PRISMTECH

Performance Critical Middleware

A Middleware-agnostic Unified Component Model

Nawel Hamouche, PhD
PrismTech France

Olivier Hachet
Thales

- Increasing interest in CBSE for distributed real-time and embedded (DRTE) systems

- Few domain specific component models have been emerging in the past few years, e.g. Autosar, Orocos, etc

- The OMG LwCCM standard is a domain-independent component model targeting DRTE systems

- No single component technology has been widely adopted

PrismTech

▶ **Portability**

  ▶ Platform, Network and middleware independence

▶ **Resource-constrained**

  ▶ Small and lightweight

▶ **Component Modeling**

  ▶ Standard modeling language (Ex. UML)

  ▶ modeling tools support

▶ **Computational Model**

  ▶ Execution threads management support

▶ **Real-time features**

  ▶ Analyzability

  ▶ Real-time services (Ex: Timers, Schedulers, …)

PRISMTECH

- ▶ Lightweight CCM is a profile for the CORBA Component Model targeting resource constrained systems
- ▶ An open, distributed, platform and programming-language independent component model
- ▶ Multiple existing implementations
- ▶ Specifies all the development cycle from design to deployment

PrismTech

Requirement	🙂	☹️
Portability	- programming-language, platform and network independent	- Strong dependency to CORBA
Resource-Constrained	- Reduced set of functionality wrt. CCM	- Not light enough - Additional CORBA footprint even when not used
Component-Modeling	- UML profile for CCM	- Lack of mature supporting MDE tools
Computational Model	- Single and multi threaded components support - Reactive execution model	- No support for RT models e.g. Periodic, Sporadic
Real-time features	- may be extended by real-time services thanks to Qos4CCM - May rely on RT-CORBA	- No native support to RT components - No standard RT services

PRISMTECH

LwCCM remains the most appropriate foundation for DRTE systems, *But, it needs to evolve into a new, simpler and middleware-independent standard that better address the business and technical needs of DRTEs*

PRISMTECH

Towards a Unified Component Model (UCM)

PrismTech

- ▶ Middleware-independent
- ▶ Simple and modular
- ▶ Extensible and Customizable
- ▶ Model-Driven Engineering enabled

PRISMTECH

▶ Container level
  ▶ Removing CORBA dependency
  ▶ No built-in support to any presumed middleware
  ▶ Deliver middleware flexibility through the connectors

▶ Component level
  ▶ CORBA-independent language mapping
  ▶ Middleware-independent connectors API (ex: Event API, Request/Reply API, Data-centric API, etc)
    ▶ Middleware-dependent connector implementation (ex: CORBA, DDS, etc)

▶ Deployment level
  ▶ Middleware-independent deployment interfaces of the components
  ▶ The component model should be independent from the deployment tool middleware

PrismTech

▶ **Removing CORBA types from components.idl**

  ▶ Cookie valuetype, CORBA::OctetSeq, CORBA::Object

▶ **An UCM component may have only facets, receptacles and extended ports**

  ▶ Facets and Receptacles ports express provided and used <u>local interfaces</u>

  ▶ Extended ports are the <u>only means</u> to express middleware-mediated communications

▶ **Removing events ports**

  ▶ Providing <u>Event-oriented extended port and connector</u> definition

▶ **Remote interfaces with the deployment tool**

  ▶ How to make CCMHome, Navigation interfaces and attributes remotely accessible by the deployment tool ?

PRISMTECH

▶ Reduce dramatically the complexity of current CIF and container programming model

▶ Enhance extensibility and modularity to reduce footprint

▶ Define standard profiles to modularize the component model implementation

PrismTech

- ▶ Only keyless homes supported
- ▶ No supported interfaces
- ▶ Removing CORBA will simplify the component implementation framework and the container programming model
  - ▶ Removing all interfaces/operations related to CORBA references and servants managements

PrismTech

▶ Open and extensible enough to support the variety of services, middleware and communication paradigms used in DRTE systems

▶ Ability to integrate new technical services

  ▶ Extensible set of non functional services

▶ Ability to integrate new connectors and middlewares

▶ Ability to integrate different execution/computation models for the component, Periodic, Aperiodic, Sporadic, …)

▶ Ability to evolve to a domain-specific model by customization

PRISMTECH

▶ Extensibility support with the Generic interaction Support and QoS4CCM

▶ Leveraging Qos4CCM specification to support container extensibility

  ▶ An interception framework for interception-based services

  ▶ A registration framework for "pure" services

▶ Connectors integration already stated by DDS4CCM

▶ How supporting different execution models ?

PRISMTECH

- ▶ Standard meta-models for components specification, implementation, packaging, assembly, configuration and deployment
- ▶ Promotes MDE tools support
- ▶ Tools and vendor independent Models
- ▶ Mature and productive modeling tools is key for UCM

PRISMTECH

▶ LwCCM Specifies all the development cycle from design to deployment

▶ Deriving a middleware-independent UCM meta-model from the CCM meta-model

▶ Using the Deployment&Configuration specification for components deployment

PRISMTECH

- DRTE systems still needs component technologies

- Openness, extensibility, modeling, platform, network and middleware independence are key features for the next successful component technology for DRTE systems

- The UCM standard is already on the road to meet these challenges

PRISMTECH