ELETTRONICA
MIND IS THE FIRST DEFENCE

Using MDA to support software-firmware co-design and co-verification on Virtual Platforms

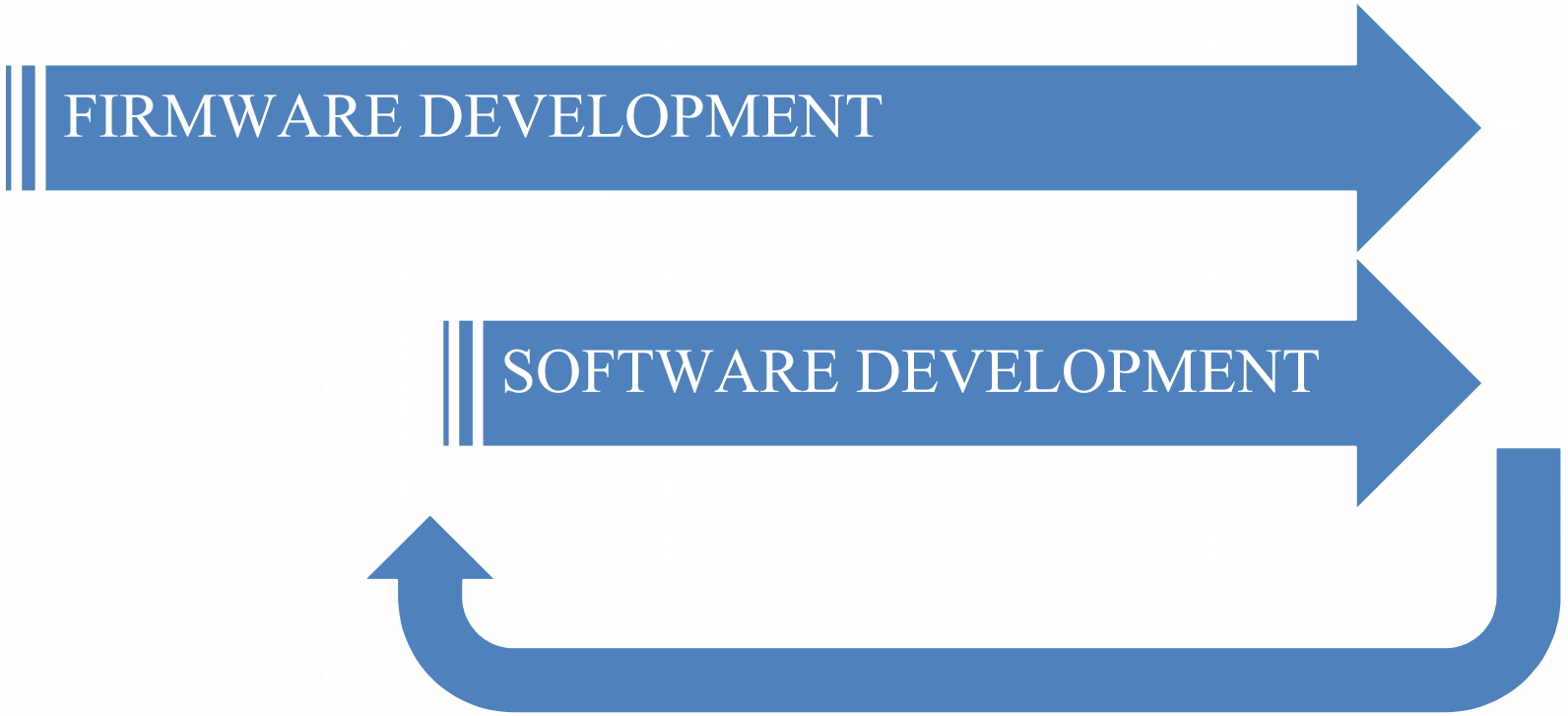
D.Perillo, F.Chirico

elt

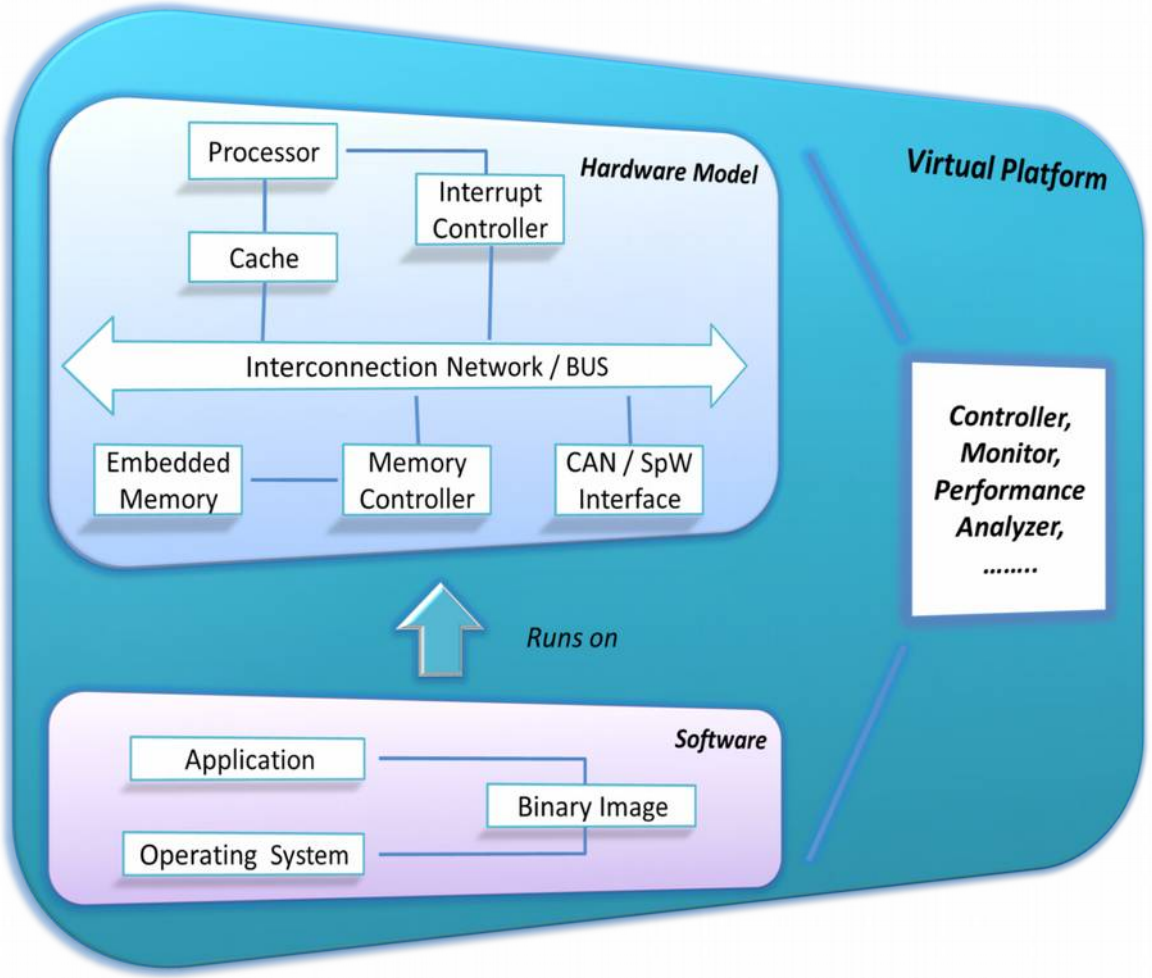
Context: The Problem

- In Embedded Systems Software and Firmware components (SWCI & FWCI) are tightly coupled.
- SWCI are developed when FWCI aren't available.
- IV&V usually takes place at Unit or System level: problems discovered at this level can have heavy impact on Project Schedule.
- Software is difficult to test against regressions: FW and HW must be available and tests cannot be performed in an automatic way (i.e. Continuous Integration environments)

