Experience in spacecraft on-board software development

Juan A. de la Puente, **Alejandro Alonso**, Juan Zamorano, Jorge Garrido, Emilio Salazar, Miguel A. de Miguel

aalonso@dit.upm.es

Universidad Politécnica de Madrid

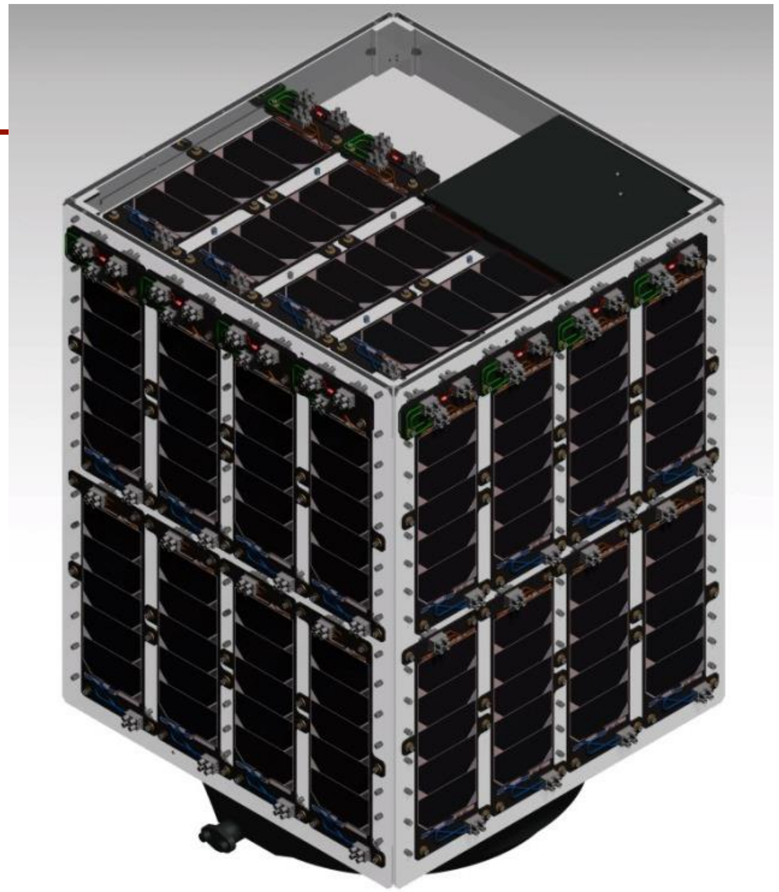
Ada-Europe 2014, Paris, France

Introduction

- **Aim**: Describe on-going work and experiences of STRAST group
- Long time experience in the group:
  - Currently oriented towards mixed-criticality partitioned systems, development tools, real-time kernels, and language features.
- UPMSat-2: **micro-satellite** used for experimenting with technologies and acquiring experience
- Two approaches:
  - **Monolithic**
  - **Partitioned**: FP7 MULTIPartes project (www.multipartes.eu)

