Final Presentation Days, December 2013

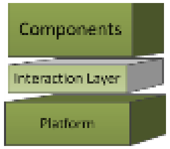
Thursday, December 12

…..

12:30 Lunch Break

13:30 OBSW reference architecture consolidation
Contractor: SSF/Terma ~ ESA TO: Andreas Jung

14:30 Methods and Tools for On-Board Software Engineering



On- board software reference architecture consolidation

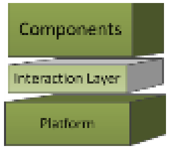
Victor Bos (SSF)
Daniel Stefl, Martin Cupak (ESC)
Gérald Garcia (Thales)

Poul Hougaard, Maria Ieronymak, Nicholas Mercredy (Terma)
Ana-Elena Rugina, Benjamin Jusot-Chauny, Marie de Roquemaurel, Marie-Aude Esteve, Marie-Helene Deredempt, (Astrium)
Petr Hnetynka , Pavel Jezek, Tomas Bures, Michal Malohlava, (Charles University)

Outline of Presentation

- **Findings summary**
- **What is the on-board software reference architecture?**
- **Background and objectives**
- **Process**
- **Findings**

Presenters:
Victor Bos
Poul Hougaard



Finding Summary

- Gap between functional chain analysis and OSRA (component model).

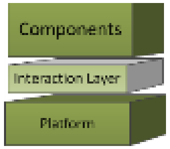
- The level of details required to apply OSRA to real missions revealed omissions.

- Separation of concern requires combinations of concerns. This is not covered in the baseline documents.

- Building blocks are nothing but components with a business case.

- Component model is not expressive enough.

☹



Finding Summary

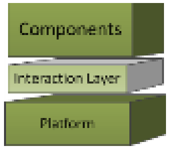
- Generally, the component model is sound and fit for the purpose of developing real-time embedded applications

- Current state of OSRA is very promising and provides a very good basis upon which our (and other people’s) findings can be studies and resolved

