

The control system for the new FAIR accelerator complex

- Technical Design -

Ralph C. Bär
03. March 2009
MAC meeting at GSI

Design considerations, General Strategy

- ◆ The present GSI control system is not adequate for the FAIR chain of accelerators. The expected full functionality of FAIR demands a substantial revision of the present system (HW base performance, timing, BD DAQ limits, ...)
- ◆ New functions shall not be retrofitted to the present system
 - ❖ a new control system has to be elaborated and tested on several milestones
 - ❖ only one system: new system will control both GSI and FAIR machines
 - ❖ build substantially on proven principles and solutions of the existing system
- ◆ The present control system (injectors) cannot be completely replaced (big investment)
 - ❖ present CS will be modernized, obsolete technology replaced
 - ❖ integration of present control system (FE, middleware, timing) into the new controls architecture
- ◆ The new CS shall be tested on the existing accelerators

Control System Design Base

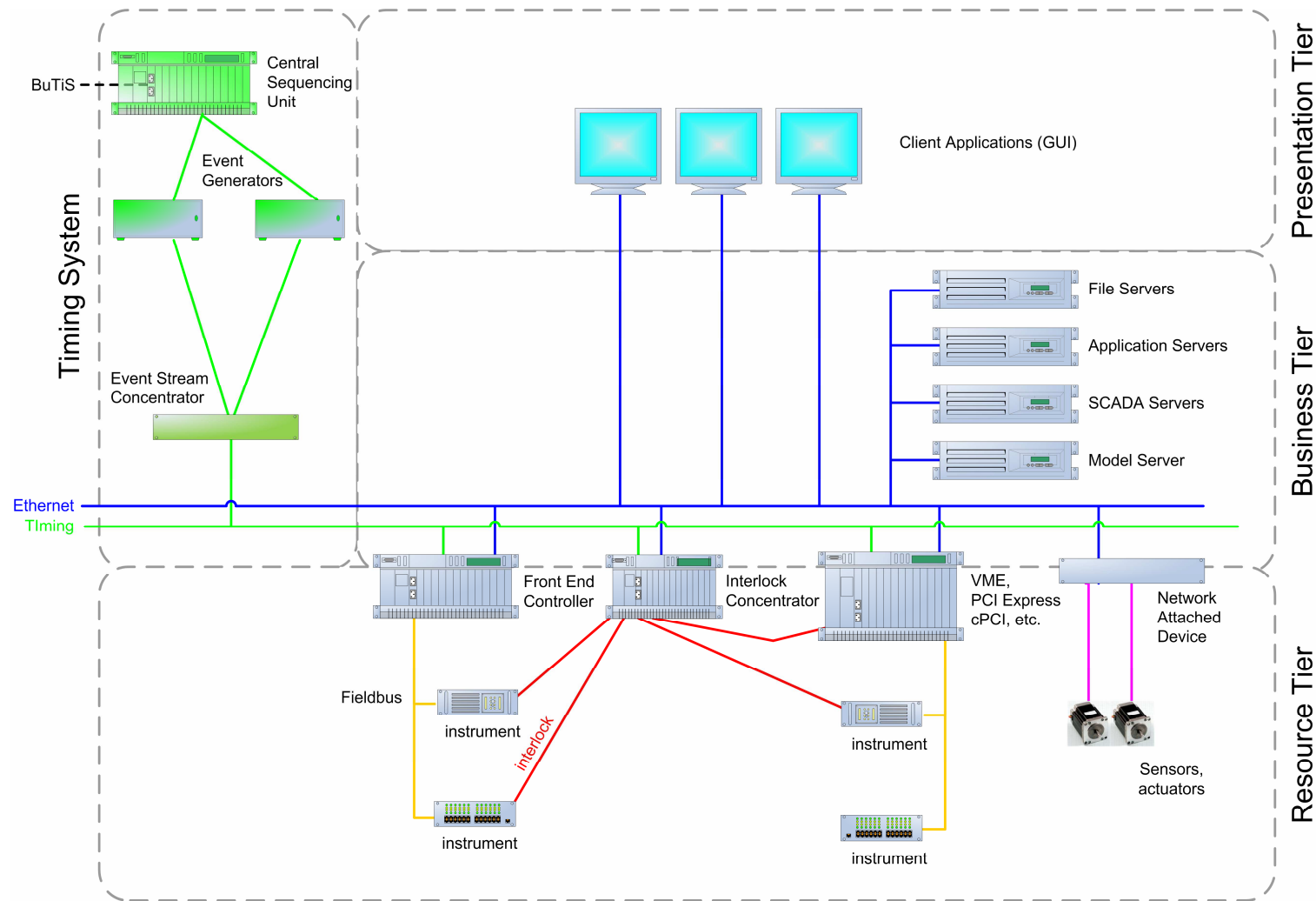
Control system design considerations

- ◆ Controls requirement analysis (→done, general requirements clear)
- ◆ Design will follow strict policy of standardization
 - ❖ technologies, interfaces, HW base, software frameworks, coding schemes, etc
 - ❖ strong management support from FAIR project management needed