© 2014 Alejandro Alonso

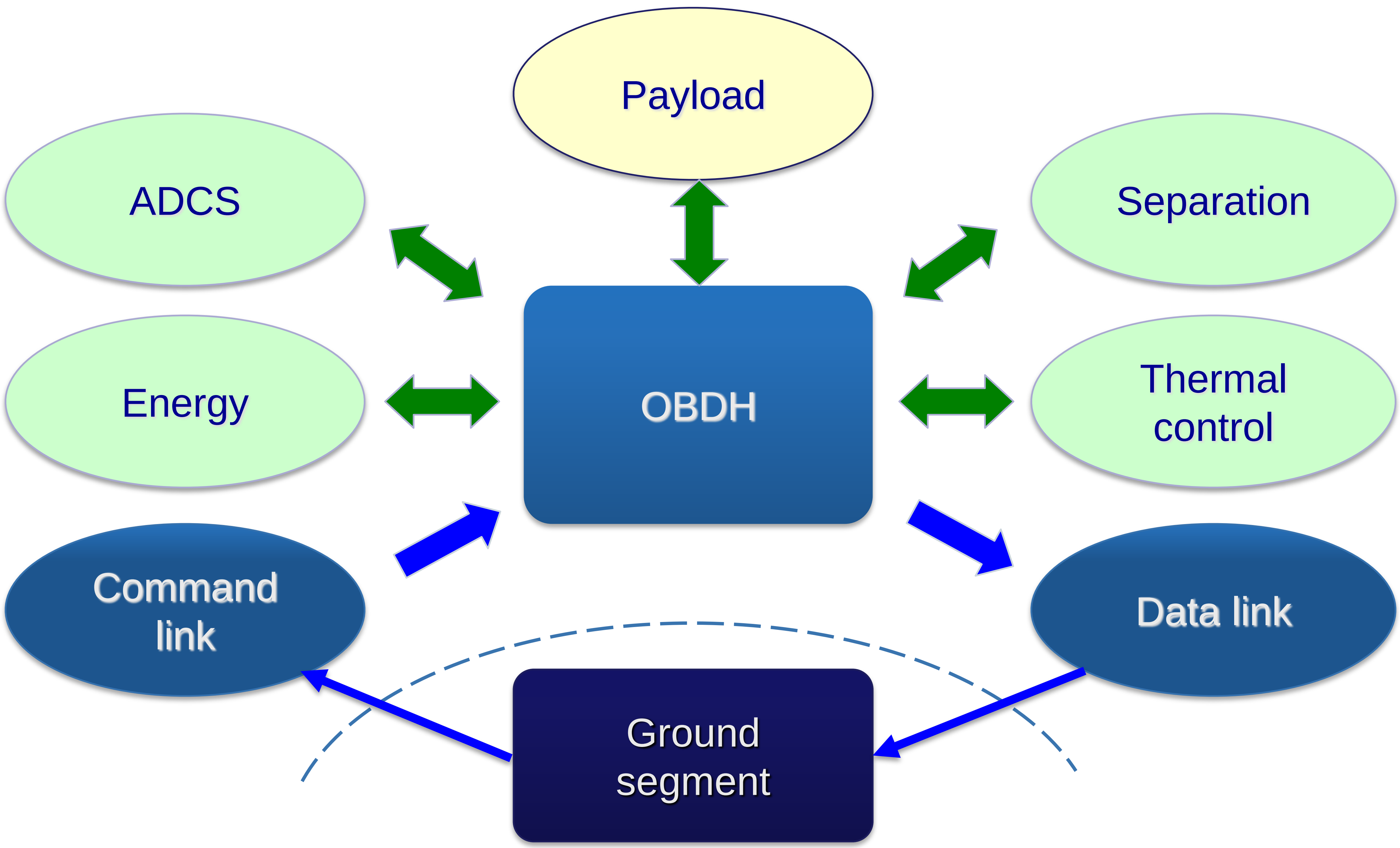
1. Introduction to UPMSat2



- Satellite developed at **UPM**
  - Collaboration with **industry**: Tecnobit
- Get **knowledge** and **experience** on space technology
- **Experiment** with **own** technologies:
  - Research, Teaching, Demonstration
- Collaborate with industries in the space domain
  - Payload experiments: Attitude control, solar cell, magnetometer, solar sensors, etc.
- Expected launch in **2015**

© 2014 Alejandro Alonso
dit UPM datsi

On-Board Data Handling (OBDH)



