Model Driven Development of High Integrity Applications Based on Reusable Assets

Miguel A. de Miguel
Technical University of Madrid
Topics

♦ Modelling Tool Assets and Modelling Tool Configurations
  • Modelling tools: one asset for multiple tools

♦ Modelling Assets for High Integrity Applications
  • UML2RTSJ : RTSJ applications in UML
  • UML2AdaRavenscar : Ada Ravenscar Applications in UML
  • MAST_RMA : Modelling MAST in eclipse modelling tools
  • MARTE1x1 : MARTE Profile
  • MARTE_SA_Gen : Integration of MARTE, RTSJ, AdaRavenscar and MAST

♦ Application example: tele-presence robot
Modelling Tool Assets

- ERMA provides support to reuse *same modelling tools* in *multiple modelling tools*
  1. Install ERMA
  2. Import Assets
  3. Deploy Assets
  4. Install Assets

- The **deployment** is key. It customizes
  1. UML Profiles
  2. Model libraries
  3. Commands & Inter Assets Communication
  4. Software Behaviours
Modelling Assets For Development of High Integrity Applications

- **Software development**
  1. Generators for RTSJ
  2. Generators for Ada

- **Models Analysis**
  3. Scheduling Analysis

- **UML for Real-Time**
  4. MARTE 1.1
  5. Edition support

- **Tool Integration**
  6. Mapping from MARTE 2 MAST
  7. MARTE 2 RTSJ
  8. MARTE 2 Ada
Topics

✨ Modelling Tool Assets and Modelling Tool Configurations
  • Modelling tools: one asset for multiple tools

✨ Modelling Assets for High Integrity Applications
  • UML2RTSJ : RTSJ applications in UML
  • UML2AdaRavenscar : Ada Ravenscar Applications in UML
  • MAST_RMA : Modelling MAST in eclipse modelling tools
  • MARTE1x1 : MARTE Profile
  • MARTE_SA_Gen : Integration of MARTE, RTSJ, AdaRavenscar and MAST

✨ Application example: tele-presence robot
UML2RTSJ: RTSJ applications in UML

The kernel of UML2RTSJ is a Java AST metamodel
- Represents Java programs as.ecore models

UML model library for representation of
- RTSJ library
- Some basic Java libraries

QVT transformations from UML 2.2 L0 to javaxmi

MOF2Text generators from javaxmi to Java (Jamaica)

Three basic commands
- Load UML library (javax.rtsj)
- Generate javaxmi from UML package
- Generate Java from javaxmi Model
UML2RTSJ : RTSJ applications in UML

- Cruise Control Example modelled with UML/RTSJ: AsyncEvent, RealtimeThread, MonitorControl, ...
- UML model elements handled are UML Infrastructure (Classes, Association, Operations, Instance specifications, ...) and some UML 2.2 L1 elements: Interfaces
- We create the model (based on RTSJ UML model library) and then
  - Generate Java AST (this the L0 of Java applications)
  - Generate Java code
UML2AdaRavenscar : Ada applications in UML

- Ada Ravenscar centre is the UML neutral model of Ada applications
- Generates pattern based plain UML output model
  - Only standard UML 2.2 types and data types
  - Most common real-time patterns (periodic, sporadic, shared...)
  - Generic UML classes with direct translation to Ada generic packages
  - MOF2Text generators to Ada Ravenscar
MAST (1.3.8) Asset

- Integration of MAST into eclipse
- **EMF Models**
  - MAST Modelling - representation of MAST input models in ecore
  - MAST Results - representation of MAST output models in ecore
- **Artefacts** - mast analysis and convert programs
- **Editors** - edition based on navigators of MAST analysis and MAST result models
- **Commands** for invocation of MAST analysis
- Parameters for **configuration** of analysis
  - Unified time unit
  - MAST analysis method
MARTE1x1 : MARTE Profile

- MARTE1x1 asset was designed as interchange language from CHESS ML to transformators to analysis/code
  - MARTE 1.1 profiles
  - MARTE Model Library
  - Tool independent URIs