- ◆ Strictly modular design with well defined interfaces
- ◆ Make use and adapt available ACS solutions/frameworks whenever available
- ◆ Use todays standards and technologies
 - ❖ Java, C++, Databases, Network, CORBA, Linux, Python, etc.
- ◆ CS is designed as state-of-the-art OO decentralized distributed system and follows the “standard 3-tier architecture”

❖ Resource tier

❖ Business tier

General System Architecture



Equipment Interfaces and Control

◆ Equipment Interfaces

- ❖ limited set of equipment interface standards will be supported
- ❖ standard digital backplane (proprietary communication bus, but FAIR-standard interface to at least all power supplies (→ACU), RF and kicker system: →compatibility, upgrade path GSI→FAIR)
- ❖ network attached devices
- ❖ RS 232/485, fieldbus
- ❖ PLCs and digital I/O
- ❖ timing receiver boards

◆ FE equipment controllers

- ❖ FAIR standard FE device controller: custom & cost effective solution (HD experience)
- ❖ single board computers (VME, PCI, PCIE, μ TCA) as needed for equipment control
- ❖ Note: Controls package comprises all FE controllers except BD and Cryogenics

◆ Software framework for FE system programming

- ❖ decided to use FESA from CERN
- ❖ provides standard and coherent software solution, maintained by CO dep.
- ❖ allows equipment specialists (e.g. BD) to develop FE driver classes (distributed development)