© 2014 Alejandro Alonso

Requirement Specification

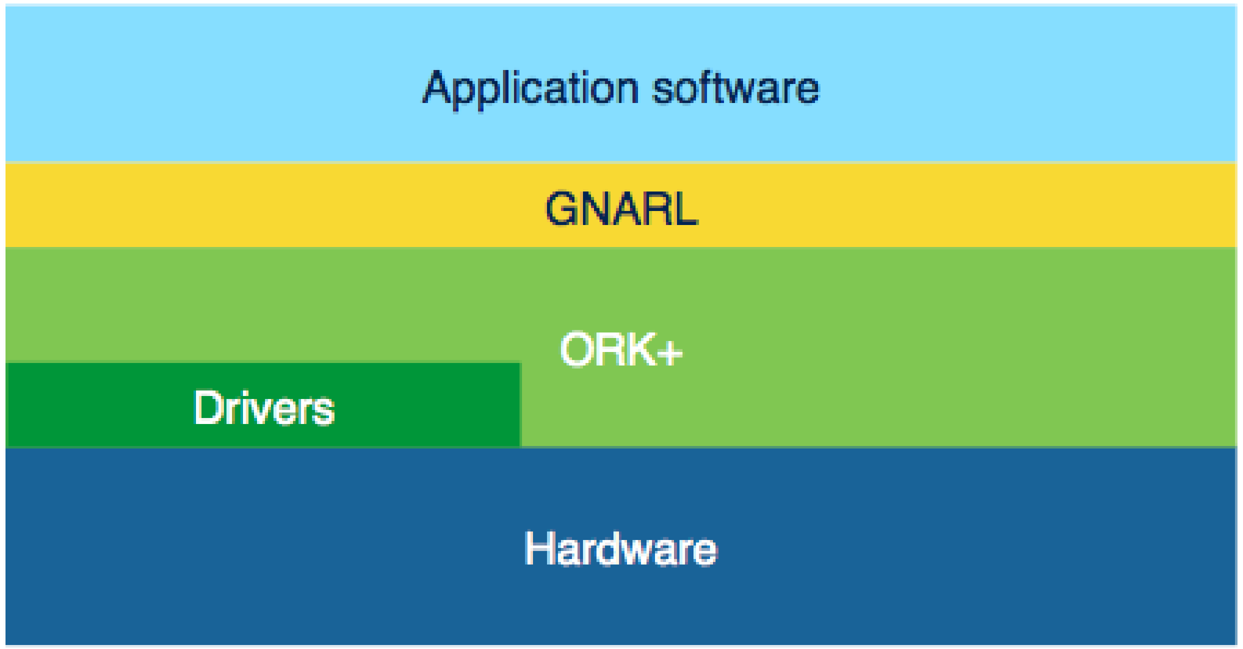
ID	Requirement
SYS-3	***The system will manage the operating mode of the satellite as defined in the state machine***
PMC-1	***The system shall acquire housekeeping data at regular intervals***
PMC-2	***Housekeeping data shall be validated with respect to a validity range***
PMC-3	***Housekeeping events: variable out of range, sensor error***
ACS-1	***Attitude control to be run periodically***
TTC-2	***TM messages to be sent when satellite is visible from ground station***
TTC-3	***TM messages: State, Housekeeping, Events/Errors, Experiments***
TTC-4	***TC should be decoded and executed, either immediately or when programmed***
TTC-5	***TC messages: Open link, change mode, change configuration parameter, resend message***
PFC-1	***RT behaviour to be defined for: Event and mode control, data acquisition, ADCS, TM&TM***

