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)

MADES PROJECT– FP7 248864

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

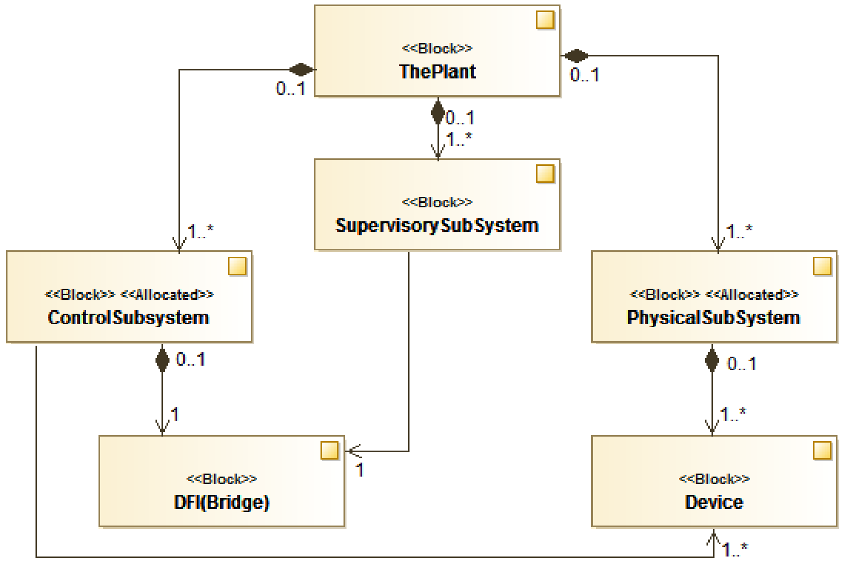
INTRODUCTION

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

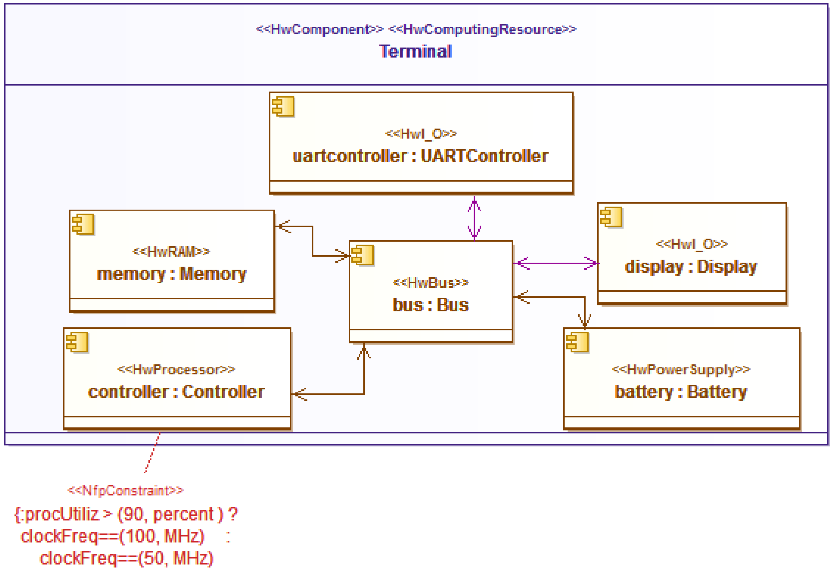
SysML

- 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

TXT e-solutions

SOFTEAM Think Object

THE UNIVERSITY of York

POLITECNICO DI MILANO

THE Open GROUP

CASSIDIAN AN EADS COMPANY

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)

Technology/IT industry

- Softeam (FR)
- TXT e-solutions (IT)

Standardization and dissemination

- Open Group (UK)

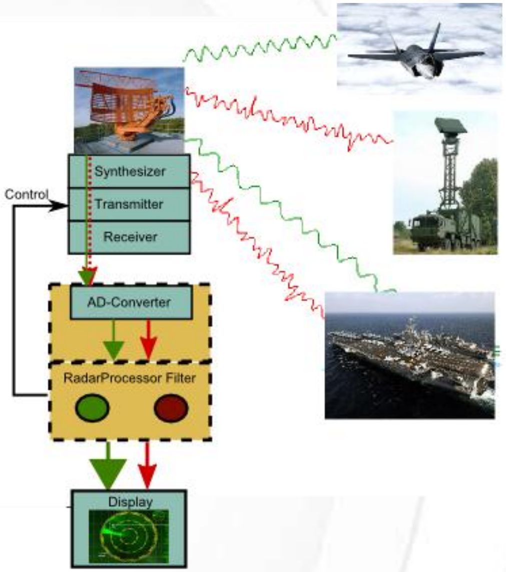
MADES Approach

Design activities exploit a dedicated language developed as an extension to OMG's MARTE and SysML profiles

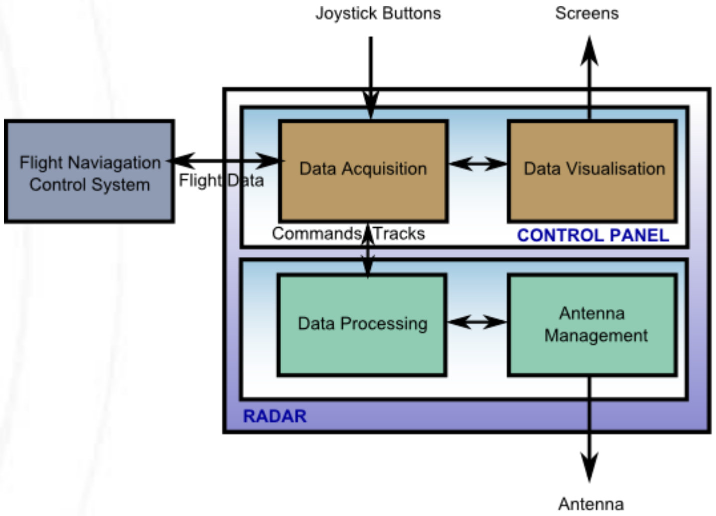
Validation include the verification of key properties on designed artifacts, closed-loop simulation

Code generation addresses both hardware description languages and conventional programming languages

MADES Case Studies



A ground based radar processing unit provided by Cassidian



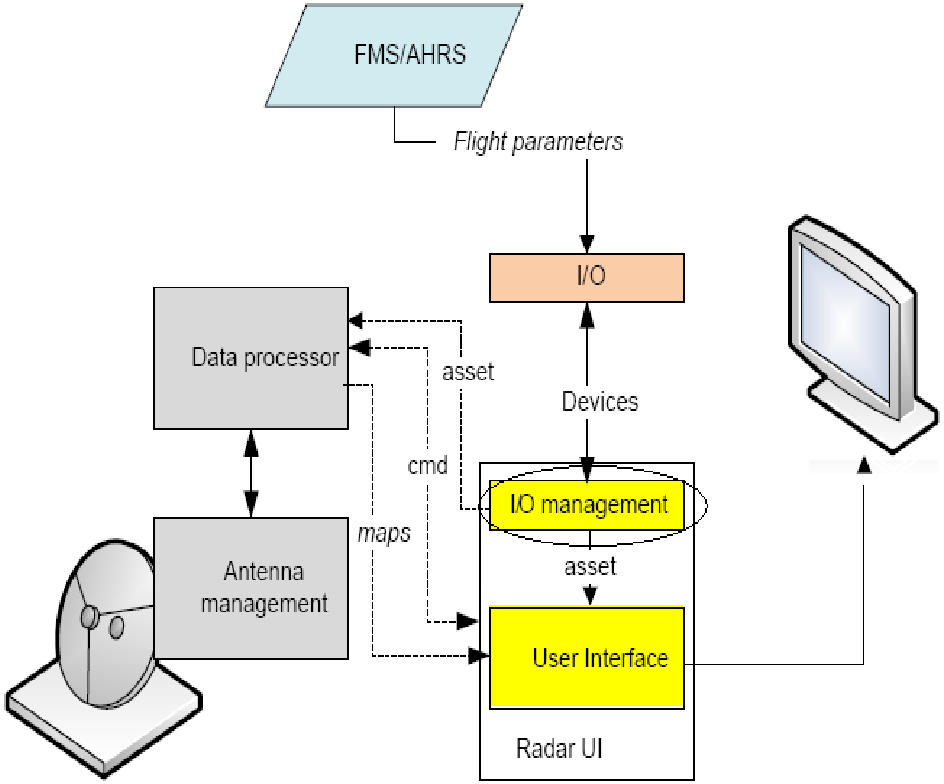
An onboard radar control unit provided by TXT

MADES Case Studies: TXT e-solutions

Onboard Radar System User Interface

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.