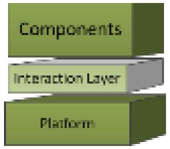
- Ideas reused in ASIM project

- OSRA offers potential for systematic reuse at different levels

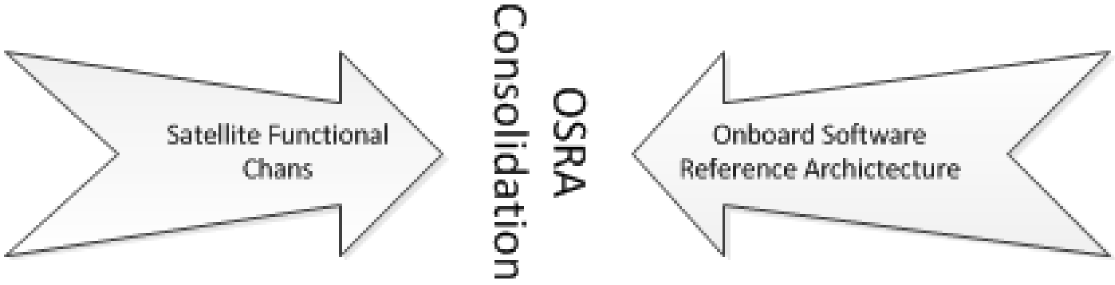
☺ ☺ ☺



Background and objectives



Objective



Satellite Functional Chans

OSRA Consolidation

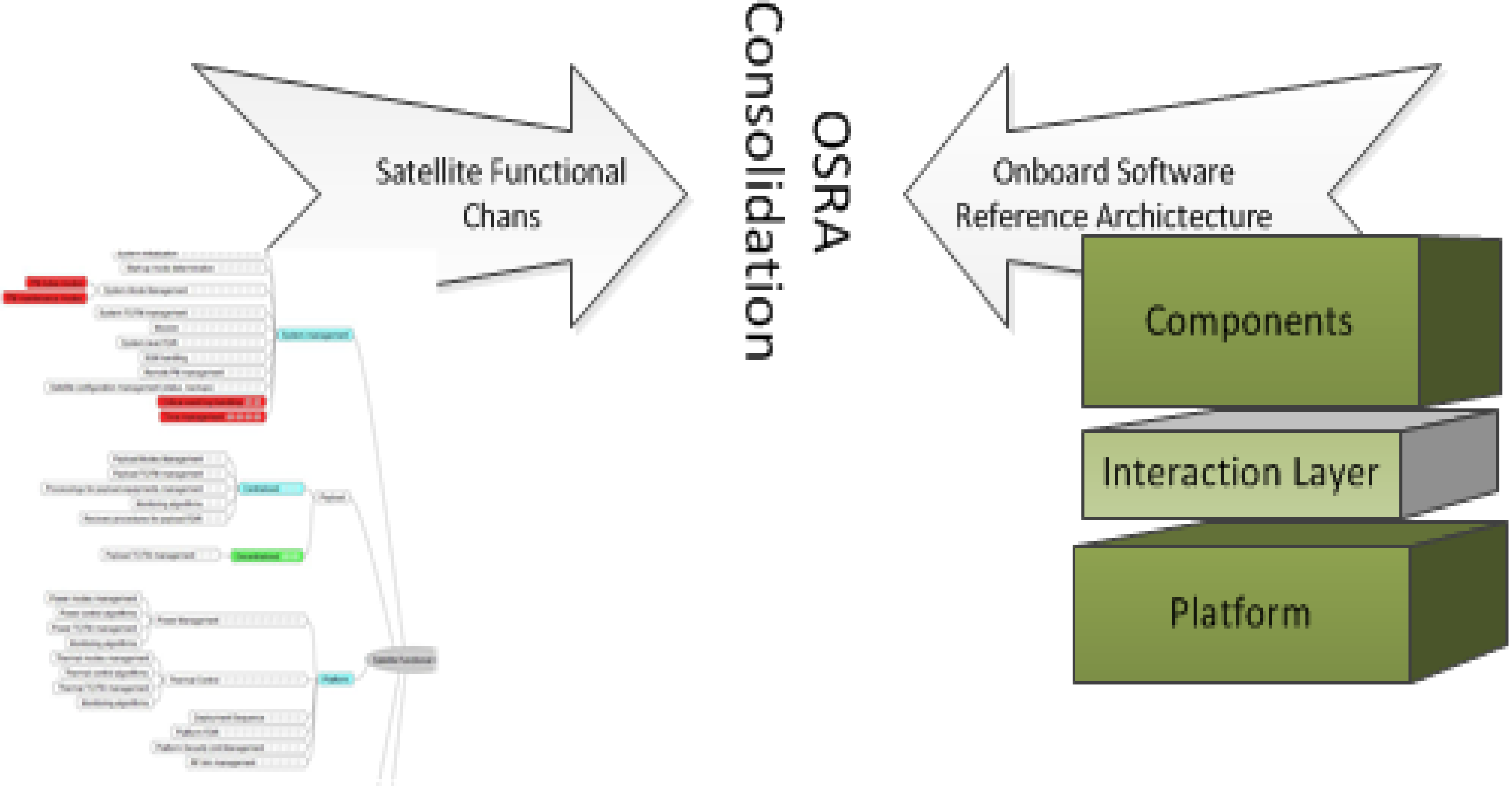
Onboard Software Reference Archictecture

The objective of the activity is to identify all building blocks and interfaces of the core onboard software reference architecture and to verify and consolidate the software architectural concepts described in the SAVOIR-FAIRE document and developed in COrDeT2.

- Parallel projects with same objective, but distributing the functional chains.

