THE FAIR CONTROL SYSTEM – SYSTEM ARCHITECTURE AND FIRST IMPLEMENTATIONS

Ralf Huhmann*, Ralph C. Bär, Dietrich Hans Beck, Jutta Fitzek,
Günther Fröhlich, Ludwig Hechler, Udo Krause, Matthias Thieme
GSI Helmholtz Centre for Heavy Ion Research, Darmstadt, Germany

Abstract

The paper presents the architecture of the control system for the Facility for Antiproton and Ion Research (FAIR) currently under development. The FAIR control system comprises the full electronics, hardware, and software to control, commission, and operate the FAIR accelerator complex for multiplexed beams. It takes advantage of collaborations with CERN in using proven framework solutions like FESA, LSA, White Rabbit, etc. The equipment layer consists of equipment interfaces, embedded system controllers, and software representations of the equipment (FESA). A dedicated real time network based on White Rabbit is used to synchronize and trigger actions on equipment level. The middle layer provides service functionality both to the equipment layer and the application layer through the IP control system network. LSA is used for settings management. The application layer combines the applications for operators as GUI applications or command line tools typically written in Java. For validation of concepts already in 2014 FAIR's proton injector at CEA/France and CRYRING at GSI will be commissioned with reduced functionality of the proposed FAIR control system stack.

SYSTEM OVERVIEW

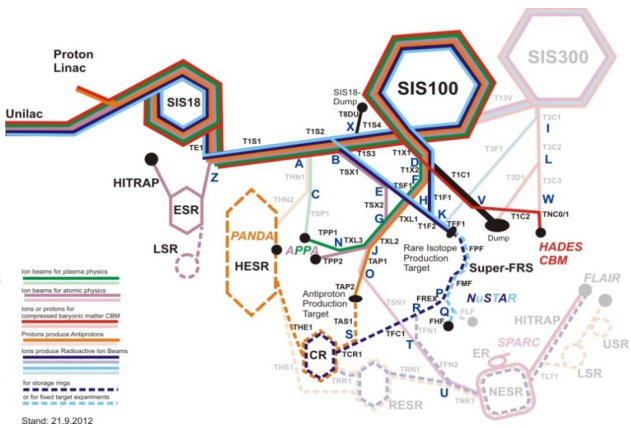

Figure 1: FAIR parallel beam operation © P. Schütt, GSI, Darmstadt.

The Facility for Antiproton and Ion Research (FAIR) supports operation of multiple beams in parallel (Fig. 1), i.e. several experiments can be served simultaneously. Common upstream accelerators are used virtually multiplexed by switching between predefined settings. Approx. 4000 devices are used in the facility, producing heavy ion and proton beams. Core research areas of the international accelerator facility are Nuclear Structure, Astrophysics, Antiproton Physics, High Energy Nuclear Physics, Atomic and Plasma Physics. The control system must support the multiplexed operation on all layers (Fig. 2). In order to trigger synchronized device action and to select coherent data settings in real time a dedicated Timing System based on a White Rabbit [1] network is used.

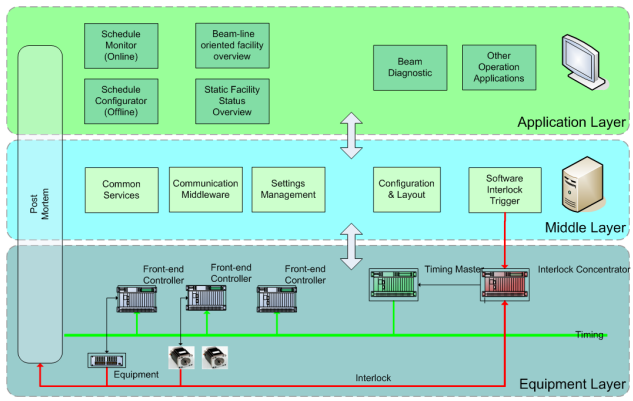

Figure 2: FAIR control system overview.