MADES METHODOLOGY

End User Requirements

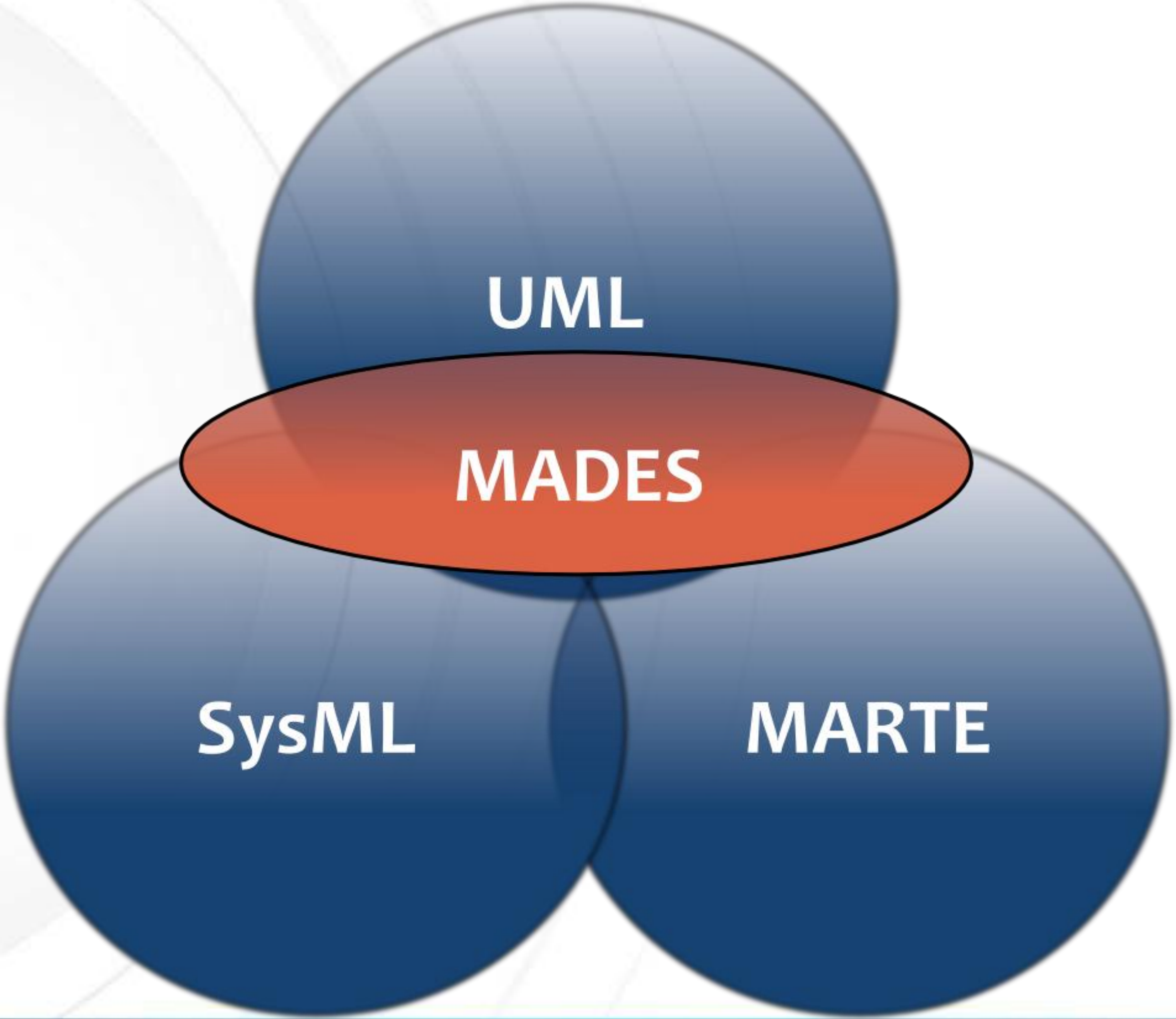
- System Specification
  - Requirements modeling
  - Functional system design
- System Co-Design
  - Software design
  - Hardware design
  - Allocation
  - Timing and Scheduling
  - Behavior
- Verification, Simulation, Code Generation and Synthesis

Challenges for MADES

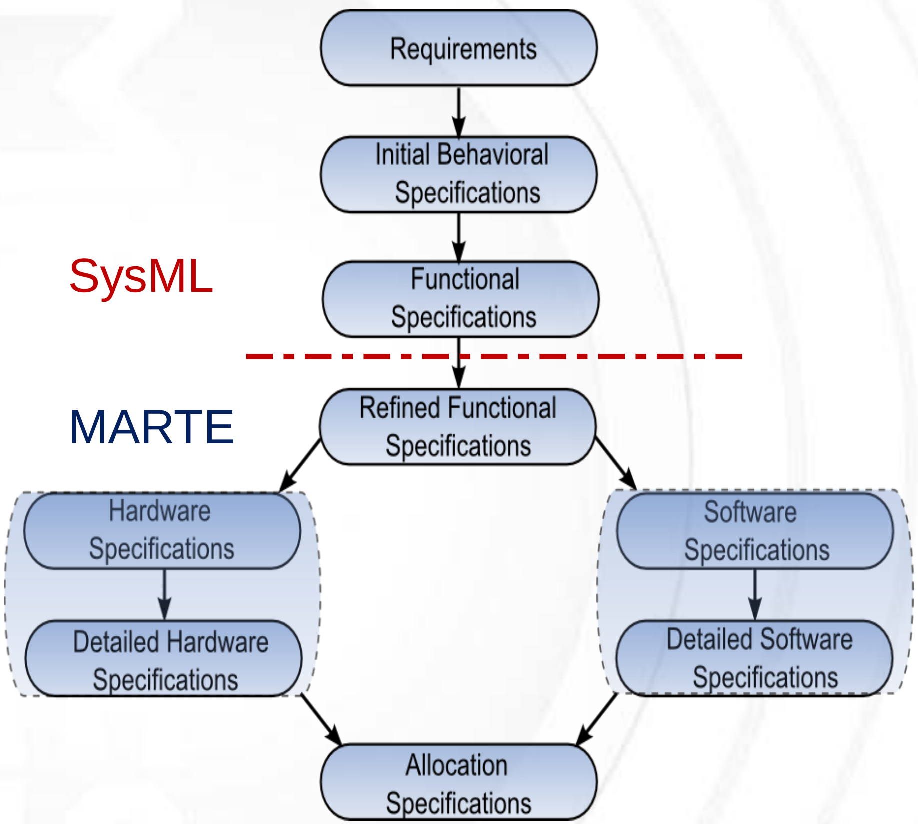
- MARTE : designed to enable flexibility
  - **Drawback:** Large number of concepts (700+ pages of specs)
  - Same concept can be applied to different UML elements (classes, instances, connectors, ports, attributes)
  - Same design may be modeled in different ways using MARTE concepts from different 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

