MyCCM
A Component Based Approach for Real-Time & Critical Systems

Olivier Hachet
Olivier.hachet@thalesgroup.com

Thales Communications
1) Path toward CBSE for RTE

2) The MyCCM Framework

3) Future orientation
Path Toward CBSE
RTE Systems Characteristics

- Execution platform heterogeneity
- Complex internal communication and interaction schemes
- Reuse of code is not obvious
  - Intricated with underlying technologies
- Complex to integrate and tune
- Difficult to test
- Domain specific method and tools
Targeted Improvements for RTE Systems

- Capitalize on software architectures
  - Better structuration of software
  - Reuse of business code

- Common approach for several targets (HW, OS, MW)
  - Software Design tooled up process
  - One source running on multi-target
  - For near-real-time to hi-integrity runtime environments
  - Complementary IVV tooled up process

- Standalone to highly distributed
- Static to Reconfigurable

CBSE with Component/Container/Connector principles allows to reach these goals
From R&T to industrial adoption

Research Projects
- ARTEMIS
- ITEA2
- ANR
- Information Society Technologies
- Les Pôles de Compétitivité

Domain Needs
- Space
- Software Radio
- Optronic
- Electronic Warfare
- ...

CBSE dedicated to
Real Time Embedded Systems
- Component Framework Basis
- Real Time constraints
- Safety and Security
- Connection with MDE

MyCCM
- C++ (optro, vetro, SDR)
- Ada (space, radar)
- Java (ECS)
- C (railway)

Standardisation
- iWCCM (2003)
- DDS4CCM (2008)
- QoS4CCM (RTF 2007)
- ...UCM (2013 ...)

Informations confidentielles / propriétés de Thales. Tous droits réservés. / Thales confidential / proprietary information. All rights reserved.
MyCCM Framework
Component Based Approach

Non-structured code

Structured code, with abstraction API

Tooled Process (MDE)

MyCCM Approach enables the generation of technical code (communications, deployment, etc…)

Thales Communications
A common Architecture Description Formalism

- Based on the Lightweight CCM Abstract Component Model
  - Standard interaction patterns + Programming model
  - D&C for deployment

- Specific Adaptations for RTE
  - Non CORBA environments, static deployment
  - Very low footprint

- Built-in Extension mechanisms (Connectors)
  - Ability to define new interaction patterns and implementations
MyCCM Framework Architecture

Software Architecture

- Rhapsody
- MelodyCCM
- RSA / CX

Shared Architectural Principles

- EMF Component Model MetaData Structure
- Domain or Environment specific Extensions

NF Code Generation & Deployment on Targets

- Back-end ...
- Engagement Systems
- SDR/SCA
- Space / Weapon Systems
- Optronic / Vetro / EWR...
  - Common Generation Rules
  - Platform Specific Generation Rules
  - NF Generated Code + Deployment facilities
  - ...
MyCCM applied to SDR Signal processing

Front-end
- UML Modeler
- Graphical DSL (Melody, Advance)
- Textual DSL (IDL, XM, COAL)

Back-end C++ for SDR Digital Signal Processors
- Platform Specific Generation Rules
- Common Generation Rules
- Generated Code

Generated Code
- Common Generation Rules
- Platform Specific Generation Rules

MyCCM runtime (Abstraction layer)

Specific Processor Libs
- Error management
- threads
- allocators
- Method objects

Thales DSP OE
- MHAL Com
- Broker

SCA Lib

RTOS

Domain Specific Middleware and Services

POSIX AEP
MyCCM applied to Space Domain

Front-end
- UML Modeler
- Graphical DSL (Melody, Advance)
- Textual DSL (IDL, XML, COAL)

EMF Component Model MetaData Structure

Domain Specific Metadata

Back-end Ada for Space Domain
- Platform Specific Generation Rules
- Common Generation Rules
- Generated Code

Common Generation Rules

Generated Code

- Platform Specific Generation Rules
- Common Generation Rules
- Generated Code

Generated Code

MyCCM runtime (Abstraction layer)
- Task Mngt
- Semaphore Mngt

Comm. Space PUS

Message Passing

Software Bus

Domain Specific Middleware and Services

RTOS- Proprietary

Textual DSL
(IDL, XML, COAL)

Graphical DSL
(Melody, Advance)

UML Modeler

EMF Component Model MetaData Structure

Domain Specific Metadata

Generated Code

Common Generation Rules

Platform Specific Generation Rules

Generated Code

Generated Code

Generated Code

Generated Code

Front-end

Back-end Ada for Space Domain

MyCCM runtime (Abstraction layer)

Domain Specific Middleware and Services

RTOS- Proprietary
A tooled approach to software engineering
- Model Driven
- Component-Based Architecture Approach
- Promotes the use of Automatic code Generation

Focus on value added code
- Reduce the amount of time and effort on non value-added code

Improve reusability
- Identify reusable application blocks
- Integration of legacy code

Reduce integration costs
- Ensure Interfaces are properly defined
- Ease testing activities

Do not impose a framework but integrate domain expertise in a CBSE tooled approach “Make Your CCM”
**Optronic (C++)**
- Since 2007
- 3 programs

**Vetronic (C++)**
- Since 2009
- 2 programs

**Electronic Warfare (C++)**
- Since 2011
- 2 programs

**Control Station (Java)**
- Since 2008
- 1 program

**Software Defined Radio (C++)**
- Since 2012
- Support for GPP and DSP targets

**Space Domain (Ada)**
- since 2010
- Used in 5 programs

**Weapon systems (Ada)**
- To be deployed in 2013

**Railway Domain (C)**
- Under evaluation
- Targeted deployment in 2014
Future Orientations
A CBSE based on Component / Container / Connector

- A MDE component-based design
  - Based on a meta-model capturing CBSE main concepts
  - Extensible (capability to add new features for specific needs)
  - Target several execution frameworks

- Middleware agnostic
  - Interactions fully managed by connectors
    - Interaction patterns instead of specific interfaces
  - Optional compatibility with IDL/IDL3

- Support native languages for component implementation
  - Remove CORBA specific language mappings (or Optional)
  - C++11 or subset supported by current compilers
  - Java

- Reduce dramatically the complexity of the programming model
Future Component Framework Vision

Software Architectural View

Interaction pattern (Pub/Sub, synchronous Data flow, …)
Future Component Framework Vision

Execution Framework View

Component

Connector Fragment

Services access points

Cmp Fwk runtime & Services

Domain Specific Services (Thread, memory, mutex,....)

Com MW

Operating Environment

Operating System 1

Hardware 1

Com MW

Domain Specific Services (Thread, memory, mutex,....)

Cmp Fwk runtime & Services

Operating Environment

Operating System 2

Hardware 2

Thales Communications & Security
Promote an Open Source core MyCCM (C++ first)

- **Select the core building blocks**
  - Meta model, runtime, code generators, deployment
  - Ensure modularity on meta models, generators, deployment
  - Allowing to add new features as « plug-in »

- **Identify the appropriate Open Source community**
  - Providing facilities to host and promote the project
  - With LGPL or EPL license policies style

Contribute to its evolution toward UCM Standard

- **COMET project (French R&T) with Prismtech**
  - Make MyCCM middleware agnostic and contribute to lwCCM evolution

- **Commit new services and connectors when necessary**
Thank you!