- Consolidate OSRA using actual or previous missions

Objective



Satellite Functional Chans

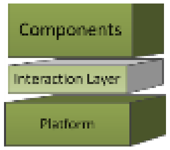
OSRA Consolidation

Onboard Software Reference Architecture

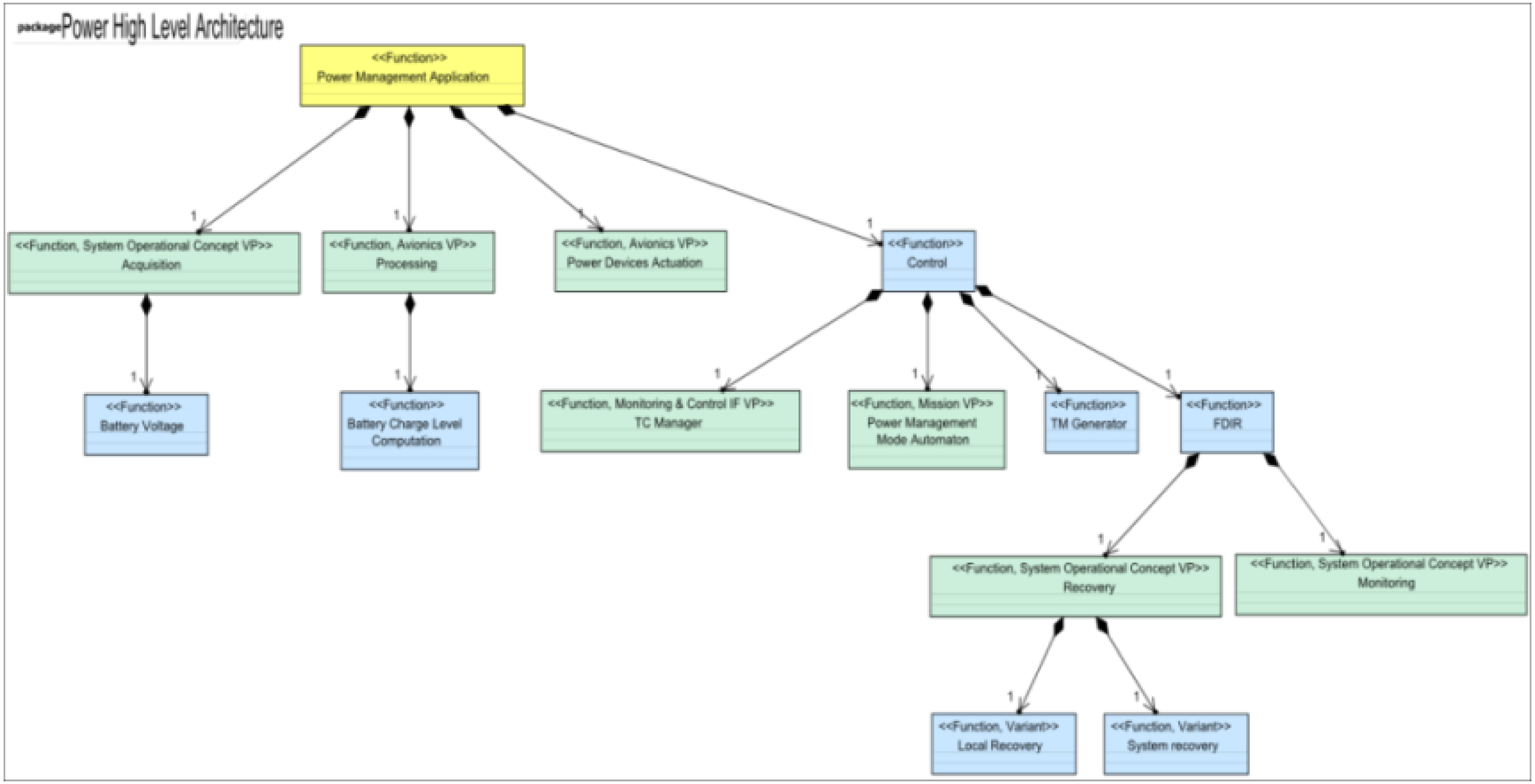
Components

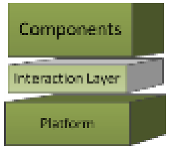
Interaction Layer

Platform

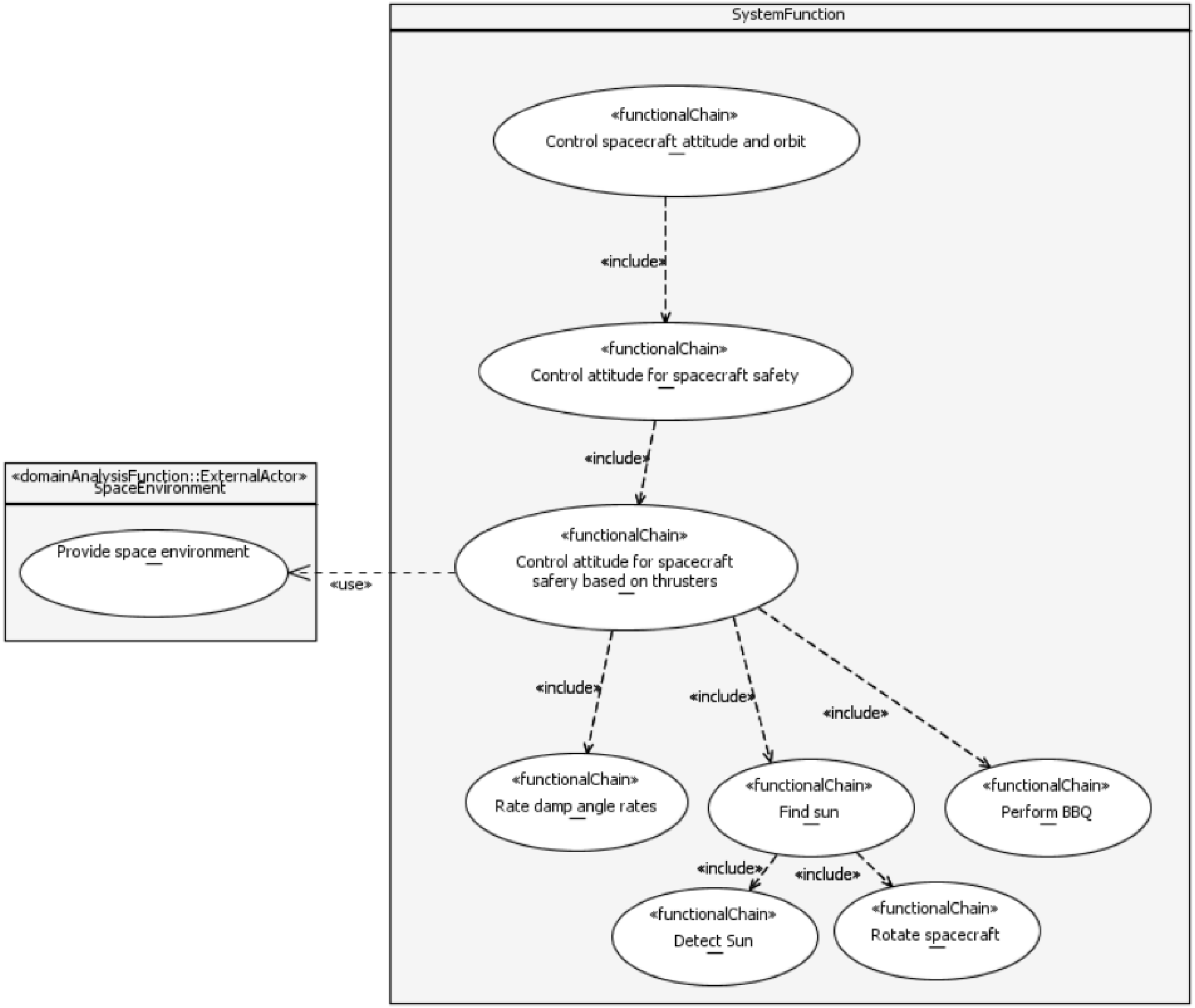


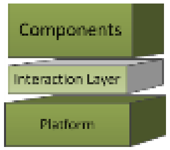
Functional Chains


package Power High Level Architecture
<<Function>>
Power Management Application
1
<<Function, System Operational Concept VP>>
Acquisition
1
<<Function, Avionics VP>>
Processing
1
<<Function, Avionics VP>>
Power Devices Actuation
1
<<Function>>
Control
1
<<Function>>
Battery Voltage
1
<<Function>>
Battery Charge Level Computation
1
<<Function, Monitoring & Control IF VP>>
TC Manager
1
<<Function, Mission VP>>
Power Management Mode Automaton
1
<<Function>>
TM Generator
1
<<Function>>
FDIR
1
<<Function, System Operational Concept VP>>
Recovery
1
<<Function, System Operational Concept VP>>
Monitoring
1
<<Function, Variant>>
Local Recovery
1
<<Function, Variant>>
System recovery

