THE NEW CONTROL SYSTEM FOR THE FUTURE LOW-EMITTANCE LIGHT SOURCE PETRA 3 AT DESY: SPRINTING TO THE FINISH

R. Bacher, Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany

Abstract

At DESY the existing high-energy physics booster synchrotron PETRA 2 will be transformed into a 3rd-generation light source (PETRA 3). In addition the technical systems and components of the pre-accelerators LINAC 2 and DESY 2 will be improved. Within the scope of this project, the control system and the front-end electronics will be upgraded. Besides a report on the current project's status, the paper emphasizes the basic conceptual ideas and discusses their implications and how they lead to novel features and development tools.

INTRODUCTION

After having switched off the proton-lepton collider HERA 2, the booster PETRA 2 is currently being transformed into the high-brilliance 3rd-generation light source PETRA 3 [1]. The design values for the new storage ring are 6 GeV for the particle energy and 100 mA for the current. The transverse particle beam emittance is expected to be 1 nmrad. Fourteen undulator beam lines operated by HASYLAB (Hamburger Synchrotron-strahlungslabor), EMBL (European Molecular Biology Laboratory) and the GKSS research centre will provide photons for various experiments (X-ray diffraction and imaging, high-energy resolution spectroscopy, material science, X-ray absorption and resonant scattering as well as structural biology).

Within the scope of the PETRA 3 project, the accelerator control systems and fractions of the front-end electronics of PETRA as well as of the electron/positron pre-accelerators LINAC and DESY 2 will be upgraded almost simultaneously. While beam operation in the pre-accelerators will restart already in summer 2008, user beam operation at PETRA 3 is expected to start not before January 2009.

FRONT-END ELECTRONICS AND DEVICE INTERFACES

The upgrade of the front-end electronics is a particular challenge of the project. Fractions of the old existing DESY-proprietary legacy SEDAC field-bus network and the attached electronics have to be preserved, while hundreds of magnet or kicker power supply and vacuum controllers will be exchanged by modules based on CANopen industrial field-bus technology. In addition, the modern TwinCAT [2] soft Programmable Logic Controllers are becoming increasingly popular. Off-the-shelf electronics for beam diagnostics, signal digitizing and video capturing have to be integrated seamlessly.

CANopen-based Front-End Electronics

New developments concentrate on CANopen as field-bus interface standard. We have established a hardware standard using industrial 3U-Euro crates with general purpose electronic boards which we have developed based on the the Coldfire and HCS12 microcontroller [3] families. The corresponding Vector [4] CANopen implementations have been adapted to our needs. In order to communicate with the CANopen stack the developer of the application software registers the user specific variables in the CANopen object dictionary and provides the user specific code for a predefined set of call-back functions. In addition, a processor board based on Altera NIOS II is under development.

To improve flexibility and to ensure pin compatibility with the device hardware not being replaced, the general purpose processor boards are connected with user-specific boards implementing the corresponding electrical and mechanical interfaces. The cables to the user-specific equipment are connected at the back of the crate.

The CANopen bus lines are connected to PC104-like systems running embedded Linux and acting as control system device servers and as CANopen bus masters.

To facilitate field-bus operation and diagnosis the TICOM [5] software library has been developed. TICOM (TINE-based (see below) CANopen Manager) provides almost all bus master functionality requested by the CANopen standard and an efficient connectivity to the accelerator control system The TICOM server provides APIs to access CANopen devices and their corresponding object dictionaries. The TICOM viewer is a Java client application offering a user-friendly visualization of the bus topology and status. Optionally, the process data flow can be monitored.

Front-End Device Access with the Common Device Interface

The Common Device Interface CDI [6] has been developed to provide both a common interface to various front-end electronic standards and an interface to the accelerator control system (Fig. 1). It offers bus plugs for CANopen, SEDAC, RS232, Siemens SIMATIC PCS7/Ethernet and Beckhoff TwinCAT ADS. A plug for Libera [7] beam position modules is currently under development.

The devices attached to the corresponding buses are registered using entries in the CDI database. The user specifies the name of the entry, the bus type and address parameters, the data access rule (READ, WRITE, atomic WRITEREAD, …) and format (short, long, text, …) as well as data manipulation parameters (mask, pattern, limit, calibration rule).