Standard FAIR Front-End Controller

FAIR front end controller summary

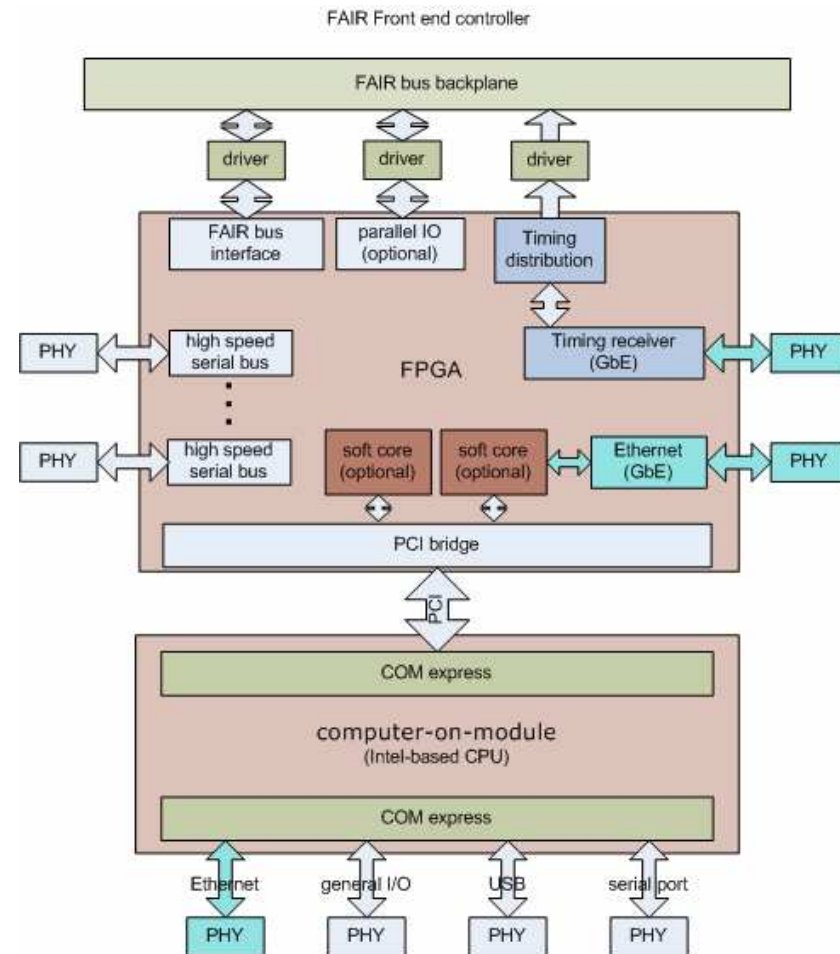
- ◆ Single-board computer (SBC) with Linux as Operating system
- ◆ Platform for front-end software (FESA)
- ◆ Mechanical dimensions like 3U Eurocard 100 mm x 160 mm (1U=33,35 mm)
- ◆ Mounted in slotted subracks (19"-System)
- ◆ Stacked construction; motherboard (carrier) with mezzanine card

Main components:

- Intel-based computer-on-module (COM) with COM Express Interface
- Carrier board with a powerful FPGA, interface circuits and several peripheral connectors
- ◆ FPGA and COM-CPU communicate via PCI
- ◆ Communication with the controlled devices over a dedicated parallel bus interface (FAIR-bus)
- ◆ Optionally communication over high speed serial interfaces
- ◆ Controlled devices are power supplies, RF components, diagnostic components and other

Foreseen interfaces:

- Minimum two Gigabit Ethernet (GbE) interfaces for **timing** and control
- Up to four high speed serial links
- For diagnostics and maintenance USB and EIA-232
- Up to 64 bit wide bidirectional interface



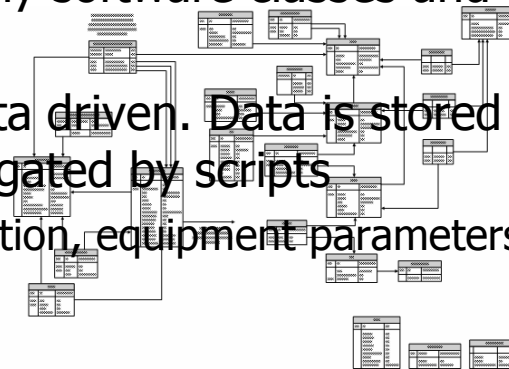
FE software architecture: FESA

- ◆ FESA – core component of the CERN control system at front-end level
 - ❖ technology: OO C++ on Linux OS
 - ❖ integrates real-time and non-RT actions on one CPU
 - ❖ easy design and deployment of FESA device classes with FESA shell and code generation tools
 - ❖ supports multiplexed operation of accelerators
 - ❖ well developed framework, broad equipment support
 - ❖ supports Intel and PowerPC architecture → easy integration on upcoming FAIR controller
 - ❖ used for new BPM system as part of SIS upgrade program
 - ❖ in progress: connection GSI→CERN timing to enable machine timing driven operation of FESA devices
 - ❖ in design phase: RDA plug-in to present FESA devices to the current GSI control system → smooth transition from present control system to FAIR control system