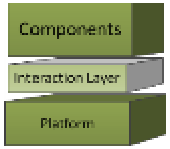


Functional Chains



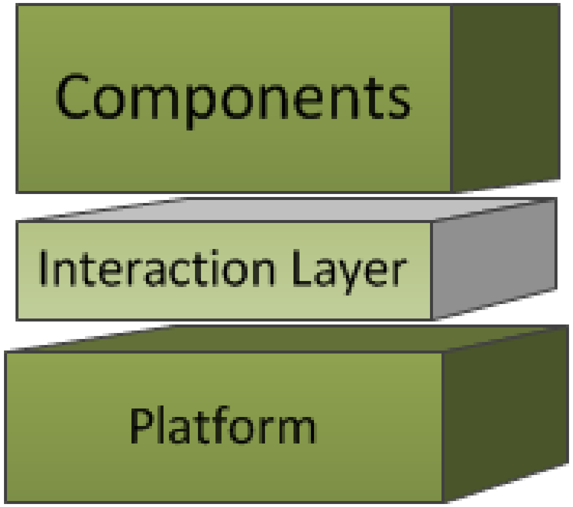


OBSW Reference Architecture

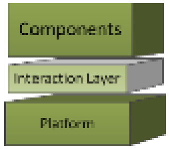


On-boards software reference architecture

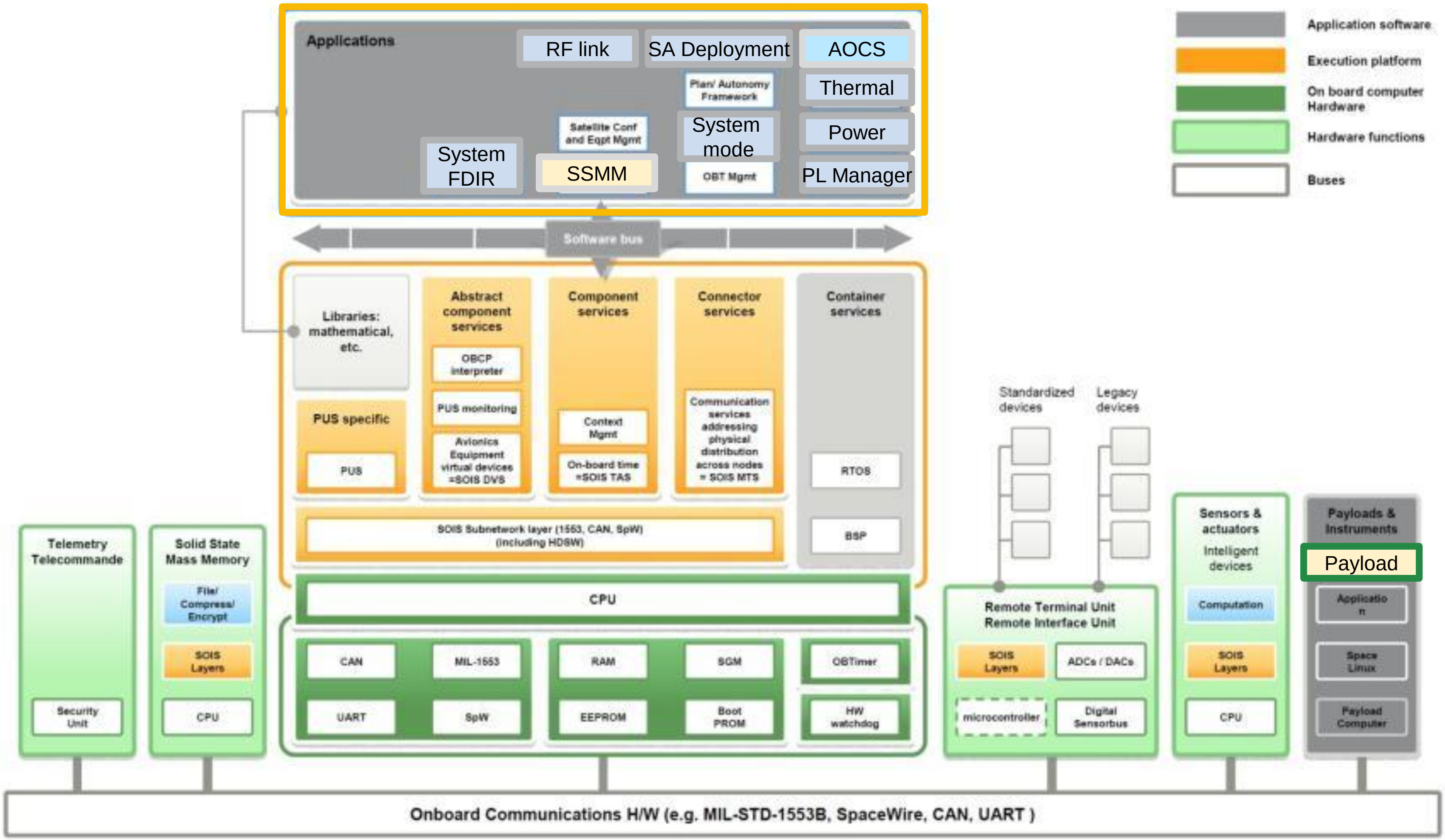
- Faster: ….
- Later: ….
- Softer: …