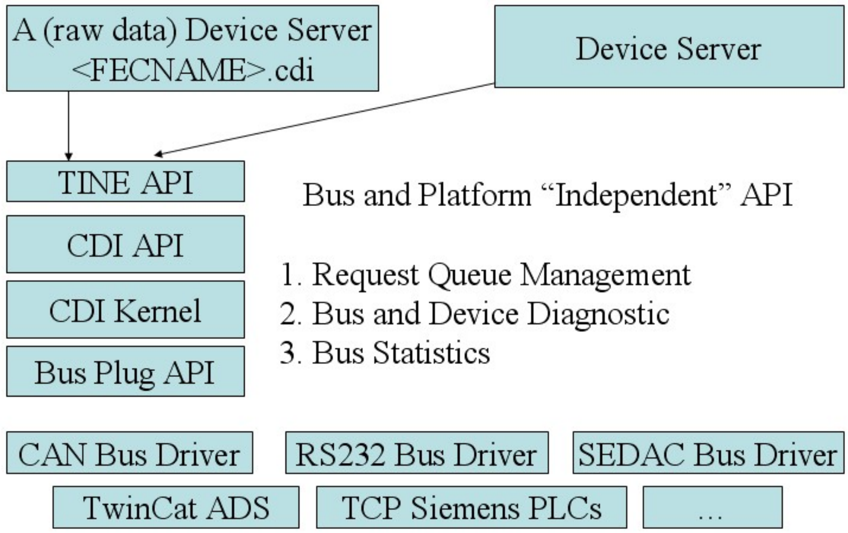

Figure 1: Architecture of the Common Device Interface

Devices of the same type can be described by template entries. Additional coding by the user is not required. On initialization, the CDI starts a TINE control system server which exports to the control system the device properties described in the CDI database.

TINE CONTROL SYSTEM SOFTWARE SUITE

The upgraded control system uses a distributed, multi-layer architecture linked by the integrating middleware or software bus TINE (Threefold Integrated Network Environment) [8], a set of communication protocols and services developed over the past years. TINE is now in a mature state. Version 4.0 has been released recently. TINE is stream-lined for efficient network communication and providing optimal flexibility for the control system users.

TINE is a multi-platform system, running on such legacy systems as MS-DOS, Win16, and VaxVMS as well as Win32, Linux, most Unix machines, MACOS, VxWorks and NIOS. TINE is also a multi-protocol system to the extent that UDP and TCP/IP as well as IPX are supported as data exchange protocols. Finally, TINE is a multi-control system architecture system, allowing client-server, publisher-subscriber, broadcast and multicast data exchange. The transmission of video frames (0.5 Mbyte each) at a 10 Hz repetition rate through a 100 MB Ethernet in multicast mode is routinely used at DESY [9].

TINE provides application programmer interfaces (APIs) for Java, VisualBasic, C/C++, LabView, MatLab and a command line interface for scripting tools. Code generating wizards available for C/C++, VisualBasic and JAVA facilitate the application development.

The TINE client/server implementation in C has been widely used for a long time while the corresponding JAVA implementation has been finished in the course of the PETRA 3 project.

Name services are provided with plug-and-play automated server registration. Address redirection allows the grouping of existing servers into virtual servers which hide potentially complex topologies from the client user.

TINE includes interfaces to several associated services. Data filtering and archiving, event handling, alarm filtering and archiving are already supported. An interface for central message processing and archiving will be developed.

The connectivity to other control systems is a unique feature of TINE. Powerful gateways are provided to seamlessly integrate EPICS, Tango and DOOCS servers.

APPLICATION DEVELOPMENT

Application development is a major and important task within the PETRA 3 control system upgrade project. Device and middle-layer servers are implemented in C/C++, Java, VisualBasic and LabView. The control room client applications for the accelerator operators, equipment experts, and scientists are implemented as rich client applications with Java, LabView and Matlab or as browser-based thin client applications using HTML and JavaScript.

Rich Client Java Applications with ACOP

According to our experience, control room applications based on the rich-client model are best suited for providing optimum visualization and performance. Examples for rich client applications are complex orbit displays which might combine position monitor readings and orbit correction tools or archive data viewer with the option to correlate different data channels in various contexts.

Framework classes for client and server applications have been developed to ensure design conformity and to handle initialisation data. To facilitate coding, the ACOP (Accelerator Component Oriented Programming) toolbox [10] is used for simple data access and rendition. The widely used ACOP chart component has been extended to a suite of different ACOP components offering a powerful graphical user interface. ACOP components are Swing components following the Java beans standard. Besides ACOP chart, a slider, a wheel switch, a dial knob, an animated label and a gauge component have been implemented. All ACOP components support design- and runtime customization such as adding a channel to a chart or dragging and dropping channel metadata to another application instance.