On-Board Computer

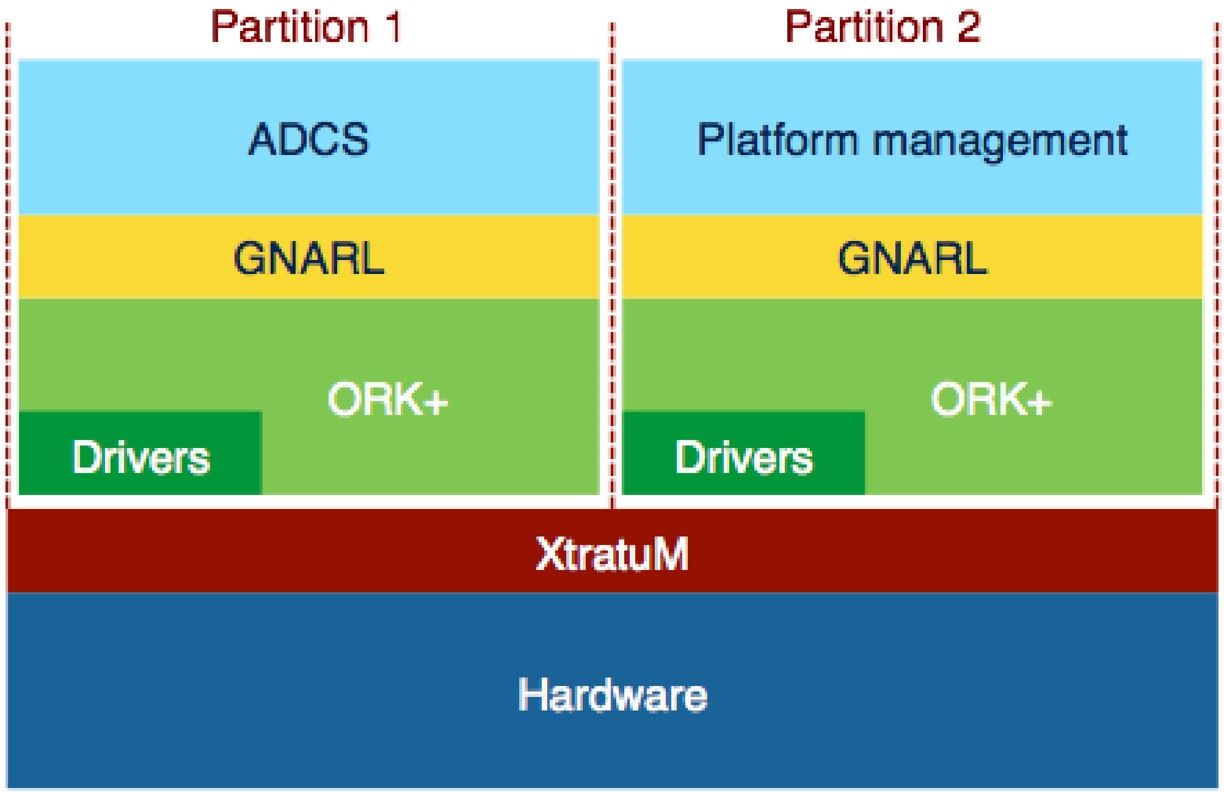
- LEON3 processor:
  - SPARC v8 RISC
  - Timers, bus and device controllers
  - Open VHDL model: Synthesized on FPGA
- 4 MB SRAM + 2 MB EEPROM
- 64 analog inputs, 104 digital I/O
- Serial interfaces: RS422, RS232, I2C, SPI
- Developed by TECNOBIT and STRAST/UPM

© 2014 Alejandro Alonso

Architecture Approaches



(a) Monolithic architecture.



(b) Partitioned architecture.