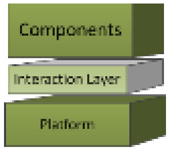


- Component Layer: On-board software applications
- Interaction Layer: tool generated
- Execution Platform: predefined set of services

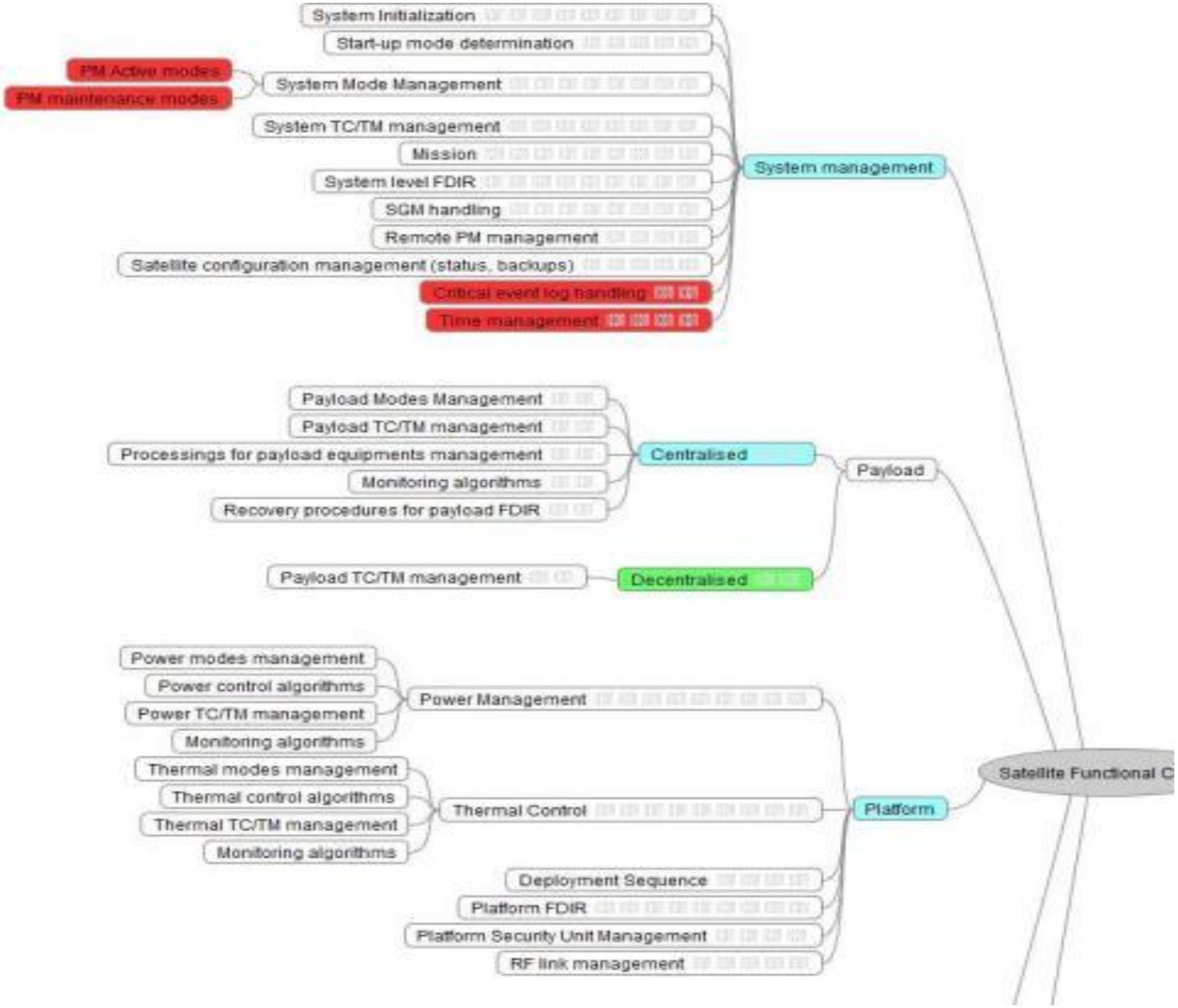


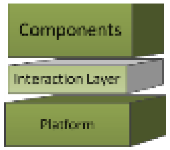
On-boards software reference architecture



