be equipped with advanced diagnostics and controls tools continuously monitoring electron beam generation, transport, and compression. The SwissFEL Injector Test Facility (SITF), which has been a major platform for designing and testing such tools since for the last few years, is going to be put out of operations by the end of 2014. After that, there will be no similar platform available on site for the SwissFEL machine component prototyping and testing. In order to supply a satisfactory workbench for electron beam diagnostics and control developments, a dedicated automated Diagnostics and Controls Test Stand (DCTS) was set up in the SwissFEL diagnostics laboratory located in the SITF building. There are several advantages of this location. For instance, being in the area of a well-established SITF control system infrastructure, any required DCTS computer network is easily configurable and supportable. Another important advantage is the direct access to modern laboratory tools for efficient equipment checking and tuning.

To ensure the reproducibility of test results in the real machine environment, diagnostics and controls projects on the stand are required to organize as close to their SwissFEL operational conditions as possible. For example, wire scanner moving components must be kept in vacuum, motor steering cables have to be about 20-30 meters long, etc.

All DCTS control system software developments must be compatible with EPICS [2], which is the principal controls tool at PSI.

#pavel.chevtsov@psi.ch

DCTS DEVELOPMENTS

From its very beginning, the DCTS project was divided into several stages based on the SwissFEL type control system hardware and software availability for beam diagnostics applications.

On stage one, which was the initial step, the basic test stand control infrastructure was set up and the development of the test procedures for all tasks implemented on the test stand began.

Basic DCTS Control Infrastructure

The main DCTS control network is a general purpose 1 Gbit class C subnet. The operator console is a Linux PC furnished with a high resolution monitor. A standard SwissFEL electronics rack houses the DCTS control equipment, which includes one motion control unit and several VME64x crates.

One VME crate, which is a primary DCTS software development crate, is equipped with a new PSI control computer (IOC), IFC_1210, designed in collaboration with IOxOS Technologies SA, Switzerland [2]. The IFC_1210 is a highly configurable FPGA platform associated with XMC, PMC, and FMC mezzanine slots for custom expansions, a powerful dual core PowerPC, and a real time OS. It is built around a high-performance switched PCI Express GEN2 architecture. The VME support firmware provides a complete VME master/slave interface. Compared to the SITF workhorse MVME-5100 single board CPU, this new IOC type offers much more computing power, a higher data throughput, and very flexible I/O features available for software developers. However, to make IFC_1210 fully operational, a lot of work must be done to design and implement EPICS device/driver support software for a variety of controls hardware components that can be handled by this IOC. It was obvious to begin with the easiest part of this work, which is porting existing PSI VxWorks-specific VME memory access software to the new IOC platform. As a result, the most critical VME drivers, including the General Purpose Memory Mapping (GPMM), became available for IFC_1210 by the time of the initial DCTS setup. This has allowed one to relatively easily integrate a variety of VME control cards into the test stand controls environment and to concentrate major software development efforts on supporting fast, not less than 250 Mega-Samples-Per-Second (MSPS), 16 bit ADC, high speed DAC, and digital IO FMC modules available from IOxOS Technologies SA.

The test stand timing system is realized in the second VME crate. It will provide control device triggering mechanisms and synchronous transmissions of DCTS

ISBN 978-3-95450-146-5

MAGNET MEASUREMENT SYSTEM UPGRADE AT PSI

P. Chevtsov[#], V. Vrankovic, PSI, 5232 Villigen, Switzerland

Abstract

The magnet measurement system at the Paul Scherrer Institute (PSI) was significantly upgraded in the last few years. At the moment, it consists of automated Hall probe, moving wire, and vibrating wire setups, which form a very efficient magnet measurement facility. The paper concentrates on the automation hardware and software implementation, which has made it possible not only to significantly increase the performance of the magnet measurement facility at PSI, but also to simplify magnet measurement data handling and processing.

INTRODUCTION

The majority of magnets for PSI accelerators and experiments are designed in house and produced by industry. Upon delivery, these magnets are systematically measured to assure that all field quality specifications are met. The measurements are done with the use of the PSI magnet measurement system the technical and operational capabilities of which have significantly advanced in the last months. As a result, it is a modern, automated, user friendly system, which consists of high precision measurement setups based on Hall probe, moving wire, and vibrating wire techniques. The setups are arranged in separate rooms of the PSI magnet measurement laboratory, which are equipped with all necessary measurement tools, control computers, and operational consoles. The magnet measurement system is integrated into the PSI controls, which is based on EPICS [1]. This fact combined with the cutting-edge data acquisition and control software developed for magnet measurement applications at PSI makes the system easy to run not only from local operational consoles but also remotely, eventually from any PC connected to the PSI network.

HALL PROBE SETUP

Basic magnetic field mapping is performed using Hall probes. Fast automated Hall probe measurements are provided by a computerized Magnet Measurement Machine (MMM) created at PSI.

The MMM is a very precise positioning device sliding on compressed air pads over a flat, carefully machined granite block. Hall probes are attached to a titanium arm with carbon fibre extensions that can move in three translation directions (X, Y, Z) and rotate in the horizontal plane (a yaw angle) and around the arm (a roll angle). Each move is performed by a dedicated stepping motor. Positions are determined by Inductosyn® linear encoders with one half-micron accuracy. We note that the MMM measurements are performed in any translation direction while two rotations are used only for the proper probe positioning. Therefore, a measured field map corresponds to a line, a plane, or a volume in a Cartesian coordinate system.

#pavel.chevtsov@psi.ch

The Hall probe setup is installed in a temperature controlled (± 0.1° C) room, which is important for accuracy of this kind of measurements.

Hall probe potentials are recorded by an Agilent 3458A digital multimeter (DMM device). Periodic probe calibrations are done with the use of a Nuclear Magnetic Resonance based Metrolab PT2025 teslameter (NMR device). Both devices can be remotely controlled via the GPIB bus. Magnet current values are set by PSI digital power supply controllers.

The main Hall probe setup control hardware is implemented in a VME-64x standard. The control computer (IOC) is a MVME-5100 single board CPU running the VxWorks real time kernel. The IOC also runs the EPICS database and control software (MMM server), which handle all Hall probe setup components and provide the information about the state of these components and all measurement data. The stepping motors and their encoders, which are based on Inductosyn® sensors, are interfaced via a MAXv-8000 card and its transition module. In-house developed Industry Pack (IP) control modules PSC-IP2 sitting on a Hytec 8002 carrier board provide the access to magnet digital power supply controllers. A variety of digital (DIO) signals are required for MMM operations. They are handled by Hytec 8505 IP modules. DIO signals are used, for example, to trigger DMM devices and to support a special manual mode (as opposed to its normal mode that is automatic), in which the MMM is positioned by the magnet measurement personnel pressing buttons on a remote control unit that looks similar to a conventional TV remote control. We note that the manual mode is especially valuable for initial MMM positioning with respect to any magnet to measure.

MMM measurements are performed in a "continuous scan on the fly" mode, which means that the machine doesn't stop to make a particular measurement. This allows one to finish a complex field mapping process for each magnet in a relatively short time frame, which is typically one day or less.

The MMM control is done with the use of a specially designed graphical user interface (GUI) tool, which is called mmmgui. The tool is based on a well-known Qt framework [2], which automatically makes it computer platform independent. At PSI, the tool can run on any 32 or 64 bit Scientific Linux PC console. The mmmgui control software (MMM client) is multi-threaded. Each thread deals with a particular MMM operational mode, which is supported by a dedicated mmmgui panel implemented as a standard Qt tab. Threads communicate to each other over a shared memory synchronized with the MMM EPICS database.

ISBN 978-3-95450-146-5

STATUS OF CONTROL SYSTEM FOR THE TPS COMMISSIONING

Y. S. Cheng, Jenny Chen, C. Y. Wu, P. C. Chiu, C. Y. Liao, K. H. Hu, Demi Lee,
Y. T. Chang, C. H. Huang, S. Y. Hsu, C. H. Kuo, C. J. Wang, K. T. Hsu
National Synchrotron Radiation Research Center, Hsinchu 30076, Taiwan

Abstract

Control system for the Taiwan Photon Source (TPS) project has been implemented. The accelerator system began to be commissioning from third quarter of 2014. Final integration test of each subsystem will be done. The EPICS was chosen as the TPS control system framework. The subsystems control interfaces include event based timing system, Ethernet based power supply control, corrector power supply control, PLC-based pulse magnet power supply control and machine protection system, insertion devices motion control system, various diagnostics, and etc. The standard hardware components had been installed and integrated, and the various IOCs (Input Output Controller) had been implemented as various subsystems control platforms. Development and test of the high level and low level software systems are in final phase. The efforts will be summarized at this report.

INTRODUCTION

The TPS [1] is a latest generation of high brightness synchrotron light source which is being in construction at the National Synchrotron Radiation Research Center (NSRRC) in Taiwan. It consists of a 150 MeV electron linac, a booster synchrotron, a 3 GeV storage ring, and experimental beam lines. Civil construction had started from February 2010. The construction works are approximately finished in half of 2013. Accelerator system installation and integration was proceeding in later 2013. The control system environment was ready in half of 2014 to support subsystem final integration test and commissioning without beam. Commissioning with beam was started from August 2014.

Control system for the TPS is based on the EPICS framework [2]. The EPICS toolkit provides standard tools for display creation, archiving, alarm handling and etc. The big success of EPICS is based on the definition of a standard IOC structure together with an extensive library of driver software for a wide range of I/O cards. The EPICS toolkits which have various functionalities will be employed to monitor and to control accelerator system.

The control system consists of more than a hundred of EPICS IOCs. The CompactPCI (cPCI) IOC will be equipped with input/output modules to control subsystems as standard IOC or the TPS control system. The power supply and fan module of the cPCI crate will be hot-swapped. Adopting cPCI platform for EPICS IOCs provides us a chance to take advantages of local IT industry products with better supports and low cost. The other kinds of IOCs are also supported by the TPS control system, such as BPM IOC, PLC IOC, various soft-IOC and etc. The IOCs, control consoles and servers are also running Linux operation system.

To achieve high availability of the control system, emphasis has been put on software engineering and relational database for system configurations. Data channels in the order of 10^5 will be serviced by the control system. Accessibility of all machine parameters through control system in a consistent and easy manner contributes to the fast and successful commissioning of the machine. High reliability and availability of TPS control system with reasonable cost and performance are expected.

CURRENT STATUS

Installation for the control system is almost done. The system is ready to do final integration with subsystems. The software environment is ongoing final revision to connect various subsystems. Control related applications are on-going before subsystem ready in June/August 2014. The progress of the control system is summarized in following paragraphs.

Networking

Mixed of 1/10 Gbps switched Ethernet are deployed for the TPS control system [3]. The Gigabit Ethernet connection will be delivered at edge switches installed at control and instruments area (CIA). One CIA corresponding to one cell of the storage ring, there are 24 CIAs in total. The control network backbone is a 10 Gigabit link to the control system computer room. Private Ethernet is used for Ethernet based devices access which will support fast Ethernet and GbE. Adequate isolation and routing topology will balance between network security and needed flexibility. The file and database servers are connected to the control and intranet network, allowing the exchange of data among them. Availability, reliability and cyber security, and network management are focus in the design phase.

Equipment Interface Layer

There are several different kinds of IOC at equipment layer to satisfy various functionality requirements, convenience, and cost consideration. Most of the devices and equipments will be connected to cPCI IOCs with EPICS running directly. The 6U cPCI platform was chosen for the EPICS IOC. To simplify various developments at construction phase, only 6U modules are supported for the machine control system. The cPCI EPICS IOC equipped with the latest generation CPU board will be standardized as ADLINK cPCI-6510 CPU module [4]. The CPU module equipped with Intel Core i7

ISBN 978-3-95450-146-5

NETWORK ARCHITECTURE AT TAIWAN PHOTON SOURCE OF NSRRC

Chih-Hsien Huang, Yin-Tao Chang, Yung-Sen Cheng, Chang-Hor Kuo and Kuo-Tung Hsu
NSRRC, Hsinchu 30076, Taiwan

Abstract

A robust, secure and high throughput network is necessary for the 3 GeV Taiwan Photon Source (TPS) in NSRRC. The NSRRC network divides into several subsets according to its functionality and includes CS-LAN, ACC-LAN, SCI-LAN, NSRRC-LAN and INFO-LAN for the instrumental control, subsystem of accelerator, beam-line users, office users and servers for the information office respectively. Each LAN is connected via the core switch by routing protocol to avoid traffic interference. Subsystem subnets connect to control system via EPICS based channel-access gateways for forwarding data. Outside traffic will be block by a firewall to ensure the independence of control system (CS-LAN). Various network management tools and machines are used for maintenance and troubleshooting. The network system architecture, cabling topology and maintainability will be described in this report.

INTRODUCTION

Taiwan photon source (TPS) [1] is a 3 GeV synchrotron radiation facility with ultra-high photon brightness and extremely low emittance in National Synchrotron Radiation Research Center (NSRRC). The construction began in February 2010, and the commissioning started in the third quarter of 2014.

A high secure and robust network is necessary for such a new accelerator. In order separate the traffic of various kinds of users, 5 layer-3 middle switches are used. TPS control system used for the operations of accelerators and beamlines is implemented by the experimental physics and industrial control system (EPICS) [2, 3] software toolkit. Control devices are connected by the control network and integrated with EPICS based input output controller (IOC).

THE NETWORK ARCHITECTURE OF NSRRC

The NSRRC campus hosts a 1.5 GeV Taiwan light source (TLS), several buildings for office and experimental users, and TPS building. In order to maintain the network functionality of the existed infrastructure and transport to TPS new infrastructure, it is gathered by a middle-layer switch labelled with NSRRC-LAN that the exited wired network including the control system of TLS in the first step. The bandwidth of the backbone upgrades to 10 Gbps and upgrades to 1 Gbps for users gradually. During the upgrade, vlan introduces to the setting of switches in order to replace the use of the public internet protocols (IPs).

In the building of TPS, three subnets including CS-LAN, ACC-LAN and SCI-LAN are created for the control system, subsystems of accelerator and beam-line, shown in Fig. 1. The office automation users belong to the NSRRC-LAN. The subdomain of each LAN is setting in a layer-3 switch. The traffic between middle switch and core switch is through the routing protocol. Single mode fibers supporting 10/40 Gbits/sec link between network rooms and equipment areas at reasonable cost. 10 G links are setup as backbone at this moment.

Control and subsystem network services are available in 24 control instrumentation areas (CIAs) with an individual edge switch in the inner ring area. Major devices and subsystems are installed inside CIAs. There are 4 network rooms and one network & server room outside the ring. These provide the network for the experimental users and the fiber adapter of timing system and intranet network between the middle-layer switch and edge-layer switch, which is installed within beam line station.

There are many servers (e. g. e-mail server, AD server, employee portal server, Web server) belonging to the information office that serve for the employee or various kinds of users. An INFO-LAN will be created for this kind of usage after the servers are moved to TPS. Two subnets with one public IP and one private IP will be signed in the INFO-LAN for the internet and intranet servers.

The traffic of the wireless LAN (WLAN) is independent of the wired network in order to insure the normal operation of the wireless network while the local wired network breaks down. Each access point (AP) broadcasts four service set identifiers (SSIDs), i.e. NSRC-Staff, NSRRC-TEL, NSRRC-Roaming and NSRRC-Guest for staff, IP phone, roaming and guest users. Each user must be certificated before using the wireless. For the staff users, the traffic is direct into core switch without passing through firewall for the intranet access. However, for the Roaming or guest users, the traffic must pass through the firewall and the usage of the intranet is limited by the policy of the firewall.

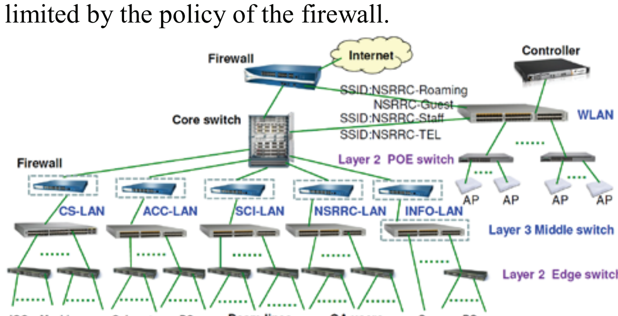

Figure 1: The network infrastructure in NSRRC campus.

ISBN 978-3-95450-146-5

BPM CONTROL, MONITOR, AND CONFIGURATION ENVIRONMENTS FOR TPS BOOSTER

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

Abstract

Booster synchrotron for the Taiwan photon source project which is a 3 GeV synchrotron light source constructed at NSRRC is in commissioning. The BPM electronics Libera Brilliance+ [1] are adopted for booster and storage ring of Taiwan Photon Source (TPS). The acceptance test had been completed in 2012 [2]. The provided BPM data is useful for beam commissioning where it can be used to measure beam position, rough beam intensity along the longitudinal position and also for tune measurement. This report summarizes the efforts on BPM control, monitor and configuration environment.

INTRODUCTION

The TPS is a state-of-the-art synchrotron radiation facility featuring ultra-high photon brightness with extremely low emittance [3]. Civil constructions had been completed in early 2013. The TPS accelerator complex consists of a 150 MeV S-band linac, linac to booster transfer line (LTB), 0.15–3 GeV booster synchrotron, booster to storage ring transfer line (BTS), and 3 GeV storage ring. The booster has 6 FODO cells which include 7 BD dipoles with 1.6 m long and 2 BH dipoles with 0.8 m long in each cell. Its circumference is 496.8 meters and it is concentric with the storage ring in the same tunnel. Libera Brilliance+ electronics has been adopted for the position measurement for booster and storage ring. It has provided precise beam position measurement and useful tools during booster commissioning in progress. This report will summarize the booster BPM related environment and functionalities.

BOOSTER BPM LAYOUT AND FUNCTIONALITIES

The TPS booster ring has six cells where each cell is equipped with 10 BPMs which can be used to measure beam position and rough beam intensity along the longitudinal position. Fig. 1 shows the mechanical drawing of booster BPM which shapes 35x20 mm elliptical and button diameter 10.7 mm. The calibration factor Kx and Ky is 8.25 and 9.66 mm respectively.

The conceptual functional block diagram of the BPM electronics is shown in Fig. 2. It will provide several data type for different application. ADC and TBT data is acquired on demand by trigger; 10 Hz slow data is for DC average orbit and 10 kHz fast data could be applied for booster ramping orbit or fast orbit feedback application. It is also embedded with EPICS IOC for control, monitor and configuration. The timing AMC module would provide functionalities of synchronization, trigger, interlock and post-mortem.

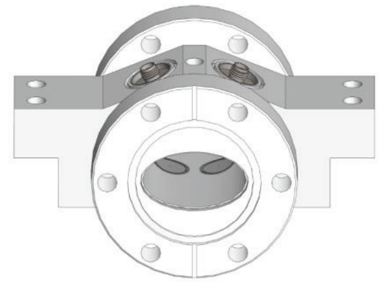

Figure 1: Button-type BPM for TPS Booster.

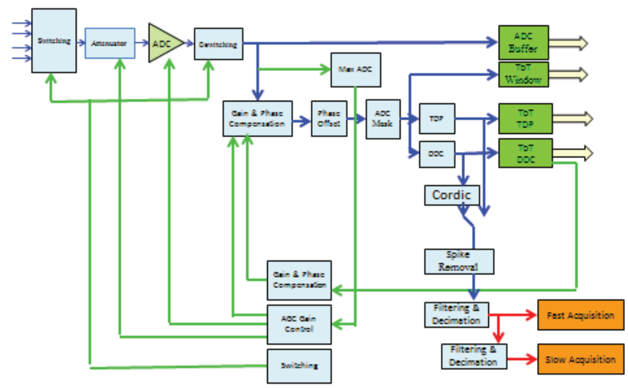

Figure 2: BPM platform functional block diagram.

FUNCTIONALITY TEST AND MANAGEMENT ENVIRONMENT

To support operation of the BPM electronics, functionalities like cold start, shutdown, housing, control system interface should meet the requirements. The delivered units also had been performed functionality and performance test to ensure compliance with this specification.

Cold Start and Shutdown

All BPM platforms and BPM electronics have been performed cold and shutdown test. Only minor problems encounter. Non-booting issue due to incorrectly shut-down procedure is reported to the vendor. Uploading the new FPGA image would fix it.

EPICS Interface

The EPICS interfaces for all BPM platforms are tested. Several defects is identified and then resolved after firmware upgrades. GUI based on EDM and Matlab is also developed for configuration and monitor.

Housekeeping

The BPM platform provides power supply status monitoring, temperature monitoring, fan status monitoring and fan control. The status looks good from the system maintenance point of view. EDM pages are

A MODULAR PERSONNEL SAFETY SYSTEM FOR VELA BASED ON COMMERCIAL SAFETY NETWORK CONTROLLERS

D.M. Hancock, B.G. Martlew
STFC Daresbury Laboratory, Warrington, UK

Abstract

STFC Daresbury Laboratory has recently commissioned VELA (Versatile Electron Linear Accelerator), a high performance electron beam test facility. It will be used to deliver high quality, short pulse electron beams to industrial users to aid in the development of new products in the fields of health care, security, energy and waste processing and also to develop and test novel compact accelerator technologies. In the early stages of the design it was decided to use commercial Safety Network Controllers and I/O to implement the Personnel Safety System in place of the electro-mechanical relay-based system used on previous projects. This provides a high integrity, low cost solution while also allowing the design to be modular, programmable and easily expandable. This paper describes the design and realisation of the VELA Personnel Safety System and considers its future development. In addition, the application of the system to the protection of high-power laser systems and medical accelerators will also be discussed.

INTRODUCTION

The VELA photoinjector consists of a 2.5 cell S-band RF gun using a copper photocathode, driven by a sub-100fs UV laser, designed to provide low emittance, short pulses of electrons. VELA features a suite of diagnostics including YAG screens for transverse beam profiles, a variety of emmittance measurement devices, an energy spectrometer dipole and a transverse deflecting cavity for bunch length and longitudinal phase-space measurements [1]. The machine is housed inside a 2m thick concrete enclosure designed to provide shielding for future higher energy upgrades. The accelerator vault is split into 3 distinct areas; the accelerator room, beam area 1 and beam area 2 and is designed in such a way that set-up work can be carried out in either of the beam areas whilst the accelerator is operational. The basement area beneath the accelerator is also shielded. Construction of VELA began in September 2011 with first beam achieved in April 2013 and the first experimental users in September 2013. The general layout of VELA is shown in Fig. 1

PERSONNEL SAFETY SYSTEM OVERVIEW

The VELA Personnel Safety System (PSS) controls the generation of ionising radiation by enabling the operation of the electron gun and RF cavities. The gun and RF system may only be operated when the injector room and basement areas have been searched and interlocked. The PSS also controls the transmission of the electron beam to beam area 1 and beam area 2 by enabling the operation of radiation shutters and dipole magnets.

Search Procedure

A two-person search system is employed and each area (accelerator room, basement and beam areas) can be searched independently. A card reader is used to limit initiation of a search to those trained in the correct search procedure. Pairs of search buttons (which must be pressed in sequence) guide the search team along a pre-defined route which covers the entirety of the shielded area.

Warning Devices

'Secret until lit' warning signs and audible warning devices alert staff to the status of the accelerator. Illuminated signs use arrays of LEDs (light emitting diodes) which have a low failure rate (the manufacturers quote a MTBF of 100,000 hours which equates to >10 years continuous use). All warning devices are 'fail-safe' and this function is achieved by the use of current sensing circuits to detect current flowing through the devices and these circuits provide interlock inputs to the PSS . Sensors used for flashing signs and audible warning devices have a build-in time delay to allow for the variation in current. The sensors have an adjustable current threshold which gives an indication that the device is working when the current threshold is exceeded.