- ◆ Collaboration
 - ❖ 2 GSI people at CERN (as part of core team for FESA 3.0)
 - ❖ FESA team at GSI: 4 people (2 controls, 2 beam instrumentation)
 - ❖ Cosylab as external contractor (BPM system, RDA plug-in)

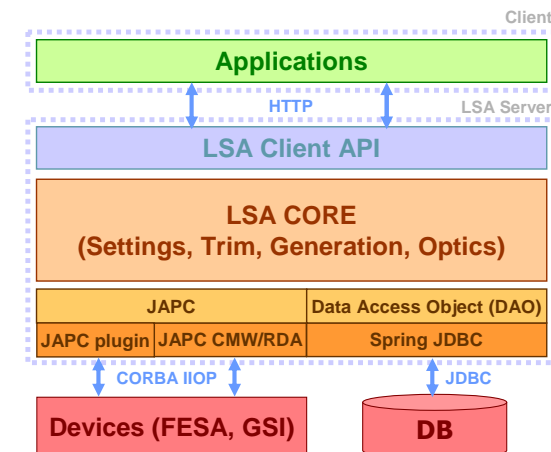
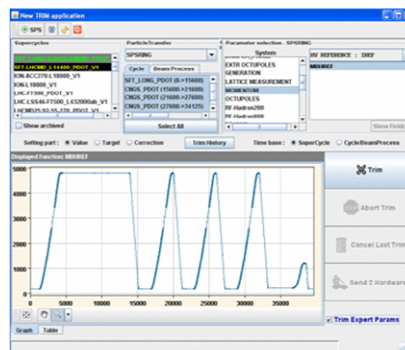
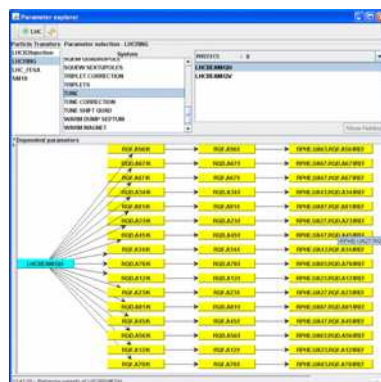
Applications and Business Tier Software

- ◆ Modular and distributed software architecture with clear separation between user interface (GUI), control core/service and device access
 - ❖ application programs provide I/F to the operator or end-user (operation environment) and must handle a great variety of controls tasks (GUI, data processing, etc.)
 - ❖ application programs rely on core system services: centralized processing power, system coordination, DB services, etc.
- ◆ All applications and core services are foreseen to be implemented in Java Technology with Swing classes for GUI.
- ◆ For Application environment and machine setting generation and management CERN solutions are foreseen to be used (LSA). Many software classes and GUI widgets can be adopted from CERN.
- ◆ Databases: CS and its building blocks are highly data driven. Data is stored only once. If needed in several instances, data is propagated by scripts
 - ❖ e.g. magnet data, calibration curves, lattice information, equipment parameters
- ◆ Operation concept for FAIR: Not worked out yet



Settings Generation & Management: LSA

- ◆ LSA – core component of the CERN control system at operating level
 - ❖ highly data driven, DB is the master and contains all needed information on optics, devices, cycles etc. ONE DB for all accelerators.
 - ❖ parameters are organized in hierarchies (from physics to hardware)
 - ❖ consistent settings generation and management for all levels
- ◆ with LSA, machines are operated on physics level in a consistent way
 - ❖ devices are accessed using an abstraction layer that hides middleware



- ❖ Used technologies: Java, Swing GUIs, Spring Framework, Oracle DB, 3-tier architecture