© 2014 Alejandro Alonso

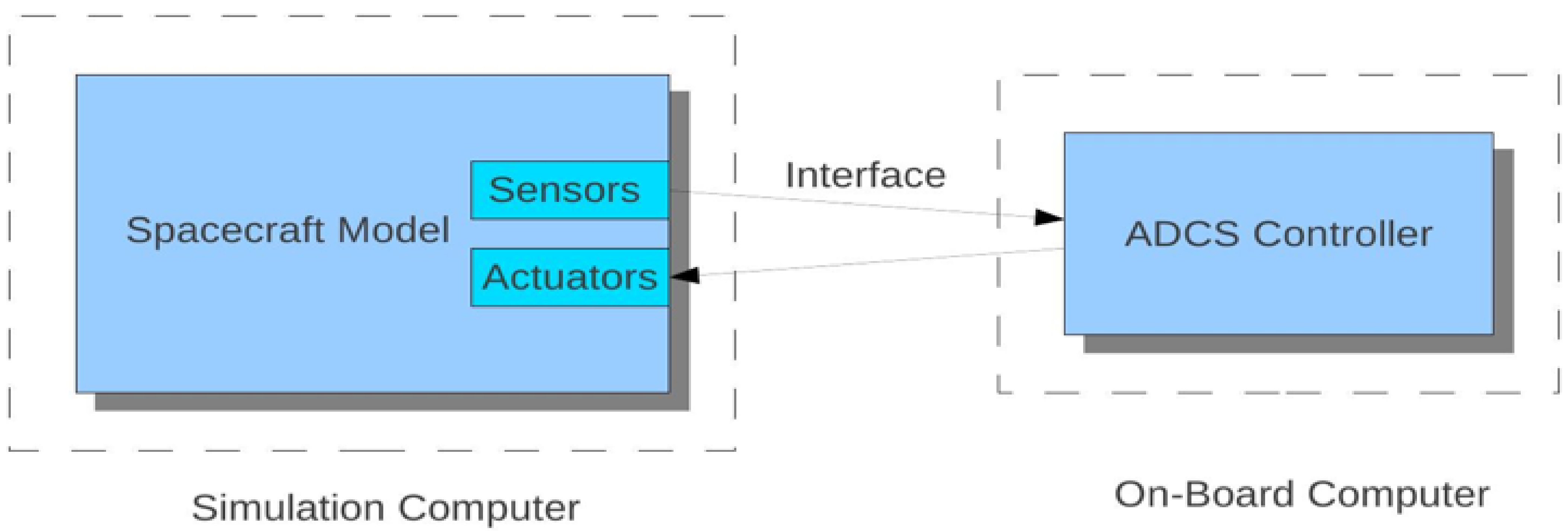
2. UPMSat-2 Development: monolithic

- ESA sw engineering standards for flight missions
- Tools and methods for the flying OBDH:
  - **TASTE** toolset:
    - Supported modeling languages: Simulink, SDL, and uses AADL
    - Generates Ravenscar Ada Code
  - **GNAT Pro** for LEON3 from AdaCore
    - Additional tools like GnatCheck and AUnit
    - Includes an evolution of the ORK kernel (UPM)
  - **RapiTime**: measuring WCET from Rapita Systems
  - Code generation tool for MATLAB / **Simulink** of MathWorks
  - Development of **Ravenscar drivers** for UART, I2C, SPI, FLASH memory, digital inputs/outputs, RTC and ADC.