Safety Console

A 'safety console' located in the control room provides a number of PSS keys which enable individual accelerator functions. These are:

- Shielding key – removal prevents a search of the accelerator room and basement
- RF Mode Key – removal prevents operation of the RF system
- Run Key – removal prevents operation of the gun by disabling the photoinjector laser shutters
- BA1 Key – removal prevents a search of beam area 1
- BA2 Key - removal prevents a search of beam area 2

ISBN 978-3-95450-146-5

BENEFITS, DRAWBACKS AND CHALLENGES DURING A COLLABORATIVE DEVELOPMENT OF A SETTINGS MANAGEMENT SYSTEM FOR CERN AND GSI

R. Mueller, J. Fitzek, H. Huether, GSI, Darmstadt, Germany
G. Kruk, CERN, Geneva, Switzerland

Abstract

The settings management system LSA (LHC Software Architecture) [1] was originally developed for the LHC (Large Hadron Collider). For FAIR (Facility for Antiproton and Ion Research) a renovation of the GSI control system was necessary. When it was decided in 2008 to use the LSA system for settings management for FAIR, the middle management of the two institutes agreed on a collaborative development. This paper highlights the insights gained during the collaboration, from three different perspectives: organizational aspects of the collaboration, like roles that have been established, planned procedures, the preparation of a formal contract and social aspects to keep people working as a team across institutes. It also shows technical benefits and drawbacks that arise from the collaboration for both institutes as well as challenges that are encountered during development. Furthermore, it provides an insight into aspects of the collaboration which were easy to establish and which still take time.

INTRODUCTION

The idea of collaboration between CERN and GSI on the settings management system was first discussed in 2006 between the Controls groups management of both institutes. At that time the development of LSA was well advanced but completion of functionality necessary for the LHC startup in 2008 still required a significant amount work. While GSI was evaluating different possibilities for a new control system that could be used for FAIR, it was agreed that two software engineers from GSI will spend 18 months at CERN to help with the development and commissioning of LSA.

The common development had clear benefits for both institutes. For CERN it was a reinforcement of the LSA team by two skilled developers in the view of the upcoming deadline and a fresh view on the system by new external people, while for GSI it was an excellent occasion to gain a valuable experience, learn about the system and main ideas behind, and to evaluate its possible use at GSI.

Since the first impressions about the portability of LSA were very positive, after the 18 months of joint development at CERN, the two GSI developers deployed a dedicated version of LSA in their home institute for further evaluation. The first goal was to prepare a working version of an LSA-based control system for the existing SIS18 synchrotron, which would help other GSI developers and users get deeper insight in the system and be a final verification of applicability of LSA to the whole accelerator complex at GSI.

After positive feedback and successful machine development sessions on SIS18, it was decided to use LSA for FAIR and continue the collaborative development of LSA by both institutes [2–5].

BUILDING THE TEAM

The initial 18 months of common development in one place was certainly beneficial in building a solid foundation for further collaboration as it created strong bonds within the team who worked together towards shared goals. It gave everyone a good overview about the technical aspects of the control systems in both institutes as well as a good insight into the work processes of the colleagues from the other institute, their constraints and their deadlines. This certainly helped in increasing the mutual understanding and in taking certain decisions related to the collaborative development.

To keep the collaboration active and to maintain the team spirit developed during those 18 months in the longer term, it was perceived as important to maintain a regular contact. This goes beyond exchanging ideas via mail or phone calls to also having in person discussions during regular visits. For the collaboration between CERN and GSI, these take place twice a year and are used to discuss, agree and schedule major changes to be applied in the system.

DEVELOPMENT PROCESS

Modularization

Even though LSA had been organized in a modular way from the beginning, during the common development in 2007 and 2008 it has been further restructured to split the code base into generic and CERN-specific parts, allowing possible extensions by GSI.

Figure 1 shows the current package hierarchy. The common modules in the middle contain the generic settings-management framework and have no outward dependencies, so they can be compiled and released independently. The domain objects and logic related to particular accelerator models, types of equipment, infrastructure (such as timing) and operation modes are implemented in the institute-specific parts (on the left and right side in the Figure 1).

The institute-specific extensions are developed and released independently by CERN and GSI. However all changes done in common modules are subject to review and acceptance by developers from both institutes.

ISBN 978-3-95450-146-5

ePlanner SOFTWARE FOR MACHINE ACTIVITIES MANAGEMENT

Bakshi Sanjai Kumar Srivastava, Rajesh Kumar Agrawal, Pravin Fatnani,
Raja Ramanna Centre for Advanced Technology (RRCAT), Indore, India

Abstract

For Indus-2, A 2.5 GeV Synchrotron Radiation Source, operational at Indore, India, the need was felt for software for easily managing various related activities for avoiding communication gaps among the crew members and clearly bringing out the important communications for machine operation. Typical requirements were to have the facility to enter and display daily, weekly and longer operational calendars, to convey system specific and machine operation related standing instructions, to log and track the faults occurring during the operations and follow up actions on the faults logged etc. Overall, the need was for a system to easily manage the number of jobs related to planning the day to day operations of a national facility. The paper describes such a web based system developed and in use regular use and found extremely useful.

OVERVIEW

Indus-1 and Indus-2, the Synchrotron Radiation Sources (SRS) at RRCAT Indore are national facilities operated round the clock to provide synchrotron radiations to users as well as carrying out machine studies. The concerned groups among which good information communication is considered essential include the beam line users, the operation crew, the machine sub-system experts and the management. Keeping sync among various groups is important and necessary for effective management of various activities and smooth running and utilisation of the facility. Large number of sub systems and frequent changes often required in various systems also call for effective monitoring of the system changes and communicating the same to all concerned. The Eplanner software was conceived to minimise various difficulties faced in day to day operation of this facility.

SYSTEM REQUIREMENTS

Based on the previous experience and the feedback received from various system experts and machine operation crew members following basic requirements were considered before developing the ePlanner software:

(1) It should be possible to use the software by multiple users simultaneously connected over campus network.
(2) The ePlanner should be easy to use for especially non computer experts.
(3) Only authenticated & authorized users should be able to use the specified modules of ePlanner. However it should be possible to get the logged data without authentication in read only mode.
(4) It should be possible to query & retrieve the specific information from stored historical data in chronological order.

SOFTWARE DESCRIPTION

Access to ePlanner has been protected with password so that only authenticated users could use the system. Each Section of ePlanner has different user group. System checks user's credentials using institute's central e-mail server. Hence users may use the software using their official e-mail login and password. This approach was adopted to avoid storing duplicate credentials for same person. Essential functions of ePlanner are depicted in Figure 1. ePlanner provides functionalities for Work Plan Management, Machine Shutdown Management, Beamline Booking Management, Standing Instructions Management, Electronic Notice Board (eNoticeBoard) and Information Display.

Figure 1: ePlanner Functions.

It is felt by Indus operation crew members that Indus Fault Logbook (FLogbook) which is a separate application used for tracking the details of faults occurring during round the clock operation of Indus-1 and Indus-2 should be merged with ePlanner so that all the information could be managed from a single software package.

Data Input

Using HTML & JavaServer Pages (JSP) ePlanner provides the form (having text boxes, check boxes, drop down list, etc.) for entering/logging the related information. Logged textual data with attached document (if any) may also be sent immediately to system persons concerned through e-mail. A unique log-id is generated

ISBN 978-3-95450-146-5

RECENT HIGHLIGHTS FROM COSYLAB

M. Plesko, F. Amand, Cosylab, Ljubljana, Slovenia

Abstract

Cosylab was established 13 years ago by a group of regular visitors of the PCaPAC. In the meantime, it has grown to a company of 90 employees that covers the majority of accelerator control projects. In this talk, I will present the most interesting developments that we have done in the past two years on a very different range of projects and I will show how we had to get organized in order to be able to manage them all. The developments were made for labs like KIT, ITER, PSI, EBG-MedAustron, European Spallation Source, Maxlab, SLAC, ORNL, GSI/FAIR but also generally for community software like EPICS, TANGO, Control System Studio, White Rabbit, etc. And they range from electronics development to high level software: electric signal conditioning and interfacing, timing system, machine protection system, fibre-optic communication, Linux driver development, core EPICS development, packaging, high performance networks, medical device integration, database development, all the way up to turnkey systems. Efficient organisation comprises a matrix structure of teams and groups versus projects and accounts, supported by rigorous reporting, measurements and drill-down analyses.

INTRODUCING COSYLAB

In the course of the writing of this paper we received final confirmation of a 2.5 million CHF project for Cosylab on the SwissFEL at PSI, Switzerland. Great news for us and a confirmation of Cosylab's growing involvement in the control system work on cutting edge Big Physics machines around the globe. In this paper we will start by highlighting a few particular projects we are taken on as we speak with a turnkey approach. We will then zoom in on project management aspects that are key to keeping such endeavours on track and delivering on the promise. We will finally look at operational aspects, how Cosylab is internally organized, as the organization is tailored to nature of the projects, their size, as well as our current size.

ELI-NP

The Extreme Light Infrastructure (ELI) [1] currently consists of four projects that will provide a great platform for the study of the fundamental processes that unfold during light-matter interactions. One of them is ELI-NP (Nuclear Physics) in Magurele, Romania.

Once built, the ELI-NP will be the most advanced laser and gamma beam facility in the world. It features Cosylab's first true turnkey control system, designed as such from the ground up. Coordinated efforts towards turnkey solutions have existed for several years and are being applied by Cosylab to other projects – most prominently for the current projects for ESS and MedAustron.

SOLARIS

Solaris is a project of the Polish Synchrotron Consortium (comprising 35 research institutes and universities). It received financing from the European Structural Funds, and is being constructed as we speak, with first research planned for 2015.

Specific to this project is a strong partnership with the MAX-IV project [2] in Lund, Sweden. The design of the synchrotron ring aims at maximizing reuse of the design of the 1.5 GeV storage ring of MAX-IV. This has repercussions on the design of the control system: it is based on the TANGO control system [3], chosen by MAX-IV.

Cosylab has been selected for the delivery of the control system integration services, including delivery of the timing system hardware. With this choice the Solaris team has opted not to reinvent the wheel on control system integration, just as they chose not to reinvent the storage ring design.

PROJECT MANAGEMENT ASPECTS

Incomplete Requirements

The incomplete requirements, that are so typical of ground breaking and pioneering big physics projects, including the control system part, have a tendency to stall projects before they have even started.

We have a strong philosophy about incomplete requirements, namely to accept them as a fact of life, rather than to fight them (by insisting on and waiting for clearer requirements, hence: stalling). We rather start with designing on a "straw man" design for the control system (with the requirements we have) and develop an early vertical prototype. As such we are pro-active in eliciting the actual requirements, because seeing physical things makes people think much more.

Stalling or over-analysing instead makes little sense, because no complex development ever follows the simple "waterfall" scenario with the phases like requirements, design and implementation following linearly one after the other only once the previous is finished. In reality, some requirements are dropped during the way, new come in, and what's more, development cycles often follow a spiral and pass several times through all the phases.

ISBN 978-3-95450-146-5

MANAGING THE FAIR CONTROL SYSTEM DEVELOPMENT

Ralph C. Bär, Frédéric Ameil
GSI Helmholtz Center for Heavy Ion Research, Darmstadt, Germany

Abstract

After years of careful preparation and planning, construction and implementation works for the new international accelerator complex FAIR (Facility for Antiproton and Ion Research) at GSI have seriously been started. The FAIR accelerators will extend the present GSI accelerator chain, then being used as injector, and provide anti-proton, ion, and rare isotope beams with unprecedented intensity and quality for a variety of research programs.

The accelerator control system (ACS) for the FAIR complex is presently being designed and developed by the GSI Controls group with a team of about 50 soft- and hardware developers, complemented by an international in-kind contribution from the FAIR member state Slovenia.

This paper presents requirements and constraints from being a large and international project and focusses on the organizational and project management strategies and tools for the control system subproject. This includes the project communication, design methodology, release cycle planning, testing strategies and ensuring technical integrity and coherence of the whole system during the full project phase.

FAIR PROJECT

FAIR, the new Facility for Antiproton and Ion Research is a new international accelerator facility for the research with antiprotons and ions. It is being built at GSI in cooperation of an international community of countries and scientists. The accelerator facilities significantly extend the present GSI accelerator complex, then being used as an injector for the FAIR machines.

In October 2010, nine countries signed the international agreement on the construction of FAIR under international law. FAIR will be financed by a joint international effort of so far 10 member states. The Federal Republic of Germany together with the local federal state of Hesse will provide the major part of the budget. International partners in Europe and overseas will substantially contribute as well, about 30% of the construction costs, some of them already being shareholders of FAIR. The countries will contribute both in kind, by supplying facility components, and in cash.

FAIR will be realized in several modules. The funding for the modularized start version, 1 billion Euro in 2005 prices, has been acquired. The FAIR start version [1] comprises the superconducting SIS-100 ring, CR, HESR, SFRS, Proton-linac and about 3.5 km of beam line.

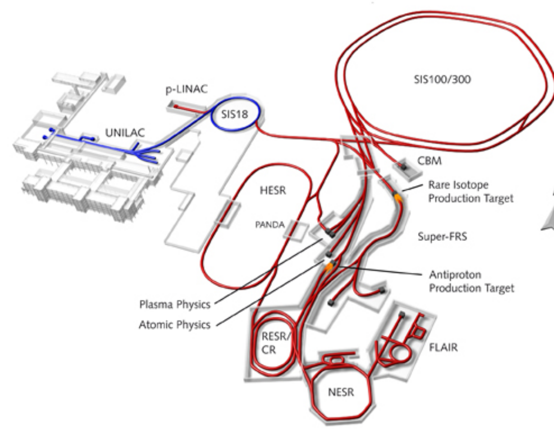

Figure 1: Schematic overview GSI (blue, existing) and FAIR (red, to be built) accelerators and beam lines.

FAIR PROJECT MANAGEMENT

Considering the substantial scope of the FAIR project (with its three sub-projects accelerators, experiments and civil construction), the technical and organizational complexity as well as the high financial investments needed, the FAIR council and mainly the German funding agency has required for a high and professional standard project management.

Consequently, in the past years GSI has established a project organization and management that is adequate for a project of this dimension. In 2012 GSI has been completely reorganized in order to fully focus on the FAIR construction phase. Besides the line-hierarchy, a matrix-like project organization was introduced:

- Project Leader FAIR@GSI (PL)
- Machine Project leaders (MPL)
- Work Package Leaders (WPL)

For the accelerator subproject, a Project Leader (PL) was appointed and a Project Management Office was built up. For the individual machines (e.g. SIS-100, CR, HESR, etc.) 7 Machine Project Leaders were introduced. For the technical subsystems (e.g. power converters, RF systems, vacuum, beam instrumentation, magnets, etc.) about 100 Work Package Leaders (WPL) have been installed and assigned to the respective machines.

Integrated Project Planning

To manage time schedules and resources of the GSI contribution to FAIR, Integrated Project Planning group has been established and project planning tools introduced and customized for FAIR. For resource-loaded project planning, an MS Project 2010 server environment has

ISBN 978-3-95450-146-5

STATUS OF INDUS-2 CONTROL SYSTEM

Pravin Fatnani#, Anurag Bansal, Rajesh Kumar Agrawal, Kirti G. Barpande, Amit Chauhan, Sampa Gangopadhyay, Pankaj Gothwal, Ashesh Mangaldas Gupta, Musuku Janardhan, Bhavna Nitin Merh, Rohit Mishra, Chandrashekhar Purushottam Navathe, Kutubuddin Saifee, M Seema, Yogendra M Sheth, Bakshi Sanjai Kumar Srivastava, Rishipal Yadav

Raja Ramanna Centre for Advanced Technology (RRCAT), Indore, India

Abstract

Indus-2 is a 2.5 GeV Synchrotron Radiation Source at Indore, India. With 8 beamlines operational, several more under installation & commissioning and 5 insertion devices planned, the machine is operated in round the clock mode for users. With implementation of orbit, tune and bunch feedback systems and many new systems in planning, machine is constantly evolving and so is the control system. The control system software is based on WINCCOA SCADA running on windows PCs and also integrates with other software modules in Labview and Matlab. The control hardware is a combination of VME based control stations interconnected over Ethernet and Profibus. Some recent system enhancements include parameter deviation alarms, transient data capture system, database improvements and web services. Paper takes a stock of the control system and its evolution along with new systems recently introduced.

INTRODUCTION

Over the years, Indus-2 has been consistently graduating to newer milestones. Among various other enhancements, these relate to addition of new front ends and beam lines, regular operation of various beam lines, round the clock operation of machine, improvement of orbit stability through implementation of orbit control systems, tune feedback system to stabilise the machine tune points, improvements in ramping and cycling procedures, faster data logging, automation and monitoring of various auxiliary systems like LCW, compressed air system, cavity precision chillers, solid state RF amplifiers, addition of vacuum chamber temperature alarms, bunch by bunch feedback system to counter instabilities at higher beam currents, addition and remote operation of diagnostic beamlines, machine and sub-system diagnostics etc. Now the machine operates with global slow orbit feedback (SOFB) system controlling the orbit to within 30 microns. With tune feedback system implemented, bunch by bunch feedback system integrated with the control system, both local and global fast orbit feedback (FOFB) systems demonstrated, the system is now being prepared for regular use of combined operation of SOFB along with FOFB. Average horizontal orbit drift correction using RF frequency correction will also be integrated. Two undulators will be received and installed soon.

fatnani@rrcat.gov.in

The preparations for system integration of undulators and its control system with machine control system are on the anvil. The control system for Indus accelerators is designed, developed, and maintained by the Accelerator Control Section, RRCAT.

OVERVIEW

The control system works on client server model and enables functional and physical separation and placement of hardware and software modules across the entire range of control system components. Using WinCCOA [1] SCADA on Windows client and server machines interconnected over ethernet switched network to two layers of VME controllers via Profibus, the distributed Indus-2 control system (Fig.1) monitors about 10000 I/Os in all. User Interfaces are mostly built around the SCADA for managing the complex requirements in an integrated manner. The intermediate, supervisory layer and the lower equipment interface layers are based on VME controllers running a multitasking real time operating system. Ethernet (100 MBPS) and PROFI bus (750 K baud) communications are used between L1-L2 and L2-L3 respectively. The modular control system hardware is designed around VME bus.

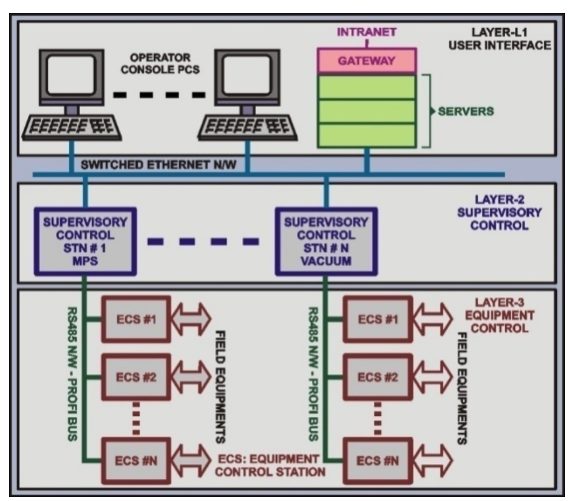

Figure 1: Indus-2 Control System Architecture.

ISBN 978-3-95450-146-5

CONCEPTUAL DESIGN OF THE CONTROL SYSTEM FOR SPring-8-II

R. Tanaka, T. Matsushita, T. Sugimoto, JASRI/SPring-8, Hyogo, Japan
T. Fukui, RIKEN/SPring-8, Hyogo, Japan

Abstract

The SPring-8 storage ring was inaugurated 17 years ago in 1997. The storage ring is an 8-GeV synchrotron that functions as a third-generation light source, providing brilliant X-ray beams to a large number of experimental users from all over the world. In recent years, discussions have been held on the necessity of upgrading the current ring to create a diffraction-limited storage ring at the same location. Now, a plan to upgrade the storage ring, called SPring-8-II, has been launched. First, new beam optics capable of storing beams of 6 GeV was designed using a five-bend magnet system to obtain smaller electron beam emittance that would produce coherent X-rays that are brighter than those produced by the current ring. The design of a control system that would meet the performance requirements of the new ring has also started. Equipment control devices are based on factory automation technologies such as PLC and VME, whereas digital data handling with high bandwidths is realized using telecommunication technologies such as xTCA. In this paper, we report on the conceptual design of the control system for SPring-8-II on the basis of the conceptual design report proposed by RIKEN.

INTRODUCTION

SPring-8, which is a third-generation light source, has been in service for more than 17 years. Out of the 140,000 users in the SPring-8 community, approximately 4,500 users come to the site for synchrotron radiation experiments every year. In the past, productive scientific results have been obtained and shared with the community. Recently, an XFEL facility, SACLA, was constructed at the same site as SPring-8 and inaugurated for public use in 2012. The SPring-8 site is now unique in the sense that it is the only location to have both the SR light source and XFEL.

Experiments at SPring-8 provide good measurements of static phenomena with crystal samples because of incoherent X-ray beams. On the other hand, SACLA can measure fast-moving dynamical phenomena even for non-crystal samples such as thin-film proteins, with 10-fs pulses destroying the samples. This difference represents the characteristic features of the two light sources. On the basis of the proposed conceptual design report (CDR) [1], we can infer that a wide gap exists between the two machines.

To narrow the present gap, a new project involving a new storage ring is planned; that is, the current storage ring will be replaced by a new storage ring at the same location. In the proposed project, a 1-GeV linac, which currently serves as the injector for SPring-8, will be replaced with an 8-GeV SACLA linac. The CDR says, "*We know how it happens but we do not know why it happens. We should provide a tool to offer answers to the question, why*". This is the motivation of the SPring-8-II project.

The control system plays an essential role in the working of large accelerator facilities currently operational in the world. The controllability and operability of the facility strongly depend on the architecture and implementation of the control system; hence, the current control system of SPring-8 will be upgraded suitably to fulfil the performance requirements of the SPring-8-II storage ring, as described in this paper.

SPring-8-II Project

The CDR of the SPring-8-II project is available on the Web [1]. The basic idea is to replace the current 8-GeV storage ring with a low-emittance 6-GeV storage ring at the same location by reusing the present machine tunnel. The C-band linac of SACLA will be used as a full-energy injector for the new ring. To transport electron beams to the new ring, a beam transport line, XSBT, is constructed, as shown in Fig. 1. The X-ray beamlines for the present undulators will be retained. The blackout period (construction period) is expected to be one year or less.

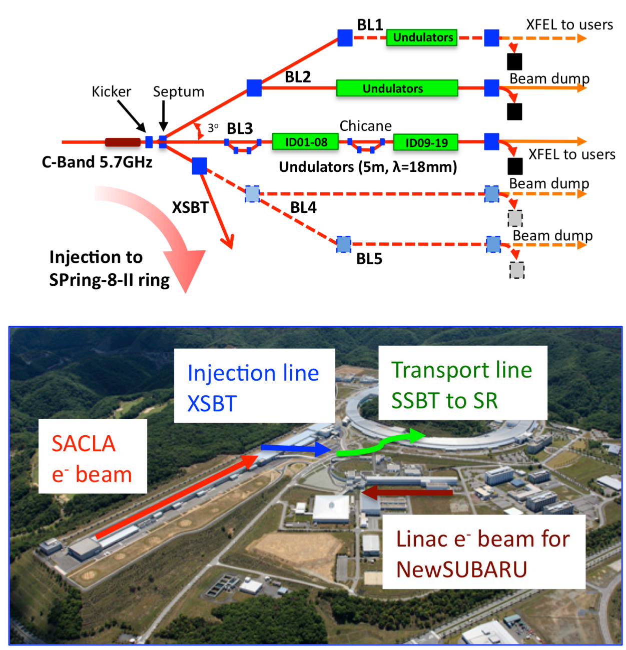

Figure 1: Injection line to SPring-8-II by XSBT at SACLA (upper), and beam transport route (lower).

ISBN 978-3-95450-146-5

COMMON DEVICE INTERFACE 2.0

P. Duval, H. Wu, DESY, Hamburg, Germany
J. Bobnar, Cosylab, Ljubljana, Slovenia

Abstract

The Common Device Interface (CDI) is a popular device layer [1] in TINE control systems [2]. Indeed, a de facto device server (more specifically a 'property server') can be instantiated merely by supplying a hardware address database, somewhat reminiscent of an epics IOC. It has in fact become quite popular to do precisely this, although the original design intent anticipated embedding CDI as a hardware layer within a dedicated device server. When control system client applications and central services communicate directly to a CDI server, this places the burden of providing useable, viewable data (and in an efficient manner) squarely on CDI and its address database. In its initial release variant, any modifications to this hardware database needed to be made on the file system used by the CDI device server (and only when the CDI device server was not running). In this report we shall describe some of the many new features of CDI release 2.0, which have drawn on the user/developer experience over the past eight years.

CDI AND TINE

Although the Common Device Interface (CDI) can be used outside of the TINE control system it is nonetheless strongly coupled to the TINE libraries as well as the TINE application programmer's interface (API) and the TINE naming convention and hierarchy. It is worthwhile to discuss some of these aspects in order to better understand the discussion that follows.