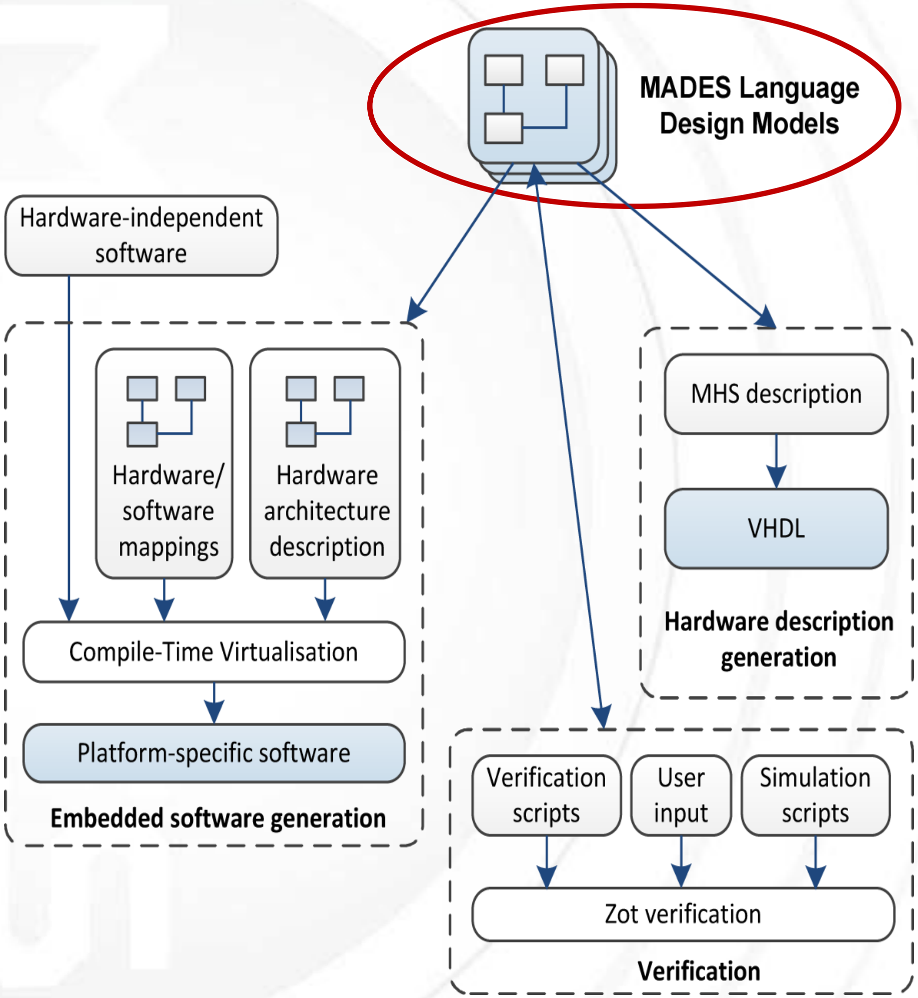


MADES Methodology



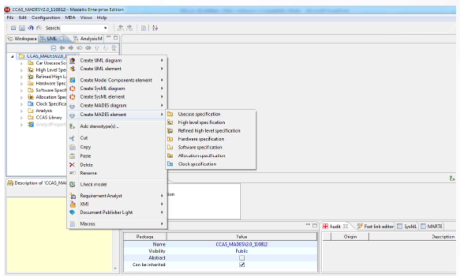
- Effective SysML/MARTE subset
  - SysML for functional specifications
  - MARTE for non functional and co-design specifications
  - UML behavioral specifications supported
- Integration of Verification and Validation (V&V) + code generation concepts
- Dedicated diagrams
- Influence on future revisions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

MADES Implementation Approach

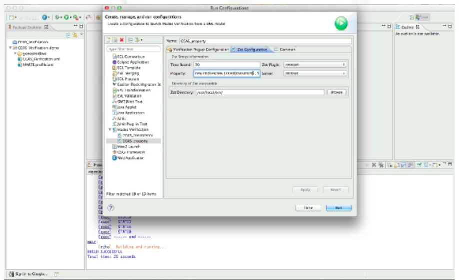


- Generic MADES methodology, guidelines and examples to guide system designers:
  - Modeling, Verification
  - Code generation, Synthesis
- Reducing ambiguities
- Reducing design time and costs
- Reinforce formality for Validation and Verification

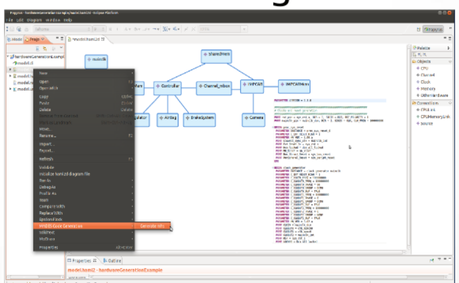
MADES Implementation Approach



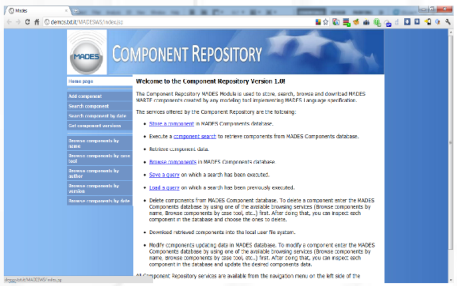
Modeling



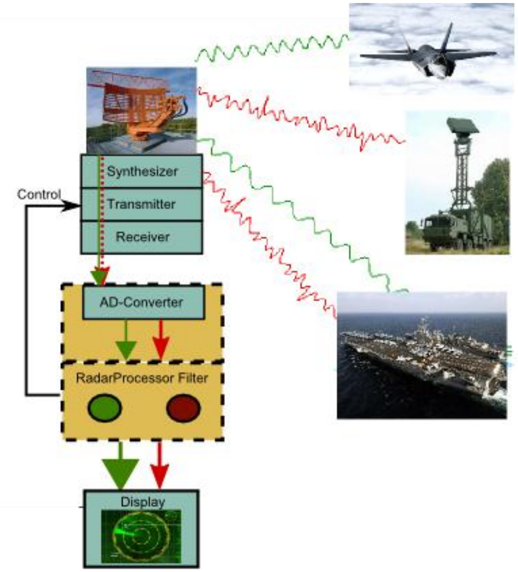
Verification



Code Generation

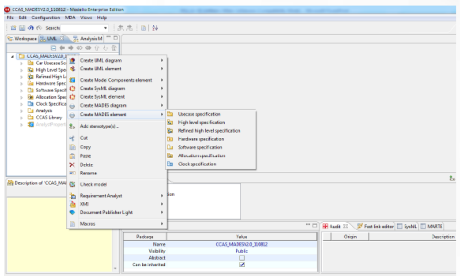


Component Repository

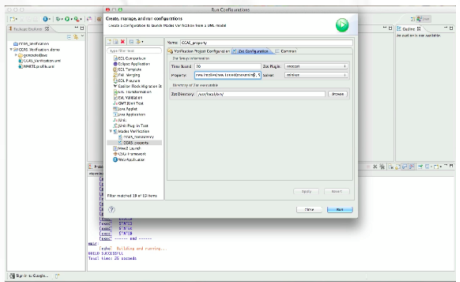


A ground based radar processing unit provided by Cassidian

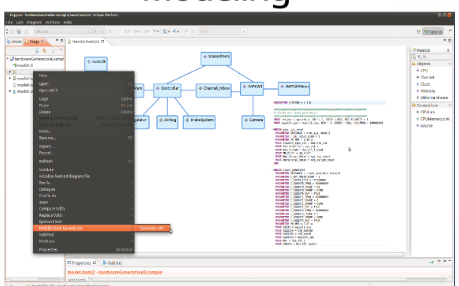
MADES Implementation Approach



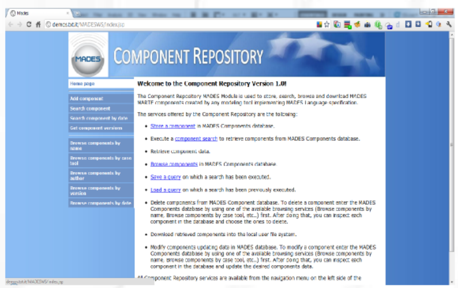
Modeling



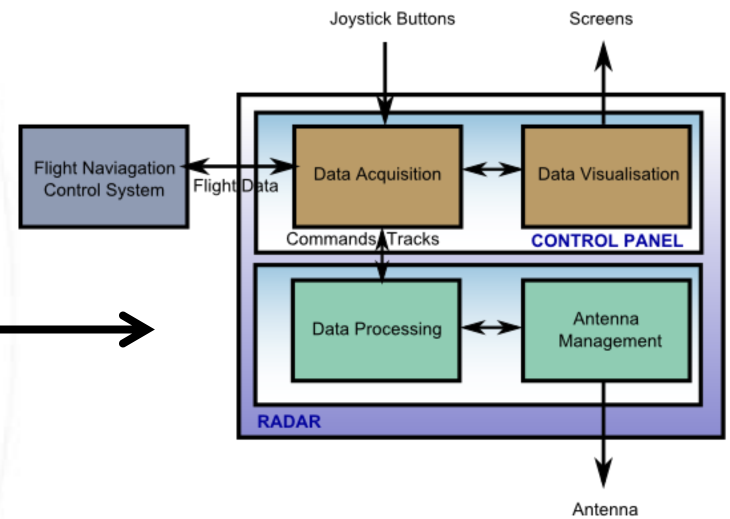
Verification



Code Generation



Component Repository



An onboard radar control unit provided by TXT

MADES Diagrams

Requirements Specification

Requirements diagram
SysML based requirements for initial system functional requirements

Initial Behavior Specification

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

Functional Specification

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

MADES Diagrams

Refined Functional Specification

<u>Refined Functional Specification diagram</u>
Refinement of SysML concepts into MARTE aspects, addition of non functional properties

Hardware Specification

<u>Hardware Specification diagram</u>
Description of generic hardware : nodes, memories, communication channels, clocks etc