© 2014 Alejandro Alonso

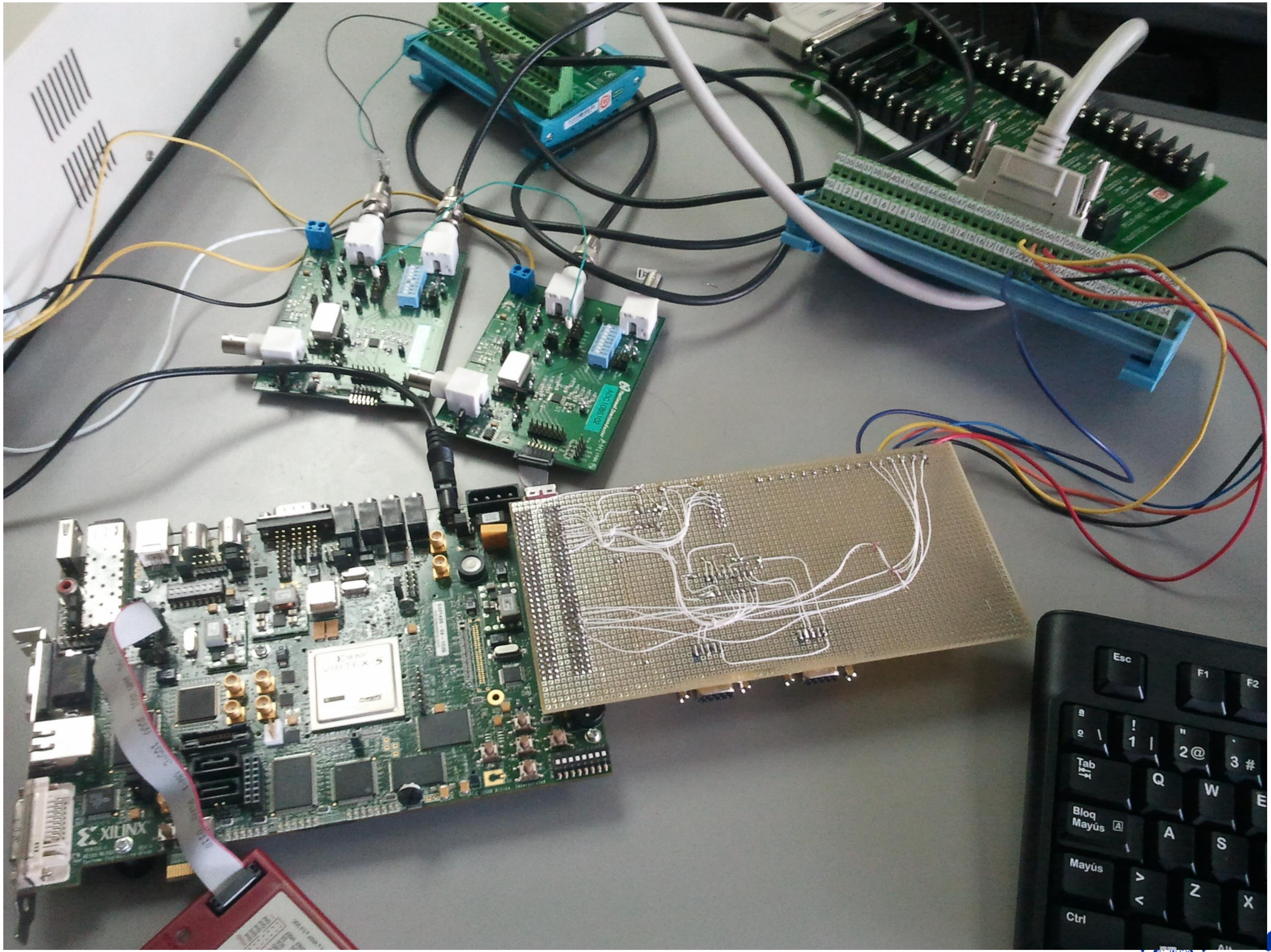
Software Validation Facility

- Platform for testing control attitude
  - Hardware in the loop
  - System interacts with a simulation of satellite behaviour
- Sotfware MATLAB and Simulink with Toolboxes for Control System y Data Acquisition among others.
- Boards for analog and digital inputs/outputs