TINE itself does not require any specific hardware device layer in order to provide control system services. On the other hand TINE is weakly coupled to CDI in that specific CDI hooks are embedded in the TINE library. This in turn allows a TINE device server to utilize embedded CDI services for hardware access.

CDI Hardware Server

A very simple manifestation of embedded CDI services is the so-called CDI hardware server, which is essentially a generic TINE device server providing access to the hardware devices contained in the CDI database. Such a server provides no additional device control intelligence beyond that which can be configured in the database. Originally it was imagined that although the hardware server would be a very useful tool for testing hardware, developers would design device servers based on direct data acquisition via embedded services within a single framework. In practice, the hardware server itself has become the mainstay of hardware access for most TINE device servers. In most cases a device server with specific control intelligence is designed as an effective middle layer server communicating with a front-end CDI hardware server. In many cases, however, client applications communicate directly with CDI servers and there is no additional device server in the picture at all!

At this juncture we should point out that the CDI hardware server should properly be termed a property server and not a device server. Control systems are sometimes categorized into those which provide a database-driven paradigm (such as EPICS [3]) or a device-server paradigm (such as DOOCS [4] or TANGO [5]). Although TINE falls into the device-server camp it also supports property servers. Traditional device-servers treat instances of equipment as named devices and these devices have properties which one can access via the device server. A property server on the other hand considers services and information to be designated as properties located on some host, and such a service property will likely apply to a set of keywords.

The services a CDI server offers of course include bus access properties, such as sending and receiving on a hardware bus. For such properties, the keywords correspond directly to named hardware addresses, referred to as CDI devices. Other services include bus and template information as well as database management services.

CDI SPECIFICS

CDI operates on a plug-and-play basis, making use of bus-plug interfaces to the device hardware. A bus plug is a hardware specific shared library which encapsulates the details of the hardware bus I/O behind the CDI API. The CDI shared library is told which bus plug libraries to load via a CDI manifest database.

A CDI address database provides the cross-reference information necessary to instantiate the named hardware devices that CDI will export. The use of CDI address templates can greatly facilitate this instantiation. As a database format, CDI uses comma-separated value (CSV) files, which are easy to view in any text editor and fit seamlessly into any spreadsheet application such as Excel.

A CDI address database snippet is shown in Fig. 1. Here one can see how templates make life easy. Templates are defined by specifying the bus name 'TEMPLATE' and providing both the template name and template field name separated by a colon. If a device instance such as 'PU01I' specifies a template <BLM> in its address parameters then it will automatically expand into multiple CDI I/O devices given by instance name and template field name separated by a dot. The bus itself in this case is the in-house DESY bus SEDAC. The initial entry gives the special bus name 'FIELDBUS' which

ISBN 978-3-95450-146-5

INEXPENSIVE SCHEDULING IN FPGAS

W. W. Terpstra, D. Beck, M. Kreider, GSI, Darmstadt, Germany

Abstract

In the new scheme for machine control used within the FAIR project, actions are distributed to front-end controllers (FEC) with absolute execution timestamps. The execution time must be both precise to the nanosecond and scheduled faster than a microsecond, requiring a hardware solution. Although the actions are scheduled at the FEC out of order, they must be executed in sorted order. The typical hardware approaches to implementing a priority queue (CAMs, shift-registers, etc.) work well in ASIC designs, but must be implemented in expensive FPGA core logic. Conversely, the typical software approaches (heaps, calendar queues, etc.) are either too slow or too memory intensive. We present an approach exploiting the time-ordered nature of our problem to sort in constant time using only a few memory blocks.

INTRODUCTION

In a schedule-driven control system, pending actions include an execution timestamp. When the timestamp matches the current time, the responsible front-end controller (FEC) executes the action. Within the scope of the FAIR project, many physically distributed FECs will execute actions in concert. Actions are coordinated and distributed by a central unit, the data master, which controls beam production.

Unfortunately, the data master is quite complicated. The actions it requires a FEC to take may be delivered in an order different than the execution order. This paper describes how a FEC takes an incoming set of out-of-order actions and outputs them in sorted order. In principle, a single FEC may control many devices attached to many interfaces. However, for the purposes of this paper, we will concern ourselves only with the actions delivered by a FEC through a single interface. We also omit message processing.

Concretely, a FEC processes action tuples (a,x) , where a is the action to execute and x is the time at which it must be executed. At any given time t, the FEC has a set P_t of pending/yet-to-be-executed tuples. That is, $x \geq t$ for all $(a,x) \in P_t$. At time t, the FEC must output a if $(a,t) \in P_t$; this is illustrated in Figure 1. Obviously, it is physically impossible for a single interface to execute two actions concurrently. The data master would never ask a FEC to do something impossible. Therefore, we can assume $x = y \to a = b$ for $\{(a,x),(b,y)\} \subseteq P_t$.

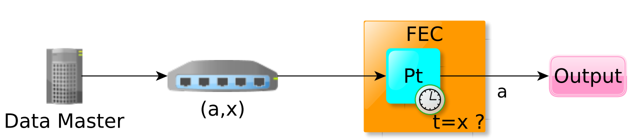

Figure 1: Actions (a,x) flow from the data master to the FECs. They are stored in P_t until $t = x$ and then output.

For FAIR, the control system is required to have nanosecond precision. This means that at least the execution/output of actions must be synchronized by hardware. Furthermore, the data master distributes the schedule via gigabit Ethernet. The FECs must be capable of accepting the schedule at the full rate, which suggests hardware may be required here as well. FECs include an FPGA and therefore we choose to solve the problem of receiving the schedule, sorting it, and outputting it, all in the FPGA.

AN FPGA CRASH COURSE

We can program an FPGA to contain customized hardware. Like an application-specific integrated circuit (ASIC), we can implement any digital circuit consisting of logic gates and registers. Unfortunately, FPGAs only have a limited number of comparatively slow logic elements to implement core logic (logic gates and registers). FPGAs also include many block memories, which are essentially small SRAM chips embedded inside the FPGA. These block memories have the same density and performance as they would in an ASIC. For these reasons, a good FPGA design tries to minimize core logic by leveraging block memory.

Digital logic is generally clocked. Registers take their values on the rising edge of a clock signal. In a modern FPGA with a reasonably complicated design, the clock speed is generally limited to <500MHz. That means that each clock cycle takes >2ns. The particular sorting circuit presented here can run at 325 MHz (≈3ns period) on an Altera Arria V chip. To achieve nanosecond precision, this means we need to be able to output an action in very few clock cycles. Furthermore, we need to accept new tuples at a similar rate.

Our goal is thus to accept a tuple (a,x) on clock cycle t to compute $P_{t+1} = P_t \cup \{(a,x)\}$. Furthermore, if there is $(a,t) \in P_t$, then we output a. In other words, on every clock cycle, we need to potentially accept a new action into the buffer and output the action whose timestamp is due.

APPROACH

Sorting a set of $n = |P_t|$ numbers requires $O(n \log(n))$ comparisons. If we must sort one timestamp every cycle, that means $\log(n)$ comparisons per cycle. For FAIR, timestamps are 64-bit numbers. In a modern FPGA, comparing even a single pair of 64-bit numbers in one clock cycle would reduce the maximum performance to ≈125 MHz. A hardware technique called pipelining can spread this work between multiple clock cycles. In our implementation, a 64-bit comparison is done in 3 steps to achieve the target performance. Unfortunately, this makes the comparator quite expensive. While there are techniques to spread out the work of all $\log(n)$ comparisons across multiple cycles using pipelining [1], these approaches cost significant hardware.

ISBN 978-3-95450-146-5

TESTBED − AUTOMATED HARDWARE-IN-THE-LOOP TEST FRAMEWORK

P. Maslov, K. A. Meyer, K. Žagar, Cosylab, Ljubljana, Slovenia

Abstract

In a big physics facility such as ITER or ESS, the control system is typically updated at least 3 times a year. This means that prior to each minor release all components should be tested. For testing DAQ drivers, a test plan should be written, based on which a manual test is performed. The idea behind the TestBed suite is to execute tests automatically. Our TestBed is a PXI chassis which contains an embedded controller running the CODAC control system on a Scientific Linux operating system and a DAQ board capable of generating and acquiring analog and digital signals. It provides an easy-to-use framework written in Python and allows for the quick development and execution of automatic test scripts.

ARCHITECTURE

From the hardware perspective, each system under test (SUT) is physically connected to the TestBed (TB) (Fig. 1) with a connector board using a predefined pin configuration. Both SUT and TB are connected to the LAN (not shown in Fig. 1).

The software part consists of three tiers:

1. Software that provides the desired functionality of a DAQ board:
   - C executables
   - EPICS device support (NDS [1] driver + IOC)
   - *LabVIEW interface*
2. Python bindings in the form of a class that reflect the given functionality of 1)
3. Automatic test cases written by the test-plan engineer using 2).

The NI-PXI6259 functionality that is supported in the TB suite includes:

- Analog input/output (static) on a desired channel
- Analog input (waveform) on a trigger
- Analog output (waveform – sine/saw/square/from file) on a trigger signal
- Configuration of the DIO port mask
- DIO diagnostics: port mask (0 – input, 1 – output) and lines state (0 – low, 1 – high)
- Digital input/output (static) on a desired line
- Device reset

The underlying connection protocol is SSH and is provided by the Python package Paramiko (the NDS implementation utilizes the Python package CaChannel).

TESTBED BASE CLASS

The Testbed Python class (Fig.2, left hand-side) provides a set of test methods to be executed on an SUT. These methods can be extended for any DAQ board and then used to quickly write automatic test scripts.

An example of such test case script showing three tests for the NI 6259 DAQ board (Fig.2, right hand-side) is:

1. test_aio_static: generate static signal on AO0 (TB), acquire static signal from AI0 (SUT), compare results.
2. test_dio_static: generate static signal on DO0 (TB), acquire static signal from DI0 (SUT), compare results.
3. test_aio_wf: generate a sine wave on AO0 (TB) on the trigger signal PFI1, acquire the waveform on AI0 (SUT) on the trigger signal PFI1, compare results.

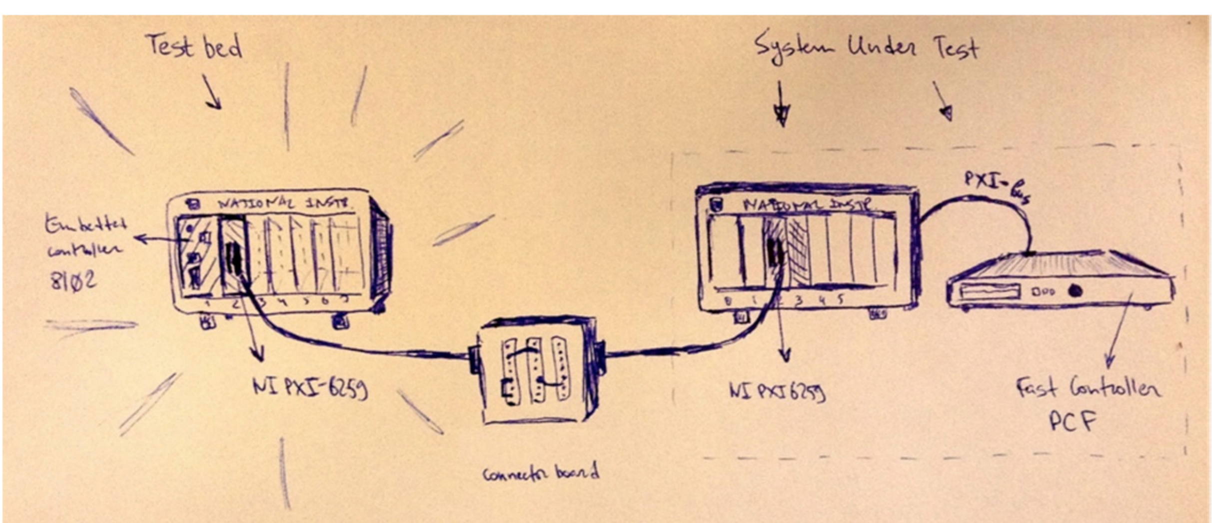

Figure 1: TestBed chassis is attached to the system under test.

ISBN 978-3-95450-146-5

LAUNCHING THE FAIR TIMING SYSTEM WITH CRYRING

M. Kreider, J. Bai, R. Bär, D. Beck, A. Hahn, C. Prados, S. Rauch, W. W. Terpstra, M. Zweig
GSI Helmholtzzentrum für Schwerionenforschung, D-64291 Darmstadt, Germany

Abstract

During the past two years, significant progress has been made on the development of the General Machine Timing (GMT) system for the upcoming FAIR facility at GSI. The primary features are time-synchronization of 2000-3000 nodes using the White Rabbit Precision Time Protocol (WR-PTP), distribution of International Atomic Time (TAI) timestamps and synchronized command and control of FAIR control system equipment.

A White Rabbit network has been set up connecting parts of the existing facility. A next generation of the Timing Master has been developed. Timing Receiver nodes in the form factors Scalable Control Unit (standard front-end controller for FAIR), VME, PCIe and standalone have been developed. CRYRING is the first machine on the GSI/FAIR campus to be operated with this new timing system and serves as a test-ground for the complete control system. Installation of equipment starts in late autumn 2014 followed by commissioning of equipment in winter.

INTRODUCTION

The primary task of the General Machine Timing (GMT) system is the hard real-time control of the GSI and FAIR accelerator complex with sub-ns precision [1]. This is a two-step process. First, the *Settings Management* [2] distributes multiple settings to concerned Front-End Computers (FECs) via the normal network of the accelerator. Activities[1] are prepared by the Front-End Software FESA [3]. Second, the GMT generates on-time actions at the FECs. Such an action triggers a prepared activity at the FEC and provides an index referencing one of the preloaded settings.

The fundamental idea behind the GMT is the concept of time-based control. The distribution of information and timely execution of activities are decoupled. As a prerequisite, all nodes of the GMT share a common notion of time provided by WR-PTP [4] via the dedicated White Rabbit network (timing network). The central component of the GMT is the Data Master (DM) [5]. At some time the DM receives an (updated) schedule recipe for the operation of the facility from the settings managements. This recipe only contains indices to and time intervals between actions. Based on this recipe, the DM controls the facility in hard real-time. This is again a two-step process. First, the DM broadcasts timing messages including the indices, but with absolute execution time stamps, via the timing network to the Timing Receivers (TR) embedded in the FECs. The messages must be distributed with an upper bound latency. Second, messages are received by the TRs where they are filtered. Relevant messages remain pending until the specified execution time when the TR performs the action. Depending on the configuration of the TR, such an action could be digital signal generation, complex activity such as ramping a radio-frequency system, or signaling an event to the front-end software via an interrupt request (IRQ).

In 2014 the primary features have been implemented in such a way that the GMT and other components of the FAIR control system can be used coherently for the control of a real machine like CRYRING. This synchrotron has in the meantime moved from its original site in Stockholm to GSI and is presently installed in a refurbished cave behind the existing Experimental Storage Ring (ESR). While the infrastructure for the installation of the control system is presently being completed, the GMT components relevant for the recommissioning of CRYRING have been implemented and tested. This paper reports on the on-going work and summarizes the present situation.

ASTERISK

For the nodes of the GMT, the timing team at GSI supports a variety of hardware types and functionality. The interfaces of the GMT provide a common “look and feel” to the users, hiding the complexity and differences between the form factors to a large extent. Although some features (e.g. a display) may not exist on all form factors, common functionality must be presented identically at the interfaces.

To guarantee these requirements, the timing team builds releases. Such a release includes hardware, gateware (FPGA code), firmware (embedded CPU code) as well as software (host system code) in a consistent way. The first version, named *Asterisk*, has been released in July 2014. It includes all features of the GMT required for the next milestones of the whole FAIR control system.

Hardware

Asterisk includes four form factors. The Scalable Control Unit (SCU) is the standard FEC for the FAIR control system and has been developed by the hardware section of the control system department [6]. Three other form factors have been developed by the department of Experiment Electronics and are intended for usage by the department of Beam Instrumentation as well as Data AcQuisition (DAQ) systems of FAIR experiments. The most important one is the PCIe module PEXARIA5, since this module represents the reference implementation based on an ARRIA V FPGA for all form factors provided through the GSI timing team. The two remaining modules, the VME board VETAR and the standalone form factor EXPLODER are still based on ARRIA II FPGAs.

[1] Example: Ramping of a magnet.

ISBN 978-3-95450-146-5

TCP/IP CONTROL SYSTEM INTERFACE DEVELOPMENT USING MICROCHIP BRAND MICROCONTROLLERS*

Christopher E. Peters#, Maria A. Power, ANL, Argonne, IL 60439, USA

Abstract

Even as the diversity and capabilities of Single-Board-Computers (SBCs) like the Raspberry Pi and BeagleBoard continue to increase, low level microprocessor solutions also offer the possibility of robust distributed control system interfaces. Since they can be smaller and cheaper than even the least expensive SBC, they are easily integrated directly onto printed circuit boards either via direct mount or pre-installed headers. The ever increasing flash-memory capacity and processing clock speeds has enabled these types of microprocessors to handle even relatively complex tasks such as management of a full TCP/IP software and hardware stack. The purpose of this work is to demonstrate several different implementation scenarios wherein a computer control system can communicate directly with an off-the-shelf Microchip brand microcontroller and its associated peripherals. The microprocessor can act as a Hardware-to-Ethernet communication bridge and provide services such as distributed reading and writing of analog and digital values, webpage serving, simple network monitoring and others to any custom electronics solution.

MOTIVATION AND THEORY

The Argonne Tandem Linac Accelerator System (ATLAS) is located at the United States Department of Energy's Argonne National Laboratory in the suburbs of Chicago, Illinois. It is a National User Facility capable of delivering ions from hydrogen to uranium for low energy nuclear research in order to perform physics analysis of the properties of the nucleus. In support of this goal, the accelerator control system and its electronics hardware have been in a state of continuous upgrade since its transition to computer control in the late 1980s [1]. In addition, the beam transport hardware and support electronics equipment has been in a constant state of change. The interfaces into the control system range from simple 0-10v analog input signals, to 50+ bit binary coded decimal, to all manner of digital communication protocols (RS-232/485, GPIB, USB, SNMP, ModBUS).

These signal types can require inefficient cable solutions, sometimes requiring hundreds of separate conductors to transmit relatively slow changing (sub ~10Hz) and relatively low accuracy requirement (2-4 significant figures) beam transport control channel points. Specifically, these types of channels stand in contrast to the high speed, high accuracy data acquisition and timing channels used by the target and detectors groups.

This work was supported by the U.S. Department of Energy, Office of Nuclear Physics, under Contract No. DE-AC02-06CH11357. This research used resources of ANL's ATLAS facility, which is a DOE Office of Science User Facility. *microchip.com #ChrisPeters@anl.gov

The result is a mismatch between the modest channel speed and accuracy requirements and the high cost of implementation. The goal of this work is not only to define a low cost control system hardware interface, but also to push that platform lower down into the custom hardware components currently developed at ATLAS (Figure 1). This will provide better focus for hardware designers to only work directly on electronics separate from the high level interface, and allows the controls group to rely more on generic and reusable interfaces.

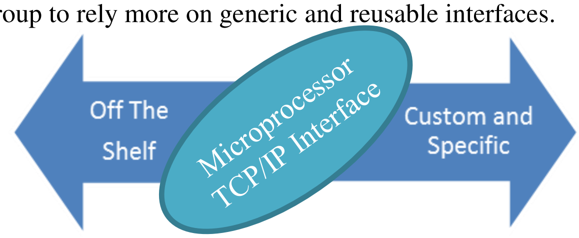

Figure 1: Microprocessor Based Interfaces Exist in the Middle of the Spectrum of Hardware Design Complexity.

MICROPROCESSOR COMPARISON TO SINGLE BOARD COMPUTERS

The goal of this work is to investigate the advantages and disadvantages of using the latest in microprocessor based platforms in a distributed control system environment. Single Board Computers (SBCs) have increasingly been viewed as a way to add various communication protocols [2] to hardware. These devices can interface with simple electronics via on-board ADCs and PWM/DACs, and also act as complex data processing platforms in their own right. This paper specifically focuses on the Microchip brand [3] and the OEM produced PIC32 Ethernet Starter Kit (PIC32 or ESK).

It is often difficult to directly compare the offerings of common SBCs like the BeagleBoard Black [4], the Raspberry Pi [5], and Industrial PCs like PC/104 [6] and microcontrollers, since the selection of each platform is often highly dependent on the application. Our goal is to determine if the more limited and focused capabilities of microprocessors are matched well to implementation into low-level custom built beam transport electronic devices.

PRODUCTS AND INITIAL EXPERIENCES

The initial steps in this research were to determine what possible COTS offerings were available to push TCP/IP communication down into the hardware design level. Initial experimentation was performed using a starter kit, however all components are available for installation on any custom developed PCB hardware.

ISBN 978-3-95450-146-5

THE ROLE OF THE CEBAF ELEMENT DATABASE IN COMMISSIONING THE 12 GeV ACCELERATOR UPGRADE*

T. Larrieu, M. Joyce, M. Keesee, C. Slominski, D. Turner, JLAB, Newport News, VA 23606, U.S.A.

Abstract

The CEBAF Element Database (CED) was first developed in 2010 as a resource to support model-driven configuration of the Jefferson Lab Continuous Electron Beam Accelerator (CEBAF). Since that time, its uniquely flexible schema design, robust programming interface, and support for multiple concurrent versions has permitted it to evolve into a more broadly useful operational and control system tool.

The CED played a critical role before and during the 2013 startup and commissioning of CEBAF following its 18-month long shutdown and upgrade. Information in the CED about hardware components and their relations to one-another facilitated a thorough Hot Checkout process involving more than 18,000 system checks.

New software relies on the CED to generate EDM screens for operators on-demand thereby ensuring that the information on those screens is correct and up-to-date. The CED also continues to fulfil its original mission of supporting model-driven accelerator setup. Using the new ced2elegant and eDT (elegant Download Tool), accelerator physicists have proven able to compute and apply energy-dependent set points with greater efficiency than ever before.

BACKGROUND

The Jefferson Lab CEBAF accelerator is a superconducting recirculating linear accelerator capable of delivering continuous wave electron beams simultaneously to multiple experimental halls. Previously capable of delivering 6GeV, CEBAF recently underwent an extensive multi-year $338M upgrade project to double its energy capacity to 12GeV and add a fourth experimental hall (fig. 1). The upgraded accelerator was commissioned successfully between November 2013 and May 2014. The CEBAF Element Database (CED) was an indispensable tool during the commissioning.

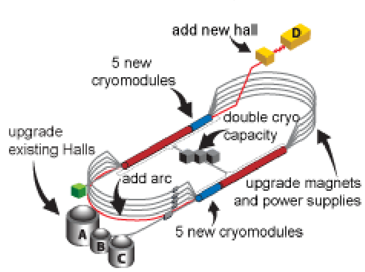

Figure 1: The CEBAF 12GeV Upgrade Project.

* Notice: Authored by Jefferson Science Associates, LLC under U.S. DOE Contract No. DE-AC05-06OR23177. The U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce this manuscript for U.S. Government purposes.

DATABASE DESIGN

The CED design is based upon a modified Entity-Attribute-Value with Classes and Relationships (EAV/CR) data model [1]. Whereas in a traditional schema adding support for new accelerator hardware would involve adding additional tables and columns to the database, the EAV/CR data model employed by the CED is introspective – defining a new class of accelerator hardware in the CED simply involves adding rows to the already-existing metadata "catalog" tables. Once defined in the catalog, existing software is fully capable of interacting with the new entities after discovering their properties from the metadata tables.

A major benefit of the EAV/CR data model is that its static schema is well suited to integration with Oracle Workspace Manager [2] to provide timestamp-based versioning of table rows, named save points, and multiple independent workspaces (akin to branches in a software version control system).

DATABASE IMPLEMENTATION

Platform

At Jefferson Lab, the CED is implemented using version 11g Standard Edition Oracle Relational Database. The database server runs Redhat Enterprise Linux 6 on x86_64 hardware. The current API library is compiled solely for 32-bit x86 architecture, though prior versions were also compiled for the Solaris 10 Sparc and HP-UX PA-RISC platforms.

Layout

To optimize performance for the different use cases, the CED database instance is distributed among three database users/schemas as shown in Figure 2. The operational schema stores the current machine configuration and is optimized for performance. The historical schema provides efficient storage and access to (read-only) historical save points (snapshots). And the development schema is used to create workspace branches where data can be edited and prepared before being promoted to the operational schema.