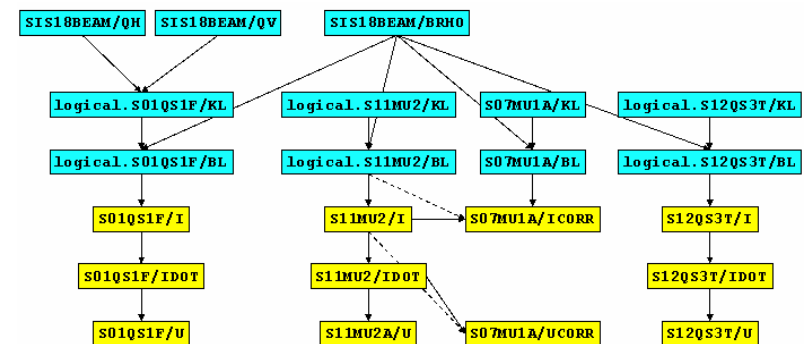
Settings Generation & Management: LSA

◆ Collaboration

- ❖ GSI sent 2 people to CERN to work inside the LSA team → 3 MY
- ❖ Ongoing collaboration on LSA development (GSI: 2 people, CERN: 4)
- ❖ *One* code base

◆ Achievements at GSI

- ❖ LSA prototype server running at GSI
- ❖ FAIR DV project started to model SIS18 in LSA
 - ◆ concentration on physics model allowed already the generation of ramps for all important SIS18 devices (after 2 months)
 - ◆ adaptations to LSA needed for GSI, current phase: identification of topics
 - ◆ integrated applications have to be written for GSI
- ❖ Device access CMW/FESA and GSI Corba MW working (get, set, monitor)



Machine Timing System

Challenges

- ◆ distributed facility (about 1 km cable length resulting in latencies of up to $\sim 10 \mu\text{s}$)
- ◆ absolute time synchronization of about $1 \mu\text{s}$ needed
- ◆ far more complex for FAIR operation, present timing system insufficient
- ◆ most machines need bunch-to-bucket transfer mechanism (message exchange)
- ◆ distributed RF, kicker systems, TOF applications need bunch-synchronous timing
- ◆ provide timing for whole campus (accelerators & experiments)
- ◆ timing system shall have hierarchical structure

Design

- ◆ General Machine Timing GMT (\rightarrow part of Controls working package)
- ◆ Bunch Timing System BuTiS (\rightarrow responsibility of RF group)
- ◆ Both systems tightly coupled

GMT: distribution technology

Technology

- ◆ Broadcast centrally generated timing telegrams in a star topology
- ◆ distribute all facility events to each single timing receiver
- ◆ key technologies: synchronous Gigabit Ethernet, PTP (IEEE 1588), UTC via GPS
- ◆ definition of new layer-2 protocol standard "**white rabbit**" to meet timing and synchronization requirements
- ◆ connections via standard fiber and copper solutions

Collaborations

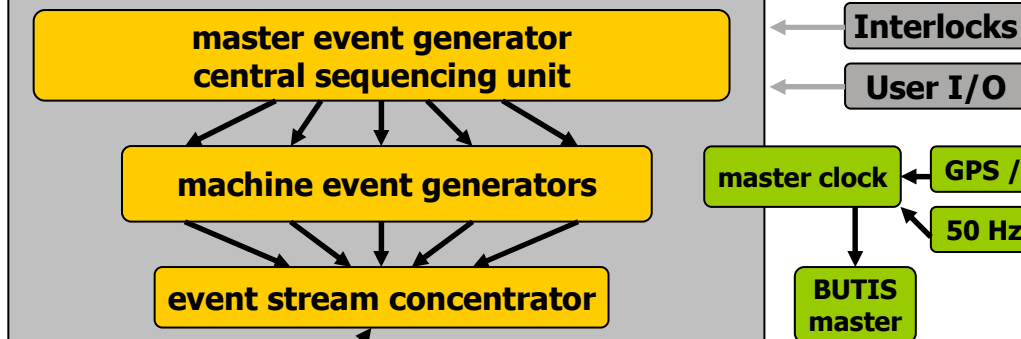
- ◆ Partners: CERN, GSI, AAS, IN2P3, ELETTRA, Cosylab
- ◆ Possible participation: NUSTAR experiments, ITER, MedAustron und whoever else is interested