© 2014 Alejandro Alonso
UPM

OBC Breadboard Model



3. Mixed-Criticality: Partitioned

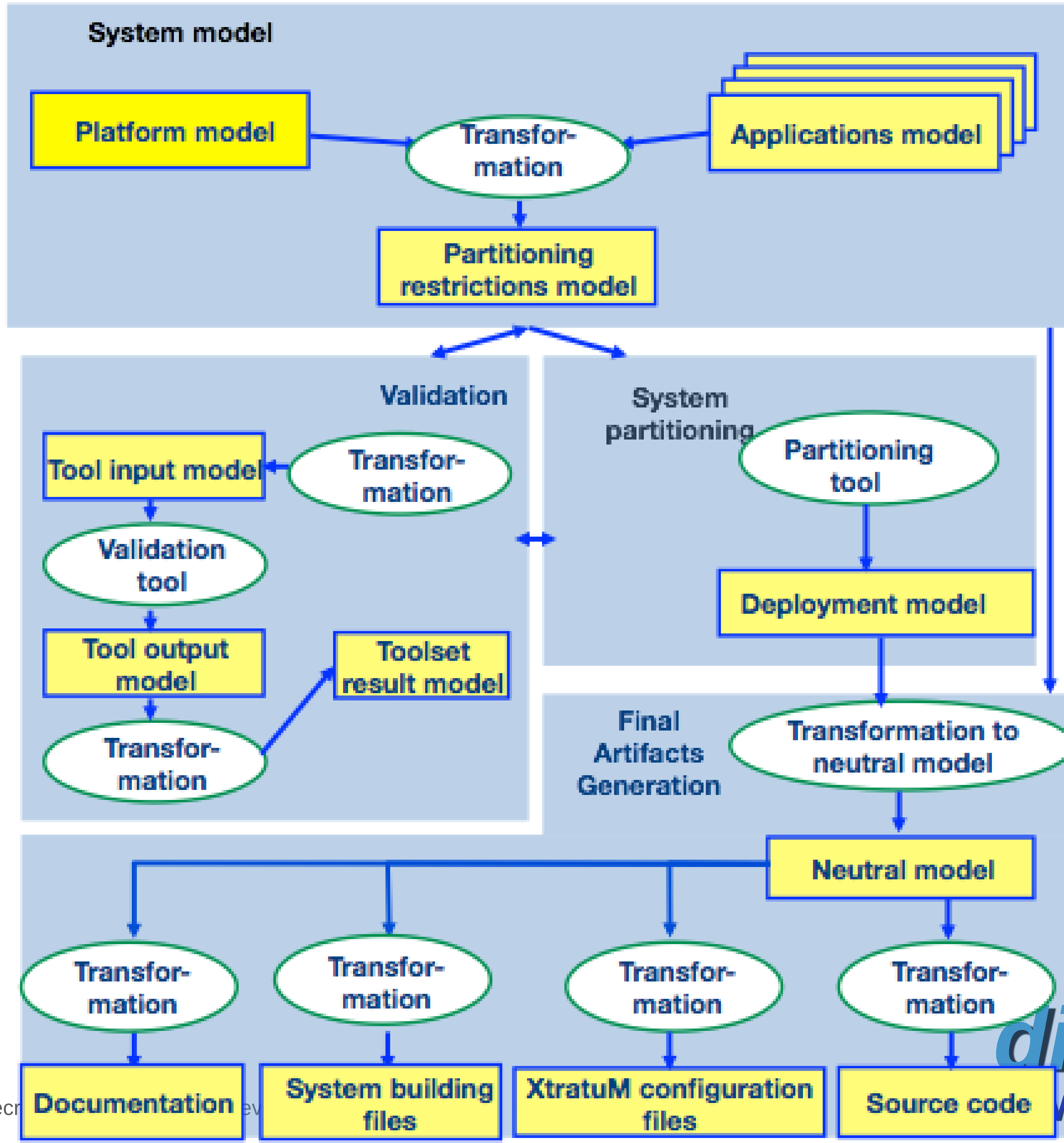
- Integration of applications of different criticality (safety, security, real-time and non-real time) in a single embedded system
- Key potential benefits:
  - **Complexity management** by means of system partition, segmentation and abstraction
  - **Reduce** number of subsystems
    - reduce overall cost, size, weight and power consumption
  - **Overcome** current scalability limitations
    - Availability of COTS multicore (e.g. P4080) and virtualization technology
- Key challenges:
  - Safety **certification** according to safety standards
  - Temporal **isolation**

3. MultiPARTES Framework:

- Development of mixed-criticality systems.
- Support for non-functional requirements (NFR)
  - Specification, validation, and transformations
  - Real-time, safety, security
- Support for partitioned systems
- Support for multi-core architectures
- System modelling
  - Support legacy applications
- Support for system deployment

