Software and Gatewa are Development for Sirius BPM Eletronics Using a Service-Orient ed Architecture

L. M. Russo, Beam Diagnostics Group
Brazilian Synchrotron Light Laboratory, LNLS, Brazil

Introduction
The Brazilian Synchrotron Light Laboratory (LNLS) is in the final stages of developing an open-source BPM system for Sirius, a 4th-generation synchrotron light source under construction in Brazil. The system is based on the MicroTCA.4 standard comprising AMC FPGA boards carrying FMC digitizers and a CPU module. The software is built with the HALCS framework and employs a service-oriented architecture (SOA) to expose a flexible interface between the gateware modules and its clients, providing a set of loosely-coupled components favoring reusability, extensibility and maintainability. In the paper, the BPM system will be discussed in detail focusing on how specific functionalities of the system are integrated and developed in the framework to provide SOA services. In particular, two domains will be covered: (i) gateware modules, such as the ADC interface, acquisition engine and digital signal processing; (ii) software services counterparts, showing how these modules can interact with each other in a uniform way, easing integration with control systems.

Generic Hardware Architecture
- Standard communication interfaces: PCIe, UART, etc.
- Hierarchical design, e.g. based on open-source Wishbone Bus Protocol
- Peripherals with minimal interaction, acting as isolated components
- Desirable to have knowledge about internal components such as: unique ID, name, address range, version, capabilities

SOA-based Software Architecture
- Software abstracts hardware components as services
- Uses a common protocol to communicate with hardware device
- Services can coordinate themselves by using an intra-Controller protocol
- Protocols acts as a flexible/extendible API

HALCS Architecture
- HALCS Framework implements SOA principles, using an Inversion of Control design paradigm
- Uses a common RPC protocol to expose services functionalities
- Broker provides discoverability and reliability to services using Mailbox messaging pattern
- Services (SMIO) register functions
- Services can use additional abstractions:
  - SMCH for external chips: AD9510 clock distributor and PLL, S67x clock oscillator, etc.
  - SMPR for external protocols: SPI, I2C, etc.
  - DEVO: Event-driven reactor engine
  - LLIO: Hardware Abstraction Layer

Links
HALCS Framework: https://github.com/lnls-dig/halcs
BPM EPICS IOC: https://github.com/lnls-dig/bpm-epics-ioc
BPM Gatewa: https://github.com/lnls-dig/bpm-gw
DSP Core: https://github.com/lnls-dig/dsp-cores
Infra Cores: https://github.com/lnls-dig/infra-cores
Timing Receiver: https://github.com/lnls-dig/timing-receiver-gw

Summary
- SOA principles applied to low-level software maximize reuse of commonly used functionalities
- Best used in conjunction with isolated hardware components with minimal interaction among each other
- Successfully deployed in the BPM and the upcoming MicroTCA.4 Timing Receiver projects