HARDWARE SIMULATION KIT FOR BEAM INSTRUMENTATION

Arkady Lokhovitskiy, Danil Kortchaguin, JINR, Dubna, Russian Federation
Michael Ludwig, CERN, Geneva, Switzerland

Introduction

For beam instrumentation front-end software consolidation in the CERN-PS AB-BDI-SW section has launched a campaign in collaboration with the Joint Institute for Nuclear Research (JINR) in Dubna (Russia). This consolidation is to a large extent re-engineering of legacy front-end software of the running CERN-PS machine. This raises the following issues: standardization, simulation of non active timing events, simulation of non available hardware, and backward compatibility. This paper describes a beam instrumentation hardware simulation, which is used to develop, test and validate instrumentation software, which are disconnected from the real hardware and machine timings.

CONSOLIDATION

The aim of this consolidation is to replace instrumentation servers in our front-end computers (FECs) which have reached the end of their lifecycle after more than a decade by completely re-engineered instrument servers for the LHC Injector chain. Directly affected are 15 FECs for beam current measurements hosting in total 279 devices and 8 FECs for beam position measurements with 357 devices, all of which are "24h/7d" mission critical. The new servers have to cover all functionality of the existing software, provide a high degree of backward compatibility in order to avoid software incompatibility in the client application layer, and provide subscription, structured properties, a data subset selection mechanism, object oriented design and code generation using graphical design tools. All these new features are needed to satisfy the increased demands of the LHC era. New servers are test-deployed and validated on a per-FEC basis under operational conditions with beam in dedicated software machine development sessions: this is normally the first time a new server runs with real hardware.

CONSTRAINTS

Standard Framework

The Front-End Software Architecture (FESA) framework [1] is the new framework which is used to overcome the current diversity in the LHC injector chain front end equipment software domain and pave the way towards LHC for efficient development, diagnostic and maintenance in this area. All data retrieving and processing is going through two types of actions: real-time (RT) actions and communication (COMM) actions. RT actions are scheduled according to timing maps and manage control and acquisition data flows between the hardware and a common shared memory region called FESA device. COMM actions are scheduled according to users' requests and transmit data to client applications.

Instrument Functional Model

An instrument server generally performs many acquisitions from different hardware modules at different moments during the production of a type of a beam[*], processes them according to the logic of the measurement and then publishes the results for each type of beam separately[†]. The result data must be available and consistent for at least the duration of one cycle (of a given type), so that the application client can pick it up before it is overwritten by the next cycle of the same type.

Backward Compatibility

FESA is not fully backward compatible with the present control system (referred to as **g**eneral **m**odules **GM**) and existing naming conventions [1]. The GM type software clients are interfaced to FESA using a system of special GM classes for each instrument which connect to a set of FESA properties for this instrument (FESA2GM adapter). Each GM class provides full inheritance to the GM super classes at the same time. Thereafter, the GM-specific communication channels: local GM access in the FEC, the common middleware servers (CMW) and the remote procedure calls (RPC) servers can be used transparently. Correspondence maps between old and new properties and devices, which provide also many-to-one relationships, are used to overcome naming incompatibilities and to regroup devices in the re-engineered instruments.

STANDARDIZATION

In the context of the consolidation project [3] we deal with a lot of hardware module types and different coding principals. At the same time there are no common patterns to standardize and simplify hardware module calls as the FESA does with an instrument design. This requires finding some solution to provide an abstraction as a standardized approach to HW interaction design in FESA, including instrument simulation. The abstraction has to solve the following issues:

- Support a vastly heterogeneous structure of equipment and coding technologies present in CERN.

[*] A beam is produced during a cycle. A sequence of several (usually different) cycles forms a supercycle.

[†] The results are multiplexed in pulse-to-pulse (=cycle) modulation (PPM) slots of the device memory.

- Reuse efficiently existing code/design and provide high expansibility to work with new hardware module types at a later time.
- Simulate a hardware module behaviour to test and validate instruments, which are disconnected from the real hardware. Exclude an instrument rebuilding at switching on/off the simulation mode.
- Ensure compatibility with FESA framework and device drivers installed on FECs of the LHC complex.

HARDWARE SIMULATION KIT

The hardware simulation kit represents a layer between the FESA framework and hardware application programming interface HW-API (see Figure 1). It's based on an object oriented approach to closely link the software to the hardware module [4][5] and allow its real control or its simulation. In addition it allows us to switch from the real world to simulation and vice-versa in a transparent manner, just by configuration.