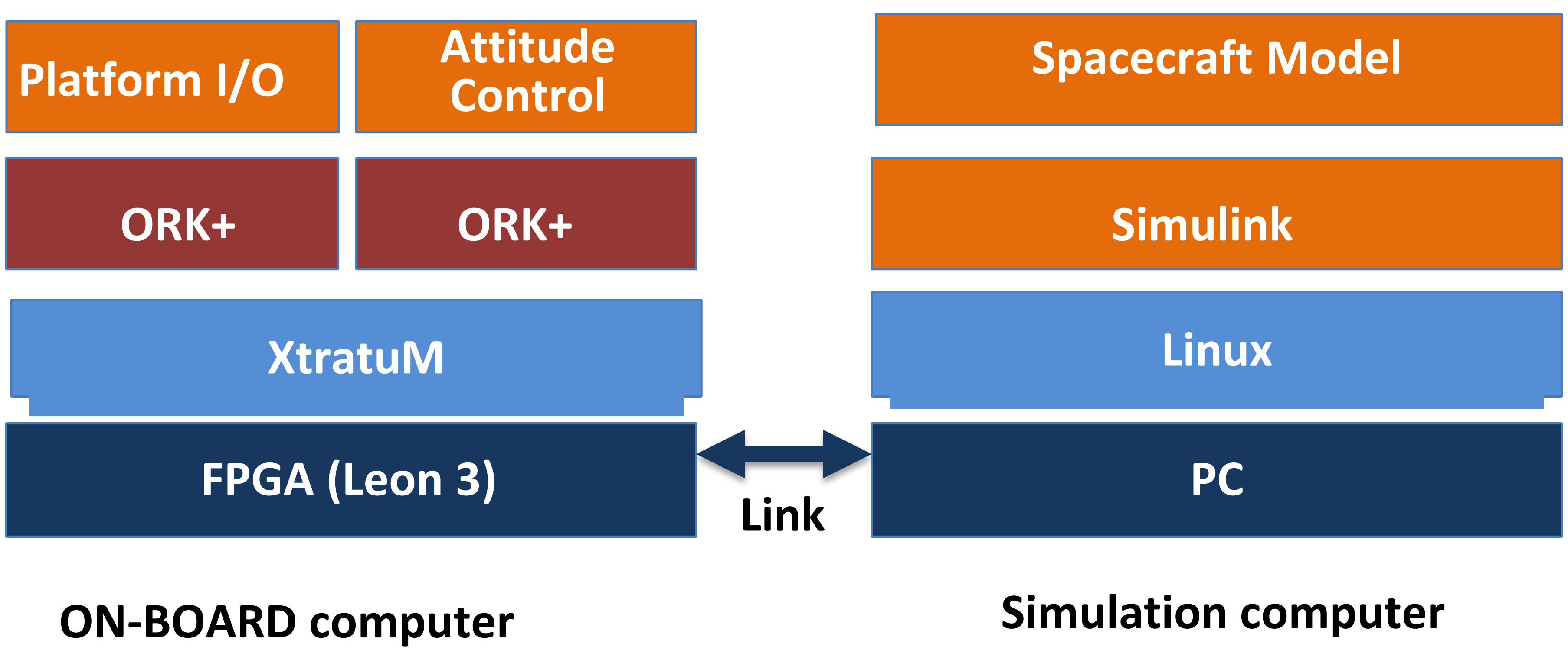
© 2014 Alejandro Alonso

Framework Architecture



Software Validation Facility

- Platform for testing control attitude
  - Hardware in the loop
  - System interacts with a simulation of satellite behaviour



Generation of Code Skeletons

- Oriented towards high integrity systems
- Compliant with the Ravenscar profile
- Compliant with: Guide for the use of the Ada programming language in high integrity systems
  - assessment of suitability of language features for analysis techniques
  - does not define a subset of the language
  - helps choice language features depending on the analysis & testing techniques to be used

© 2014 Alejandro Alonso

Periodic task body

```
package body <<PackageName>> is
task body Periodic_Task_Type is
   Canceled    :  Boolean;
   Id          :  aliased constant Task_Id   := Current_Task;
   WCET_Timer  :  Ada.Execution_Time.Timers.Timer (Id'Access);
begin
      Initialization;
      delay until Clock + Task_Offset;
      loop
            Ada.Execution_Time.Timers.Set_Handler (WCET_Timer, Task_WCET,
                  WCET_Ovr_Handler.Handler'Unrestricted_Access);
            Ada.Real_Time.Timing_Events.Set_Handler (Deadline_Overrun,
                  Clock + Task_Deadline, Deadline_Ovr_Handler.Handler'access);
            Activity;
            Ada.Real_Time.Timing_Events.Cancel_Handler (Deadline_Overrun, Canceled);
            delay until Clock + Task_Period;
     end loop;

end Periodic_Task_Type;

-- Bodies of procedures and protected objects in private part.
…
end <<PackageName>>;
```


© 2014 Alejandro Alonso

Conclusions

- MDE: allowed us to raise the abstraction level
  - Desirable more maturity in the used tools
- Use of TASTE tools: good experience
  - Allowed testing system design
  - Code generation a bit messy
- Mixed criticality systems based on partitioning
  - Great potential
  - Partitioned kernel must be qualified
  - Can support multi-core processors
  - Development of framework for supporting development
  - On-going work

© 2014 Alejandro Alonso