ISBN 978-3-95450-146-5

InfiniBand INTERCONNECTS FOR HIGH-THROUGHPUT DATA ACQUISITION IN A TANGO ENVIRONMENT

T. Dritschler*, S. Chilingaryan*, T. Farago†, A. Kopmann*, M. Vogelgesang*
*Institute for Data Processing and Electronics
†Institute for Photon Science and Synchrotron Radiation
Karlsruhe Institute of Technology, Germany

Abstract

Advances in computational performance allow for fast image-based control. To realize efficient control loops in a distributed experiment setup, large amounts of data need to be transferred, requiring high-throughput networks with low latencies. In the European synchrotron community, TANGO has become one of the prevalent tools to remotely control hardware and processes. In order to improve the data bandwidth and latency in a TANGO network, we realized a secondary data channel based on native InfiniBand communication. This data channel is implemented as part of a TANGO device and by itself is independent of the main TANGO network communication. TANGO mechanisms are used for configuration, thus the data channel can be used by any TANGO-based software that implements the corresponding interfaces. First results show, that we can achieve a maximum bandwidth of 30 Gb/s which is close to the theoretical maximum of 32 Gb/s, possible with our 4xQDR InfiniBand test network, with average latencies as low as 6 µs. This means that we are able to surpass the limitations of standard TCP/IP networks while retaining the TANGO control schemes, enabling high data throughput in a TANGO environment.

INTRODUCTION

PC-based systems have made significant advances in stability and computational power and have become viable for usage as control systems even in large installations. Driven by this influx of 'off-the-shelf' PC systems, the ESRF started to develop its own remote control system called TANGO [1] in 2003. TANGO provides transparent and uniform access to all devices on the control network. It uses the CORBA communication layer to send function calls to any remote device that provides a TANGO interface.

Since TANGO was created mainly as a control system for the synchrotron community and is distributed as an open source system, it found broad acceptance across the European synchrotron community and is now actively being developed by a large consortium, lead by ESRF. Eventually, the ANKA synchrotron at KIT has also decided to extensively use TANGO as one of the control system for their beamlines.

The new IMAGE beamline is being constructed at ANKA to allow ultra fast 3D tomography with near real-time monitoring and fast image-based control loops. The beamline development aims to permit investigations of the internal morphology and structural changes in small living organisms in 4D (3D + time) with micrometer spatial resolution and sub-second time resolution [2]. The required resolution is achieved with high-speed cameras providing over a thousand frames per second and with streaming bandwidth ranging from a few hundred Megabytes up to multiple Gigabytes per second. The image-based control is made feasible by the UFO-framework [3]. Running on GPU-based computational servers, it is able to process a few Gigabytes of tomographic data per second. The efficient delivery of this data to the computation nodes, though, is a challenge.

To comply with ANKA standards, the control system for the IMAGE beamline [4] is based on TANGO to communicate with the beamline hardware and the controls for the pixel sensors/cameras are exposed using TANGO modules. Since TANGO transports all its data using CORBA with standard TCP/IP communication, its bandwidth is not only limited by the speed of the network adapter, but also by the performance of the TCP/IP stack, especially in terms of latency [5]. Also, the effective bandwidth is further limited by the particular implementation of CORBA (omniORB [6]) used by TANGO.

In this paper, we present our approach to extend the already existing TANGO infrastructure with a secondary high-bandwidth data channel. We use InfiniBand interconnects to avoid latency and bandwidth limitations of standard TCP/IP over Ethernet. A TANGO enabled camera driver was developed that allows to transport camera data through a secondary data channel based on InfiniBand RDMA communication. With this new data channel and the TANGO control system, coupled together, a control interface was created that replicates the remote TANGO camera interface into a local system while using InfiniBand as transfer for the camera data. This effectively combines high throughput data with the flexibility of TANGO while feeling and behaving like a purely local camera system.

DATA TRANSPORT PERFORMANCE

We evaluated the performance of TANGO and omniORB under certain scenarios. For this, a TANGO device server was written that generates pictures of 512x512 pixels (265 KB data) with a static gradient pattern and a region of random noise data in the center. These pictures can then be pulled from the server over the standard TANGO interface. We connect to this device server with a device proxy client that continuously pulls the picture attribute from the device server for a certain number of iterations and creates a mean value of the observed throughput by dividing the transfered

ISBN 978-3-95450-146-5

PICOSECOND SAMPLING ELECTRONIC FOR TERAHERTZ SYNCHROTRON RADIATION

M. Caselle, M. Balzer, M. Brosi, S. Chilingaryan, T. Dritschler, V. Judin, A. Kopmann, A.-S. Müller, J. Raasch, L. Rota, L. Petzold, N. J. Smale, J., L. Steinmann, M. Vogelgesang, S. Wuensch, M. Siegel, M. Weber KIT, Karlsruhe, Germany

Abstract

The ANKA storage ring generates brilliant coherent synchrotron radiation (CSR) in the THz range due to a dedicated low-α_c-optics with reduced bunch length. At higher electron currents the radiation is not stable but is emitted in powerful bursts caused by micro-bunching instabilities. This intense THz radiation is attractive for users. However, due to the power fluctuations, the experimental conditions cannot be easily reproduced. To study the bursting CSR in multi-bunch operation an ultra-fast and high-accuracy data acquisition system for recording of individual ultra-short pulses has been developed. The Karlsruhe Pulse Taking Ultra-fast Readout Electronics (KAPTURE) is able to monitor all buckets turn-by-turn in streaming mode.

KAPTURE provides real-time sampling of the pulse with a minimum sampling time of 3 ps and a total time jitter of less than 1.7 ps. In this paper we present the KAPTURE system, the performance achieved and the integration in the ANKA control system.

INTRODUTION

At the ANKA synchrotron light source, up to 184 electron buckets can be filled with a distance between two adjacent bunches of 2 ns corresponding to the 500 MHz frequency of the accelerating RF system.

Since a few years, special user operation with reduced bunch length in the order of a few picoseconds has been available to research communities. In this mode, coherent synchrotron radiation is generated for electro-magnetic waves with a wavelength in the order of or longer than the electron bunch length. Due to this, we observe a strong amplification of the radiation spectrum in the THz band. Moreover, above a certain current threshold, a coherent modulation of the longitudinal particle distribution (microbunching) occurs due to CSR impedance [1]. This particle dynamic effect changes the characteristics of the CSR tremendously. The microbunching structures fulfil a coherence condition for shorter wavelengths. This leads to an instantaneous increase of the radiated THz power. Observations in the time domain show bursts of radiation that occur with different periodicities depending on the bunch current. The characteristics of the bursting patterns are unique for different sets of accelerator parameters [2].

The KAPTURE (KArlsruhe Pulse Taking and Ultrafast Readout Electronics) system opens up the possibility to monitor the THz radiation of all bunches in the ring over a principally unlimited number of turns, realising a new type of measurement at ANKA. In this paper we present the KAPTURE system and the integration in the ANKA environment.

KAPTURE SYSTEM

The KAPTURE system records individual pulses continuously with a sub-millivolt resolution and a timing resolution in the order of picoseconds. KAPTURE is a flexible system and can be easily configured for the requirements of any synchrotron facility.

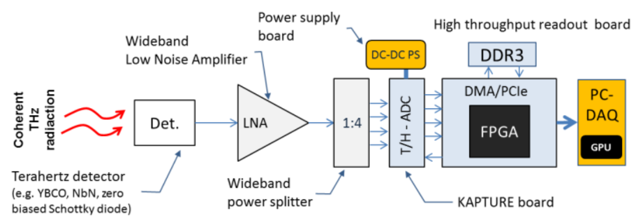

Figure 1: KAPTURE system for the detection of coherent THz radiation generated at ANKA.

The architecture of the KAPTURE system is shown in Fig. 1. It consists of a Low Noise Amplifier (LNA), a power splitter, a picosecond pulse sampling stage called "KAPTURE board", a high throughput readout board and a high-end Graphics Processing Unit (GPU). The signal from the detector is fed into a LNA and then divided in four identical pulses by a wideband power splitter. Matching of the high bandwidth required to design a novel wideband power divider architecture [3]. The LNA gain compensates the insertion loss due to the power divider with a minimal additional noise. The KAPTURE board consists of four parallel sampling channels each operating at 500 MS/s [4]. Each sampling channel receives one of the four pulses from the power splitter and acquires it with one sampled point. The sampling time between the channels is settable with an accuracy of 3 picoseconds. The final result is that each detector pulse will be sampled with 4 sample points at a programmable sampling time between 3 and 100 ps. The basic concept and the architecture of the picosecond KAPTURE board have been reported previously [5]. The high throughput readout board uses a new bus master DMA architecture connected to PCI Express logic [6] to transfer the digital samples from the KAPTURE board to a high-end GPU server. For continuous data acquisition a bandwidth of 24 Gb/s (12 bits @ 2 ns * 4 digital samples) is necessary. The DMA architecture has been developed to meet this requirement with a high data throughput of up to 32 Gb/s. The GPU computing node is used for real-time reconstruction of the pulse from the 4 digital samples. Afterwards, the peak amplitude of each pulse and the time

ISBN 978-3-95450-146-5

INTEGRATION OF INDEPENDENT RADIATION MONITORING SYSTEM WITH MAIN ACCELERATOR CONTROL

N. Kamikubota#, N. Yamamoto, J-PARC, KEK & JAEA, Tokai-mura, Ibaraki, Japan
T. Iitsuka, S. Yoshida, Kanto Information Service, Tsuchiura, Ibaraki, Japan

Abstract

The radiation monitoring system of J-PARC was constructed as a part of safety facilities. It is isolated, and has been operated independently from the main accelerator control.

In 2013, integration of the radiation monitoring system with the main accelerator control was discussed. In order not to affect to the original safety system, standard TCP/IP network connections and accesses to the central database are not allowed. During 2013-2014, we added new hardware to the existing systems, and developed device-level data-link layers to enable one-way data transfer to the accelerator control system.

In 2014, radiation monitoring data can be supervised from the accelerator control system. We understand that this is a significant improvement to realize safer operation of J-PARC accelerators and experimental facilities.

INTRODUCTION

An accelerator facility is always associated by related facilities. For example, safety facilities (a personal protection system, a radiation safety system, etc.), utilities (water cooling system, electricity distribution facility, etc.), and so on. In J-PARC accelerator complex, the radiation safety system was constructed by non-accelerator group, and had been operated independently. It has an isolated network, dedicated terminals for data view and history retrieve. Behind the fact, there exists a strong policy: “safety systems must be independent”.

In J-PARC, the radiation monitoring system is a part of the safety system. When accelerator operators and/or staff members wanted to know radiation levels of monitoring posts, they had to visit the radiation safety office. Checking of radiation monitoring data from the central control room was not possible.

SCHEME FOR DATA-SHARING

J-PARC Accelerator Complex

J-PARC (Japan Proton Accelerator Research Complex) is a high-intensity proton accelerator complex. It consists of three accelerators: a) 400-MeV linac (LI), b) 3-GeV Rapid Cycling Synchrotron (RCS), and 30-GeV Main Ring (MR). Addition to them, there are three experimental facilities: d) Material and Life Science Experimental Facility (MLF), e) Neutrino Experimental Facility (NU), and f) Hadron Experimental Facility (HD). J-PARC was constructed and has been operated jointly between two institutes: JAEA and KEK [1,2].

#norihiko.kamikubota@kek.jp

Accident of Hadron Experimental Facility

On May 23, 2013, we had a serious accident at the Hadron Experimental Facility. Radioactive materials were released out of the radiation controlled area. The dose level was estimated less than 0.17 uSv on the site boundary closest to the Hadron Experimental Facility. Moreover, 102 workers were exposed by uncontrolled radioactive materials. Out of them, 34 persons were found to receive radiation dose in the range of 0.1-1.7mSv, all below the legal limit [3,4].

A team of third-party experts (the External Expert Panel) was established in order to review and inspect the accident. In August, 2013, the official report, including preventive measures against recurrence of similar accidents, was submitted to the Nuclear Regulation Authority of Japan [5]. It reflects the recommendations of the Panel. In the report, lack of sharing of radiation monitoring information was pointed out.

Policies and Plans for Implementations

We discussed a method to enable sharing of radiation monitoring data between the radiation monitoring system and the accelerator control system. In order to keep the independency of the safety system, TCP/IP network connections are not allowed. In addition, accesses to the central database of the safety system are prohibited, to avoid increases of system loads. We accepted above policies as same as before.

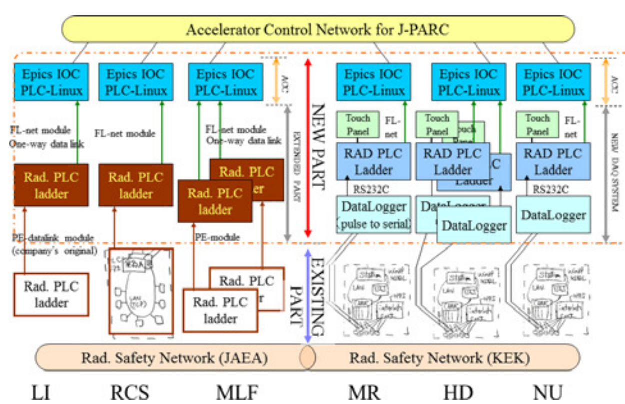

Figure 1: Overview of the data-sharing scheme.

We investigated the existing radiation monitoring systems. Our idea is that data-links to the device-level layers are possible without influences on the original safety system. The overview of the implementation scheme is shown in Figure 1.

ISBN 978-3-95450-146-5

LabVIEW PCAS INTERFACE FOR NI CompactRIO

G. Liu*, J. Wang, K. Xuan, C. Li, NSRL, USTC, Hefei, Anhui 230029, China
K. Zheng, Y. Yang, National Instruments China, Shanghai 201203, China

Abstract

When the NI LabVIEW EPICS Server I/O Server is used to integrate NI CompactRIO devices running under VxWorks into EPICS, we notice that it only supports "VAL" field, neither alarms nor time stamps are supported. In order to overcome these drawbacks, a new LabVIEW Channel Access Portable Server (PCAS) Interface is developed, and is applied to the Hefei Light Source (HLS) cooling water monitor system. The test results in the HLS cooling water monitor system indicate that this approach can greatly improve the performance of the NI CompactRIO devices in EPICS environment.

INTRODUCTION

The Hefei Light Source (HLS) is a dedicated second generation VUV synchrotron radiation light source. It was designed and constructed two decades ago. In order to improve the performance of the HLS, especially to obtain higher brightness and more straight sections, an upgrade project was started from the end of 2009. The cooling water monitor system was reconstructed in the project. The CompactRIO (Compact Reconfigurable Input Output) product from National Instruments (NI) is adopted to the cooling water monitor system because it is designed for harsh environment, and the software is easy to duplicate and deployed.

The NI CompactRIO devices are running under VxWorks. When the NI LabVIEW EPICS Server I/O Server [1] is used to integrate the NI CompactRIO devices into EPICS, we notice that it only supports "VAL" field, neither alarms nor time stamps are supported. In order to overcome these drawbacks, a new LabVIEW Channel Access Portable Server (PCAS) [2] Interface is developed.

The LabVIEW PCAS Interface is a software library which provides channel access server ability to LabVIEW code and emulates the way how an IOC manages process variables. It is based on the PCAS which implements the underlying channel access functions. There are two ambitious aims in designing the LabVIEW PCAS interface: (1) it supports all platforms where LabVIEW exists, including Windows, Linux and VxWorks; (2) it supply common records with common fields, similar usage pattern as IOC.

DEVELOPMENT

The LabVIEW PCAS Interface employs the Channel Access Portable Server distributed with EPICS base. The portable server comprises of server library and server tool. The server library refers to the software that lies beneath the C++ class-interface to the portable server. The developer only needs to know how to use the interface in order to create a server tool. A server tool is a specific application written by a developer using this interface.

There is no record in the portable server, all PVs are standalone data. In order to provide users an IOC-like interface, we encapsulate PVs in the format of <record name>.<field name>, e.g. Temp.VAL.

Figure 1 is the flowchart of a server application, including the VI interfaces and the background thread. The VI interfaces are functions we provide to users. The background thread implements all the supported CA server interfaces called by CA clients. The records store all data used by the server application. In the background thread, the Channel Access is handled by the portable server. We implement the callbacks required by underlying components.

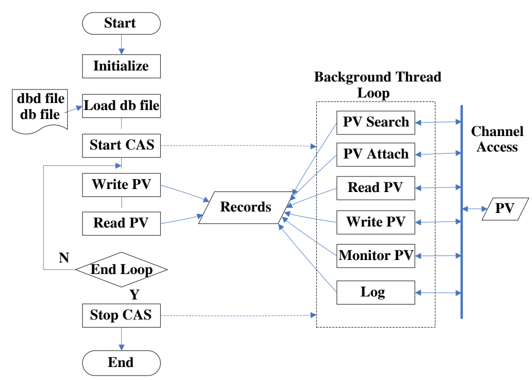

Figure 1: Flowchart of the LabVIEW PCAS Interface.

There are many record types defined by EPICS base, but we only support some common records with common fields up to now in order to limit the work load. Table 1 shows the supported record types and field. Here, "all" refers to the record type ai/ao, bi/bo, stringin/stringout and waveform.

Table 1: Record Types and Fields Supported by the LabVIEW PCAS Interface

Record Type	Fields
all	VAL/STAT/SEVR
ai/ao	HIHI/LOLO/HIGH/LOW HHSV/LLSV/HSV/LSV HYST/ADEL/MDEL
bi/bo	ZSV/OSV/COSV
waveform	NELM
ai/ao/waveform	HOPR/LOPR/PREC/EGU

*gfliu@ustc.edu.cn

HLS POWER SUPPLY CONTROL SYSTEM BASED ON VIRTUAL MACHINE

J. Wang*, G. Liu, K. Xuan, C. Li, NSRL, Hefei, China

Abstract

The Hefei Light Source (HLS) is a VUV synchrotron radiation light source. It is upgraded recently to improve its performance. The power supply control system is a part of the HLS upgrade project. Five soft IOC applications running on the virtual machine are used to control 190 power supplies via MOXA's serial-to-Ethernet device servers. The power supply control system has been under operation since November 2013, and the operation results show the power supply control system is reliable and can satisfy the demands of slow orbit feedback with the frequency of 1Hz.

INTRODUCTION

The Hefei Light Source (HLS) is a VUV synchrotron radiation light source. It is upgraded recently to improve its performance. As a part of the HLS upgrade project, all the power supplies are rebuilt, and the power supply control system is correspondingly reconstructed.

There are 190 power supplies totally. They are divided into about ten types, and used for dipole magnet, quadrupole magnet and sextupole magnet, etc. All these power supplied are designed with the unified control interface. Five soft IOC applications running on the virtual machine are used to control these power supplies via MOXA's serial-to-Ethernet device servers.

The power supply control system has been under operation since November 2013, and the operation results show that the power supply control system is reliable. The communication time is less than 50 ms, it can satisfy the demand of the slow orbit feedback with the frequency of 1Hz.

HARDWARE

The power supply control system is developed under EPICS environment, its hardware structure is shown in Figure 1.

Five softIOCs are running on the virtual machines, which is built with VMware. They communicate the power supplied via MOXA's serial-to-Ethernet device servers Nport 6650-16. All the power supplies has the unified interface, i.e. serial port with plastic fibre connection, the baud rate is 115.2kbps. A photoelectric converter with 16 ports is specially designed for MOXA Nport 6650-16, and is used between the serial device servers and the power supplies.

All the IOC applications are put on a NFS server, each softIOC is used as NFS client to share the IOC applications.

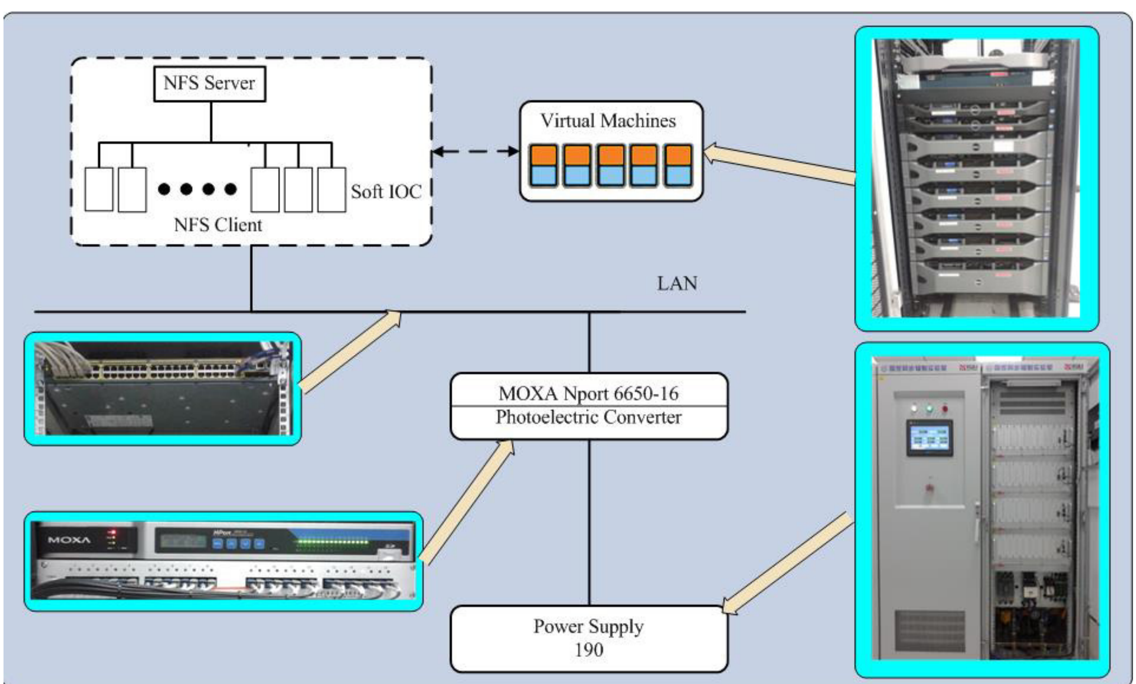

Figure 1: Hardware structure of power supply control system.

*wangjg@ustc.edu.cn

ISBN 978-3-95450-146-5

THE SOFTWARE TOOLS AND CAPABILITIES OF DIAGNOSTIC SYSTEM FOR STABILITY OF MAGNET POWER SUPPLIES AT NOVOSIBIRSK FREE ELECTRON LASER

S. S. Serednyakov
Budker Institute of Nuclear Physics SB RAS, Novosibirsk, Russia

Abstract

The magnetic system of Novosibirsk free electron laser (FEL), which contains a lot of magnetic elements, is fed by power supplies of different types. The time stability of the output current of these power supplies directly influences the coherent radiation parameters and operation of the entire FEL facility. In this connection, we developed a system for diagnostics of the power supplies state, integrated into the common FEL control system. The main task of this system is to analyze the output current of the power supplies and calculate their time stability characteristics. Besides, this system is capable to determine the amplitude and frequency of output current ripples, if any, for a particular power supply and display obtained results. This paper describes the main architecture and some other capabilities of this system, and presents examples of its usage.

INTRODUCTION

A high-power FEL based on a multiturn energy recovery linac (ERL) [1] is under construction now at Budker Institute of Nuclear Physics. The first and second phases of the project have already been commissioned and are in operation.

The magnetic system, one of most important systems of the ERL, consists of numerous magnetic elements of different types. All these elements are fed by DC power supplies of different output powers and currents. The stability of operation of the FEL and its main parameters depends on the time stability characteristics of these power supplies. In this connection, there was developed a set of particularized software tools for monitoring of the output currents of all the power supplies. The architecture of the current system is influenced by the factors below.

1. The architecture of the software of the control system for the magnetic system and the installation as a whole.
2. The type and operation characteristics of the communication bus between the control PC and the power supply control devices.
3. The capabilities and main operation modes of these power supply control devices.

HARDWARE AND ARCHITECTURE OF CONTROL SYSTEM OF MAGNETIC ELEMENTS

The magnetic system and its control system of Novisibirsk FEL are described in detail in [2]. All power supplies are controlled by modules embedded into power supply racks and communicating with the central IBM-PC via a CAN bus interface. Power supplies of different types are governed by different controllers. For low-power current sources with an output current of up to 17 amperes, pairs of multi-channel controllers are used, controlling up to 16 power supplies – 16-channel DAC and 40-channel ADC devices. For high-power current sources, each power supply is commanded by a separate controller with 1-channel DACs and 5-channel ADCs. All controllers are connected to one CAN bus line [Fig. 1].