Software Specification

<u>Software Specification diagram</u>
Description of application tasks and software aspects running on hardware platform

MADES Diagrams

Detailed Hardware Specification

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

Detailed Software Specification

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

Allocation Specification

Allocation Specification diagram
Mapping of software and hardware aspects of the system

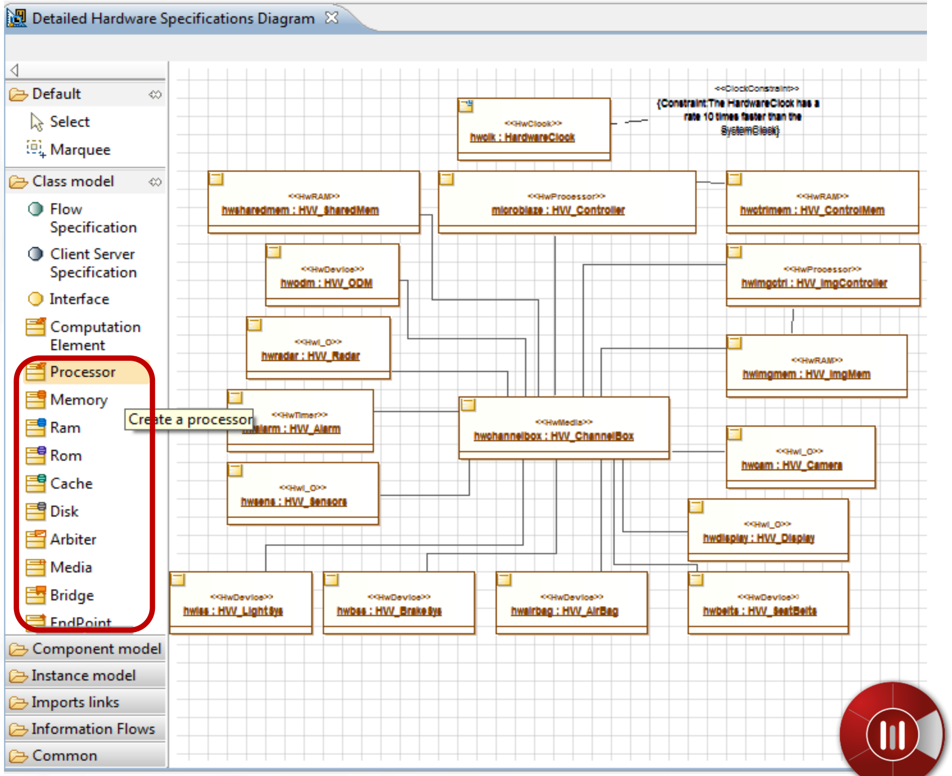
MADES Diagrams

Clock Specification

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

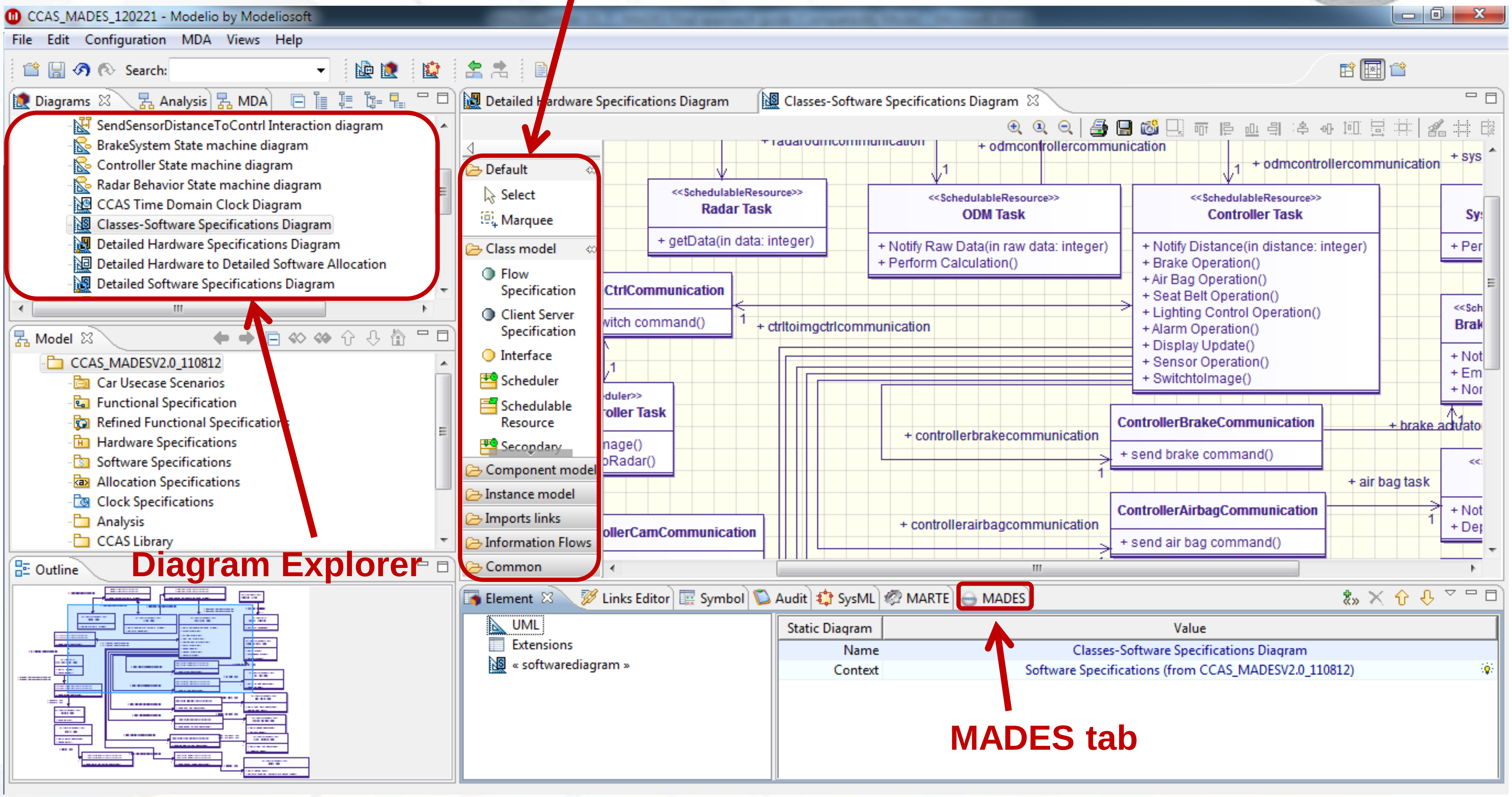
www.modelio.org

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

www.modeliosoft.com

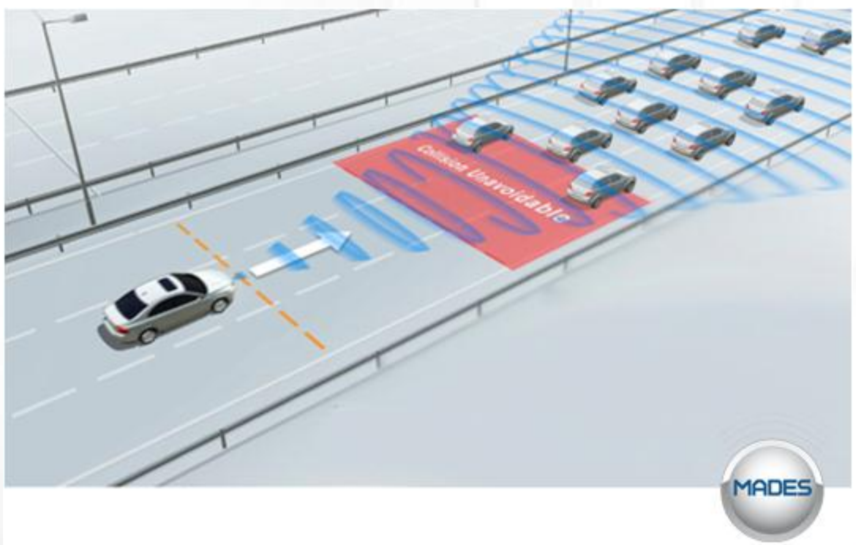
Modelio (Screenshot)

