UML, SysML and MARTE in Use, a High Level Methodology for Real-time and Embedded Systems

Alessandra Bagnato*, Imran Quadri° and Andrey Sadovykh°

* TXT e-solutions (Italy)
° Softeam (France)
Presentation Outline

- Introduction
- MADES Overview
  - End User Case Studies
- MADES Methodology
  - Car Collision Avoidance System (CCAS) example
- Feed back from End Users
- Conclusions
- Demo
INTRODUCTION
Motivations

• Need of effective design methodologies for Real-Time and Embedded Systems (RTES)

• High abstraction level based approaches are promising: reducing time to market and system complexity
  - Model Driven Engineering, UML
SysML

• For Systems Engineering
  ▪ Aspects such as **Non Functional properties**, **Time** concepts are not present for RTES specifications
  ▪ Allocation aspects

• Widely adapted in the industry with supporting tools
MARTE

- For RTES design and specifications
  - Co-Design,
  - Non Functional properties,
  - Time aspects,
  - System analysis possible
- Specification are complex to use
- Currently lacks sufficient tool support and complete methodologies
MADES OVERVIEW
The Vision

Lowering SysML/MARTE entry barriers

“MADES aims to develop a holistic, model-driven approach to improve the current practice in the development of embedded systems. The proposed approach covers all phases, from design to code generation and deployment”
Project Consortium

- Project type: Collaborative Project (STREP)
- Duration: 30 months
- Project start: February 1, 2010
- Project end: July 31, 2012
- Objective: Embedded Systems Design
- 6 partners

MADES PROJECT – FP7 248864

OMG Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems, 18th April 2012, Paris-France
Partner Roles

User needs, models and Use Cases
- Cassidian [EADS] (DE)
- TXT E Solutions (IT)

Research/development
- Politecnico di Milano (IT)
- University of York (UK)

Standardization and dissemination
- Open Group (UK)

Technology/IT industry
- Softeam (FR)
- TXT e-solutions (IT)
MADES Approach

- **Design activities** exploit a dedicated language developed as an extension to OMG’s MARTE and SysML profiles.

- **Validation** include the verification of key properties on designed artifacts, closed-loop simulation.

- **Code generation** addresses both hardware description languages and conventional programming languages.
MADES Case Studies

A ground based radar processing unit provided by Cassidian

An onboard radar control unit provided by TXT

MADES PROJECT – FP7 248864

OMG Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems, 18th April 2012, Paris-France
The radar has to follow the aircraft movements to be able to direct the antenna on the target accordingly.

The radar system receives the navigation information from the FMS (Flight Management System) and the AHRS is the inertial platform: a set of sensors managed by a computer that provides the values related to position, asset, velocity and other values.
End User Requirements

- **System Specification**
  - Requirements modeling
  - Functional system design

- **System Co-Design**
  - Software design
  - Hardware design
  - Allocation
  - Timing and Scheduling
  - Behavior

- **Verification, Simulation, Code Generation and Synthesis**
Challenges for MADES

• MARTE : designed to enable flexibility
  - **Drawback:** Large number of concepts (700+ pages of specs)
  - Same concept can be applied to different UML elements (classes, instances, connectors, ports, attributes)
  - Same design may be modeled in different ways using MARTE concepts from different MARTE packages and different UML elements
  - Current academic/industrial MARTE modeling tools are too generic: provide all concepts, no usage tips
  - Lack of guidelines and examples

• SysML and MARTE combination: poorly expressed
  - Not effective dedicated methodologies or tools
  - Need of lowering entry barriers
Challenges for MADES

UML

MADES

SysML

MARTE
MADES Methodology

- Effective SysML/MARTE subset
  - SysML for functional specifications
  - MARTE for non-functional and co-design specifications
  - UML behavioral specifications supported
- Integration of Verification and Validation (V&V) + code generation concepts
- Dedicated diagrams
- Influence on future revisions of SysML and MARTE standards
MADES Implementation Approach

- Generic MADES methodology, guidelines and examples to guide system designers:
  - Modeling, Verification
  - Code generation, Synthesis
- Reducing ambiguities
- Reducing design time and costs
- Reinforce formality for Validation and Verification
MADES Implementation Approach

Modeling

Verification

Code Generation

Component Repository

A ground based radar processing unit provided by Cassidian
MADES Implementation Approach

An onboard radar control unit provided by TXT
Requirements diagram
SysML based requirements for initial system functional requirements

UML behavioral diagrams
Use case, Activity, Sequence, State and Interaction overview for initial system behavior

Functional/Internal Functional Specification diagrams
Description of system functionality by means of SysML block and internal block descriptions
MADES Diagrams

**Refined Functional Specification diagram**
Refinement of SysML concepts into MARTE aspects, addition of non functional properties

**Hardware Specification diagram**
Description of generic hardware: nodes, memories, communication channels, clocks etc

**Software Specification diagram**
Description of application tasks and software aspects running on hardware platform
MADES Diagrams

**Detailed Hardware Specification diagram**
Refinement of hardware concepts with details closer to execution platform details

**Detailed Software Specification diagram**
Description of underlying OS (if any), refinement of software aspects

**Allocation Specification diagram**
Mapping of software and hardware aspects of the system
MADES Diagrams

Clock Specification diagram
Definition of system clock types and clocks, clock and timing constraints for hardware/software aspects