Without Virtual Platforms

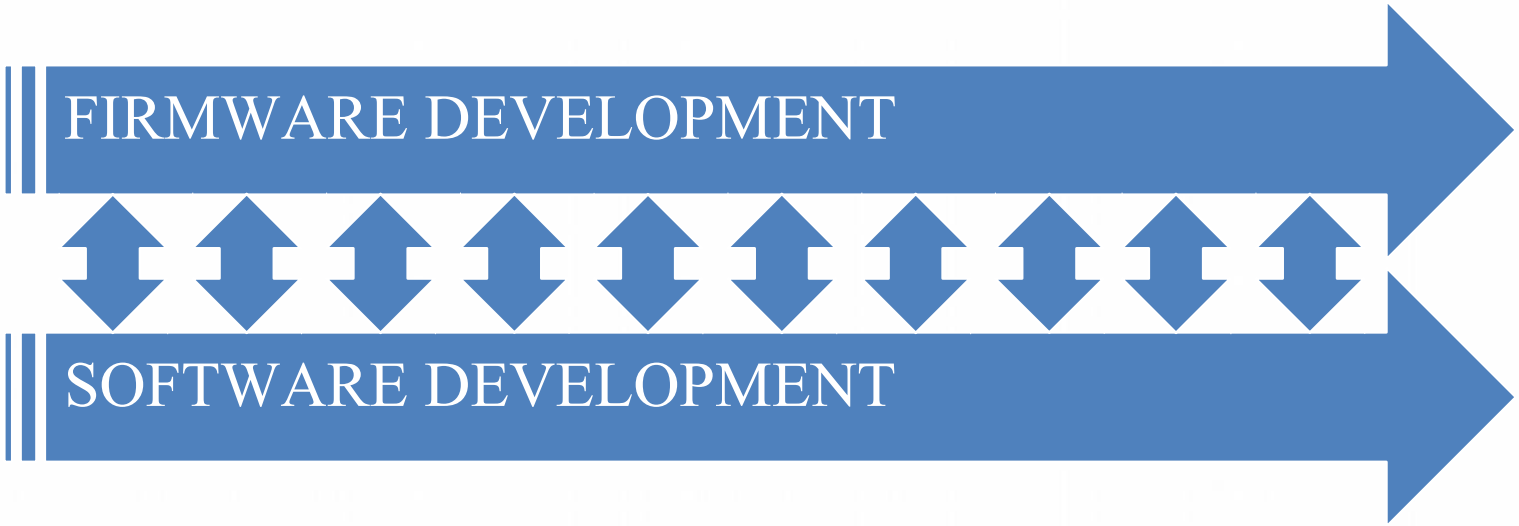


Virtual Platforms to the rescue



- A Virtual Platform is a software simulation of a computational platform
- A Virtual Platform is binary compatible with the real computational platform
- Virtual Platforms can be «enhanced» adding models that behave as missing firmware components
- Many VPs available commercially and not (SIMICS, OVP, QEMU)

With Virtual Platforms



Methodology Key Points

- Since early stages software development performed on a Virtual Platform
- Software & Firmware Engineers cooperate on Virtual Platform Firmware Models development
- Software V&V can take place since early stages of development: no last minute surprises and big design changes
- Continuous Integration environments can be used to discover regressions: fail early, fail often VS fail late, fail rarely
- Virtual Platform models development requires additional resources

Virtual Platform Models Costs Mitigation

- Use MDD approaches to reduce models costs
  - Model Skeleton Generation
  - Device Driver Generation
- Use HLS (High Level Synthesis) tools to partially generate Firmware from C, C++ or SystemC models
  - Vivado HLS
  - Mentor Graphics Catapult

MDD Technologies

- FW/SW interface represented using lightweight or heavyweight modeling
  - UML Profiles
  - Ecore Metamodels
  - XML Schema (XSD)
- OMG M2T for generation of virtual platform model and driver
  - Acceleo
  - Eclipse Modeling Framework
  - Eclipse Plugin Generation

Virtual Platform Comparison

- SIMICS (WindRiver)
  - Commercial (High Cost)
  - Documented
  - High Virtual Hardware Availability
  - DML, C, C++, SystemC
- OVP (Imperas Software)
  - Commercial (Low Cost, Free for Educational Purposes)
  - Documented
  - Medium Virtual Hardware Availability
  - C, C++, SystemC
- QEMU
  - Open Source
  - (un)Documented (need to read source and design documents)
  - High Virtual Hardware Availability
  - C

Virtual Platform (QEMU Example)

- **ADD CUSTOM FW TO QEMU VERSATILE EXPRESS PLATFORM**
- **DEVELOP LINUX KERNEL MODULE**
- **DEVELOP SOFTWARE THAT INTERACTS WITH VIRTUAL FIRMWARE**

Example (Boot and test run)