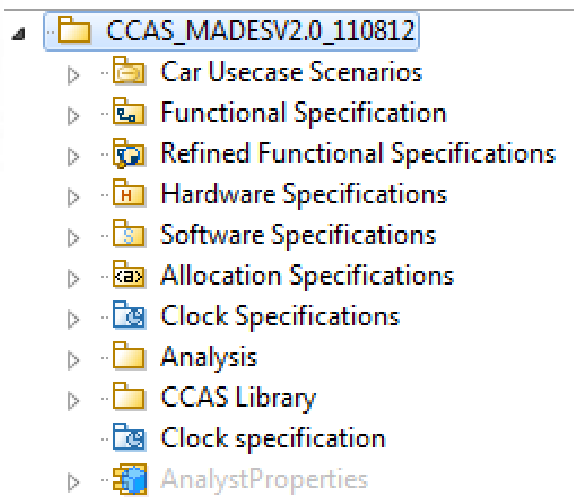
Dedicated commands in Diagram Palette



Car Collision Avoidance System

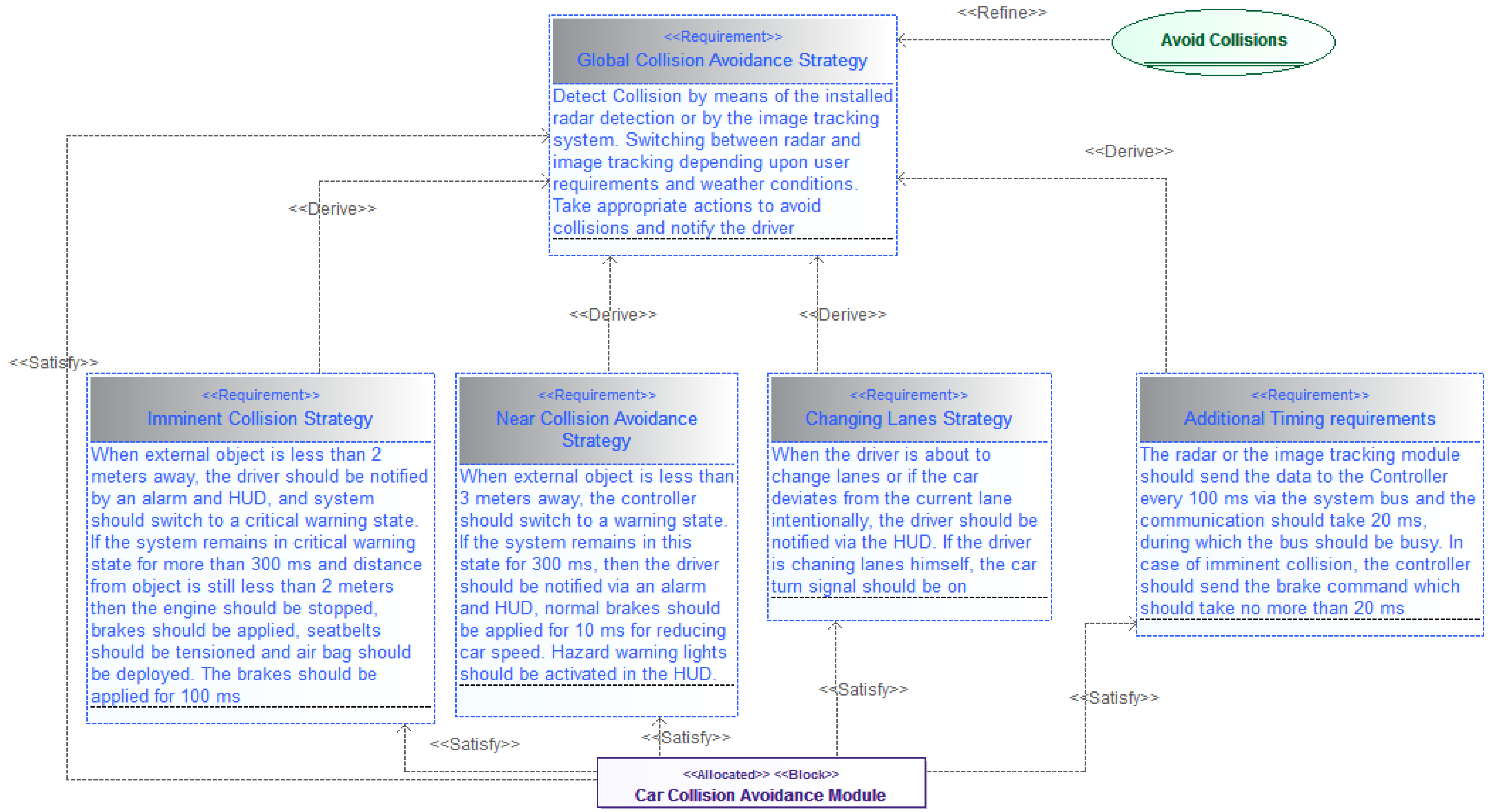
- A system able to detect and prevent collisions
  - Either using a radar tracking module
  - Or an image processing module
  - Initial version presented at OMG Technical Meeting at Arlington, VA – USA (March 2011)





