Urša Rojec

Integration of MTCA.4 Components in Medical and Scientific Applications

System Design Best Practices

Overview

- Motivation → Leveraging MTCA.4 properties to simplify system architecture

- Example 1 - Dose Delivery System for MedAustron Proton Therapy Machine
  - Handling different hardware interfaces
  - Leveraging MTCA.4 for redundant and critical signals (interlocks)
  - Leveraging MTCA.4 diagnostics

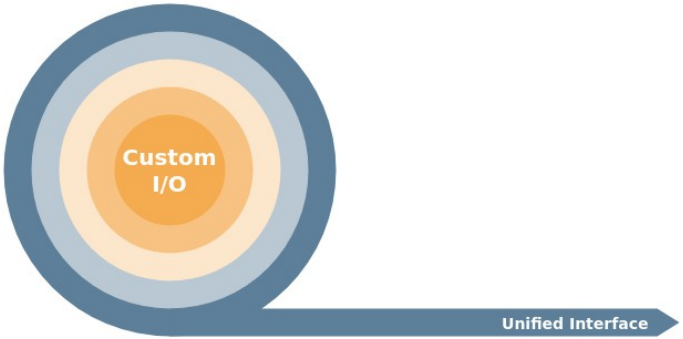
- Example 2 - Diagnostic Devices (BPM, BCM) and LLRF for ESS
  - Minimizing the code base
  - Building on common properties of such devices (high-speed DAQ)
  - Leveraging MTCA.4 RTM concept for hardware (and software) reuse

Two Different Architectural Challenges

COSYLAB

Dose Delivery System

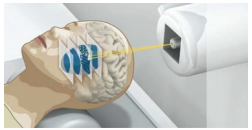
Systems that perform the same function but can have vendor-specific I/O interfaces
→ abstract away custom I/O



Diagnostics (BPM, BCM) and LLRF

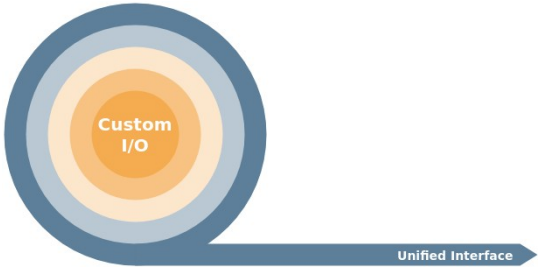
Systems that perform different functions but have a lot in common in terms of I/O - how they gather, process and output data
→ common base

Ex1: Dose Delivery System



COSYLAB

- Several components
  - Unified interface - Clear and fixed interface within DDS
  - Vendor specific I/O to the outside → FMC modules



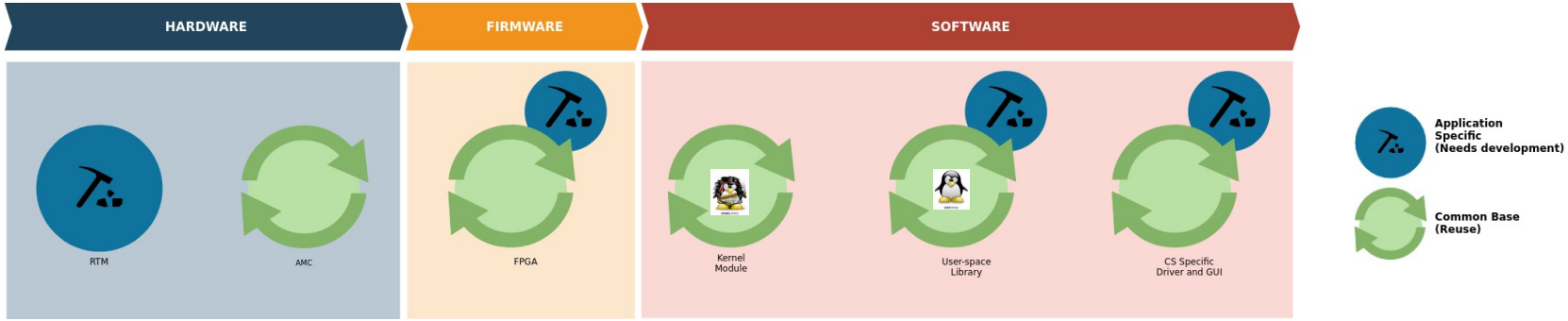
- Modularity – handles different nozzle topologies
- Redundancy of safety critical functionality
- MTCA.4 Properties used:
  - Hotplug for replacing failed boards with system on-line
  - PTP links for interlocks
  - MCH diagnostics for monitoring card status and slot occupancy → safety
  - Backplane connections – more rugged compared to cables outside

Ex2: Diagnostics (BPM, BCM) and LLRF

- Custom analog front-end on RTMs
- Reuse digitizer board

- Reuse software base, add only application specific functionality
- Software is not immune to entropy
  - Keep things in functionally complete and managable chunks

- Add custom common features to the firmware (e.g. timestamping support)



Generic DAQ Solution

COSYLAB

ToDo at the beginning ✓

- Identify devices with similar principle of operation (non-physics)
- Select board that matches requirements
- Develop the generic support

ToNotDo ✗

- Unify just for the sake of unifying - devices need to have something in common → remember entropy!

Benefit in immediate support when prototyping (data display and export, GUIs, scripting and Matlab interface) and well tested core SW base

Generic DAQ Solution

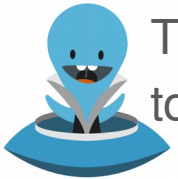
COSYLAB

ToDos ✓

- Be strict about interface specification
- Software and firmaware collaboration on interface spec 

ToNotDos 

- Exceptions to the point that they become common-practice
- Sacrifice final simplicity to save development time on one device
- Treat every system as special



This pays off in maintenance, debugging, amount of different spares that need to be kept on stock, training of new employees and less headaches in general

QUESTIONS?