

# A first dive into COrDeT

Jérémie POULY (CNES)

**ADCSS 2014**



# SOMMAIRE

- **INTRODUCTION**
- **FIRST THOUGHTS ON CORDET**
- **MARTE TO SUPPORT CORDET**
- **CHESS TO SUPPORT CORDET**
- **CONCLUSION & FUTURE WORKS**

# INTRODUCTION

- Space systems – and OBSW – are more and more complex
  - ◆ “Divide and conquer” paradigm
  - ◆ “Building blocks” approach
- CNES represented in the SAVOIR-FAIRE working-group (Paul ARBERET)
  - ◆ CNES shares the rationale and objectives of COrDeT
- Component-based software engineering is necessary
  - ◆ Relies on early interface definition between OBSW functions
    - » Could be the main objective by itself
  - ◆ Enables independent validation of OBSW parts
  - ◆ Facilitates import/export of components between actors of the space ecosystem
    - » Rules out one of the major source of concern during SW integration phase
- Difficulties to promote global component-based approach at CNES
  - ◆ Local experimentations are ongoing
    - » « Myriad-evolution » mass-memory component
  - ◆ COrDeT joint effort gives a legitimacy

# CNES EVALUATION

- CorDeT high-level objectives and technical orientations are shared by CNES (that participated in their definition)
- CNES decided to perform a first technical evaluation
  - ◆ Centered on the OSRA specification
  - ◆ To evaluate the technical concepts according to different criteria
    - » Mainly feasibility and maturity of each concept
  - ◆ To evaluate the OSRA specification (COrDeT2 « R6 » deliverable) as a standalone specification
- ➔ Two ongoing CNES studies
  - ◆ MARTE – COrDeT (ATOS)
    - » COrDeT analysis
    - » Capability of MARTE profiles to support COrDeT concepts
  - ◆ CHESS – COrDeT (CSSI / INTECS)
    - » COrDeT analysis
    - » Consistency of CHESS methodology and tools with COrDeT recommendations

# SOMMAIRE

- INTRODUCTION
- FIRST THOUGHTS ON CORDET
- MARTE TO SUPPORT CORDET
- CHESS TO SUPPORT CORDET
- CONCLUSION & FUTURE WORKS

# EVALUATION CONTEXT

- Focus on OSRA specification
- Analysis based on available documentation
  - ◆ COrDeT2 « report 6 » OSRA specification (and other COrDeT2 documents)
  - ◆ COrDeT training material based on COrDeT2
  - ➔ COrDeT3 documentation unavailable when the studies started
  - ➔ ESA support to focus on established concepts not questioned in COrDeT3
    - ◆ « Monitoring and Control Services » not considered
- Evaluation centered on the following concepts
  - ◆ Three-layers architecture
    - » Component layer
    - » Interaction layer
    - » Execution platform layer
  - ◆ Component model

# PRELIMINARY IMPRESSIONS (1/2)

- Three-layers architecture (PIM, PSM, Execution platform)
  - ◆ Separation of concerns (functionnal properties VS. non-functionnal)
  - ◆ Components are independant from the execution platform
  - ◆ Interaction layer is automatically generated and connected to services provided by the selected execution platform
  - ◆ **Architecture particularly adapted to SW evolutions and re-use**
  
- Component Model
  - ◆ Typical CBSE description language, most features are classical
  - ◆ **From an object-oriented programming perspective (as opposed to functional programming)**
    - » Data types do not accept operations (« methods »)
    - » Unusual interface definition (allowing for data fields declaration)
  - ◆ **Hierarchical components not covered in the OSRA (on purpose)**
  - ◆ **Mode management (to perform scenario based schedulability analysis) not adressed (on purpose)**

## PRELIMINARY IMPRESSIONS (2/2)

- Interaction layer
  - ✦ Automatically binds the component layer to the execution platform layer
  - ✦ Containers and connectors are generated according to a computational model
  - ✦ **Computational model remains an open and high-level concept in COrDeT**
  - ➔ **Containers and connectors lack specification**
    - » Specification seems to highly rely on ADA Ravenscar profile
  - ➔ **Specify interaction layer dependency on the computational model**
    - » Support « pseudo computational models »
- Execution platform
  - ✦ Set of services for I/O abstraction and data handling
  - ✦ Internal structure of execution platform out of the scope of COrDeT
  - ✦ Services offered either as pseudo-components or API
  - ✦ Should be provided with a tool to automatically generate the interaction layer
  - ✦ **Execution platform services should be explained in more details**

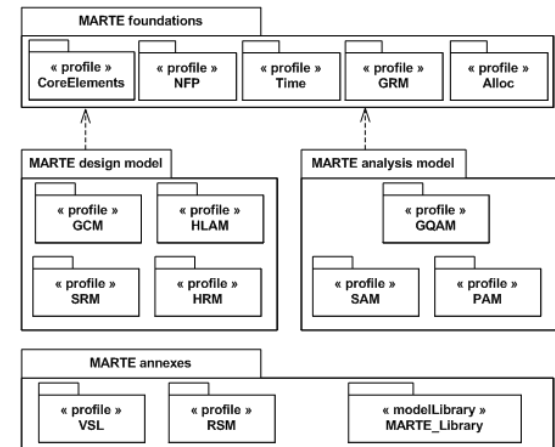


# SOMMAIRE

- INTRODUCTION
- FIRST THOUGHTS ON CORDET
- **MARTE TO SUPPORT CORDET**
- CHESS TO SUPPORT CORDET
- CONCLUSION & FUTURE WORKS