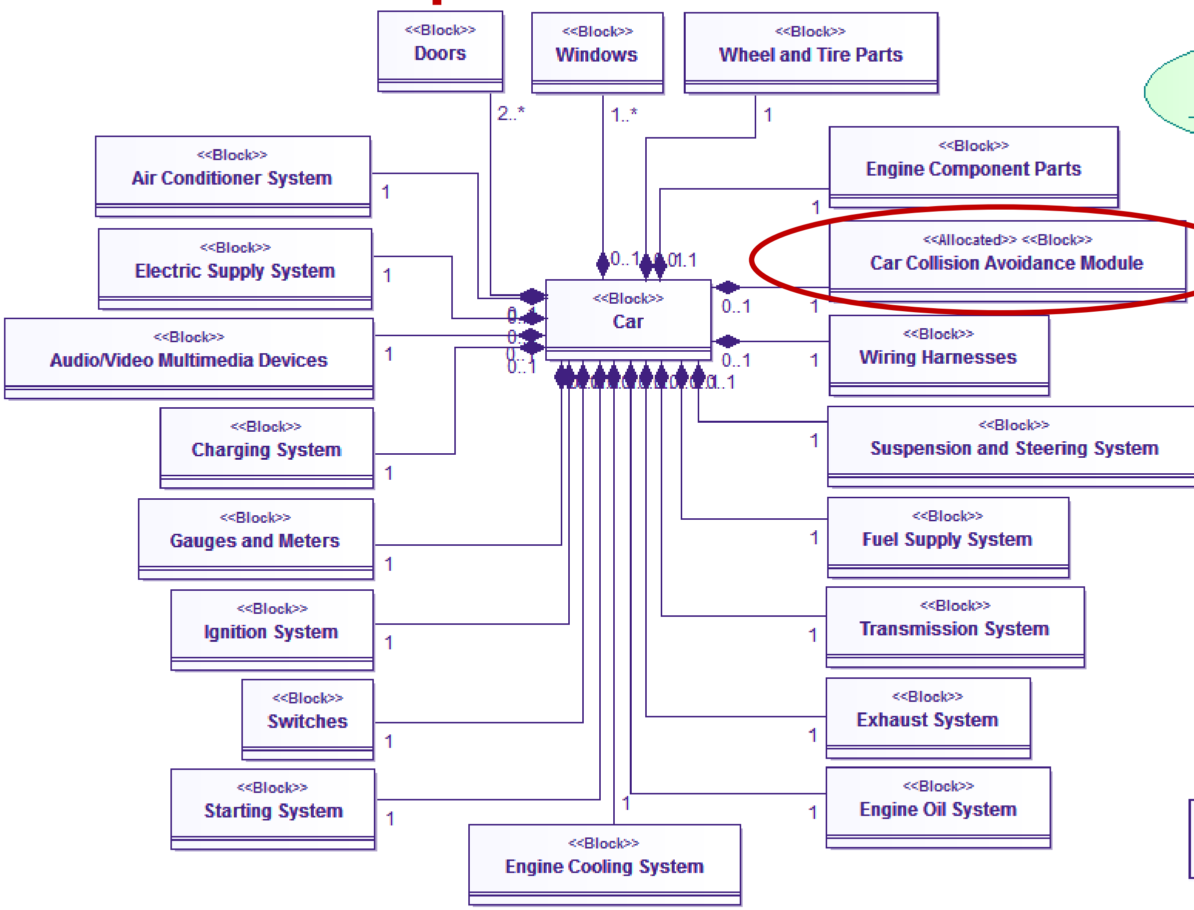
Extract of CCAS Specifications

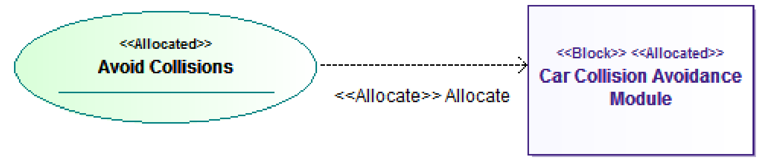
System Requirements



Extract of CCAS Specifications

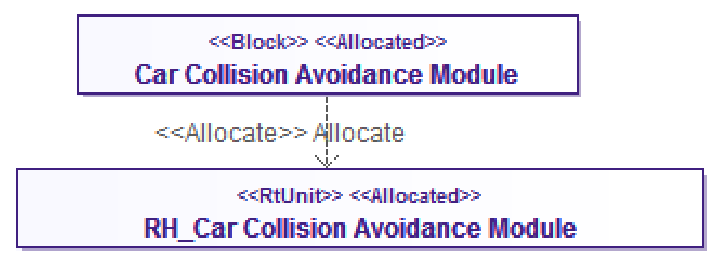
Functional Specifications



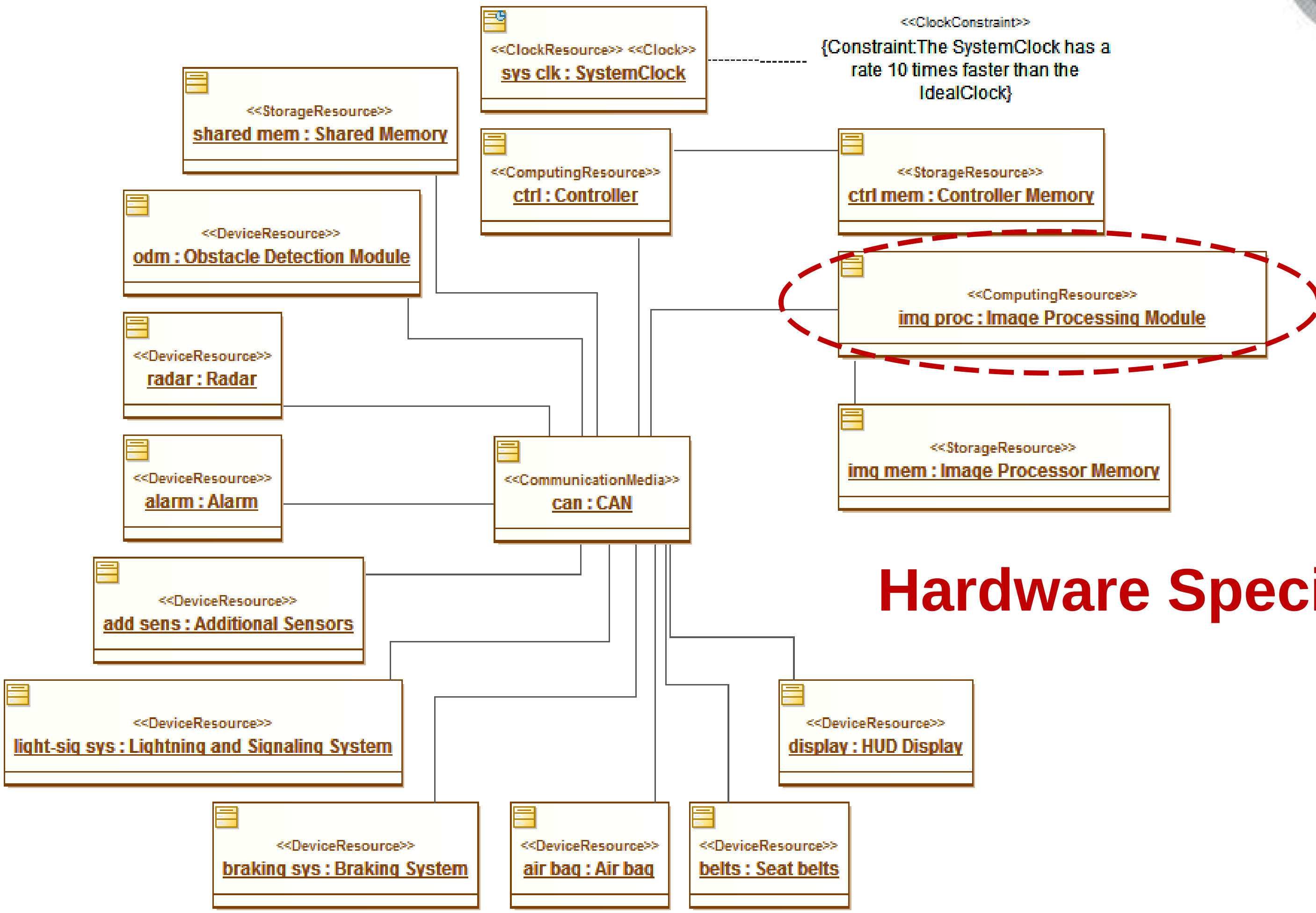


Mapping

Refined Functional Specifications

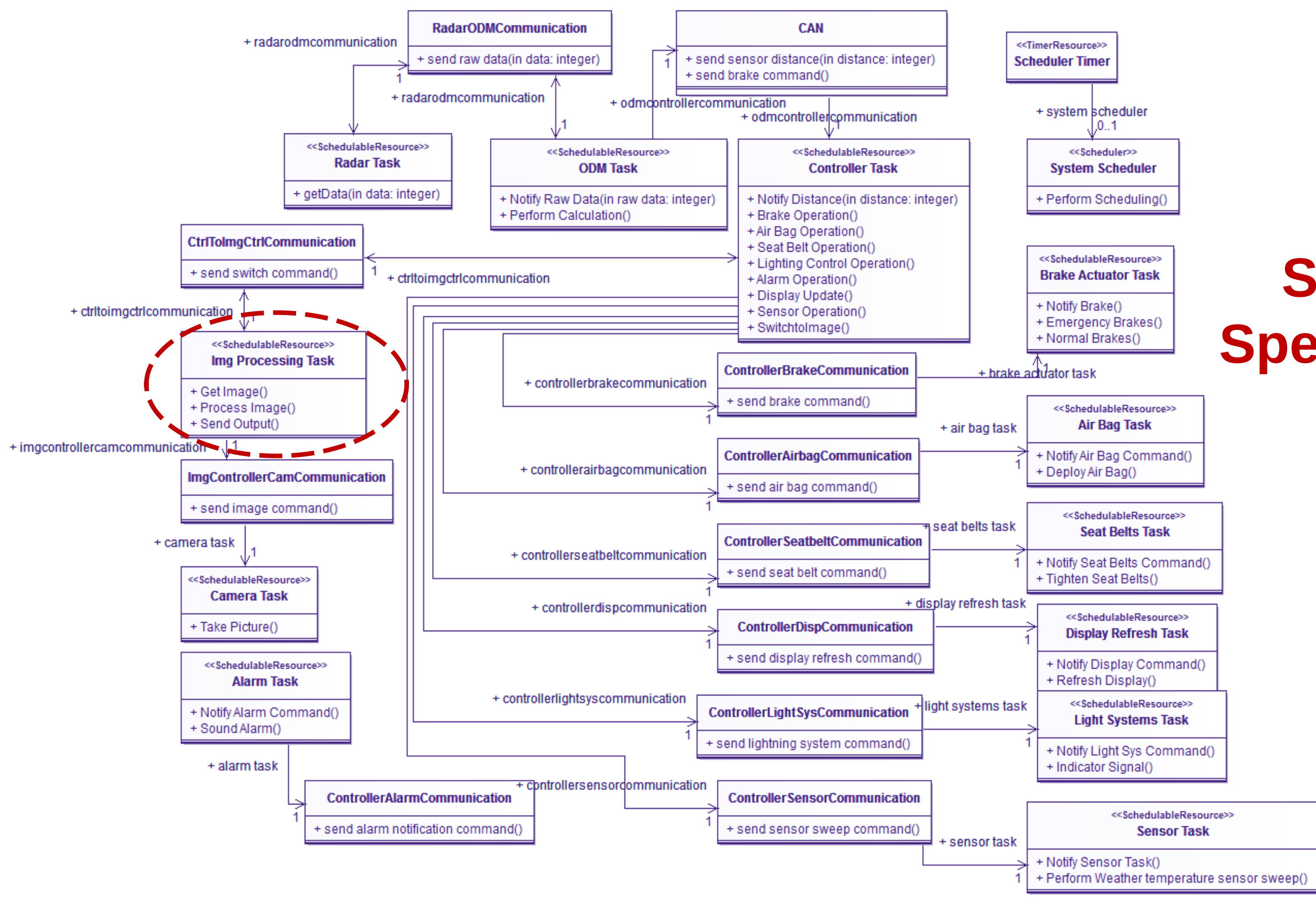


Extract of CCAS Specifications



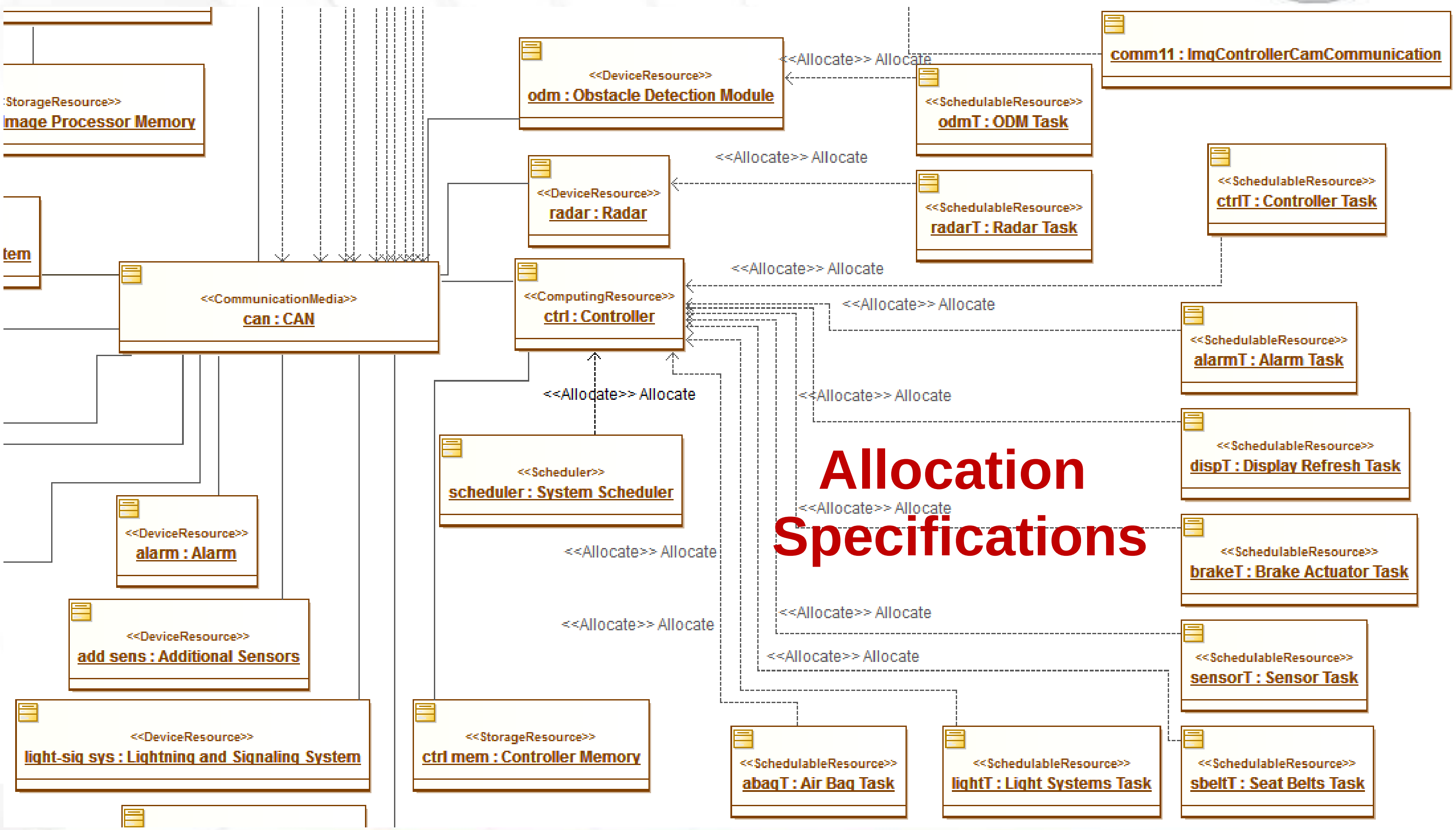
Hardware Specifications

Extract of CCAS Specifications

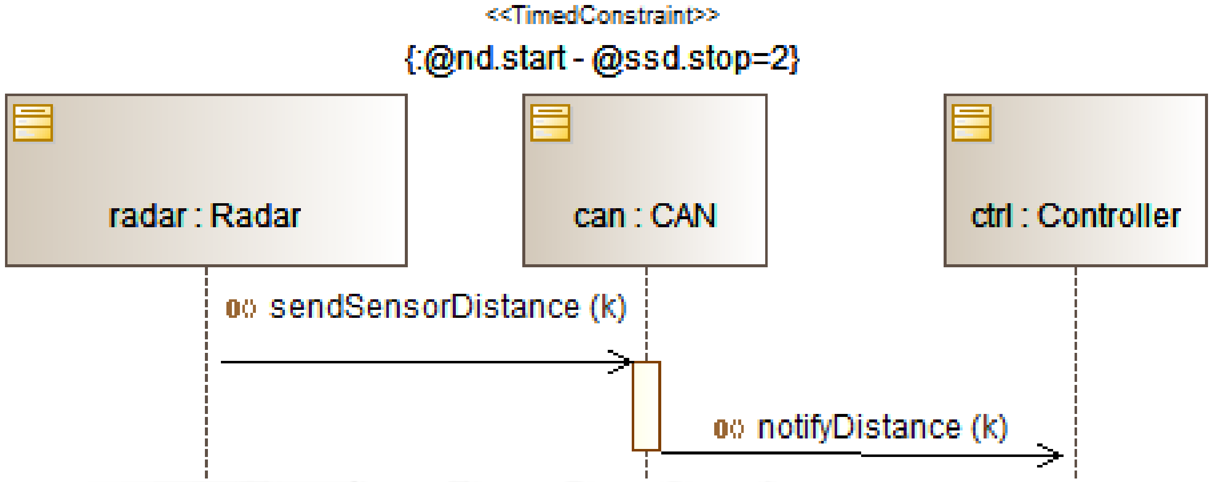


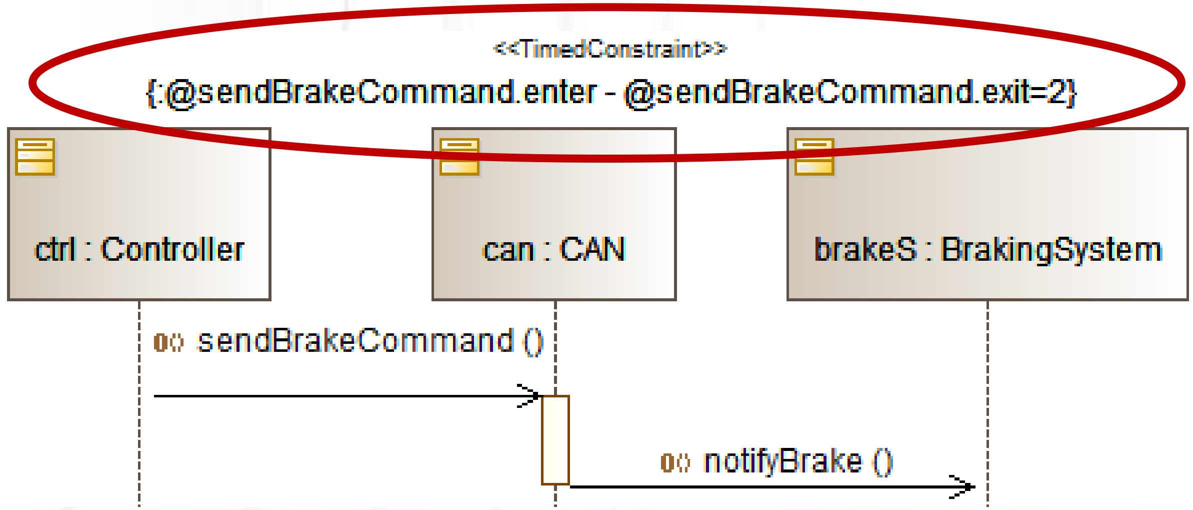
Software Specifications

Extract of CCAS Specifications



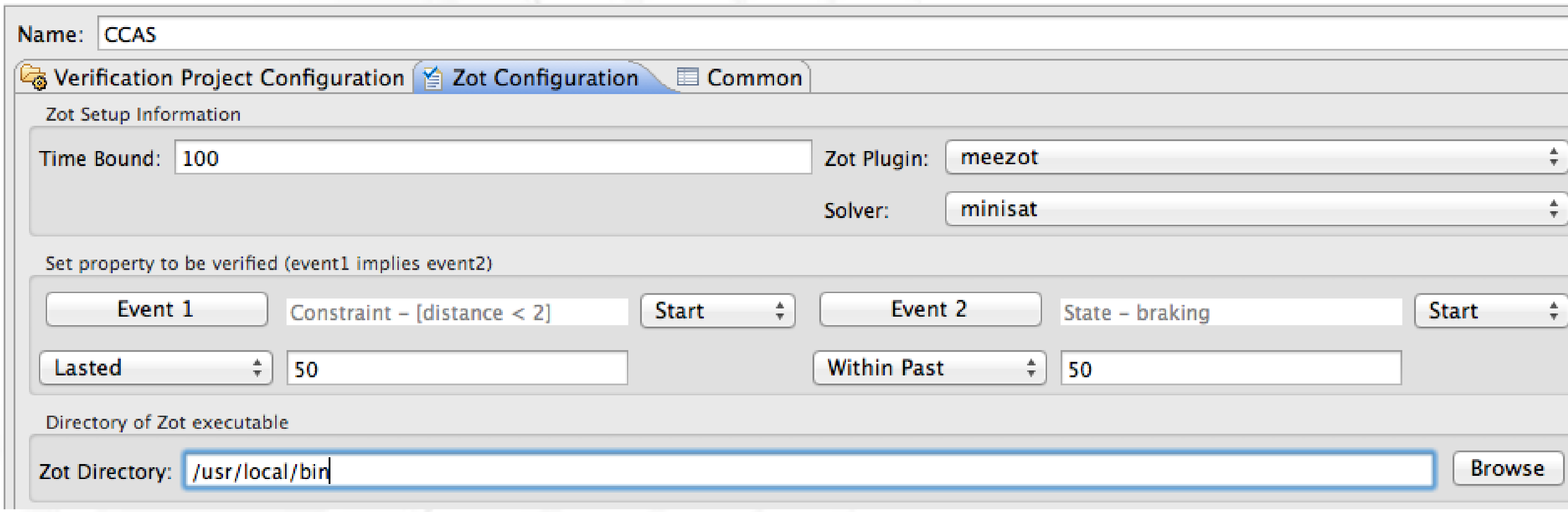
MADES: Verification and Validation





MADES: Verification and Validation

Temporal Logic: Example of temporal properties



MADES: Verification and Validation

Problems | Javadoc | Declaration | Console

```
GenCCAS build.xml [Ant Build] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Mar 2
     [exec]   37.462327 seconds of total run time (36.888518 user, 0.573809 system)
     [exec]   [ Run times consist of 0.605 seconds GC time, and 36.858 seconds non-GC time. ]
     [exec]   81.92% CPU
     [exec]   97,130,708,872 processor cycles