Thin Client Applications with the Web2cToolkit

Browser-based thin client applications are used to display overview information on the status of the various beam lines or to provide a simple access to control system data of common interest.

The Web2cToolkit [11] is a framework for thin "ajaxian" control system clients. AJAX is an acronym for Asynchronous JavaScript and XML. Applications designed with the Web2cToolkit framework are platform independent and accessible from everywhere without firewall constraints. They provide secure user authentication and specific rights for authorized users. Additional client-side software installation is not required. In general, the Web2cToolkit framework is designed to be control system independent. Currently, it provides a native TINE plug and access to various device control network or field-bus systems. The Web2cToolkit framework includes a synoptic display, a synoptic display editor and an archive viewer.

A Web2c synoptic display application is configured from a set of components such as static or dynamic labels, images or indicators, trend charts or histograms and ticker or time messages. Each configuration is saved in a corresponding xml-type configuration file. The application is displayed by the web browser and communicates asynchronously with a Java servlet (Fig. 2).

REFERENCES

[1] http://petra3.desy.de/
[2] http://www.beckhoff.de/
[3] http://www.freescale.com/
[4] http://www.vector-informatik.de/
[5] P.K. Bartkiewicz, T. Delfs, S.W. Herb, B. Pawlowski, DESY, Integration of CANopen-based Controllers with TINE Control System for PETRA 3, this conference (ID 1457)
[6] P. Duval, H.G. Wu, DESY, Using the Common Device Interface in TINE, PCaPAC 2006, Newport News, USA
[7] http://www.i-tech.si/
[8] P. Duval, The TINE Control System, Overview and Status, this conference (ID 1449)
[9] P. Duval, M. Lomperski, G. Trowitsch, S. Weisse, DESY, Status of a Versatile Video System at PITZ, PETRA 3, and EBML, this conference (ID 1562)
[10] J. Bobnar, I. Kriznar, Cosylab and P.K. Bartkiewicz, P. Duval, H.G. Wu, DESY, The ACOP Family of Beans: A Framework Independent Approach, this conference (ID 1424)
[11] http://adweb.desy.de/mst/web2cToolkit/

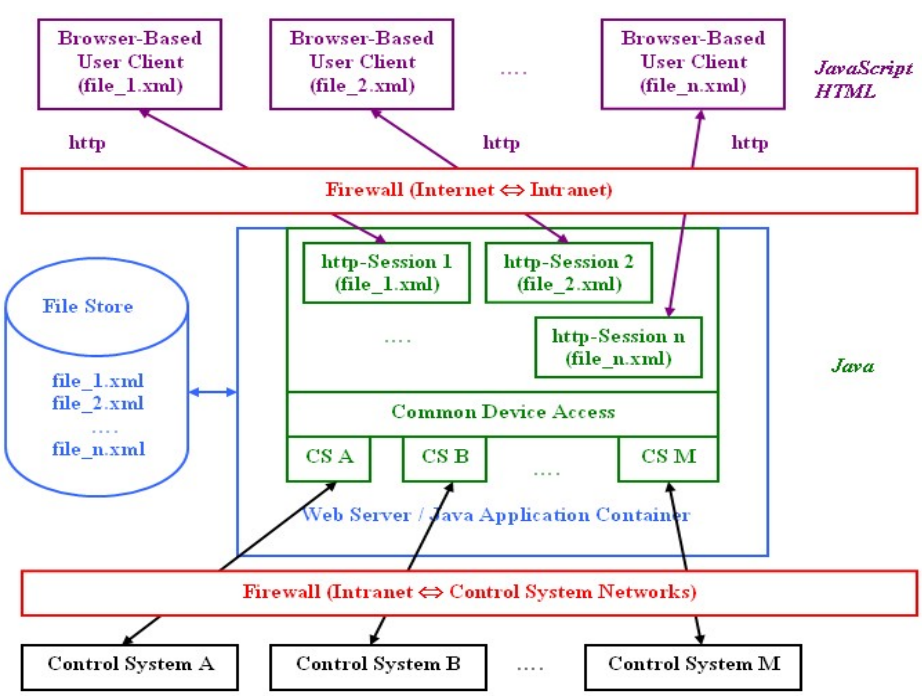

Figure 2: Web2cToolkit architecture