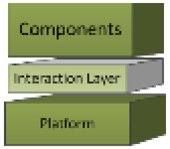


Mindmap


System Initialization
Start-up mode determination
PM Active modes
PM maintenance modes
System Mode Management
System TC/TM management
Mission
System level FDIR
SGM handling
Remote PM management
Satellite configuration management (status, backups)
Critical event log handling
Time management
System management
Payload Modes Management
Payload TC/TM management
Processings for payload equipments management
Monitoring algorithms
Recovery procedures for payload FDIR
Centralised
Payload
Payload TC/TM management
Decentralised
Power modes management
Power control algorithms
Power TC/TM management
Monitoring algorithms
Power Management
Thermal modes management
Thermal control algorithms
Thermal TC/TM management
Monitoring algorithms
Thermal Control
Deployment Sequence
Platform FDIR
Platform Security Unit Management
RF link management
Platform
Satellite Functional C

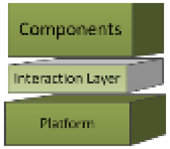

Components
Interaction Layer
Platform

Process (of the projects)



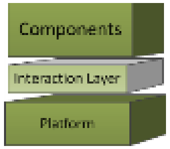
The Process

- Mindmap (mission analysis)
- Domain engineering approaches
- Functional chain analysis
- OSRA verified by mapping functional chains.
- Academic verification of software reference architecture.
- Relation to ECSS-E-ST-40C
- Building block example