Table 1: Parameters of the Power Supply Controllers

Device	*CANDAC16*	*CANADC40*	*CDAC20*
Qty	24	24	13
PS Number	16	16	1
PS Imax	17	17	2500
ADC channels	---	40	5
DAC channels	16	---	1

Above is presented a comparative table of main controller parameters (Table 1.). As seen, all controllers containing an ADC module have multiple input channels. In the regular operation mode, these modules are working in the multi-channel mode, i.e. serial measurement of voltage in all input channels and sending of the values measured to the control PC via the CAN bus interface. In this case, the ADC works in the cyclic mode: once the measurement of the last channel is completed, a new measurement cycle begins from the first channel. The architecture of the set of commands for all these controllers and the CAN bus protocol allow operation of numerous controllers in such mode.

For diagnostics purposes, another measurement mode – "single-channel mode" – was realized. When switched to this mode, the controller starts serial measurements of a single selected channel, until it receives from the central PC a command "STOP" or a command "start new measurement cycle". As mentioned above, switching

ISBN 978-3-95450-146-5

PyPLC, A VERSATILE PLC-TO-PC PYTHON INTERFACE

S. Rubio-Manrique, G. Cuní, D. Fernandez-Carreiras, Z. Reszela, A. Rubio,
CELLS-ALBA Synchrotron, Barcelona, Spain

Abstract

The PyPLC Tango Device Server provides a developer-friendly dynamic interface to any Modbus-based control device. Raw data structures from PLC are obtained efficiently and converted into highly customized attributes using the python programming language. The device server allows to add or modify attributes dynamically using single-line python statements. The compact python dialect used is enhanced with Modbus commands and methods to prototype, simulate and implement complex behaviours. As a generic device, PyPLC has been versatile enough to interact with PLC systems used in ALBA Accelerators as well as to our Beamlines SCADA (Sardana). This article describes the mechanisms used to enable this versatility and how the dynamic attribute syntax allowed to speed up the transition from PLC to user interfaces.

INTRODUCTION

ALBA[1], member of the Tango Collaboration[2][3], is a third generation Synchrotron in Barcelona, Europe. It provides light since 2012 to users through its 7 beamlines,with 2 more under construction.

Programmable Logic Controllers from several vendors (B&R, Pilz, …) are used for acquisition, protection and motion within our Tango Control System[4]. PLC's are the main component of equipment and personnel protection systems, but they are also used in accelerators and beamlines for vacuum/temperature diagnostics and motion control. The most complex system managed by PLC's is the Equipment Protection System (EPS)[5].

Equipment Protection System at ALBA

The EPS is an autonomous system ensuring the safe operation of all elements in ALBA accelerators and Beamlines. It generates both interlocks and operation permits, following the logics previously defined between the Control section and the Accelerators and Experiments divisions and programmed by Control engineers.

EPS uses 58 B&R CPU's and 110 periphery cabinets to collect more than 7000 signals. In addition to the main purpose of protection, several hundreds of signals distributed across the whole system are acquired for diagnostics and control of movable elements: temperatures, vacuum sensors, position encoders and switches, electrovalves, ...

Other PLC-based Systems

The Modbus protocol and Tango devices are also used to control PLC's in the RF circulators, bakeout controllers, water cooling system, air conditioning in the experimental hutches and overall Personnel Safety Systems, on which the Tango Control System have just read access. The same control interface is used to communicate with all this subsystems, with certain customization depending on the control needs.

PLC TANGO DEVICES

An interface between Tango Control System and our PLC-based subsystems was needed for three main purposes:

- Supervision of autonomous systems based on PLC's (EPS, PSS).
- Configuration of the EPS settings and interlock thresholds during commissioning.
- Integrate critical and diagnostics signals into our Control services like Archiving, Taurus UI, Alarms, Beamlines SCADA (Sardana) [6][7].

To achieve a successful integration of the PLC signals into our control system it was needed to automate the creation of Tango Attributes in the PLC device servers. The commissioning work-flow required a regular update of I/O and variables lists in the PLC's, and existing UI's and services like archiving had to be capable to keep pace with each update of the attribute list.

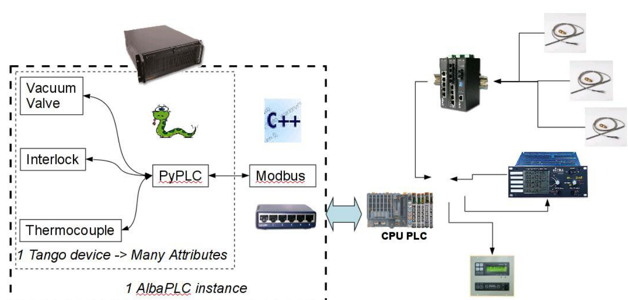

Figure 1: PyPLC Device Architecture.

DEVELOPING A PLC TANGO DEVICE

The first device server developed at ALBA for communicating with PLC's was a C++ device server, ModbusPLC, running on top of the Modbus Device developed at the ESRF; that already implements all basic Modbus commands. The ModbusPLC C++ Tango device allowed to create and remove attributes depending on Tango property values [8], exporting as many integer attributes as 16 bits registers were mapped in Modbus.

This implementation presented several drawbacks:

- Hides the diversity of signals available from the PLC (digital inputs, flag registers, integers, 16 bits floats, 32 bit values spanning multiple registers).
- Too rigid for showing complex elements (needed many attributes to represent a 3-position valve).

ISBN 978-3-95450-146-5

A REAL-TIME DATA LOGGER FOR THE MICE SUPERCONDUCTING MAGNETS

J.T.G. Wilson, STFC Daresbury Laboratory, Warrington, UK

Abstract

The Muon Ionisation Cooling Experiment (MICE) being constructed at STFC's Rutherford Appleton Laboratory will allow scientists to gain working experience of the design, construction and operation of a muon cooling channel. Among the key components are a number of superconducting solenoid and focus coil magnets specially designed for the MICE project and built by industrial partners.

During testing it became apparent that fast, real-time logging of magnet performance before, during and after a quench was required to diagnose unexpected magnet behaviour. To this end a National Instruments Compact RIO (cRIO) data logger system was created, so that it was possible to see how the quench propagates through the magnet. The software was written in Real-Time LabVIEW and makes full use of the cRIO built-in FPGA to obtain synchronised, multi-channel data logging at rates of up to 10 kHz.

This paper will explain the design and capabilities of the created system, how it has helped to better understand the internal behaviour of the magnets during a quench and additional development to allow simultaneous logging of multiple magnets and integration into the existing EPICS control system.

MICE

The Muon Ionisation Cooling Experiment (MICE) is an international collaboration of particle and accelerator physicists from Europe, the US and Japan. It seeks to design, build and operate a muon ionisation cooling channel, which given the consequence of the short muon lifetime that makes traditional cooling techniques inappropriate, is an essential technology for the design of a muon collider or neutrino factory[1].

The MICE cooling channel is of the same design as the cells proposed for the neutrino factory and consists of 3 absorber coil modules with low density absorbers inside a focusing magnetic field and 2 RF-coupling coil modules. It is being built on a dedicated muon beam from the ISIS accelerator at Rutherford Appleton Laboratory.

MAGNET QUENCHES

Upon testing of the first Focus Coil magnet a series of unexpected magnet quenches were occurring meaning that the magnets were not able to reach the power levels specified by the design requirements.

There was already a quench detection system installed, however, this was only designed for machine protection, shutting down the magnets in the event of a quench and was thus not designed to monitor all of the individual coil power levels. This also led to doubts as to whether there were actually magnet quenches occurring or if the quench detection system was not functioning correctly. To be able to properly address the unexpected quenches and prove the integrity of the quench detection system it was decided that a further diagnostic tool was required to monitor the power levels on each of the coils of the magnet so that the starting point of a quench could be determined and it's propagation through the rest of the magnet analysed.

The proposed solution to this was to create a standalone logging system that could capture magnet performance data before, during and after a quench. Because of the unexpected and unpredictable nature of the magnet quenches (testing could be running for hours before experiencing a quench) a system that simply logged the values from the magnets as soon as they were turned on would create far too much unnecessary data. Similarly, a system that started logging only once it had received the signal from the quench detector would miss vital information because data showing the quench starting to build up on the coils was needed for diagnosis of the fault. The solution to this was to have a system with a 'rolling capture window'. This 'window', or buffer, would temporarily save the data (i.e. in RAM) and once the buffer was full would start to overwrite the oldest data in the buffer with the newest. This would allow for the system to have already captured and be temporarily holding the data showing a voltage differential building up to a quench which, after receiving a signal from the quench detection system, it could amend with the data during and after a quench.

THE DATA LOGGER SYSTEM

Hardware

CompactRIO is a reconfigurable embedded control and acquisition system. The CompactRIO system's rugged hardware architecture includes I/O modules, a reconfigurable FPGA, and an embedded controller [2].

For this application it was decided that NI 9222 4 channel C-series modules [3] were needed, despite the greater cost than other similar analog cards, to achieve the necessary sample rates. These cards can provide up to 500 kS/s per channel at a 16-bit resolution. These were coupled with the NI 9103 chassis [4] which provides 4 C-series module slots and the NI 9012 controller [5].

Additional external electronics were needed to reduce the input voltages down to the +/- 10 Volts that the NI

BEAM DATA LOGGING SYSTEM BASE ON NoSQL DATABASE AT SSRF*

Y.B. Yan, Y.B. Leng#, L.W. Lai, Z.C. Chen, H.L. Geng Shanghai
Synchrotron Radiation Facility (SSRF) Shanghai Institute of Applied
Physics, Chinese Academy of Sciences
Shanghai 201204, P.R. China

Abstract

To improve the accelerator reliability and stability, a beam data logging system was built at SSRF, which was base on NoSQL database Couchbase. The Couchbase is an open source software, and can be used both as document database and pure key-value database. The logging system stores beam parameters under predefined conditions. It is mainly used for the fault diagnosis, beam parameters tracking or automatic report generation. The details of the data logging system will be reported in this paper.

OVERVIEW

Shanghai Synchrotron Radiation Facility (SSRF) is a low emittance third generation light source located at Shanghai, China. It includes a 150MeV LINAC, 150MeV to 3.5GeV Booster, LINAC to Booster transfer line, Booster to storage ring transfer line and 3.5GeV storage ring. To improve the accelerator reliability and stability, a beam data logging system was built, mainly based on the beam instrumentation system.

SSRF beam instrumentation system consists of more than 200 devices, which covered the beam position measurement, beam charge & current measurement, beam size & length measurement, fill pattern measurement and so on [1]. All these parameters are very important during the accelerator commissioning, operation and machine studies. More than 20k scalar process variables and hundreds of 2k-points waveform records are published online every second. With proper storage and analyze tool-kits, these data could be invaluable. Otherwise the potential of various new electronics will be wasted.

On the other hand, various hardware and software failures have been recorded during the past few years, such as global orbit disturbance, random glitch or offset jump of individual position readings [2]. All these failures affected the reliability and stability of the entire machine. There were no effective tools to analyze the reason due to lack of adequate raw data. The regular sampling rate of achieved data is about 1Hz. History of broadband data such as turn-by-turn (several hundreds kHz) orbit data or bunch-by-bunch data are required in this case. Due to the huge size, the data are not likely to be stored periodically. A logging system, which stores the raw data under some predetermined conditions, is urgently needed.

* Work supported by the Knowledge Innovation Program of Chinese Academy of Sciences and National Natural Science Foundation of China (No. 11105211, 11305253)
# lengyongbin@sinap.ac.cn

SYSTEM ARCHITECTURE

The data logging system is based on the Couchbase [3], which is an open source, distributed NoSQL database. It provides key-value or document access with low latency and high sustained throughput. The system architecture is shown in Figure 1.

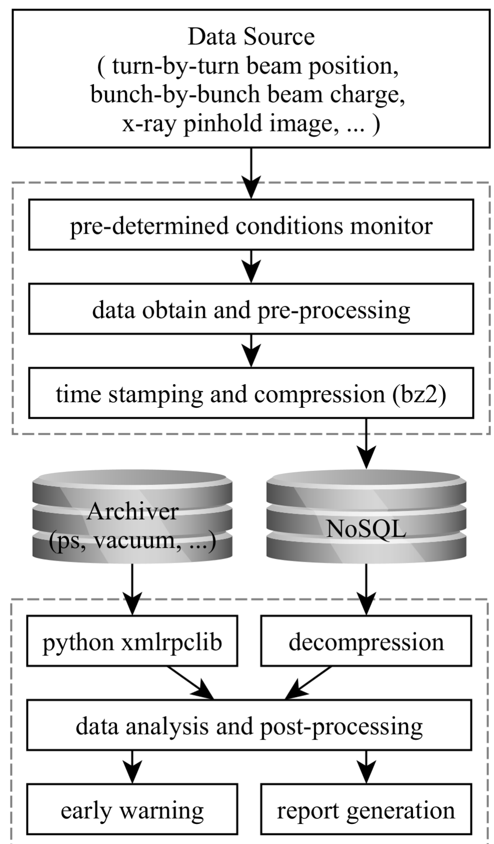

Figure 1: The architecture of data logging system.

In this system IBM System x3550 M4 server and IBM Storwize V3700 storage system are adopted, which is a cost-effective option to achieve high performance. All the software run on the Linux operating system, and written using Python, a widely used general-purpose, high-level programming language.

Data Source

For the particle accelerator, the data can be divided into two categories, hardware device related and beam related parameters. The hardware device related (such as vacuity,

ISBN 978-3-95450-146-5

NEW DATA ARCHIVE SYSTEM FOR SPES PROJECT BASED ON EPICS RDB ARCHIVER WITH PostgreSQL BACKEND

M. Montis, M. Giacchini, S. Fantinel, INFN/LNL, Legnaro, Italy
M. Bellato, INFN sez. Padova, Padova, Italy

Abstract

SPES project[1] is a ISOL facility under construction at INFN, Laboratori Nazionali di Legnaro, which requires the integration between the accelerator systems actually used and the new line composed by the primary beam and the ISOL target. EPICS[2] has been chosen as main Control System framework for the project; as consequence, a migration from the actual control system to the new one is mandatory in order to reuse the actual system for the new facility. One of the first implementation realized for this purpose is the Archiver System, an important service required for experiments. Comparing information and experiences provided by other Laboratories, an EPICS Archive System based on PostgreSQL is implemented to provide this service. Preliminary tests are done with a dedicated hardware and following the project requirements. After these tests, the system is going to be moved in production, where it will be integrated with the first subsystem upgraded to EPICS. Dedicated customizations are made to the application for providing a simple user experience in managing and interact with the archiver system.

INTRODUCTION

In a complex and extended installation like an accelerator, where different sub-systems work simultaneously and share information each other, the requirement of having available all the data under control, both online and offline, is mandatory: it results useful for production (such as post-analysis processes) and maintenance. The Channel Archiver is an archiving application realized for EPICS based control systems where a dedicated machine properly configured can archive any kind of process variable available in the control system network through the transparent communication protocol based on TCP/IP standard provided by EPICS, the EPICS Channel Access.

In the principal laboratory where EPICS is used as main control system framework, the original Channel Access Archiver, designed in 2006, is largely used. However in these few years different new solution based on Database backend are growing and starting to be a new standard for data archive service.

In this scenario, the EPICS RDB (Relational DataBase) Archiver with PostgreSQL Database has been chosen as main archive service for the SPES Project.

THE EPICS RDB ARCHIVER

During initial development, the test bench realized for the archiver system was composed by a single server equipped with all the hardware and the software required for a stand-alone test bench. In a next step, the hardware involved in the study case was upgraded and extended, in order to provide a full machine ready for production. The last set of tests has done using a part of the real control system environment under upgrade at LNL[3][4].

Hardware

The machine used to tests and deploy the archive service is a server equipped with 2 esa-core processor working in Hyper Treading, 32GB RAM and 2TB disk space, in order to have the maximum resources available both for development and production steps. It also has redundant power supply as required in this kind of environment.

Preliminary tests were performed on this host, using it as EPICS server (running an IOC) and EPICS client (the archiver application) due to analyse the correct configuration of the client tool. Later, extending tests to a real environment where different EPICS server provides different Process Variables (PVs), the hardware used for this represented in Figure 1.

Software

The focus followed in the software definition was having a machine with the minimum amount of applications and services required to perform this task, due to minimize the maintenance.

Following the guidelines adopted by LNL related to production hosts, RHEL compilant Linux was chosen as Operative System for the Archive machine. Over this OS, EPICS base was installed for providing the environment required to execute tests. At the same time, this machine was used to compile and develop the Archive tools (Archive Config Tool and Archive Engine) used to realize the final service; as consequence, a minimal graphic interface was installed for working with the Eclipse IDE and the EPICS CSS source code.

For the Archiver system, source code related to SNS CSS version 3.1.5 has been used. As suggested Ruizhe

ISBN 978-3-95450-146-5

DEVICE CONTROL DATABASE TOOL (DCDB)

Pavel Maslov, Matej Komel, Klemen Žagar, Cosylab, Ljubljana, Slovenia

Abstract

In a physics facility containing numerous instruments, it is advantageous to reduce the amount of effort and repetitive work needed for changing the control system (CS) configuration: adding new devices, moving instruments from beamline to beamline, etc. We have developed a CS configuration tool, which provides an easy-to-use interface for quick configuration of the entire facility. It uses Microsoft Excel as the front-end application and allows the user to quickly generate and deploy IOC configuration (EPICS start-up scripts, alarms and archive configuration) onto IOCs; start, stop and restart IOCs, alarm servers and archive engines, etc. The DCDB tool utilizes a relational database, which stores information about all the elements of the accelerator. The communication between the client, database and IOCs is realized by a REST server written in Python. The key feature of the DCDB tool is that the user does not need to recompile the source code. It is achieved by using a dynamic library loader, which automatically loads and links device support libraries. The DCDB tool is compliant with CODAC (used at ITER and ESS), but can also be used in any other EPICS environment.

DCDB ARCHITECTURE

The DCDB-tool uses MySQL relational database together with the BLED [1] schema (a set of tables representing the whole facility).

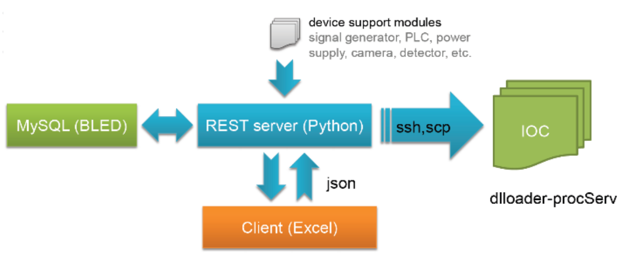

Figure 1: DCDB architecture.

The backend is a typical web-server, which is realized with a combination of the following python modules: flask-restful (REST server), sqlalchemy and pymysql (database communication layer), and paramiko (ssh). The front-end is a Microsoft Excel plugin written in C# using .NET technology. IOCs are Linux machines running EPICS and procServ [2]. The client-server communication is based on the exchange of JSON objects (strings).

DEVICE SUPPORT MODULES

Device support modules are created using standard commands that come with the Maven ITER plugin and Java API for managing CODAC development unit life-cycle (packages m-iter-plugin, m-codac-unit-api), supplemented by dlloader support (Fig.2).

```
bled@bled:~$ mvn newunit -Dunit=m-BeamPositionMonitor
bled@bled:~$ cd m-BeamPositionMonitor
bled@bled:~$ mvn newdlloader
bled@bled:~$ mvn clean compile test package
```

Figure 2: The procedure of creating EPICS device support libraries.

Device support modules should contain a library (lib/*.so), an EPICS database definition file (dbd/*.dbd files), an EPICS database template/substitution file (db/*.db) and three scripts: init.cmd (contains the string *require <module>, <version>*), init-pre.cmd (a set of configuration fields or macros to be extracted into the BLED database, plus epics shell commands to be executed before IOC initialization) and init-post.cmd (epics shell commands to be executed after IOC initialization).

```
.
├── db
│   └── BeamPositionMonitor.db
├── dbd
│   └── BeamPositionMonitor.dbd
├── init.cmd
├── init-post.cmd
├── init-pre.cmd
└── lib
    └── linux-x86_64
        ├── libBeamPositionMonitor.a
        └── libBeamPositionMonitor.so
```

Figure 3: Files to deploy.

In order to load the information about the device support module into the BLED database, use the *bled* helper script that you should invoke from the module's $(TOP) directory. It will extract the information about the module stored in the pom.xml and init-pre.cmd files and send it to the REST server over the network.

DYNAMIC LIBRARY LOADER

Dynamic library loader (or dlloader)[1] is an EPICS-based tool that allows you to load EPICS device support libraries by just adding its' definitions in the startup script (st.cmd). Hence, the integrator is freed from the necessity of compiling (recompiling) their EPICS applications/IOCs.

The DCDB-tool uses dlloader to dynamically load device support modules (described in the previous chapter). As it was already mentioned there is no need to recompile the source code, since all we have to provide is a st.cmd script, which in turn is generated by the REST server.

1. The concept of dynamically loadable device support modules, including the require function is developed by Dirk Zimoch (PSI).

ISBN 978-3-95450-146-5

STATUS OF OPERATION DATA ARCHIVING SYSTEM USING Hadoop/HBase FOR J-PARC

N. Kikuzawa#, H. Ikeda, Y. Kato, N. Ouchi J-PARC, Tokai-mura, Naka-gun, Ibaraki, Japan
A. Yoshii, NS Solutions Corporation, Shinkawa, Chuo-ku, Tokyo, Japan

Abstract

J-PARC (Japan Proton Accelerator Research Complex) consists of much equipment. In the Linac and the 3 GeV rapid cycling synchrotron ring (RCS) in J-PARC, data of about 64,000 EPICS records have been collected for control of these equipment. The data volume is about 2 TB every year, and the total data volume stored has reached about 10 TB. The data have been being stored by a Relational Database (RDB) system using PostgreSQL since 2006 in PostgreSQL, but it is becoming that PostgreSQL is not enough in availability, performance, and flexibility for our increasing data volume.

We are planning to replace PostgreSQL with Apache Hadoop and Apache HBase to accumulate enormous operation data produced from the Linac and the RCS in J-PARC. HBase is so-call NoSQL, which has scalability to data size at the cost of the high broad utility of SQL. HBase is constructed on a distributed file system provided by Hadoop, a cluster with advantages including automatically covering its cluster nodes' breakdowns and easily adding new nodes to expand its capacity. The new database system satisfies high availability, high performance, and high flexibility of storage expansion.

The purpose of this paper is to report the present status of this archive system.

INTRODUCTION

J-PARC is controlled with a lot of equipment, and we have been archiving a time series of operation data provided from about 64,000 EPICS records for the Linac and the RCS since 2006 [1]. PostgreSQL has been used in the present data archiving system, but it has some problems of capacity, extensibility, and data migration. In order to deal with these problems, we proposed a next-generation archive system using Apache Hadoop [2], a distributed processing framework, and Apache HBase [3], a distributed database [4].