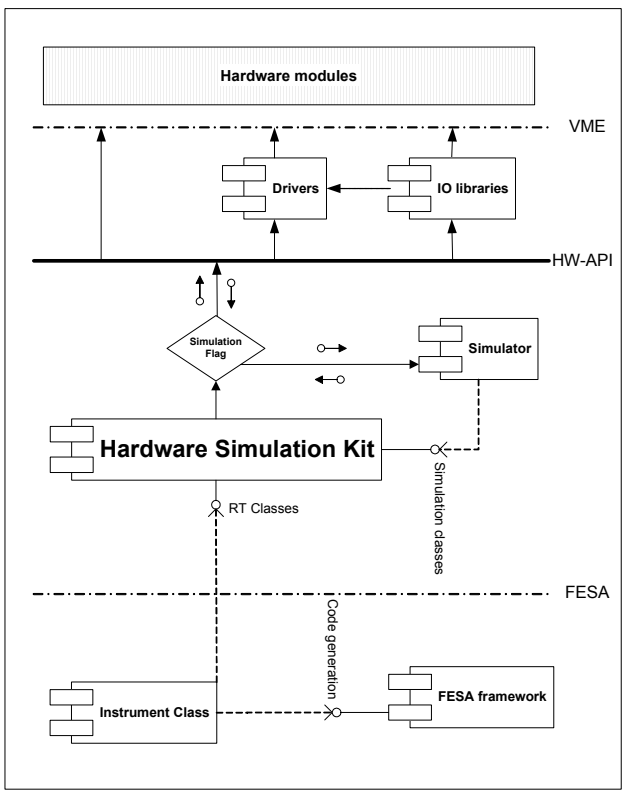

Figure 1: Component cooperation diagram. Here, a dashed line means a relation, a solid line with an arrow – an invocation, a solid line with a circle – a public interface, a circle with an arrow – data flow direction.

It provides:

- standardization of hardware interaction design within the FESA framework,
- the simulation functionality to efficiently develop and test instrumentation servers which are partially or entirely disconnected from real hardware,
- an extension interface to develop specific classes for new hardware module types and thus extend the simulation toolkit .

The hardware simulation kit implements pairs of related classes for each hardware module type with a set of default methods to perform read/write or in/out actions: one class is to be used in an RT process to both modes working with real hardware and simulation, the second one is to design an instrument specific simulator that has to be a separate process running on the same FEC where instrument binaries were deployed. All the methods calling a real hardware are virtual and can be redefined in derived classes in course of an instrument development

Beside read/write and in/out all other types of actions controlling hardware (initialize, restart, start and stop scanning, etc.) can have unpredictable function and coding principals. As described above the standardization nevertheless requires giving an interface to extend a base class functionality adding instrument specific actions. To solve this problem in the hardware simulation kit they have been excluded from a device class scope and implemented as an independent branch of the class hierarchy (see Figure 2 – Device command components).

The hardware simulation kit also contains a set of abstract template classes imposing a protocol to add new types of hardware modules. They are a public interface of Abstract Template Classes component (see Figure 2). Have implemented all the abstract methods in a derived class for a new hardware module type will get available for end-users.

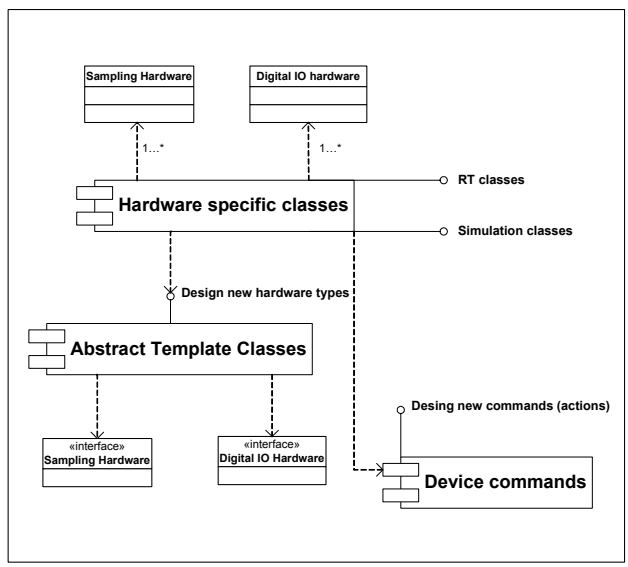

Figure 2: The hardware simulation kit component diagram (a dashed line means a relation (1…* - one-to-many), a solid line with a circle – a public interface).

SUMMARY

Simulation of instrumentation hardware (including timing [6]) is essential in order to efficiently develop and test instrumentation servers within the consolidation scenario for the LHC injector chain. The constraints of standardization, backward compatibility and instrument functionality are addressed, so that the consolidated instrument can be deployed on a per-FEC basis with minimized impact on operations.

At present a few instrument servers based on the hardware simulation kit are in progress. They measure or simulate a beam current and orbits using the following types of standard hardware modules: MPV908, SIS3300, DPRAM, and ICV196. They allow us to access the efficiency of the proposed implementation.

REFERENCES

[1] "FESA Essentials", http://project-fesa.web.cern.ch/
[2] "Guidelines and Conventions for Defining Interfaces of Equipment Developed Using FESA", AB-Note, CERN, 2005.
[3] "Consolidation Programme for CERN Accelerators", P. Bonnal, R. Forrest, J. Poole, AB-Note-2005-018, CERN, 2005.
[4] "MPV908 Operating Manual – Analog I/O Series ", Pentland System LTD, 1996.
[5] "General Sampler Class Specifications", C.H. Sicard, F. di Maio, PS-CO-Note-94-62, CERN, 1994.
[6] "The 8ESA Simulation Timing Event Source", M. Ludwig, CERN, 2005.