- Dedicated commands in each MADES diagram to speed up the design process
  - Implemented in Modelio Open Source CASE tool
  - Valuable input from partners to improve design experience
  - Increased efficiency, decrease in design time
Modelio goes Open Source

- Open source community
  - Forums, Wiki, Projects
- UML and BPMN
- Wide range of modules
  - TOGAF, SoaML, Java
  - SysML, MARTE, MADES

- Commercial solutions
  - Business Architect
  - System Architect
  - Developers
- Warranties
- Support
- Services

www.modelio.org
www.modeliosoft.com

MADES PROJECT – FP7 248864
OMG Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems, 18th April 2012, Paris-France
Modelio (Screenshot)

Dedicated commands in Diagram Palette

Diagram Explorer

MADES tab

OMG Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems, 18th April 2012, Paris-France
Car Collision Avoidance System

• A system able to detect and prevent collisions
  ▪ Either using a radar tracking module
  ▪ Or an image processing module
  ▪ Initial version presented at OMG Technical Meeting at Arlington, VA – USA (March 2011)
System Requirements

**Global Collision Avoidance Strategy**
Detect collision by means of installed radar detection or by the image tracking system. Switching between radar and image tracking depending upon user requirements and weather conditions. Take appropriate actions to avoid collisions and notify the driver.

**Imminent Collision Strategy**
When external object is less than 2 meters away, the driver should be notified by an alarm and HUD, and system should switch to a critical warning state. If the system remains in critical warning state for more than 300 ms and distance from object is still less than 2 meters then the engine should be stopped, brakes should be applied, seatbelts should be tensioned and airbag should be deployed. The brakes should be applied for 100 ms.

**Near Collision Avoidance Strategy**
When external object is less than 3 meters away, the controller should switch to a warning state. If the system remains in this state for 300 ms, then the driver should be notified via an alarm and HUD, normal brakes should be applied for 10 ms for reducing car speed. Hazard warning lights should be activated in the HUD.

**Changing Lanes Strategy**
When the driver is about to change lanes or if the car deviates from the current lane intentionally, the driver should be notified via the HUD. If the driver is changing lanes himself, the turn signal should be on.

**Additional Timing requirements**
The radar or the image tracking module should send the data to the controller every 100 ms via the system bus and the communication should take 20 ms, during which the bus should be busy. In case of imminent collision, the controller should send the brake command which should take no more than 20 ms.

**Car Collision Avoidance Module**
Extract of CCAS Specifications

Allocation Specifications

MADES PROJECT – FP7 248864
OMG Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems, 18th April 2012, Paris-France
Temporal Logic: Example of temporal properties

Name: CCAS

Verification Project Configuration

Zot Setup Information

Time Bound: 100
Zot Plugin: meezot
Solver: minisat

Set property to be verified (event1 implies event2)

Event 1
Constraint – [distance < 2]
Start

Event 2
State – braking
Start

Lasted
50

Within Past
50

Directory of Zot executable

Zot Directory: /usr/local/bin

Browse
MADES: Verification and Validation

Verification Results

MADES PROJECT– FP7 248864
Focus on the image processing module and its related task:

- Would benefit from custom architectures
- Computation requirements are high
- Image manipulation code is available in standard benchmarks
- Involves data movement
MADES: CCAS Hardware Generation

- Automatic generation of implementation hardware from MADES design models

OMG Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems, 18th April 2012, Paris-France
MADES: CCAS Software Generation

Architecture-neutral code
Easy to write, but wouldn’t work on the real hardware

written for

Virtual platform
Simple, supported by the language

translated to

Architecture-specific code
Hardware specific, would be hard to write manually

for execution on

Actual platform

C / Java / etc. Assume this simple architecture

Many different actual platforms may be targeted

OMG Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems, 18th April 2012, Paris-France
MADES: CCAS Software Generation

- Integrated into the Eclipse development environment

OMG Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems, 18th April 2012, Paris-France
MADES: Synthesis on Execution Platform

Synthesis
Place & Route

Programming

OMG Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems, 18th April 2012, Paris-France
END USER FEED BACKS
Evaluation and Procedure followed:
- Evaluation of First & intermediate versions of the tools
- Feedbacks provided on unique MADES diagrams and stereotypes
- Final version of the tools has been released taking into account end user feedbacks
- Final evaluation started from beginning of March 2012 with real life end user case studies
Case Study

- Modeling of RTES using MARTE and SysML through MADES Language subset,
- Component modeling and storage features,
- Non-functional properties specification and verification,
- Usage of a specific set of unique diagrams for expressing different aspects related to a system,
- First experimentation with MADES Integrated environment
CONCLUSIONS
Conclusion

• MADES : SysML/MARTE methodology
  ▪ Combined usage of the two complex profiles
  ▪ High level Modeling, Verification & Validation, Code generation and eventual platform implementation on FPGA

• MADES diagrams and guidelines available
  • 10 dedicated diagrams + 5 UML behavioral diagrams
    – Available as open source at http://www.modelio.org/
  • Rapid prototyping of RTES

• Possible positive influences on future versions of OMG specifications

Lowering SysML/MARTE entry barriers
Thanks!

Alessandra Bagnato
TXT e-solutions
alessandra.bagnato@txtgroup.com

Imran Quadri
SOFTEAM | ModelioSoft
imran.quadri@softeam.fr

MADES Project Web Site:
http://www.mades-project.org/

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 248864.