CORE COMPONENTS

Timing System

The timing system spans a Gigabit Ethernet network which is physically separated from the control system's IP network. Beside network infrastructure the timing network comprises one central timing master (TM) [2] and facility wide distributed timing receiver nodes (≈2000). The TM is clock master of absolute time for all timing receivers, clock and time synchronization are achieved by using White Rabbit (WR), which employs Gigabit-Ethernet, IEEE 1588-2008 (PTP), precise knowledge of the link delay, and Synchronous Ethernet [1, 3]. Today, synchronization in the one nanosecond range with a jitter in the low picoseconds range is achieved. Additionally, the TM centrally distributes timing event messages over the timing network to the timing receivers. The TM's interface to the upper layers, including its schedule of events to send, is modeled as a FESA device (see below). In particular, the schedule of the timing master is pre-supplied by LSA (see below). The timing event messages dispatched in advance by the TM carry *absolute*

* r.huhmann@gsi.de

ISBN 978-3-95450-139-7

execution time and encode an event type and data index. The event messages are queued and then punctually executed on the timing receiver site. The type and data index is used to trigger the appropriate action and to select the pre-supplied settings for the equipment. Available actions of the timing receiver include FESA real time actions (precision ≈100µs), direct equipment control via FPGA soft CPU (precision ≈ 1µs), and digital signal generation for nanosecond precision [3].

Equipment Controller

As a standard equipment controller the SCU (Scalable Controller Unit) [3, 4] has been developed. Fig. 3 shows the functional blocks of the SCU, it combines an x86 COM Express Board and an ™Altera Arria II FPGA. A parallel bus interface (SCU bus) allows to control up to 12 slave boards which connect to equipment via standardized or proprietary interfaces. Beside the SCU bus master the FPGA hosts a standard FAIR Timing Receiver for synchronization of equipment settings. The x86-Board hosts a Linux OS with real time patch to run FESA classes [5].

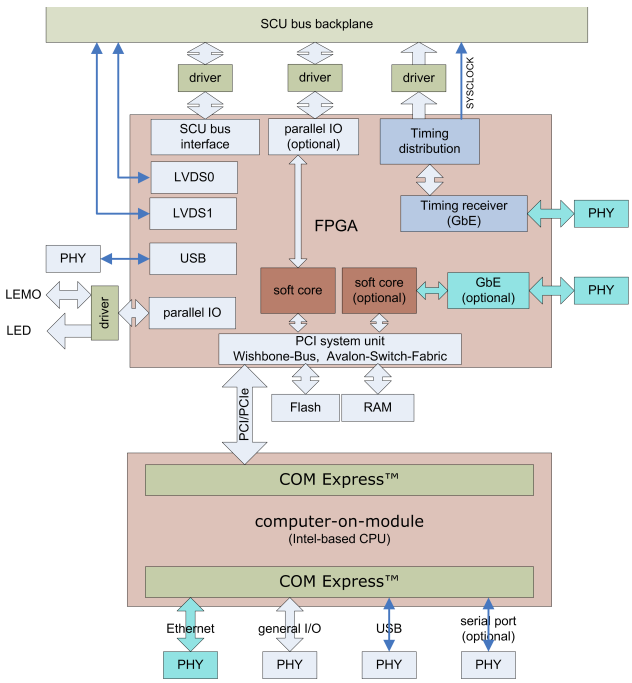

Figure 3: Functional blocks of the SCU.

Frontend Software (FESA)

FESA (Frontend Software Architecture) [5] is a framework developed at CERN and is now developed further in collaboration with GSI for the FAIR project. It is a toolbox to model abstract device objects where equipment's process variables (sensors and actuators) are represented as properties. The specific equipment access is implemented in C++ by the developer and is linked by the toolchain to the device model to build a so called FESA class (Fig. 4). Then, one or more FESA classes are linked to

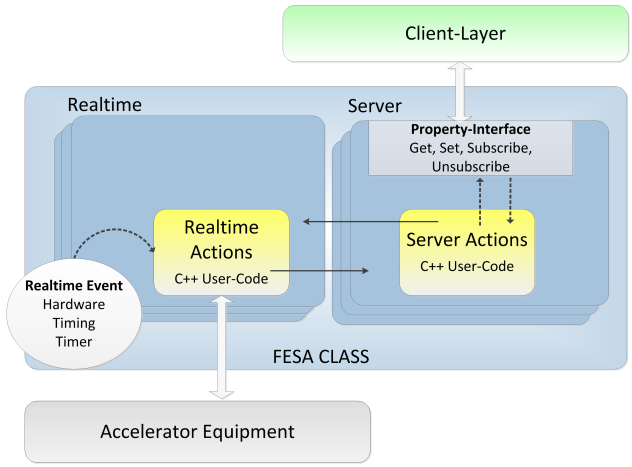

Figure 4: Functional overview of a FESA class.