Hadoop is a widely used open-source cloud framework for large scale data processing. The HBase is a distributed, scalable big data store on a cluster built with commodity hardware. One of the cores of Hadoop is a file system called HDFS (Hadoop Distributed File System), and HBase runs on it. Hadoop and HBase are scale-out architecture, and we can expand storage volume dynamically by adding new nodes. Moreover, they are designed based on the assumption of frequent breakdown

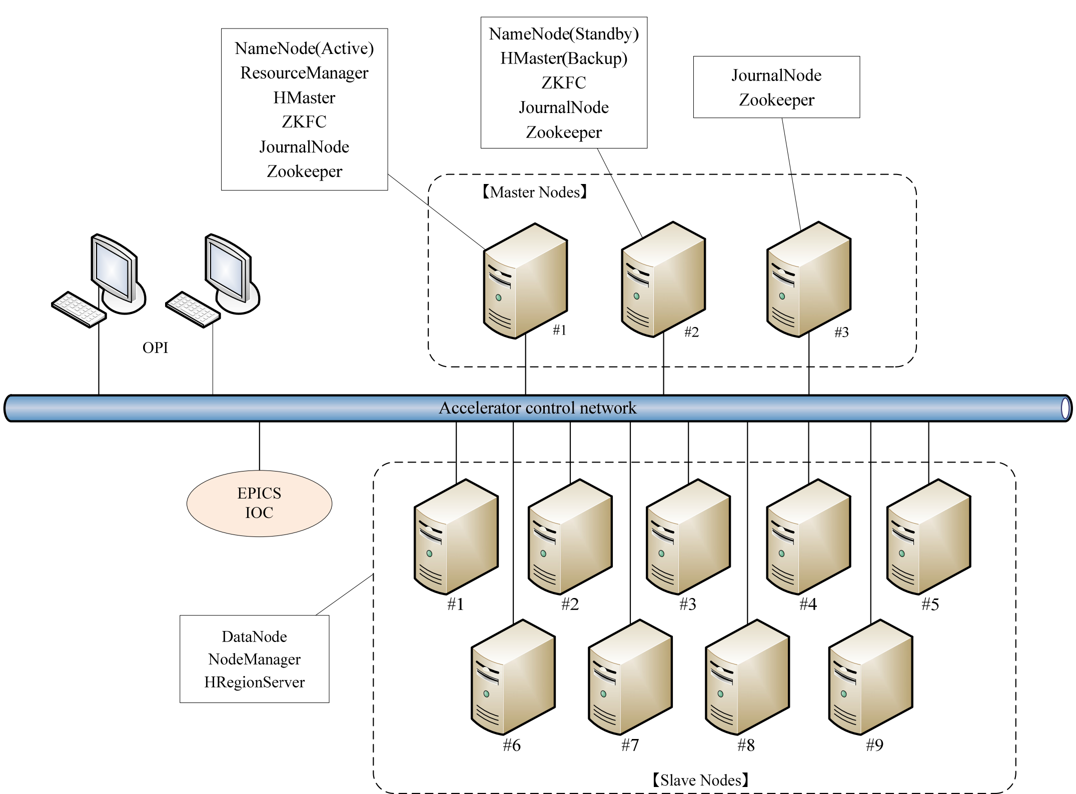

Figure 1: System configuration of the new archiving system.

#kikuzawa.nobuhiro@jaea.go.jp

ISBN 978-3-95450-146-5

MANAGING MULTIPLE FUNCTION GENERATORS FOR FAIR

S. Rauch, M. Thieme, R.C. Bär, GSI, Darmstadt, Germany

Abstract

In the FAIR control system, equipment which needs to be controlled with ramped nominal values (e.g. power converters) is controlled by a standard front-end controller called scalable control unit (SCU). An SCU combines a ComExpressBoard with Intel CPU and an FPGA baseboard and acts as bus-master on the SCU host-bus. Up to 12 function generators can be implemented in slave-board FPGAs and can be controlled from one SCU.

The real-time data supply for the generators demands a special software/hardware approach. Direct control of the generators with a FESA (front-end control software architecture) class, running on an Intel Atom CPU with Linux, does not meet the timing requirements. So an extra layer with an LM32 soft-core CPU is added to the FPGA. Communication between Linux and the LM32 is done via shared memory and a circular buffer data structure. The LM32 supplies the function generators with new parameter sets when it is triggered by interrupts. This two-step approach decouples the Linux CPU from the hard real-time requirements. For synchronous start and coherent clocking of all function generators, special pins on the SCU backplane are being used to avoid bus latencies.

DESCRIPTION OF SCU AND FG

The quadratic function generator (FG) which is described in this paper, is a VDHL macro that runs in SCU bus slave cards. At the moment, there are three slave cards with this feature: DIOB (1 FG), ADDAC1 (2 FGs) and ADDAC2 (2 FGs). The DIOB card has an digital output with 32 Bit for the FG output value. The two ADDAC cards offer an analog output with 16 Bit resolution for the FGs. The slave cards are controlled via the SCU bus from the Scalable Control Unit (SCU). The SCU is a FPGA based controller equipped with a ComExpress Board which runs linux. The communication between FPGA and ComExpress Board is done via PCIe. Inside the FPGA is a System-on-Chip (SoC) on basis of a wishbone [1] crossbar with a PCIe-to-wishbone bridge (wishbone master). The SCU bus is connected with a bridge too, that acts as a wishbone slave. Part of the SoC is LM32 cluster with a configurable number of softcore processors and shared memory (see Fig. 1). A separate crossbar is used for message signaled interrupts (MSI). A interrupt master, e.g. SCU bus bridge, sends MSIs to an interrupt slave. That can be a interrupt queue of an LM32 or the PCIe bridge. With the use of MSIs, the interrupt system is quiet flexible, because every slave can address every interrupt target.

The SCU bus is a parallel bus with 12 slave slots. The data and address lines are each 16 Bit wide. Each slave has a separate IRQ line. Inside the SCU bus bridge the IRQs are translated into MSIs. The system should be used as an arbitrary function generator with 12 independent channels. Each channel will control equipment that needs ramped nominal values. That means for example power converters and Direct Digital Synthesis (DDS) systems. The FG is configured with a set of data and interpolates then a predefined number of output values. After the interpolation is started, the FG waits for the next set of data that is provided by the linux FESA class. A brief hardware description of the FG can be found here [2]. In contrast to the older paper, a few things had to be changed for the implementation. The data path is now 64 Bit wide and both parameters, the linear and the quadratic one, can now be shifted in a 64 Bit range.

FG Inside SCU

Other then for ramped power converters, the SCU will be used to control DDS systems. This will be done with FIB cards, which are supplied from the radio frequency group. These FIB [3] cards are used as SCU slaves. But in there current revision the are not able to run a FG macro in the slaves. So a different solution had to be found. The same FG macro as used in the slaves is put behind a wishbone interface and connected to the crossbar of the SCU. The output of that FG is connected to a special wishbone master, that splits the 32 Bit output value from the FG into two 16 Bit accesses to the SCU bus. So the interpolation of the FG is done inside the SCU, instead of inside the SCU slaves. Because this modus uses a lot more bandwidth than the slave approach, the SCU bus should only be used for sending FG values. The FG macro with wishbone interface acts exactly like the FG in the slaves, only the interface is different. For the software layer they look identical.

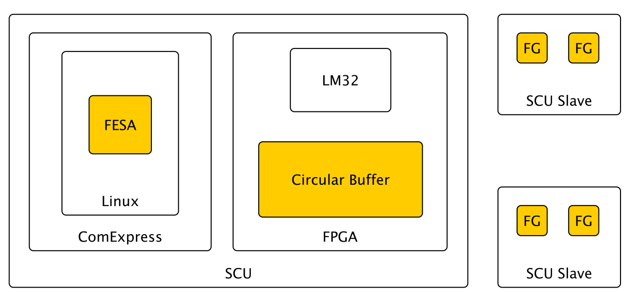

Figure 1: SCU with two slaves.

DATA SUPPLY WITH REAL TIME BOUNDARIES

The FG can be configured to interpolate in steps from 250 up to 32000. The sample frequency is configurable from 16 kHz up to 1 MHz. If the FG should now sample with 1 MHz for 250 steps that means the linux program has to provide a new data set every 250 µs. This data rate is to high for linux to be serviced reliably for 12 channels.

ISBN 978-3-95450-146-5

SETUP AND DIAGNOSTICS OF MOTION CONTROL AT ANKA BEAMLINES

Karlheinz Cerff, David Haas, Denis Jakel, KIT-ANKA, Markus Schmitt KIT-IBG, Karlsruhe, Germany

Abstract

The precise motion control in high resolution [1] is one of the necessary conditions for making high quality measurements at beam line experiments.

At a common ANKA beam line, up to one hundred actuator axes are working together to align and shape beam, to select beam Energy and to position probes. Some Experiments need additional motion axes supported by transportable controllers plugged temporary to a local beam line control system.

In terms of process control all the analogue and digital signals from different sources have to be verified, levelled and interfaced to the motion controller. They have to be matched and calibrated in the control systems configuration file to physical quantities which give the input for further data processing. A set of hard- and software tools and methods developed at ANKA over the years is presented in this paper.

INTRODUCTION

Building a new beam line, the motion control setup is accompanied by ANKA-IT from an early stage of the design phase, the factory acceptance tests of components and systems at manufacturer site and the final tests at ANKA site.

Therefore ANKA-IT proposes to the beam line designer a catalogue of up to date ANKA-proven specifications for hardware components and control software environment.

The suggestion defines objects of hardware with a range of preferred attributes which describe the components manufacturer independently. This gives a flexible response to upgrade obsolete components over years of operation or setup new systems equipped with not 'ANKA-standard' compatible components.

On the software site versatility in intermediate layer, concepts enables the embedding of OEM controllers in the ANKA control system without giving preference to the operating system they were primarily designed for.

A design guideline was created by ANKA-IT to minimize the effort to select communication software and define interface specifications. The preferences for key elements are described in the chapters below

ANKA DESIGN GUIDELINES

- The manufacturer/supplier should clearly state the proposed scope of supply for the control system and associated electronics.
- The beam line control software concept will be provided by ANKA-IT.
- Network Interfaces are preferred for beam lines but alternatively an ANKA-standard hardware Interface, s. Figure 3 is accepted.
- The motion beam line components, use VME-bus as well as various bus protocols working over Ethernet TCP/IP.
- For the system realisation, a Tango-interface is preferred, in case of not availability spec*[2] can be used. The libraries and component software drivers for setting up motion control should be documented and supplied to ANKA-IT.

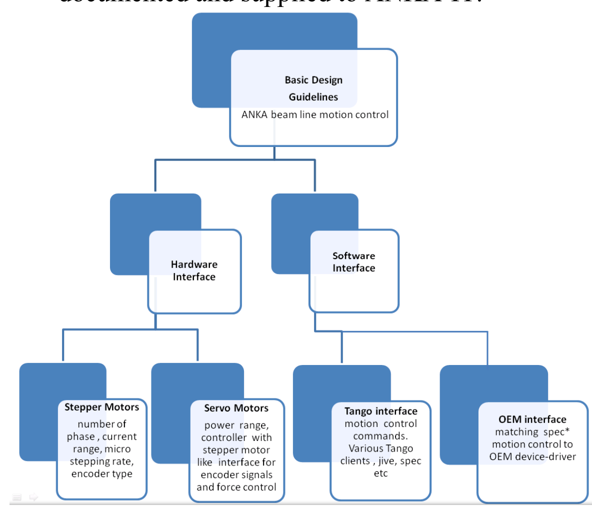

Figure 1: Overview hard- and software design guidelines for ANKA-motion control.

Since ANKA is migrating to Tango [3] for beam line control the communication between spec* and WinCCOA*[4] is fully Tango-based and all new implemented devices are controlled via Tango device servers.

DESIGN REPORT

A Tango interface and a set of geometry parameters for setup of the motor configuration should be supplied by the manufacturer in his design report. During the factory acceptance test these attributes and parameters are verified by test. The Tango interface is defined by the five device attributes: position, velocity, offset, limit switch negative direction, limit switch positive direction. The motor device is addressed by a set of specific commands given in Table 1.

*Trademark

ISBN 978-3-95450-146-5

FPGA UTILIZATION IN THE ACCELERATOR INTERLOCK SYSTEM (ABOUT THE MPS DEVELOPMENT IN THE LIPAc)

K. Nishiyama*, J. Knaster, A. Marqueta, Y.Okumura,
IFMIF/EVEDA Project Team, JAEA, Aomori, Japan
T. Narita, H. Sakaki, H. Takahashi, JAEA, Aomori, Japan
T. Kojima, Nihon Advanced Technology, Aomori, Japan
R. Gobin, CEA, Gif-sur-Yvette, France

Abstract

IFMIF (International Fusion Material Irradiation Facility) will generate 14 MeV neutron flux for qualification and characterization of suitable structural materials of plasma exposed equipment of fusion power plants. IFMIF is an indispensable facility in the fusion roadmaps since provide neutrons with the similar characteristics as those generated in the DT fusion reactions of next steps after ITER. IFMIF is presently in its EVEDA (Engineering Validation and Engineering Design Activities) phase.

As part of IFMIF Validation Activities, LIPAc (Linear IFMIF Prototype Accelerator), designed and constructed mainly in European labs (CIEMAT, CEA, INFN and SCK CEN) with participation of JAEA, is currently under installation at Rokkasho (Japan). LIPAc will accelerate a 125mA CW and 9MeV deuteron beam for a total beam average power of 1.125MW. The Machine Protection System (MPS) of LIPAc provides the essential interlock function of stopping the beam in case of anomalous beam loss or other hazardous situations, particularly critical for investment protection reasons in high power accelerators.

High speed processing is indispensable to adequately achieve the MPS main goal. This high speed processing of the signals, distributed alongside the accelerator facility, is based on FPGA technology. This paper describes the basis of FPGA use in the accelerator interlock system through the development of LIPAc´s MPS.

INTRODUCTION

LIPAc is a prototype accelerator that will reach a beam average power of 1.125 MW with deuterons in CW at 125 mA and 9 MeV. It will validate the accelerators of IFMIF [1] (125 mA in CW at 40 MeV) by demonstrating that the space charge issues can be overcome at its lowest 1st energy superconducting accelerator stage (the 40 MeV will be achieved in three additional SC stages at 14.5, 26 and 40). The involved high power and beam nature entails investment protection arguments and radiation safety aspects. The control system for LIPAc is responsible of its safety functions, the control and monitoring functions to realize these tasks efficiently target the minimization of activation induced by beam losses (driven by 'hands-on' maintenance principles) and potential hazard to the investment. In light of these requirements, the control system of LIPAc is broken down as follows: 1) Central Control System (CCS), 2) Local Area Network (LAN), 3) Personnel Protection System (PPS) to avoid unnecessary exposure to radiation, 4) Machine Protection System (MPS) for the accelerator subsystems and the facility, 5) Timing System (TS), 6) Local Control System (LCS) for the accelerator subsystems.

The remote operation for LIPAc is performed by CCS, LAN and the different LCS. The high level data (using EPICS) is transmitted by using LAN (Ethernet). All subsystems with synchronization for pulse operation are realized by using TS signals. The radiological safety for the personnel is established by PPS. And the beam inhibit (fast and slow) is realized by MPS in order to protect the accelerator and its components. This paper describes the outline and development status of MPS [2].

OUTLINE OF MPS

LIPAc produces a powerful CW deuteron beam [3] with high average beam current of 125 mA. The 9 MeV deuteron beam with its MW range beam power will be absorbed on a Cu beam dump water cooled. If an excessive beam loss event happens in an undesired manner on an accelerator component, the power would potentially cause a fatal damage. Additionally, the high inelastic cross sections of deuteron would lead, in case of excessive beam losses, to an increase of neutron and gamma induced dose rates in the radiation controlled area, with a potential dramatic impact of the targeted 'hands-on maintenance' approach. The Machine Protection System (MPS) is defined as the safety system against the accelerator troubles for the protection of the investment. The MPS has interfaces with other LIPAc subsystems, including the PPS (Personnel Protection System) Accelerator subsystems. It comprises the beam stop interlock signal from each accelerator subsystems, each of which presents interfaces with the MPS, PPS, EPICS, and emergency stop logic of subsystem. MPS realizes the beam rapid stop to minimize the effects on the beam pipe by beam loss. The target time of MPS signal transfer, which is the time taken from "MPS unit receives the interlock signal from accelerator subsystem" to "MPS sends the beam stop signal to the injector", is less than 10 μs. To achieve this fast response time, FPGA technology has been chosen.

The backbone of MPS for LIPAc is the already consolidated successful MPS unit used at J-PARC Linac. In the case of MPS for J-PARC Linac, the "Beam rapid stop" is achieved within 5 μs after "MPS units receives the interlock signal from accelerator subsystem" with high reliability experienced [4].

Therefore, the use of J-PARC Linac's knowhow as the basis of LIPAc's MPS unit is suitable for the interface between MPS and accelerator subsystem; MPS itself will realize the logic for beam stop and beam restart.

ISBN 978-3-95450-146-5

NEW DEVELOPMENTS ON THE FAIR DATA MASTER

M. Kreider, R. Bär, D. Beck, W. Terpstra, GSI, Darmstadt, Germany
J. Davies, V. Grout, Glyndŵr University, Wrexham, United Kingdom

Abstract

During the last year, a small scale timing system has been built with a first version of the Data Master. In this paper, we will describe field test progress as well as new design concepts and implementation details of the new prototype to be tested with the CRYRING accelerator timing system. The message management layer has been introduced as a hardware acceleration module for the timely dispatch of control messages. It consists of a priority queue for outgoing messages, combined with a scheduler and network load balancing. This loosens the real-time constraints for the CPUs composing the control messages noticeably, making the control firmware very easy to construct and deterministic. It is further opening perspectives away from the current virtual machine-like implementation on to a specialized programming language for accelerator control. In addition, a streamlined and better fitting model for beam production chains and cycles has been devised for use in the data master firmware. The processing worst case execution time becomes completely calculable, enabling fixed time-slices for safe multiplexing of cycles in all of the CPUs.

OVERVIEW AND SYSTEM LAYOUT

As discussed in our previous papers [1, 2], the FAIR accelerator will be a highly complex system which needs a control system to match. This suggests a design that supports high performance, flexibility and deterministic command generation and distribution. While all machine commands for beam production will be calculated from physics data ahead of time, all final scheduling and deterministic delivery is the responsibility of the Data Master (DM). The DM itself is a hybrid of an industrial PC and a Field Programmable Gate Array (FPGA) based embedded real-time system with hardware acceleration modules. We will now discuss the system layout of the current implementation and the inner workings of the sub-modules in greater detail.

CONTROL DATA

Structure

The control data received by the DM broadly resembles a flowchart for beam production. It consists of $2..I$ alternative beam production scenarios, called plans. Each plan has $1..J$ event chains in it. Chains are the basic building blocks. They each contain $0..K$ command messages. They also come with the means for simple control structures. The input format is currently XML based and converted to a binary format for the embedded system.

Time

From the start of a plan, all times are relative offsets. Each chain has a given duration, and chain start times are calculated by adding up previous durations. The only exception to the rule is a conditional wait. Here, the start time of the next chain is set to time the condition was fulfilled, plus a fixed offset. An absolute execution time is calculated for each command message at the moment it is dispatched. This is done by adding the message offsets to its chain's start time.

EMBEDDED SYSTEM

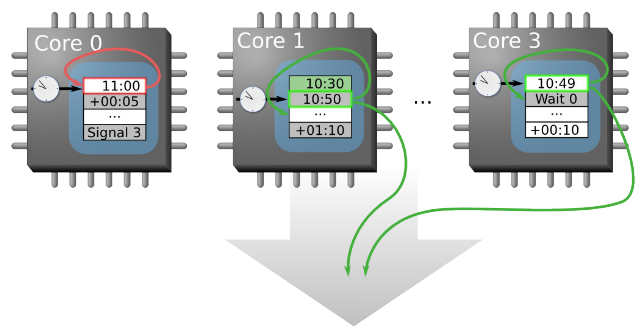

Figure 1: Scheduling commands in the Soft-CPU Cluster.

Layout and Firmware

The LM32 (Lattice Mico 32 Micro-controller) is a 32 Bit RISC processor for use in Field Programmable Gate Arrays (FPGAs) and Application Specified Integrated Circuits (ASICs) and a good choice for a control system [3]. Timer Interrupts were unsuitable for dispatch due messages, because saving and restoring all 32 registers takes considerable time and not all instructions have the same execution time. Furthermore, it would have meant the processors being idle most of the time. In order to build a deterministic system with a low reaction time, a polling approach with fixed time slices was used and multiple Soft CPUs were instantiated to deal with different parts of beam production in parallel, aided by hardware acceleration cores. Each of the LM32 runs very simple firmware with three responsibilities for each iteration. The first is synchronization, meaning checking conditions or signalling another process(or). Next comes processing the current chain. This means sending a due command message to the priority queue. The third is to check the command interface for external instructions from the control system. All worst case execution times are completely deterministic. The only exception would be dispatch, because the network interface is a shared resource for all cores. The current testbed only features one thread

ISBN 978-3-95450-146-5

FIRST IDEA ON BUNCH TO BUCKET TRANSFER FOR FAIR

J. Bai[1,2,*], D. Beck[1], R. Bär[1], D. Ondreka[1], T. Ferrand[1,3], M. Kreider[1], C. Prados[1], S. Rauch[1], W. Terpstra[1], M. Zweig[1]
[1]GSI Helmholtzzentrum für Schwerionenforschung, Darmstadt, Germany
[2]IAP, Goethe University Frankfurt am Main, Frankfurt, Germany
[3]Technical University Darmstadt, Darmstadt, Germany

Abstract

The FAIR facility makes use of the General Machine Timing (GMT) system and the Bunch phase Timing System (BuTiS) to realize the synchronization of two machines. In order to realize the bunch to bucket transfer, firstly, the source machine detunes slightly its RF frequency at its RF flattop. Secondly, the source and target machines exchange packets over the timing network shortly before the transfer and make use of the RF frequency-beat method to achieve the synchronization between two machines with accuracy better than 1°. The data of the packet includes RF frequency, timestamp of the zero-crossing point of the RF signal, harmonic number and bunch/bucket position. Finally, both machines have all information of each other and can calculate the coarse window and create announce signals for triggering kickers.

INTRODUCTION

The bunch to bucket transfer means that one bunch of particles, circulating inside the source machine, must be transferred in the center of a precise bucket and on the desired orbit of the target machine. It is realized by the General Machine Timing (GMT) system [1] and the Bunch phase Timing System (BuTiS) [2].

The main task of the GMT system is the time synchronization of more than 2000 Timing Receivers (TR) with nanosecond precision, distribution of timing events and subsequent generation of real-time actions by the TRs of the timing system located at the FAIR accelerator complex. The timing system is based on the White Rabbit (WR) network, which achieves the time synchronization by adjusting the clock phase (125 MHz carrier) and the time offset (Coordinated Universal Time – UTC) of all network TRs to that of a common grandmaster clock [3]. For the synchronization of radio-frequency (RF) components, the timimg system is complemented and linked to the BuTiS. The BuTiS is a campus wide clock synchronization and distribution system. It generates an ident impulse clock at a rate of 10 µs, a 10 MHz sinewave reference clock and a 200 MHz sinewave clock [4].

After a bunch of particles is accelerated to the top energy, the RF flattop, it must be extracted from the source machine to be injected in the centre of a bucket of the target machine without phase and energy error, e.g. Four batches of U^{28+}, each batch has two bunches (h = 2), at 200 MeV/u of SIS18 will be injected into eight out of ten buckets of SIS100 [5] (see Fig. 1). This paper explains the process of the bunch to bucket transfer. The first step is the frequency detune and the second step is the synchronization of two machines by the frequency-beat method.

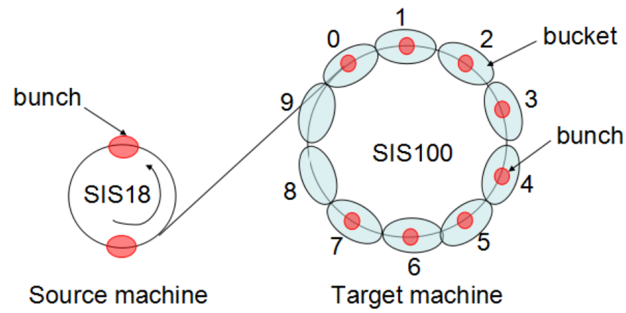

Figure 1: The bunch to bucket transfer of U^{28+} from SIS18 to SIS100.

BEAM-DYNAMICS VIEW OF THE FREQUENCY DETUNE

The first step for the bunch to bucket transfer is the RF frequency detune. In order to realize the frequency-beat between two machines, the RF frequency of the source machine has to be detuned. It means that the particles run at an average radius different by $\triangle$R from the designed orbit R. To make the frequency detuning effective, the feedback loop (i.e. the radial loop [6]) must be turned off just before the frequency detuning begins. Accepting to decenter the orbit by 8 mm for SIS18 [7]:

$$\frac{\triangle R}{R} \approx 2.4\times10^{-4} , \quad (1)$$

the RF frequency detuning at the U^{28+} 200 MeV/u [7] extraction energy ($\gamma = 1.217$) is

$$\frac{\triangle f}{f} = -\frac{\gamma^2 - {\gamma_t}^2}{\gamma^2}\frac{\triangle R}{R} \approx 5\times10^{-3} , \quad (2)$$

where $\triangle f$ is the frequency deviation for the frequency detuning, f is the RF frequency, $\gamma_t = 5.8$. The maximum RF frequency detuning is approximate to 7.5 kHz at 1.57 MHz for the U^{28+}.