# MARTE FOR CORDET (1/2)

- MARTE: Modeling and Analysis of Real-time and Embedded systems
  - ◆ UML2 profile dedicated to real-time properties
  - ◆ Stereotypes to describe a component architecture
- MARTE offers many modeling concepts
  - ◆ Large set of stereotypes for NFP
  - ◆ Compatible with schedulability analysis tools
- Mapping of COrDeT concepts onto MARTE
  - ◆ Data types, interfaces, events → OK
  - ◆ Component type → abstract component
  - ◆ Component implementation → component that inherits from abstract component
  - ◆ Component instance → « part » of a top level « system » component
  - ◆ Non functional properties → several MARTE stereotypes (rtUnit, resourceUsage, rtService, rtSpecification)



## MARTE FOR CORDET (2/2)

- Mapping of COrDeT concepts onto MARTE
  - ◆ HW topology → several MARTE stereotypes (hwProcessor, hwBus, hwSensor,...)
  - ◆ HW platform → set of UML components
  - ◆ Allocation of component instances to processing units → UML « allocate » link
  - ◆ COrDeT interaction layer is difficult to describe in MARTE
    - » The concept remains too open in COrDeT
    - » In UML/MARTE, a « container » might be a dedicated UML component, part of the COrDeT component, one or more task, integrated in the middleware,...
- Most of COrDeT concepts can be mapped onto UML/MARTE
  - ◆ Mapping is not necessarily straightforward
- However some concepts are missing in MARTE
  - ◆ Component attributes categories (MIS, CFG, DAT) are missing
  - ◆ Execution priorities and constraints can not be applied to operations
  - ◆ Ports and operations instance do not exist (might not be a problem)

# SOMMAIRE

- INTRODUCTION
- FIRST THOUGHTS ON CORDET
- MARTE TO SUPPORT CORDET
- **CHESS TO SUPPORT CORDET**
- CONCLUSION & FUTURE WORKS

# CHESS FOR CORDET

- CHESS: Composition with Guarantees for High-integrity Embedded Software Components Assembly (Artemis Project)
- Activity that just started
- Several objectives
  - ◆ Get familiar with CHESS methodology and tools
  - ◆ Assess consistency of CHESS toolset with COrDeT recommendations
    - » Check that all COrDeT concepts are covered by CHESS
    - » Analyze how CHESS may drive COrDeT choices
  - ◆ Evaluate what CHESS might add to COrDeT
    - » Functional assembly issue ?
- Preliminary impressions
  - ◆ Currently still getting familiar with the tool
  - ◆ CHESS seems map relatively well on COrDeT concepts
  - ◆ Concept of « contract » to start addressing configuration management and component functional assembly ?

# SOMMAIRE

- INTRODUCTION
- FIRST THOUGHTS ON CORDET
- MARTE TO SUPPORT CORDET
- CHESS TO SUPPORT CORDET
- CONCLUSION & FUTURE WORKS

# CONCLUSIONS

- Some COrDeT concepts may be difficult to use as written
  - ◆ Not enough details in the documentation, sometimes on purpose
- Some critical issues have not yet been addressed (to our knowledge)
  - ◆ Configuration management
    - » Handle different version of a component (variation of NFP)
    - » Describe configurable components using metadata
    - » Add dedicated attributes in component model for versionning issues ?
  - ◆ Component functional assembly
    - » Automatically validate components configurations compatibility
    - » Provide a tool to be run at interaction layer generation ?
  - ➔ Mandatory in order to deploy a building-block approach
- Still, a great work has been accomplished
  - ◆ Nearly all aspects of CBSE have been addressed (architecture, design flow,...)
  - ◆ Consensual technical choices between primes & agencies
  - ◆ Strike while the iron is hot (COrDeT4 ?)

# CNES FUTURE WORKS

- Challenge this evaluation with COrDeT3 documentation
- Two studies related to COrDeT already decided
  - ◆ COrDeT assets for SW verification and validation
    - » Explicitely assert how V&V activities benefits from CBSE as defined in COrDeT
    - » Clarify the resulting ascending part of the V-cycle
  - ◆ Apply COrDeT recommendations to the development of a PUS library
    - » Design a PUS library based on the COrDeT OSRA
    - » Evaluate the flexibility of the library (PUS services selection)
- Other studies currently in CNES R&T selection process
  - ◆ Large-scale deployment of COrDeT OSRA on 2 representative case studies
  - ◆ Development of a COrDeT tool based on SIRIUS customized for CNES
  - ◆ Beyond COrDeT: versionning & assembling issues



Thank you for your attention !

Questions ?