the run-time core to build an x86-Linux executable. The FESA classes provide a uniform interface via the object-property model and a common middle-ware to the upper layers. The device properties are set and read using synchronous or asynchronous access methods (subscription). For time multiplexed operation of the accelerators, the FESA framework supports defining multiplexed properties. Before an accelerator schedule is started the setting properties of FESA classes are pre-supplied by LSA [6] for all scheduled beams with specific settings accordingly. At runtime, FESA's real time software actions are triggered by timing events, the actual beam specific data is then selected based on information carried by the timing event message and send to the equipment. For the FAIR project the necessary interaction with the timing receiver is realized in a lab-specific timing library of the FESA framework.

LSA

LSA (LHC Software Architecture) [6] is a settings management framework which originates at CERN and is now developed further in collaboration with GSI for the FAIR project. The service written in Java is located in the control system's middle-layer and supports offline generation of machine settings, sending these settings to all involved devices, and programming the schedule of the timing system. The settings management is based on a physics model for accelerator optics (twiss, machine layout), parameter space and overall relations between parameters and between accelerators. A standardized API allows accessing data in a common way as basis for generic client applications for all accelerators. Using the LSA-API, trim-applications can coherently modify machine settings. E.g. the service generates timing constraints (e.g. ramp curve) as well as the equipment's data settings (e.g. field) for all devices derived from physics parameters (e.g. beam energy). For FAIR the framework is extended to model the overall schedule of all accelerators. Beams are described as Beam Production Chains to allow a description from beam-source to beam-target for settings organization and data correlation.

ISBN 978-3-95450-139-7

Software Services

Amongst others, the following components of the middle layer provide common service functionality to the complete stack of control system software.

The Alarm System transports, processes and visualizes error conditions (malfunctions) of hardware and software. Those alarm states are propagated from producers to consumers. Producers of alarms are logical devices or middle layer services. A consumer of an alarm might be a human operator or software to process the alarm or to trigger reactions to the alarm.

The Archiving System stores acquisition data collected and generated by the control system. It stores data on a configurable repetition rate, acquisition triggered e.g. by timing events or on-change. The service includes functionality to retrieve and filter historical data.

The Diagnostic Logging System provides the functionality to store human readable text entries generated by the complete stack of control system software in a coherent way to support efficient querying. In first order these text entries are generated for diagnostic and debugging usage.

Operation Software

The operation software is a set of coherent applications (Fig. 5) representing an integrated working environment to operators of the FAIR facility. Those applications support the operators in setting, optimizing, and monitoring the accelerators for parallel beam operation. Applications are

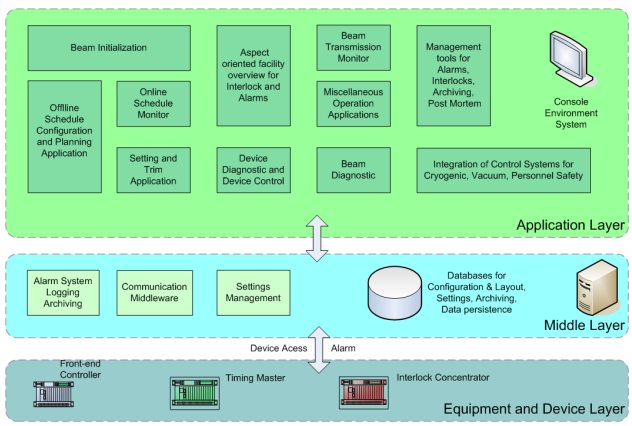

Figure 5: Operation software in the context of the control system architecture.

structured and organized generically to adapt to beam production chains, accelerator areas, and operation modes. For device layer communication the applications utilize JAPC [7] and the LSA-API. For integration with the existing GSI operation software see [8].