- There is not VSL and it supports XMI of MARTE standards

- We have included ours editors of MARTE stereotypes. This is supported in: UML2, RSA, Papyrus
**MARTE_SA_Gen** : Integration of MARTE, RTSJ, AdaRavenscar and MAST

- **MARTE_SA_Gen** purpose: integrate MARTE, MAST analysis, and RTSJ or Ada Ravenscar platform
- **MARTE_SA_Gen** includes some QVTO transformator, modules, and libraries for generation of MAST models, and to extend RTSJ/Ada generators based on MARTE
- Input Model of **MARTE_SA_Gen**
  - UML+SAM+GQAM+GRM
- Output Models
  - Ada Neutral UML
  - RTSJ Javaxmi
  - MAST model
- Round-trip transformator from MAST result to UML+SAM+GQAM
MARTE_SA_Gen : Integration of MARTE-MAST

- MARTE_SA_Gen generates MAST models based on SAM-GQAM-GMR without any assumption about execution platform
- SAM-GQAM-GRM annotates any UML models: e.g. RTSJ model with object and behaviour diagrams (e.g. sequence, activity, ...)

Technical University of Madrid
SAM-GQAM-GRM is a parallel model of software model

Both can contain inconsistencies
**MARTE_SA_Gen**: Integration of MARTE-MAST-RTSJ

- UML2RTSJ supports the mapping from UML 2.2 L0 to javaxmi
- MARTE_SA_Gen extends this for the integration of MARTE-SAM-GQAM-GRM into javaxmi models
- Java AST models based on RTSJ (javaxmi) are consistent with MAST models generated
MARTE_SA_Gen: Integration of MARTE-MAST-RTSJ

What are the differences of UML+RTSJ Model Library and UML+MARTE

- UML+RTSJ includes explicit references to RTSJ library (types, method calls, generalizations, ...)
- UML+MARTE does not include any reference to RTSJ. Any reference to RTSJ in Java code is generated based on SAM-GQAM-GRM
- UML+RTSJ are based on Java primitive types (e.g. long, void, ...)
- UML+MARTE do not use Java types.
- The consistency of MAST model and RTSJ behaviour depends on modeller
- MARTE_SA_Gen guarantees the consistency of MAST and RTSJ behaviour
Topics

♦ Modelling Tool Assets and Modelling Tool Configurations
  • Modelling tools: one asset for multiple tools

♦ Modelling Assets for High Integrity Applications
  • UML2RTSJ : RTSJ applications in UML
  • UML2AdaRavenscar : Ada Ravenscar Applications in UML
  • MAST_RMA : Modelling MAST in eclipse modelling tools
  • MARTE1x1 : MARTE Profile
  • MARTE_SA_Gen : Integration of MARTE, RTSJ, AdaRavenscar and MAST

♦ Application example: tele-presence robot
Application Example: tele-presence car

- Tele-drive for a tele-presence car. Main components:
  - Steering wheel and throttle pedal handlers
  - Cruise control system: handlers of brake, throttle, lever, speed and engine, and control
  - Combined obstacles detector: indoor localization sensor, webcam images filter, IR detector
Application Example: tele-presence car

Transformations and generators provides:

- MAST scheduling analysis for schedulable resources
- RTSJ Java code -> RTSJ code generated + logical code in Opaque Behaviors
Application Example: tele-presence car

- Modelling and programming languages:
  - UML+MARTE
  - Scheduling analysis models
  - Java+RTSJ, Ada-Ravenscar+ORK
Tools available

http://www.erca-assets.org

- Eclipse Indigo with ERMA 64K and 32K
- Eclipse Indigo 64K with ERMA and Papyrus
- ERMA plug-ins site
- ERMA plug-ins for RSA 8.0.3
- ERMA Safety-aware Assets
  - S&D Profiles and Model Libraries
  - FTA & FMECA Eclipse Languages
  - UML to FTA & FMECA transformations
  - Item Toolkit Bridges
- ERMA Assets
  - MARTE 1.1 Conform Standard and Stereotypes Editor
  - RTSJ and Ada-Ravenscar Modelling tools
  - MAST Eclipse Languages
  - MARTE to MAST, RTSJ and Ravenscar generators