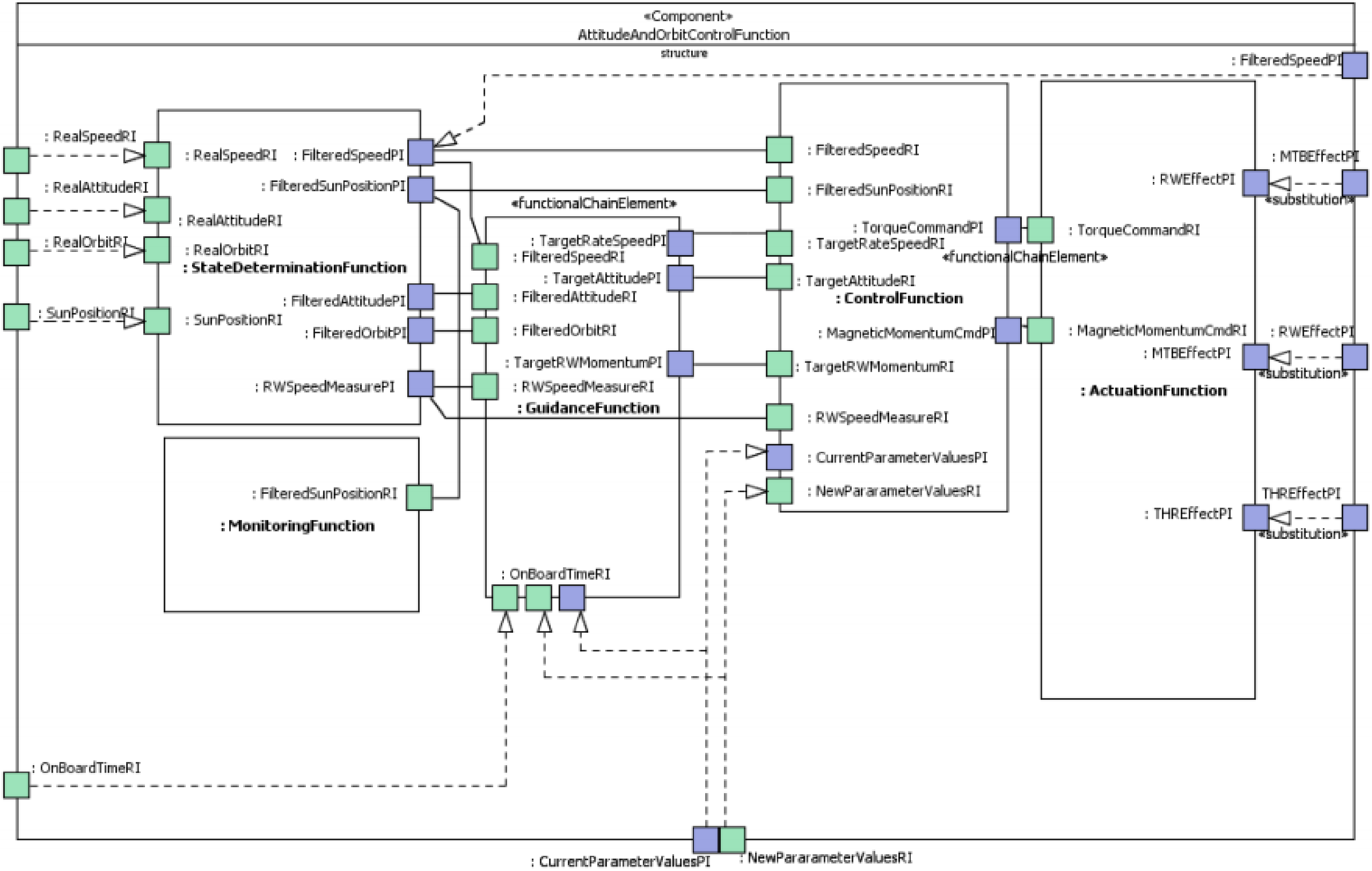


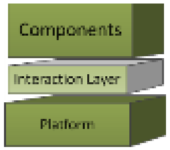
Domain Engineering

- Domain model
- Variability factors
- Functional decomposition
- Resolve variability to get to a functional decomposition
- SRS requirement generation