Interlock System

The FAIR Interlock System shall protect the accelerator from damage and minimizes radioactive contamination due to unforeseen continuous beam deposition. In case of equipment malfunction or beam-loss it is specified to either disable further beam production or to switch to a low-intensity beam on a timescale of approx. 100ms which is half of the shortest cycle of SIS18 synchrotron. Therefore, it tightly interacts with the timing master of the timing system which executes predefined schedules. Even if the interlock system may trigger a beam dump in an upstream accelerator when a malfunction downwards in the accelerator chain is detected, it is not specified as a fast beam dump system on a turn by turn base, which is instead realized by dedicated faster systems, e.g. the autonomous beam dump system of the SIS100. The interlock signal pick up and signal concentrating is realized with industrial control system technology (PLC). Processing and interaction with the timing system, as well as modeling the complete interlock system as a FESA device, is implemented on a dedicated controller running Linux.

Beam Instrumentation

Beam instrumentation of FAIR is a work-package of its own [9] but tightly integrated into the control system stack. In particular, the synchronization and triggering of beam instrumentation equipment within the beam schedule is done via utilizing the control system's Timing System [2] and integrating the corresponding Timing Receivers in the beam instrumentation equipment. Furthermore, frameworks and technology of the control system is used [10] to build the software interface (FESA [5]) of beam instrumentation equipment and to access it by client-software (JAPC [7]).

Industrial Control System (UNICOS)

UNICOS (UNified Industrial Control System) [11] is a CERN developed framework to automate the generation and integration of industrial control system components of the supervision and control layer (e.g. WinCC Open Architecture and PLC S7). The FAIR project utilizes UNICOS to build its vacuum control system and the cryogenic control system as standalone systems independent from the accelerator control system.

Other Components

RF phase synchronization for bunch to bucket transfer (BUTIS) [12] between the synchrotrons SIS18 and SIS100 is integrated into the control system but is a work-package of its own and not subject of this paper.

FIRST IMPLEMENTATIONS

The commissioning of the FAIR accelerator facility is planned for 2018. Prior, implementations of the FAIR control system stack with reduced functionality will be tested already in 2014. Firstly, for the FAIR proton injector which is currently set up by CEA/Saclay in France, and secondly, for the Swedish FAIR in-kind contribution CRYRING, a legacy standalone low energy experimental storage ring which has been transferred from the Manne Siegbahn Laboratory (MSL) to GSI/FAIR.

ISBN 978-3-95450-139-7

Proton Injector at CEA/Saclay

As equipment controllers for the power supplies the FAIR standard solution utilizing the SCU [3] with a proprietary slave board will already be used. All the injector control equipment is modeled as FESA devices. Since the proton source is operated in pulsed mode a timing system network (White Rabbit) consisting of a timing master with a simple repetitive schedule and a timing receiver which produces TTL signal is used to trigger the magnetron and beam instrumentation equipment. A specific source control application has been developed to operate the injector. For commissioning no multiplexed operation is needed.

CRYRING at GSI

For the FAIR setup, CRYRING at GSI will have two operation modes of injection. The existing source injection line of the Swedish installation, or alternatively an injection from the existing GSI Experimental Storage Ring (ESR) can be used. For the start version of CRYRING in 2014 the beam will be injected from the ESR and the corresponding injection line and the ring itself will be controlled by working prototype components of the FAIR control system. The Swedish equipment (power supplies, electron-cooler, ring and linac RF) will be SCU controlled with proprietary interface cards. The synchronization of equipment will be completely realized by the FAIR Timing System, using a prototype Timing Master and SCU implemented Timing-Receivers. All equipment will be modeled as FESA devices. For the settings generation and management LSA and generic LSA-Applications are used. Further on, CRYRING will be an ideal test bed for hardware and software developments as well as for validation of operation concepts for multiplexed beam operation.