The relative momentum shift is

$$\frac{\triangle p}{p} = {\gamma_t}^2 \times \frac{\triangle R}{R} \approx 8\times10^{-3} , \quad (3)$$

where p is the desired momentum of particle, $\triangle p$ is the momentum shift caused by the frequency detune.

* J.Bai@gsi.de

ISBN 978-3-95450-146-5

ADEI AND TANGO ARCHIVING SYSTEM – A CONVENIENT WAY TO ARCHIVE AND REPRESENT DATA

D. Haas, S. Chilingaryan, A. Kopmann, D. Ressmann, W. Mexner, KIT, Karlsruhe, Germany

Abstract

Tango offers an efficient and powerful archiving mechanism of Tango attributes in a MySQL database. The tool Mambo allows an easy configuration of all to be archived data. This approved archiving concept was successfully introduced to ANKA (Angströmquelle Karlsruhe).To provide an efficient and intuitive web-based interface instead of complex database queries, the TANGO Archiving System was integrated into the "Advanced Data Extraction Infrastructure ADEI". ADEI is intended to manage data of distributed heterogeneous devices in large-scale physics experiments. ADEI contains internal preprocessing, data quality checks and an intuitive web interface, that guarantees fast access and visualization of huge data sets stored in the attached data sources like SQL databases or data files. ADEI and the Tango archiving system have been successfully tested at ANKA's imaging beamlines. It is intended to deploy the whole system at all ANKA beamlines.

INTRODUCTION

ANKA is a third generation synchrotron light source operated by the Karlsruhe Institute for Technology (KIT). ANKA is operating sixteen beamlines and three more are under construction.

The control system of the ANKA beamlines is based on Tango [1], which has been a convenient and reliable control system. Tango offers an archiving system [2] which allows logging all Tango-attributes. In 2014 this archiving system was evaluated at ANKA to log the data of an experiment.

All logged data of a beamline and the experiment should be presented and retrieved in a modern, state of the art web interface. This offers the user a convenient way to access the data. ADEI a web based interface for database query, developed by the Institute for Data Processing and Electronics (IPE) of KIT fulfils exactly these requirements. Using ADEI as a viewer respectively analysis tool and connecting it to the databases of the Tango archiving system creates a platform to track and to monitor the status of a beamline.

For testing the environment, the system was developed and implemented at two beamlines Topo-Tomo [3,4] and Image at ANKA.

ADEI

The "Advanced Data Extraction Infrastructure (ADEI)" has been developed to provide ad-hoc data exploration capabilities to a broad range of long-running physical experiments dealing with time series data [5]. Such experiments have varying characteristics and often composed from multiple subsystems developed by different vendors. As a result, the underlying storage engines and the data formats often differ even between subsystems of a single experiment. On the other hand, the users want to get uniform access to all the data. Easy correlation of data produced by any components of the system is desirable. Beside this, operators need a tool providing the possibility to examine all collected data, checking the integrity and validity of measurements.The ADEI architecture shown in Figure 1 is modular. New data sources can easily be included. The backend provides the desired uniform access to the data. The web-based front-end allows quick inspection of data archives. The communication between front-end and back-end is realized using the AJAX (Asynchronous JavaScript + XML) paradigm.

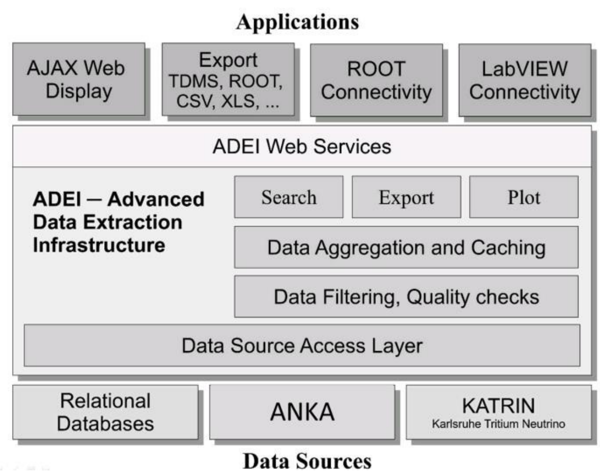

Figure 1: Architecture of Advanced Data Extraction Infrastructure ADEI. Data Source Access Layer unifies access to the time series stored in different formats. After data filtering and quality checks the data is aggregated and stored in intermediate caching databases. Access to the data is provided by the ADEI library and web services are used to communicate with client applications.

The backend consists of multiple components organizing the data flow from the data source to the client application. The Data Access Layer hides details of the underlying data sources and provides other components of the system with uniform access to all types of data. This is released using independent source drivers implementing ADEI data access interface. Furthermore, the data is passed through the chain of the configured data processing plugins which analyze the data, control the data quality, and optionally apply correction coefficients or filter out bad values. Hence, the rest of the system can fully rely on approved data quality.

ISBN 978-3-95450-146-5

WEB BASED MACHINE STATUS DISPLAY FOR SIAM PHOTON SOURCE

N. Suradet, C. Thamtong, C. Preecha, S. Klinkhieo, P. Klysubun
SLRI, 111 University Avenue, Muang District, Nakhon Ratchasima, 30000, Thailand

Abstract

A new machine operation status broadcasting system has been developed for Siam Photon Source (SPS), a 1.2 GeV synchrotron light source in Thailand. The system is implemented using web-based interface, and broadcasts the information over the SPS website, mobile application, as well as local TV network within the SPS facility, allowing users as well as technical personnel to easily access a variety of information related to the machine via web browsers and other mediums. The new system also provides supporting message services for alarm, event notification, and other operational necessities. In this report, the design of web and mobile applications, which are based on HTML5, CSS3, and adopts PHP, AJAX, Bootstrap framework (for responsive design), jQuery, High charts JS, Twitter widget, and others, will be described. The details of the hardware and software configurations, users requirements and satisfactions, as well as suggestions on further improvements, will be presented.

INTRODUCTION

The Siam Photon Source (SPS) is a synchrotron light source operated by Synchrotron Light Research Institute (SLRI), and is located in Nakhon Ratchasima, Thailand. The first light was achieved back in December 2001. At present, the machine is operating at 1.2 GeV in decay mode with a maximum electron beam current of 150 mA. Three insertion devices, a permanent magnet planar undulator, a hybrid multipole wiggler, and a superconducting magnet wavelength shifter, are currently in operation, providing synchrotron radiation from infrared to hard x-rays to synchrotron light users.

The original machine operation status broadcasting system was developed back in 2000, providing the operation status of the machine, for e.g. beam current, beam lifetime, beam energy, to users, who can access the provided information through the internal cable TV system within the facility. Each display channel receives the machine status data from a LabVIEW program located on a computer server. Since this system was available only for on-site users, another system was developed in 2006 to provide the machine status information via the internet. The fundamental language used to create this web-based system was basic static HTML. The displayed beam current and lifetime chart was captured from a NI LabVIEW window.

This web-based system has two main disadvantages. First, it consumes quite a bit of the network bandwidth because the whole web page had to be constantly updated, and the size of the chart image was quite large. Secondly, the system cannot display the data in real-time. We found it necessary to develop a new system that is more robust, more responsive, and more accessible. The new system has to meet the following requirements:

- The core system is based on web technology.
- The web layout is able to present the contents clearly and accurately across multiple types of devices (PC, mobile phones, tablets, etc.) with diverse display resolutions.
- The data is constantly updated every 5 seconds, but the network traffic must be kept low.
- The beam current and beam lifetime chart is generated by the browser on the client side. The displayed data can be exported to a CSV file.
- The system is capable of broadcasting notification messages.

SOFTWARE ARCHITECTURE

The machine status data originates from a variety of sources. These sources/hardwares are interconnected via an assortment of interface standards (OPC, GPIB, RS-232, etc.). A data logging program written with LabVIEW and installed on an acquisition server is employed to continuously gather all the machine data and log them into a database. The logging interval is 5 seconds. Open source database MySQL [1] was chosen for our purpose. LabVIEW MySQL connector toolkit [2] allows LabVIEW to communicate with MySQL (version 4.1 or later) via the TCP/IP protocol. It is a part of the LAMP (Linux-Apache-MySQL-PHP) platform that has to be installed on the web server.

When the user opens the SPS machine status web page, the browser on the client side will make a request for the PHP webpage to the web server. The web server responds by sending HTML, JavaScript, and CSS scripts to the client for processing, so that the execution is performed by the client browser. We use AJAX (Asynchronous JavaScript And XML) [3,4] to help refresh the web page for updating the data. AJAX runs a background operation which extracts the data from the database in the XML/JSON data format every 5 seconds. It updates the data field of the web page without reloading the whole page, thereby substantially reducing the traffic demand on the network. Fig. 1 shows the architecture of the machine status broadcasting system.

ISBN 978-3-95450-146-5

REDESIGN OF ALARM MONITORING SYSTEM APPLICATION BeamlineAlarminfoClient AT DESY

S.Aytac, DESY, Hamburg, Germany

Abstract

The alarm monitoring system 'Beamline-AlarminfoClient' is a very useful technical-service application at DESY, as it visually renders the locations of important alarms in some sections (e.g. fire or other emergencies). The aim of redesigning this application is to improve the software architecture and allow the easy integration of new observable areas including a new user interface design. This redesign also requires changes on server-side, where alarms are handled and the necessary alarm information is prepared for display. Currently, the client manages alarm data from 17 different servers. This number will increase dramatically in 2014 when new beam lines come into play. Thus creating templates to simplify the addition of new sections makes sense both for the server and client. The client and server are based on the Tine control system and make use of the Tine-Studio utilities, the Alarm Viewer and the Archive Viewer. This paper presents how the redesign is arranged in close collaboration with the customers.

INTRODUCTION

BeamlineAlarminfoClient (BAiC) is a visualization of emergency alarms in different areas. Currently it is used in the experimental halls PETRA III, FLASH and Photon-Science (PS) with 17 areas to be monitored.

The start of the first project was in 2004 and only used for monitoring FLASH area. Then PS alarms were added and at finally PETRA III. Since 2012 the FLASH extension project has been running and separated to FLASH-I and FLASH-II with independent FEL sources. In 2014 the PETRA III extension project began, with additional halls in the north of the storage ring and one in the east. The near future will give us ~30 areas to be monitored.

Aim of the Software Project

The functionality of the application is mostly defined by the customers and is regarded as an additional diagnostic for the technical-service personnel at DESY.

The followings alarms are monitored and displayed:

1. gas (concentration, magnetic- & exhaust valves)
2. fire
3. water
4. emergency call
5. emergency stop
6. common errors

The GUI is split into main and area views (outlined in Figure 1). If an area alarm is identified, a popup window of this area becomes visible, the active alarm is listed in the area table, and the alarm location toggles on the area plan. When the alarm is cancelled the popup becomes invisible and the alarm is listed in the main view alarm table of the last 72 hours.

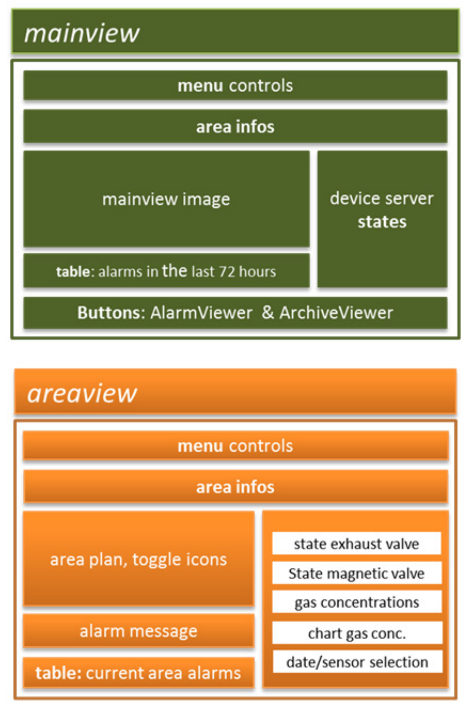

Figure 1: BeamlineAlarminfoClient GUI functionality of main and area view.

The Tine-Tool *Archive Viewer* is used for trending gas concentrations and *Alarm Viewer* for long time archiving of emergency alarms.

Motivation

Due to an unexpected increase in monitored areas, there is now a unique source code for every area server in the previous project. In 2013 a redesign of the project led to more maintainable software.

The main objective of this redesign is to reduce the project-code changes to a minimum of entries resulting from a new area as well as to reduce the communication links between server and client. In addition an upgrade of the graphical user interface is planned.

SYSTEM ARCHITECTURE

Every building displays one or more client application. The locations of these displays are at the entry of every building. Hence they are readily visible for technical-service personnel.

ISBN 978-3-95450-146-5

CONTROL SYSTEM SOFTWARE ENVIRONMENT AND INTEGRATION FOR THE TPS

Y. S. Cheng, Jenny Chen, C. Y. Liao, P. C. Chiu, C. H. Huang, C. Y. Wu, C. H. Kuo, K. T. Hsu
National Synchrotron Radiation Research Center, Hsinchu 30076, Taiwan

Abstract

The TPS (Taiwan Photon Source) is the latest generation 3 GeV synchrotron light source, and the commissioning starts from third quarter of 2014. The EPICS is adopted as control system framework for the TPS. The various EPICS IOCs have implemented for each subsystem. The control system software environment has been established and integrated specifically for the TPS commissioning. The various purposed operation interfaces have been created and mainly include the function of setting, reading, save, restore and etc. The database related applications have been built, and the applications include archive system, alarm system, logbook, Web and etc. The high level applications which are depended upon properties of each subsystem have been developed and are in test phase. The efforts will be summarized at this report.

INTRODUCTION

The TPS [1] is a latest generation of high brightness synchrotron light source which has been under construction at the National Synchrotron Radiation Research Center (NSRRC) in Taiwan since February 2010. The civil construction works are finished in later 2013. The TPS consists of a 150 MeV electron Linac, a 3 GeV booster synchrotron, and a 3 GeV storage ring, and the accelerator system installation and system integration had started from later 2013. The Linac and transport line (Linac-to-Booster) system had commissioned from half of 2014, and the booster ring is commissioning from the third quarter of 2014. The storage ring will schedule to commission from later 2014.

The EPICS (Experimental Physics and Industrial Control System) [2] framework was also selected as control system infrastructure for the TPS project. The EPICS platform has been gradually built and tested to control and monitor the subsystems of TPS. The various database records have been created for accessing the I/O data and setting parameters at the IOC (Input Output Controller) layer. Adopting the EPICS channel access mechanism with specific toolkits, the data can be accessed between the IOCs and the clients.

During the implementation process of the EPICS support for various subsystems, the operation interfaces of each subsystem are also developed according to the different operation methods. To simulate the operation process, the various operation interfaces are needed to integrate. The centralized management of the EPICS related files is also adopted, and the mechanism of save and restore will be continuously developed. The efforts will be summarized as following.

SOFTWARE ENVIRONMENT

The client consoles are adopted the Linux operation system. All of the EPICS base, modules, extensions and specific operation toolkits are installed at the Linux system. All EPICS related files at control consoles are mounted from the file server by using the NFS service to simplify software version management. Various directories are created and saved into various versions of related files for various hosts and purposes. Various directories provide a mount point for hosts mounted according to various purposes. The directories include EPICS base, modules, extensions, GUIs, saved data, temporary data and etc.

Several file servers are established to share the loading of NFS file service. The hosts mount specific file server according to its location and purpose. By loading testing, the NFS file service is divided into three parts. Two servers provide the NFS service for all IOCs and consoles; the other server is for engineer development.

SAVE AND RESTORE MECHANISM

To readily restore a set of the machine parameters for subsystems during operation as well as to optimize and record working points for different machine conditions, the mechanism of save and restore is developed. The save and restore function is established by using the MATLAB with the labCA [3]. The various files of grouped PVs (Process Variables) list are created for saving the respective parameter values of each subsystem. The file with PVs and saved parameters is also selectable for resume the settings. The graphical operation interface of save and restore mechanism is shown as the Fig. 1.

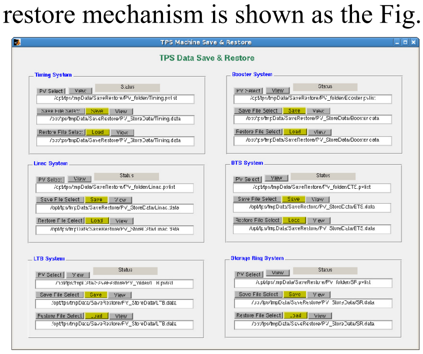

Figure 1: The OPI of save and restore mechanism.

SUBSYSTEMS CONTROL PAGES

At the commissioning phase, the main GUIs of TPS control system adopt the EDM (Extensible Display Manager) [4] toolkit to develop main graphical operation interface. The EDM OPIs of various subsystems are

ISBN 978-3-95450-146-5

POWER SUPPLIES TRANSIENT RECORDERS FOR POST-MORTEM ANALYSIS OF BPM ORBIT DUMPS AT PETRA-III

G. K. Sahoo#, P. Bartkiewicz, A. Kling, B. Pawlowski
Deutsches Elektronen Synchrotron, DESY, Hamburg, Germany.

Abstract

PETRA-III is a 3rd generation synchrotron light source dedicated to users at 14 beam lines with 30 instruments. The storage ring is presently modified to add 12 beam lines. PETRA III was operated with several filling modes such as 40, 60, 480 and 960 bunches with a total current of 100 mA at electron beam energy of 6 GeV. The horizontal beam emittance is 1 nmrad while a coupling of 1% amounts to a vertical emittance of 10 pmrad. During a user run the Machine Protection System (MPS) may trigger an unscheduled beam dump if transients in the current of magnet power supplies are detected which are above permissible limits. The trigger of MPS stops the ring buffers of the 226 BPM electronics where the last 16384 turns just before the dump are stored. These data and transient recorder data of Magnet Power Supply Controllers are available for a post-mortem analysis. Here we discuss in detail the functionality of a Java GUI used to investigate the transient behavior of the differences between set and readout values of different power supplies to find out the responsible power supply that might have led to emittance growth, fluctuations in orbits or beam dumps seen in a post-mortem analysis.

INTRODUCTION

PETRA-III [1] is a 3rd generation synchrotron light source commissioned with electron beam energy of 6 GeV and 100mA stored current at betatron tune values of 36.12 and 30.28. The horizontal beam emittance is 1 nmrad while a coupling of 1% amounts to a vertical emittance of 10 pmrad. The machine is dedicated to users for experiments from 14 beam lines with 30 end-stations. The storage ring is presently being modified to incorporate 12 new beam lines including a Superlumi beam line from dipole radiation. PETRA operates with several filling modes, such as 40, 60, 480 and 960 bunches with a beam current of 100 mA. During the normal user operation, there are unscheduled beam dumps triggered by the Machine Protection System (MPS) [2, 3]. These triggered dumps may occur before or some times after the loss of beam. The reasons for beam loss due to the MPS are of course understood. But the loss of beam prior to the beam dump by the MPS or a sudden fall of beam current, are unexpected. In these cases the reason remains unidentified or in some cases undetected. However, although the beam is lost, it leaves its signature in its post-mortem data. These post-mortem data are huge and contain a lot of information which can be extracted and analyzed in a Java Web Application MEOC [4]. Here we discuss how the Power Supply Controller (PSC) Transient Recorders are used in the post-mortem analysis to pin point the source of disturbance in magnet power supplies.

Gajendra.Kumar.Sahoo@desy.de

TECHNICAL OVERVIEW

All PETRA III magnets are driven by power supplies designed and manufactured at DESY (Fig. 1), controlled by intelligent PSCs [5].

Figure 1: PETRA-III power supply modules.

The PSC design is based on generic controller mezzanine boards (Fig. 2), designed also at DESY and widely used for other control purposes as well. The board consists of Freescale Coldfire (MCF5282) microcontroller and Altera FPGA (Flex EPF10K50), offering not only control and communication capabilities, but also enough resources needed for real time output current monitoring and transient recording.

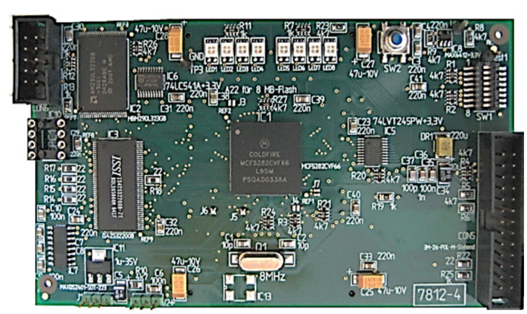

Figure 2: Generic Freescale Coldfire MCF 5282 based mezzanine card widely used for control purposes.

The 614 PSCs communicate over CAN buses (CANopen protocol [6]) with 20 front-end servers, running on PC104 Fanless Industrial Computers [7] with an embedded Linux operating system. The TINE [8, 9] network environment integrates the front-end servers with the PETRA-III control system and provides them access to central services, like data archiving, alarms and events recording systems (Fig. 3).

ISBN 978-3-95450-146-5

TPS SCREEN MONITOR USER CONTROL INTERFACE

C. Y. Liao#, Y. S. Cheng, Demi Lee, C. Y. Wu, K. H. Hu, C. H. Kuo, K. T. Hsu
NSRRC, Hsinchu 30076, Taiwan

Abstract

The Taiwan Photon Source (TPS) is being constructed at the campus of the NSRRC (National Synchrotron Radiation Research Center) and in commissioning. For beam commissioning, the design and implementation of a screen monitor system for beam profile acquisition, analysis and display was done. A CCD camera with Gigabit Ethernet interface (GigE Vision) is a standard device for image acquisition, to be undertaken with an EPICS IOC via a PV channel; display beam profile and analysis properties are made with a Matlab tool. The further instructions for the design and functionality of the GUI were presented in this report.

INTRODUCTION

Taiwan Photon Source (TPS), a 3 GeV third generation synchrotron light facility, featuring ultra-high photon brightness with extremely low emittance which is being installation at National Synchrotron Radiation Research Center (NSRRC). For beam commissioning and optimize machine operation, the two-dimensional beam-related images were recorded by screen monitor in Linac, LTB, booster, BTS and storage ring, which are widely used in synchrotron light source facility. Due to the most of machine parameters in future TPS [1] will be accessible as EPICS (Experimental Physics and Industrial Control System [2]) process variables (PVs). Thus, an analysis tool use the PVs as inputs with ability to calculate and display results in complex ways is needed. The screen monitor user control interface design and its functionality are present in this report.

LAYOUT OF SCREEN MONITOR

For the TPS beam diagnostic application distributed in Linac, LTB, BTS, booster (BR), and storage ring (SR), the screen monitor is responsible for the beam profile acquisition from YAG:Ce screen and used to analysis to find the beam characteristic data. The location and quantity of the screen monitor is listed in Table 1. The beam profile image has extensive information on beam parameters, including beam center, sigma, tilt angle and etc. The optical system contains screen, lens, and lighting system. The screen monitor assembly consists of a hollow tube, a YAG:Ce screen with 25 mm in diameter and 0.5 mm in thickness. The YAG:Ce screen is mounted at 45^{o} angle in one side to intercept the beam. A vacuum-sealed window is in the other end of the tube to extract the light. A CCD camera is mounted at a supporting tube with LEDs installed beside the CCD camera for illumination. A pneumatic device is used to move the whole assembly in or out. All of these devices are controlled remotely including the CCD power control, screen in/out control and LED lighting system. The structure of the screen monitor assembly is shown in Fig. 1. The PoE CCD camera with Gigabit Ethernet interface (GigE Vision) will be a standard image acquisition device. The CCD timing trigger clock is locked with TPS injection system, which is produced from a local timing IOC (EVR).

#liao.cy@nsrrc.org.tw

Table 1: Location and Quantity of the Screen Monitor

Location	Quantity
Linac	5
LTB	5
Booster	6+1*
BTS	5
Storage ring	1+3*

* Plus additional screen monitor are temporarily installed during the commissioning.

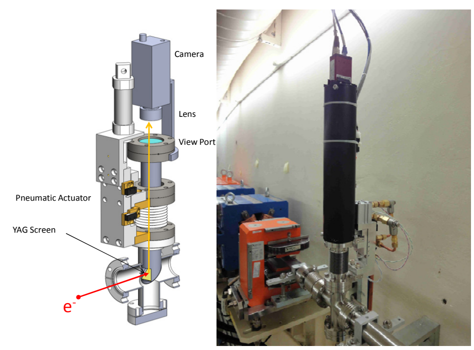

Figure 1: Screen monitor assembly in TPS booster.