Shared development and specification on the Timing System

- ◆ OHR (open HW repository)
- ◆ Use of the same hardware and participating institutes
 - ❖ Common timing receiver HW (VME, PMC, PCI, PCIe, μ TCA, simple standalone TRs)
 - ❖ Common WR switches (μ TCA form factor)
 - ❖ Common Timing Master
- ◆ Search for most common solution regarding event content and VAcc concept

GMT: event generation and distribution, timing network

FAIR timing master

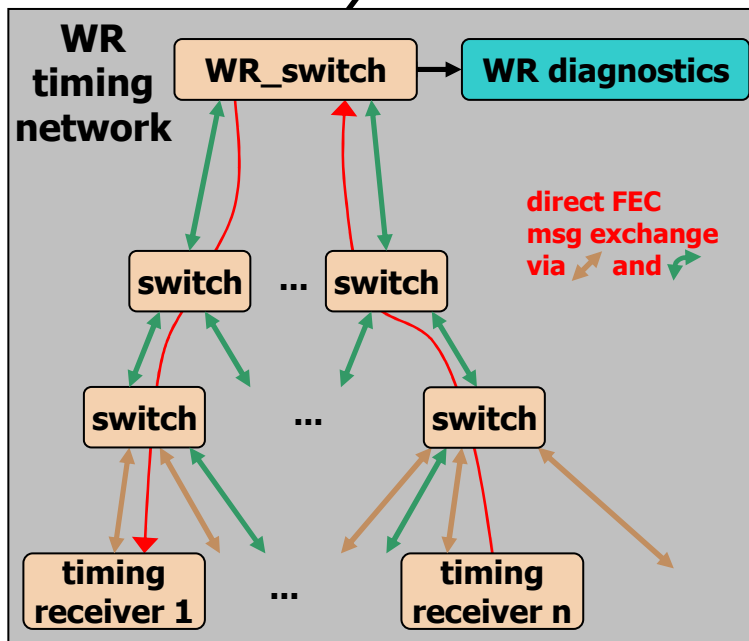


FAIR timing master

- coordination of all beams & cycles
- establishing of beam patterns
- alternative beam scenarios
- 1 EVG for each machine
- preloading of event tables & alternatives

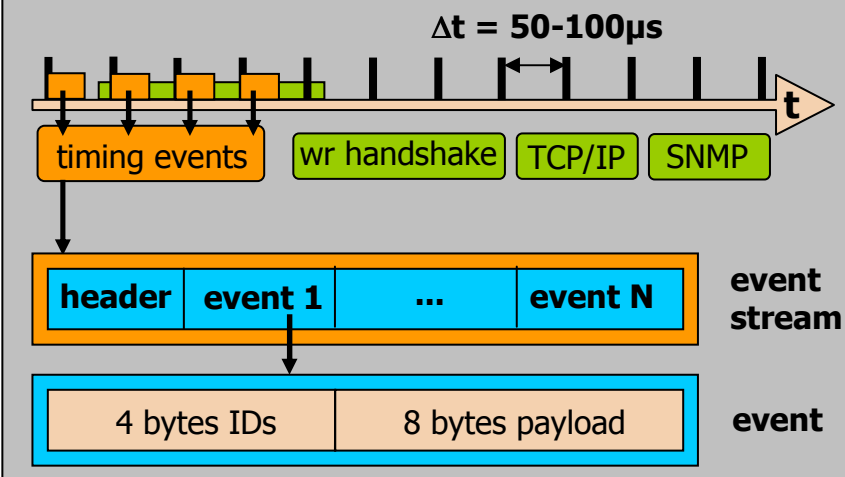
White Rabbit timing network

- true star topology
- all events broadcasted to complete facility
- event streams every 50(100) μ s
- UTC sync of all timing receivers (accuracy ~ 1 ns)
- BuTiS integration