REFERENCES

[1] J. Serrano, P. Alvarez, M. Cattin, E. Garcia Cota, J. Lewis, P. Moreira, T. Wlostowski (CERN, Geneva, Switzerland) G. Gaderer, P. Loschmidt (Austrian Academy of Sciences, Wien, Austria) J. Dedic (Cosylab, Ljubljana, Slovenia) R. Bär, T. Fleck, M. Kreider, C. Prados, S. Rauch (GSI, Darmstadt, Germany) "THE WHITE RABBIT PROJECT" Proceedings of ICALEPCS 2009, Kobe, Japan, TUC004, p. 93

[2] R.C. Bär, T. Fleck, M. Kreider, S. Mauro (GSI Darmstadt, Germany) "THE TIMING MASTER FOR THE FAIR ACCELERATOR FACILITY" Proceedings of ICALEPCS 2011, Grenoble, France, WEPMS011, p. 996

[3] S. Rauch, W. Terpstra, W. Panschow, M. Thieme, C. Prados, M. Zweig, M. Kreider, D. Beck, R. Bär (GSI, Darmstadt, Germany) "FACILITY-WIDE SYNCHRONIZATION OF STANDARD FAIR EQUIPMENT CONTROLLERS" Proceedings of PCaPAC 2012, Kolkata, India, WEPD48, p. 84

[4] Stefan Rauch, Ralph C. Bär, Wolfgang Panschow, Matthias Thieme (GSI Darmstadt, Germany) "PERFORMANCE OF THE STANDARD FAIR EQUIPMENT CONTROLLER PROTOTYPE" Proceedings of ICALEPCS 2011, Grenoble, France, WEPMN018, p. 919

[5] Alexander Schwinn, Solveigh Matthies, Dorothea Pfeiffer (GSI Darmstadt, Germany) Michel Arruat, Leandro Fernandez, Frank Locci, David Gomez Saavedra (CERN, Geneva) "FESA3 THE NEW FRONT-END SOFTWARE FRAMEWORK AT CERN AND THE FAIR FACILITY" Proceedings of PCaPAC 2010, Saskatoon, Saskatchewan, Canada, WECOAA03, p. 22

[6] J. Fitzek, R. Mueller, D. Ondreka (GSI Darmstadt, Germany) "SETTINGS MANAGEMENT WITHIN THE FAIR CONTROL SYSTEM BASED ON THE CERN LSA FRAMEWORK", Proceedings of PCaPAC 2010, Saskatoon, Saskatchewan, Canada, WEPL008, p. 63

[7] V. Baggiolini et al., "JAPC - Java API for Parameter Control", Proceedings of ICALEPCS 2005, Geneva, Switzerland

[8] R. Huhmann, G.Fröhlich, S. Jülicher, V.RW. Schaa (GSI Darmstadt, Germany) "GSI OPERATION SOFTWARE: MIGRATION FROM OPENVMS TO LINUX", Proceedings of ICALEPCS 2011, Grenoble, France, MOPMS014, p. 351

[9] M. Schwickert, P. Forck, P. Kowina, T. Giacomini, H. Reeg, A. Schlörit, GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany "BEAM DIAGNOSTIC DEVELOPMENTS FOR FAIR" Proceedings of DIPAC 2009, Basel, Switzerland, WEOA04, p. 424

[10] T. Hoffmann, M. Schwickert, (GSI, Darmstadt, Germany), G. Jansa (Cosylab, Ljubljana, Slovenia) "FESA AT FAIR - THE FRONT-END SOFTWARE ARCHITECTURE" Proceedings of PAC09, Vancouver, BC, Canada, FR5REP009, p. 4794

[11] E. Blanco Vinuela, J.M. Beckers, B. Bradu, Ph. Durand, B. Fernandez Adiego, S. Izquierdo Rosas, A. Merezhin, J. Ortola Vidal, J. Rochez, D. Willeman CERN, Geneva, Switzerland "UNICOS EVOLUTION: CPC VERSION 6" Proceedings of ICALEPCS 2011, Grenoble, France, WEPMS011, p. 786

[12] M. Zipfel, P. Moritz, "RECENT PROGRESS ON THE TECHNICAL REALIZATION OF THE BUNCH PHASE TIMING SYSTEM BUTIS" Proceedings of IPAC 2011, San Sebastián, Spain

ISBN 978-3-95450-139-7