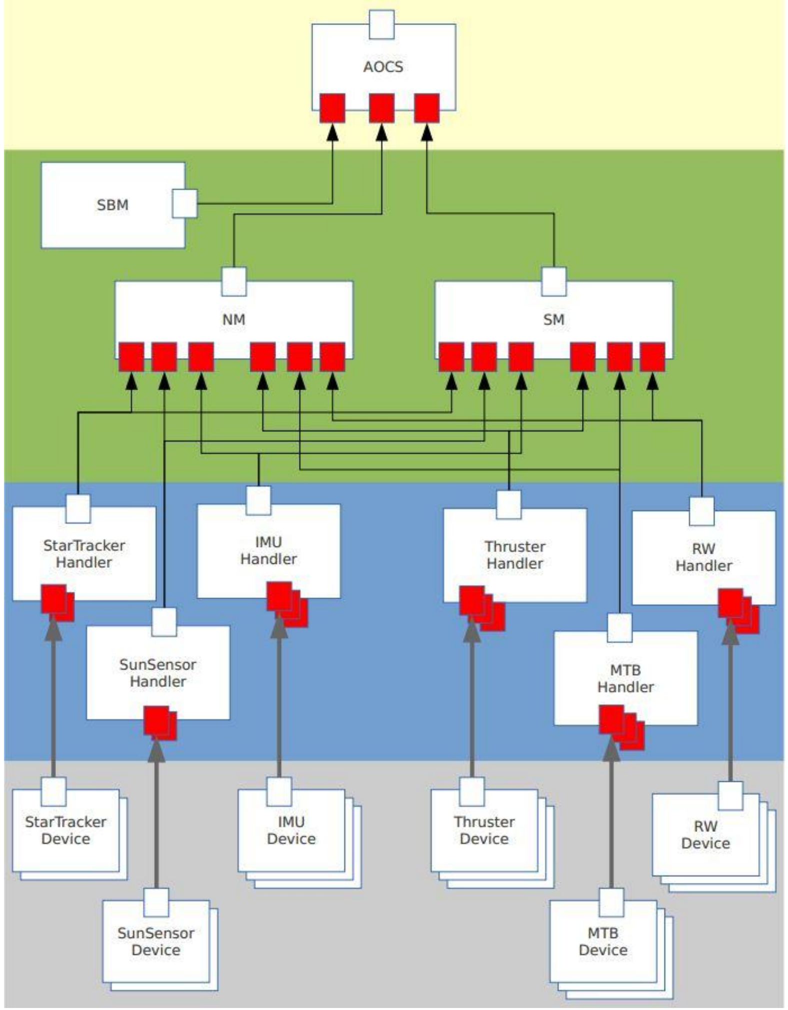


Functional decomposition

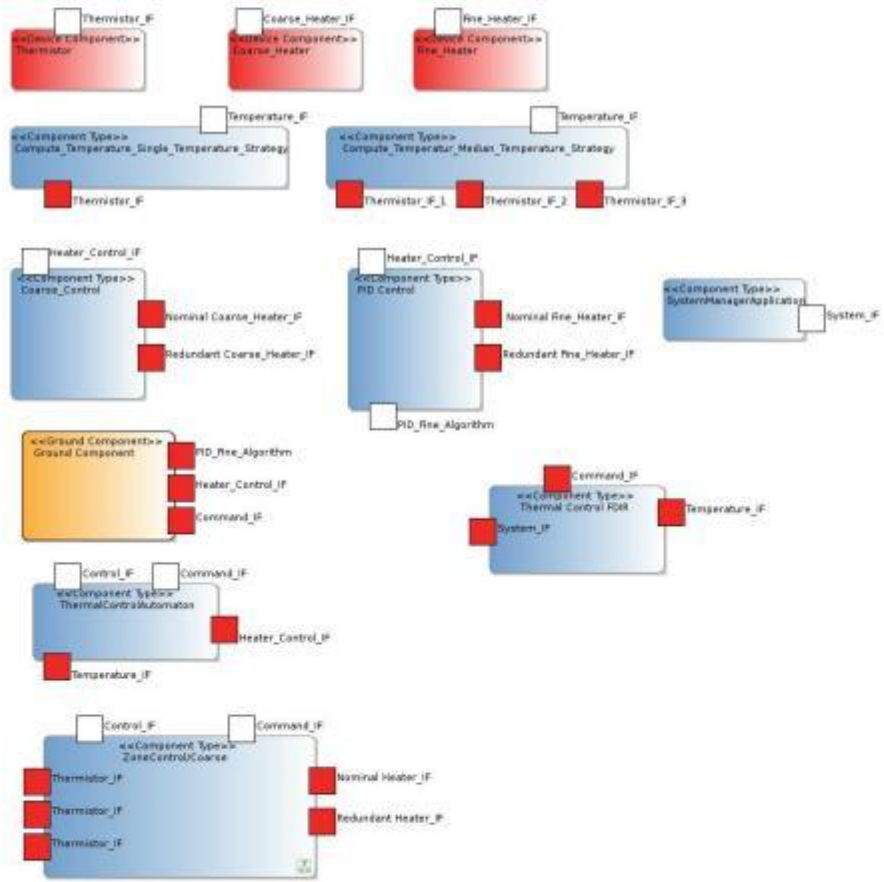




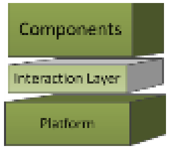
Example Functional Chain Mapping



AOCS Example from SSF



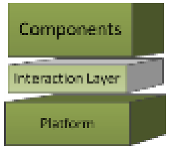
Screendump from SCM – instance diagram - on Thermal Example (Terma)



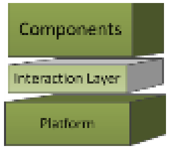
Mapping Experience

Experience gained by paper exercise and SCM tool usage.

- Ad-hoc approach from functional decomposition to OSRA component architecture
  - Based on experience of OBSW development
  - Following the steps of the OSRA design flow
    - But start from component instances
  - Manually checking if all functions are covered
- Expressive power of OSRA
  - How to specify order of component operations?
  - How to do mode handling?
  - How to combine concerns (views)?
  - How to do load-balancing?
- Issues and recommendations recorded as part of our consolidation efforts

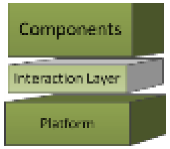


Building blocks



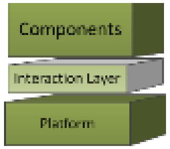
Building block (SSF)

- Has a clear, open, well-defined, specified, documented function and Interfaces
- Meets defined performance, operation and other requirements
- Is self-contained so as to be compatible with utilization at higher integration levels, e.g. board, equipment, subsystem
- Composability and Compositionality of its properties shall be guaranteed
- Has a TRL and quality level which can be assessed
- Is applicable in an envelope of well defined physical and software Environment
- Results from a process that can be repeated with guarantees
- Is worth developing, i.e. utilization is envisaged at least for the bulk of the ESA missions
- Is designed for reuse by different users, in different projects (it may be configurable depending on the variability factors)
- Can be made available off-the-shelf, under defined conditions.


Components
Interaction Layer
Platform

Building block (Terma)

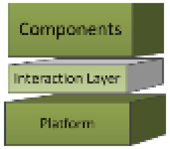
- *Is intended for reuse*
- *Has a well-defined objective, well understood in the target domain (building blocks are identified during the functional chain analysis).*
- *Is completely specified:*
  - *Has a clear and well-defined interface specification*
  - *Has a clear and well-defined specification of resource requirement*
  - *Has a clear and well-defined specification of performance guarantees*
- *Has a complete specification of validation and verification needs*



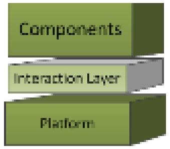
Components as Building blocks

- OSRA components have the technical properties of BB
- Non-technical properties are not elaborated much
  - they seem to highlight business potential

Conclusion: A building block is a component with business case



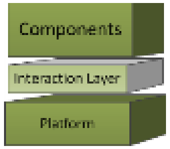
Reuse



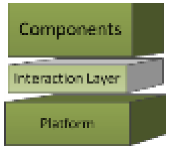
Reuse

- ECSS comprehensively covers reuse:
  - Reuse of software
  - Software developed to be reused
- ECSS does leaves process and documentation freedom
- OSRA approach makes reuse more systematic

- Types of reuse in OSRA setting
  1. Functional reuse
  2. Component type reuse
  3. Implementation reuse
     1. On different execution platforms
     2. On the same execution platform



Conclusions



Findings

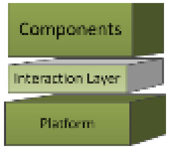
- Gap between functional chain analysis and OSRA (component model).

- The level of details required to apply OSRA to real missions revealed omissions.

- Separation of concern requires combinations of concerns.

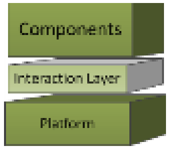
- Building blocks are nothing but components with a business case.

- Component model is not expressive enough.



Findings

- The OSRA component model is sound and fit for the purpose of developing real-time embedded applications
- Current state of OSRA is very promising and provides a very good basis upon which our (and other people’s) findings can be studies and resolved
- Ideas reused in ASIM project
- OSRA offers potential for systematic reuse at different levels


Components
Interaction Layer
Platform