fiber copper n<2000

timing events / distribution



Industrial Controls

- ◆ Some technical subsystems are not time-critical and highly industrial related
 - ❖ Cryogenic system, vacuum control, motors & drives, monitoring systems, HV generators, ...
- ◆ Control of such systems is foreseen to be realized based on a common industrial automation system (SCADA) with PLC based controls
- ◆ UNICOS framework from CERN - based on commercial SCADA product PVSSII - is foreseen (Siemens S7 PLCs already in use at GSI today)
- ◆ Slow control system could run autonomously if needed (cryogenics, vacuum)
- ◆ Full and seamless integration into the machine CS
- ◆ Working model (e.g. Central & Local Cryogenics):
 - ❖ Controls group provides central system installation, PLC technology, provides training and implementation support
 - ❖ Equipment specialists/groups: Responsible for sensors, signal processing and implementation of controls functions using the system framework
 - ❖ Experience at other labs: Industrial partners can be contracted to follow technical and functional specifications to deliver turn-key solutions
- ◆ Present situation: No manpower to start on Industrial Controls yet

Interlocks, Machine Protection

Machine protection system (MPS) is responsible for protecting parts of the machine, the machine itself, experimental and target equipment from damage by reacting on input signals from various systems.

- ◆ Requirements for machine protection are presently not precisely defined
 - ❖ Study group needs to work out scenarios and reaction consequences
- ◆ MPS design features:
 - ❖ must operate independently of the CS to ensure maximum level of protection
 - ❖ Provide VAcc sensitive reconfigurable interlock masks
 - ❖ Status/interlock signal concentration needed
 - ❖ Acts on emergency systems and the timing system (event sequencer)
- ◆ To provide debugging information
 - ❖ all interlocks must be precisely time-stamped (→ timing system)
 - ❖ all relevant systems must feature post-mortem dump capabilities

No detailed system design possible at this time

PLC based system envisaged, alternatively single board computers with DIO

Present CS integration activities

Present Developments (not complete)

- ◆ Modernization of the present CS: FE and middleware
 - ❖ Replacement of obsolete hardware, reimplementation of FE system (Linux OS, modular OO approach in C++, replacement of proprietary network protocols, use CORBA standard, etc.)
 - ❖ Required activities to integrate the present system into the new CS architecture
- ◆ Migration from OpenVMS OS to Linux
 - ❖ Application software are migrated to Linux platform, interoperability with new Java solutions are worked out and tested
- ◆ Prototype developments with FESA (BPM project with BD group, Cosylab)
 - ❖ Vertical slice (Application, DB, middleware, FESA FE system)
 - ❖ Implementation of FESA environment
- ◆ Project started to develop a gateway to access FESA equipment from the present CS (opposite direction already existing)
 - ❖ Will allow early FESA developments and testing in production environment
- ◆ Implementation of SIS18 setting generation with LSA

Summary Status

- ◆ Design of the new CS has started
 - ❖ Requirement analysis completed (first iteration, constant verification)
 - ❖ General system design, next: →engineering design
 - ❖ Prototyping: test at GSI machines, smooth transition
- ◆ Evaluation of adequate and available solutions done, collaborations started
 - ❖ FESA for the FE system framework
Status: decided, 2 FTE presently @CERN)
 - ❖ LSA for settings generation & management (physics model)
Status: proposal by Controls dep. and study group, to be decided, approval by MAC requested; 2 FTE in CO dep.
 - ❖ Timing system (co-development GSI/CERN/others)
Status: developments started, very promising (3.5 FTE active)
 - ❖ UNICOS for industrial controls
Status: not started, manpower presently not available
 - ❖ Standard FE controller (GSI development)
Status: in development
- ◆ Modernization of present GSI CS for integration into FAIR (Status: ongoing)

Controls: No technical risks, just a lot of interesting work