USER INTERFACES DEVELOPMENT

Camera Parameters and Screen Position Control GUI

Based on the areaDetector module (R1-9-1) [3-4] which provides a general-purpose interface for area (2-D) detectors in EPICS, it is easy to construct a camera control panel by using the EDM tool. The TPS main control panel is shown in Fig. 2, the screen monitor launch page is marked. The screen monitor user interfaces for the TPS Linac, LTB, BR, BTS, and SR are shown in Fig. 3, which can switch in between in one GUI. The camera parameters of exposure time and gain can be configured in this panel, the camera location also shown below. The trigger mode selection include that the Free-Run for simply monitor the image and Sync-In for synchronization of linac injection (3 Hz). This EDM panel only offer lunch the screen, control the CCD parameters and simple monitoring features but do not perform any calculations. The screen position control also can perform in this GUI, the flow state as shown in Fig. 4.

ISBN 978-3-95450-146-5

BEAMLINE DATA MANAGEMENT AT THE SYNCHROTRON ANKA

A. Vondrous#, T. Jejkal, D. Ressmann, W. Mexner, R. Stotzka
KIT, 76344 Eggenstein-Leopoldshafen, Germany

Abstract

We present a data management architecture consisting of beamline data management (BLDM) and data repository to enable data management at the synchrotron facility ANKA. Nearly each measurement device writes data with a different format, size and speed on storage devices that are distributed over the synchrotron facility. The operators perform some data management tasks manually and individually for each measurement method. In order to support the operators, users and data analysts to manage the datasets, it is necessary to collect the data, aggregate metadata and to perform ingests into the data repository. The data management layer between the measurement devices and the data repository is referred to beamline data management, which performs data collection, metadata aggregation and data ingest. Shared libraries contain functionalities like migration, ingest or metadata aggregation and form the basis of the BLDM. The workflows and the current state of execution are persisted to enable monitoring and error handling. After data ingest into the data repository, archiving, content preservation or bit preservation services are provided for the ingested data. The data repository is implemented with the KIT Data Manager. In summary, BLDM can connect the existing infrastructure with the data repository without major changes of routine processes to build a data repository for a synchrotron.

INTRODUCTION

Data management is essential for science in the information age. Preserving primary data produced in scientific investigations is not only important to establish scientific integrity [1, 2, 3]. The commodity for gaining knowledge and competitive advantages is data. Extracted relations from data stocks provide insights and help to understand phenomena. Beside good scientific practice and efficient data analysis methods, science also benefits from sharing data.

Astronomy is the textbook example for increasing the scientific outcome by sharing data. As an example, according to the Hubble space telescope bibliography [4] about 50% of the published papers related to measurements of the Hubble space telescope are based on publically available data.

To create the basement for responsible data preservation, the European commission [5] introduces data management requirements for funding scientific investigations. Large facilities like synchrotrons produce a not negligible amount of valuable data, such that we try to find a convenient way to establish data management capabilities for synchrotrons.

The following two examples of biology and materials science illustrate the reasons for an elaborated data management at a synchrotron facility.

The value of measurement data produced in a synchrotron is not only determined by hardware, experience and operational methods, it is also determined by the application. As an example, measurements of biological specimens reveal insights into the biomechanical processes of insects as shown e.g. by van de Kamp et al. in [6]. The value to the biology community is higher than the actual costs because the insights answer questions, confirm argumentation chains and create new perspectives. Storing the raw and derived data is of interest to preserve the findings for further analysis.

Sometimes, the scientific application does not exist yet and the value of the measurement data cannot be properly estimated. Measurements in the field of materials science have the potential to be of use for science and industry in the future. The crystallography open database [7] is one example for preserving crystallographic data and providing access to structured datasets.

Storing data in a structured manner for long-term usage is a challenge in the synchrotron context. The broad spectral range of the produced light and the number of measurement huts (beamlines) enable the usage of diverse measurement methods simultaneously.

Manual data management by beamline operators with logbooks or with spreadsheets is common practice. Tasks as search, retrieval, analysis or conversion are becoming demanding considering large files, many files or distributed storage locations.

To preserve valuable data, to automate data management tasks and to support operators and users at the synchrotron ANKA, a data repository is going to be implemented. The data repository keeps track of all datasets and provides data services like search, preservation, analysis, publication, curation, processing or migration.

Those benefits of a data repository are not for free. It is necessary to agree on data structures, to define metadata schemes and to aggregate the metadata. In addition, the data management has to be aware of the synchrotron specific infrastructure properties. Finally, the data management should interfere as less as possible to enable smooth beamline operation.

In this work we present data management from the source to the data repository considering synchrotron specific requirements.

METHOD

We divide the data flow from the data source (camera or detector) to the archive into two management areas. The first area is beamline data management (BLDM),

Corresponding author: alexander.vondrous@kit.edu

ISBN 978-3-95450-146-5

RENOVATING AND UPGRADING THE Web2cToolkit SUITE: A STATUS REPORT

R. Bacher, DESY, Hamburg, Germany

Abstract

The Web2cToolkit is a collection of Web services. It enables scientists, operators or service technicians to supervise and operate accelerators and beam lines through the World Wide Web. In addition, it provides users with a platform for communication and the logging of data and actions. Recently a novel service, especially designed for mobile devices, has been added. Besides the standard mouse-based interaction it provides a touch- and voice-based user interface. In addition, Web2cToolkit has undergone an extensive renovation and upgrading process. Real WYSIWYG-editors are now available to generate and configure synoptic and history displays, and an interface based on 3D-motion and gesture recognition has been implemented. Also the multi-language support and the security of the communication between Web client and server have been improved substantially. The paper reports the complete status of this work and outlines upcoming development.

INTRODUCTION

The Web2cToolkit [1] is a collection of Web services, i.e. servlet applications and the corresponding Web browser applications, including

1. *Web2cViewer*: Interactive synoptic live display to visualize and control accelerator or beam line equipment,
2. *Web2cViewerWizard*: Graphical WYSIWYG-editor to generate and configure synoptic displays,
3. *Web2cArchiveViewer*: Web form to request data from a control system archive storage and to display the retrieved data as a chart or table,
4. *Web2cArchiveViewerWizard*: Graphical WYSIWYG-editor to generate and configure archive viewer displays,
5. *Web2cGateway*: Application programmer interface (HTTP-gateway) to all implemented control system interfaces,
6. *Web2cMessenger*: Interface to E-Mail, SMS and Twitter,
7. *Web2cLogbook*: Electronic logbook with auto-reporting capability,
8. *Web2cManager*: Administrator's interface to configure and manage the toolkit, and
9. *Web2cToGo*: Interactive display especially designed for mobile devices [2] embedding instances of all kinds of Web2cToolkit services.

The Web2cToolkit provides a user-friendly look-and-feel and its usage does not require any specific programming skills. By design, the Web2cToolkit is platform independent. Its services are accessible through the HTTP/HTTPS protocol from every valid network address if not otherwise restricted. A secure single-sign-on user authentication and authorization procedure with encrypted password transmission is provided.

The Web 2.0 paradigms and technologies used include a Web server, a Web browser, HTML5 (HyperText Markup Language), CSS (Cascading Style Sheets) and AJAX (Asynchronous JavaScript And XML). The interactive graphical user interface pages are running in the client's native Web browser or in a Web browser embedded in a mobile app or desktop application. The interface is compatible with almost all major browser implementations including mobile versions. The Web2cToolkit services are provided by Java servlets running in the Web server's Java container. The communication between client and server is asynchronous. All third-party libraries used by the Web2cToolkit are open-source.

The Web2cToolkit provides interfaces to major accelerator and beam line control systems including TINE [3], DOOCS [4], EPICS [5] and TANGO [6]. The toolkit is capable of receiving and processing video frames or a continuous series of single images.

In addition the toolkit provides an environment to support and test various Human-Machine-Interface (HMI) types including mouse, touch, speech and 3D-gestures depending on the capabilities of the underlying platform [7] and the proper design and handiness of multi-modal accelerator control system applications.

IMPROVED AND NOVEL FEATURES

Recently, the Web2cToolkit suite has been substantially renovated and upgraded. Besides smaller modifications such as supporting IP6-compliant client address encoding the objectives of this process include

- Improving the security of the communication between Web client and server
- Improving the multi-language support of the Web2c Viewer,
- Providing a specific user repository holding all user-defined configurations within the web server's directory structure,
- Supplying wizard applications to graphically design Web2cViewer synoptic displays and Web2cArchiveViewer history pages.
- Redesigning the interface for extending the Web2c Viewer servlet with code provided by the user,
- Redesigning the interface for connecting other accelerator and beam line control systems with the Web2c Viewer servlet, and

ISBN 978-3-95450-146-5

OpenGL-BASED DATA ANALYSIS IN VIRTUALIZED SELF-SERVICE ENVIRONMENTS

V. Mauch, M. Bonn, S. Chilingaryan, A. Kopmann, W. Mexner, D. Ressmann
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Abstract

Modern data analysis applications for 2D/3D data samples require complex visual output features which are often based on OpenGL, a multi-platform API for rendering vector graphics. They demand special computing workstations with a corresponding CPU and GPU power, enough main memory and fast network interconnects for a performant remote data access. For this reason, users depend heavily on available free workstations, both temporally and locally. The provision of virtual machines (VMs) accessible via a remote connection could avoid this inflexibility. However, the automatic deployment, operation and remote access of OpenGL-capable VMs with professional visualization applications is a non-trivial task.

In this paper, we discuss a concept for a flexible analysis infrastructure that will be part in the project ASTOR, which is the abbreviation for "Arthropod Structure revealed by ultrafast Tomography and Online Reconstruction". We present an Analysis-as-a-Service (AaaS) approach based on the on-demand allocation of VMs with dedicated GPU cores and a corresponding analysis environment to provide a cloud-like analysis service for scientific users.

INTRODUCTION

Particle accelerators like ANKA [1] provide synchrotron radiation for the investigation of solid material and biological samples. X-ray imaging techniques produce large data sets of up to several 100 Gigabytes. The subsequent processing of the data requires special analysis applications whose features and performance depend heavily on available CPU, GPU and memory resources. Typical analysis application like Amira [2] or VG Studio MAX [3] provide visual output based on OpenGL [4], a multi-platform API for rendering vector graphics. Up to now, several conditions concerning the user analysis process have complicated a flexible analysis workflow:

- transfer, access and storage of huge data sets
- required amount of computing and memory resources
- OpenGL/DirectX capable GPUs for visual output, eventually also CUDA [5] or OpenCL [6] are necessary
- expensive workstation licenses for commercial analysis software

All aspects mentioned above prevent the use of standard end-user devices for analysis and result in the operation of dedicated workstations for scientists. The required professional workstations need to provide a high-throughput network connection to the data set storage, licensed analysis software and hardware setup. This scenario implies further disadvantages for users. They have to rent free time ranges for the workstation usage. Therefore they have to be present in special computing rooms within their institution shared with other users. Due to the operation of powerful hardware combined with an increased heat generation, inside the computing room one can often sense a high background noise. Furthermore, the workstations are typically configured with limited guest account privileges, which is why users have to consult an administrator to install additional software packages for a special purpose.

To improve this situation, one part of the project ASTOR deals with the automatic deployment of virtual machines (VMs) with remote connections supporting the rendering and display of OpenGL features. Scientists should be able to use a web portal with an overview of their available data sets and allocate virtual resources for their analysis on-demand. In this paper we discuss an Analysis-as-a-Service (AaaS) concept and present a first prototype implementation for the synchrotron community.

CONCEPT

The replacement of static localized stand-alone workstation towards a flexible analysis infrastructure concept for scientists is achievable with an Infrastructure-as-a-Service (IaaS) approach. Besides the provision of virtualized analysis environments, a suitable analysis workflow has to be defined which also considers the previous data detection and recording, data set access and the final result archiving. The complete analysis workflow could be considered as a novel Analysis-as-a-Service (AaaS) concept, defining a cloud-like service for individually customized analysis processes.

The main part of this concept is the intelligent integration of virtualized analysis environments. However, until a few years ago, the provision of remote connections to VMs supporting OpenGL or other complex visual APIs was impossible, as most virtualization solutions only came up with simplified graphic interfaces within a guest system. Furthermore the remote access protocol is a key aspect with regard to lossless data compression and smooth transfer of visual information. It must ensure a low-latency user interaction without any disturbing lag effects, as most analysis processes require several hours of work. Currently, there are just a few complete solution suites available offering the provision of virtualized workstations for professional visualization applications:

- Citrix XenDesktop [7]
- Microsoft RemoteFX [8]
- VMware Horizon View [9]

ISBN 978-3-95450-146-5

MAKING IT ALL WORK FOR OPERATORS

I. Kriznar, Cosylab, Ljubljana, Slovenia
S. Marsching, Aquenos GmbH, Baden-Baden, Germany
E. Hertle, E. Huttel, W. Mexner, N. J. Smale, A.-S. Mueller,
KIT, Eggenstein-Leopoldshafen, Germany

Abstract

As the control system of the ANKA synchrotron radiation source at KIT (Karlsruhe Institute of Technology) is being slowly upgraded it can become, at key stages, temporarily a mosaic of old and new panels while the operator learns to move across to the new system. With the development of general purpose tools, and careful planning of both the final and transition GUIs, we have been able to actually simplify the working environment for machine operators. In this paper we will explain concepts, guides and tools in which GUIs for operators are developed and deployed at ANKA.

INTRODUCTION

The machine control system of the synchrotron radiation source ANKA at KIT (Karlsruhe Institute of Technology) is migrating from the ACS CORBA based control system to the Ethernet TCP/IP devices with an EPICS server layer and visualisation by Control System Studio (CSS). This migration is driven by the need to replace ageing hardware. Approximately 500 physical devices, are being gradually replaced (or have their I/O hardware changed) and are integrated to the EPICS/CSS control system. [1]

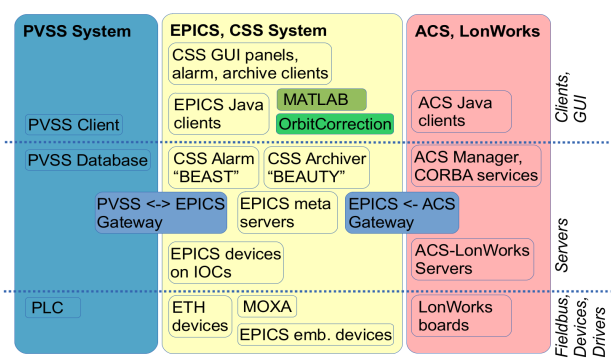

Figure 1: Patchwork or different components and technologies in ANKA control system.

The storage ring is generally operated at an energy of 2.5GeV with a typical beam current of 200 mA and a lifetime of 20 hours. Due to the finite lifetime the storage ring is emptied of electrons twice per day (8:00 and 18:00) and refilled. The refilling process involves a two stage accumulation process. The machine is then left unattended until the next injection time, which leads to two important demands on the control system, namely, robust and sensitive alarm notification in the case of reduced machine performance and a very intuitive GUI machine interface; as a single operator will only attend to the machine for approximately 10 hours per 2 months.

We can already say that the ANKA control system is based on EPICS. All core services and tool-kits are done with EPICS/CSS. However, there are still additionally other control systems, which need to be used in parallel or integrated into EPICS for various reasons. Namely PVSS and ACS (Figure 1).

GENERAL CONVENTIONS

CSS is based on Eclipse RPC and on Java, for which there exist several good references for building functional and pleasant GUI applications [2, 3]. In addition to these extensive documents ANKA has several simple guidelines that address most frequent mistakes, which are encountered:

- Use default look and feel, default fonts and sizes as provided by CSS/Eclipse.
- Design elements must be consistent across all applications and components. Buttons, check-boxes, radio-boxes, combo-boxes have well known and expected functionality; Don't misuse them or change behaviour. Don't change their labels, functionality and position in runtime ether.
- Some colours are reserved, borders are reserved for alarm notifications. It is important to use them only for designated purposes, don't confuse users with design choices that are similar to reserved use.
- Leave just the right space between components, organize them so they are equally spaced and lined to same base-lines. There are tools in CSS visual composition toolbar that help you do just that. This point should be trivial and obvious, but for some reason it is often not followed.

THE ANKA CLIENTS BUNDLE

ANKA has several distribution channels for different generations of control system clients. Distribution channels are kept in Subversion repository. Installation and updating are performed through Subversion operations on target computers. Subversion was chosen for its convenience over competing solutions. Subversion can effectively keep history of changes and it is used primary for one way distribution, so something like Git would not be fully utilized and would increase complexity for simple update operations.

Subversion distribution channels of control system clients covers: old ACS client distribution, the PVSS distribution and the main ANKA client distribution. The main ANKA client distribution, called ANKA-Clients, is kept in two branches: the main trunk is used for storing stable releases running on all operator's computers. The

ISBN 978-3-95450-146-5

HOW THE COMETE FRAMEWORK ENABLES THE DEVELOPMENT OF GUI APPLICATIONS CONNECTED TO MULTIPLE DATA SOURCES

R. Girardot, G. Viguier, K. Saintin, M. Ounsy, A. Buteau
SOLEIL Synchrotron, Gif-sur-Yvette, France

Abstract

Today at SOLEIL, our end users require that GUI applications display data coming from various sources: live data from the Tango control system, archived data stored in the Tango archiving databases and scientific measurement data stored in NeXus/HDF5 files. Moreover they would like to use the same collection of widgets for the different data sources to be accessed.

On the other side, for GUI application developers, the complexity of data source handling had to be hidden. The COMETE framework has been developed to fulfil these allowing GUI developers to build high quality, modular and reusable scientific oriented GUI applications, with consistent look and feel for end users.

COMETE offers some key features to software developers:

- A data connection mechanism to link the widget to the data source
- Smart refreshing service
- Easy-to-use and succinct API
- Components can be implemented in AWT, SWT and SWING flavours.

This paper will present the work organization, the software architecture and design of the whole system. We'll also introduce the COMETE eco-system and the available applications for data visualisation.

CONTEXT

The SOLEIL ICA team is in charge of the control systems and data visualisation and reduction GUI for accelerators and 30 beamlines.

During the first years of SOLEIL construction, ICA team was focused on developing GUI application and TANGO [1] devices for the control systems.

Then the focus was set on providing data storage and management applications for technical and scientific data.

- For the technical data the Tango Archiving service [2] was developed with very demanding requirements on the GUI for data extraction and visualisation. Today a volume of about 10 TB of data coming from more than 30 000 Tango attributes are stored in Oracle databases.
- For scientific data, it was early decided to use the NeXus data format [3] to record measurements and metadata on all our beamlines. SOLEIL beamlines produce daily thousands of experimental NeXus files with sizes ranging from a few MB up to 100 GB.

Moreover, users needed to use uniformed GUI to view their data, whatever the origin.

Control system and supervision GUI applications were first developed using the ATK [4] toolkit which is intimately linked to TANGO.

On the other hand, for technical and scientific data visualisation and reduction, there was no toolkit available to quickly develop this kind of software not TANGO based.

It is in this context that SOLEIL launched the COMETE project [5] to propose a multi data source toolkit to help software engineers to develop applications independently of the data source.

THE COMETE SOLUTION

Architecture

COMETE is a framework composed of three parts:

1. A set of graphical components (widgets) that are completely dissociated from a data source or even a data type.
2. A data source compatible with the graphical component, each corresponding to a data type.
3. Between the widgets and the data source, a mediation layer in charge of adapting and transmitting data.

DataConnectionManagement

The DataConnectionManagement module is a layer that allows connection between two abstract entities, called “**Target**” and “**Data Sources**”.

This module implements a Mediator pattern (Figure 1), as well as various other patterns such as Strategy, Observer etc. Mediator was chosen instead of MVC pattern because our two entities had to be completely independent from each other, to allow easily adding new widgets and sources.

The sources are produced by factories, which include some data refreshment mechanism.

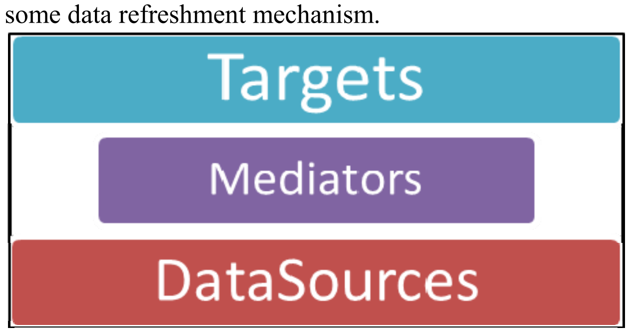

Figure 1: DataConnectionManagement pattern.

ISBN 978-3-95450-146-5

PANIC, A SUITE FOR VISUALIZATION, LOGGING AND NOTIFICATION OF INCIDENTS

S. Rubio-Manrique, F. Becheri, G. Cuní, D. Fernandez-Carreiras, C. Pascual-Izarra, Z. Reszela,
CELLS-ALBA Synchrotron, Barcelona, Spain

Abstract

PANIC is a suite of python applications focused on visualization, logging and notification of events occurring in ALBA Synchrotron Control System. Build on top of the PyAlarm Tango Device Server it provides an API and a set of graphic tools to visualize the status of the declared alarms, cre- ate new alarm processes and enable notification services like SMS, email, data recording, sound or execution of Tango commands. The user interface provides visual debugging of complex alarm behaviors, that can be declared using single-line python expressions. This article describes the architecture of the PANIC suite, the alarm declaration syntax and the integration of alarm widgets in Taurus user interfaces.

INTRODUCTION

ALBA[1], member of the Tango Collaboration[2], is a third generation Synchrotron lightsource in Barcelona, Europe. It provides synchrotron light since 2012 to users in its 7 beamlines, with 2 more under construction.

PANIC is an Alarm System running on top of the Tango Control System to provide periodic evaluation of user-specified alarm formulas, automatic actions and notification whenever formulas evaluate to True, logging of the control system status when this occurs and later monitoring and supervision of the evolution of the system.

Elements of the PANIC Alarm System have been deployed at ALBA[3] since the start of the construction phase. Developed to provide stand-alone monitoring during the vacuum installation of the accelerators it evolved into a versatile system in which many tools interact to provide not only a monitoring tool, but a supervisor service on top of a Tango Control System.

Other Alarm Systems existed already in Tango. PANIC was inspired on Elettra[4] (C++) and Soleil (Java) alarm systems; but focused on exploiting the versatility of python[5] to process rules on runtime, allowing operators and engineers to develop complex logics in rules[6].

THE PANIC ECOSYSTEM

The PANIC Alarm System is completely integrated in Tango[7]. Although it can work without a Tango Database using files as configuration, it develops its complete functionality when interacting in a complete Tango Control System.

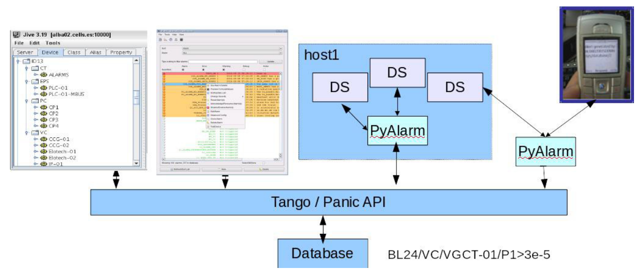

Figure 1: Architecture of the PANIC Alarm System.

The different elements that are part of a PANIC Alarm System (Fig. 1) are:

- PANIC Api: Alarms Database API and rule evaluator. Provides a unique view of the system and coherence between all devices and UI's. It encapsulates all the behavior that is later executed in the User interface or Device Servers.
- PyAlarm Device Server: Tango Device Server that, on top of the PANIC Api, executes alarm rules periodically and trigger the configured actions.
- ProcessProfiler Device Server, a Tango Device to inspect current performance of a linux system, exporting memory and cpu usage of all running processes to be used as source for Alarms.
- Festival Device Server: providing speech and pop-up notification in user terminals.
- PANIC User Interface: manager of the PyAlarm devices, editor of formulas and browser of the Tango Control System from alarms point of view.
- PANIC Tau Toolbar: simple Alarm viewer, restricted to alarms related to attributes shown in a running Tau application.
- Taurus Search Bar: it provides a search engine that indexes relationships between devices, attributes, properties, labels and alias within Tango.
- MySQL Databases: instead of having its own schema PANIC relies in databases of the Tango Control System for configuration (Tango DB) and logging (Tango Archiving DB).
- Snap Viewer: widget that browses the Tango Snapshoting Database (developed by Soleil Institute) where alarm logs and related attribute values are stored.
- Alarm NoSQL database: alternative database for unified configuration and logging, under development by Max IV team.

ISBN 978-3-95450-146-5