     [exec]   334,639,824 bytes consed
     [exec]
     [exec] This was zot2cnf. Solver time coming next.
     [exec] WARNING! DIMACS header mismatch: wrong number of variables.
     [exec] ============================[ Problem Statistics ]=============================
     [exec] |                                                                             |
     [exec] |  Number of variables:        846304                                         |
     [exec] |  Number of clauses:         2562352                                         |
     [exec] |  Parse time:                   1.30 s                                       |
     [exec] |                                                                             |
     [exec] ============================[ Search Statistics ]==============================
     [exec] | Conflicts |          ORIGINAL         |          LEARNT          | Progress |
     [exec] |           |    Vars  Clauses Literals |    Limit  Clauses Lit/Cl |          |
     [exec] ===============================================================================
     [exec] |       100 |   333463   921920  2306926 |   377413       92      13 | 60.598 % |
     [exec] |       250 |   265106   744355  1884535 |   415154      165      12 | 68.675 % |
     [exec] ===============================================================================
     [exec] restarts              : 3
     [exec] conflicts             : 331            (65 /sec)
     [exec] decisions             : 3498           (0.00 % random) (691 /sec)
     [exec] propagations          : 11872880       (2344689 /sec)
     [exec] conflict literals     : 3454           (17.60 % deleted)
     [exec] Memory used           : 156.06 MB
     [exec] CPU time              : 5.06373 s
     [exec] SATISFIABLE
     [exec] Evaluation took:
     [exec]   10.876 seconds of real time
     [exec]   0.005957 seconds of total run time (0.001064 user, 0.004893 system)
     [exec]   0.06% CPU
     [exec]   23,088,525,848 processor cycles
     [exec]   32,672 bytes consed
```

Verification Results

MADES: CCAS Subset

- 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

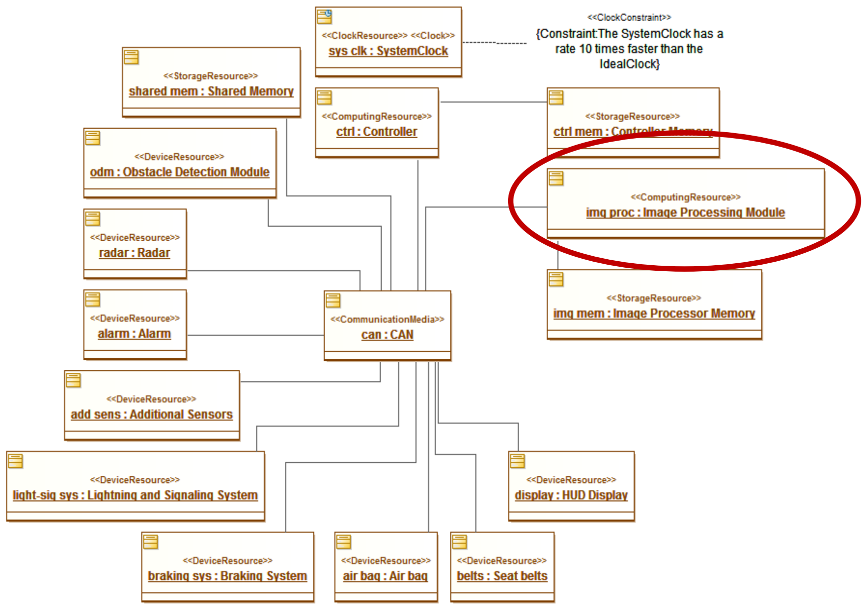
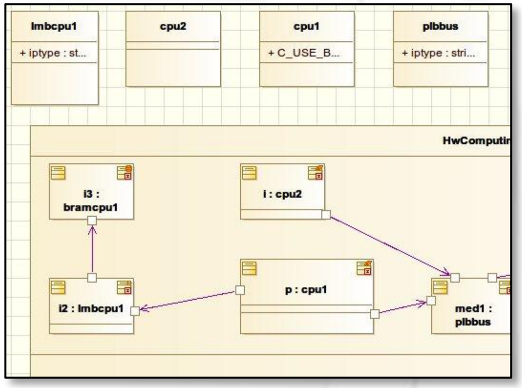


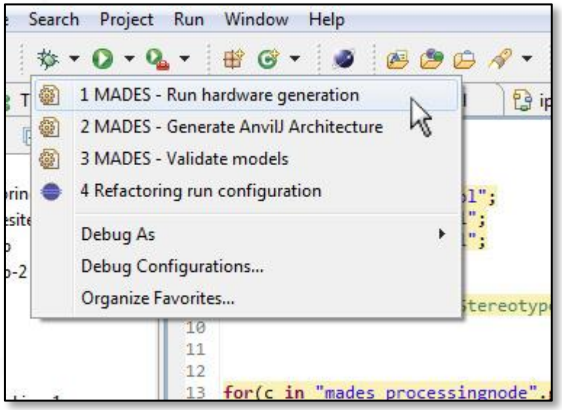
Image Processing module in CCAS

MADES: CCAS Hardware Generation

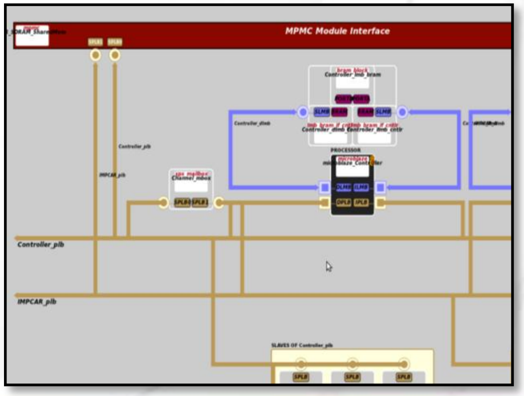
- Automatic generation of implementation hardware from MADES design models



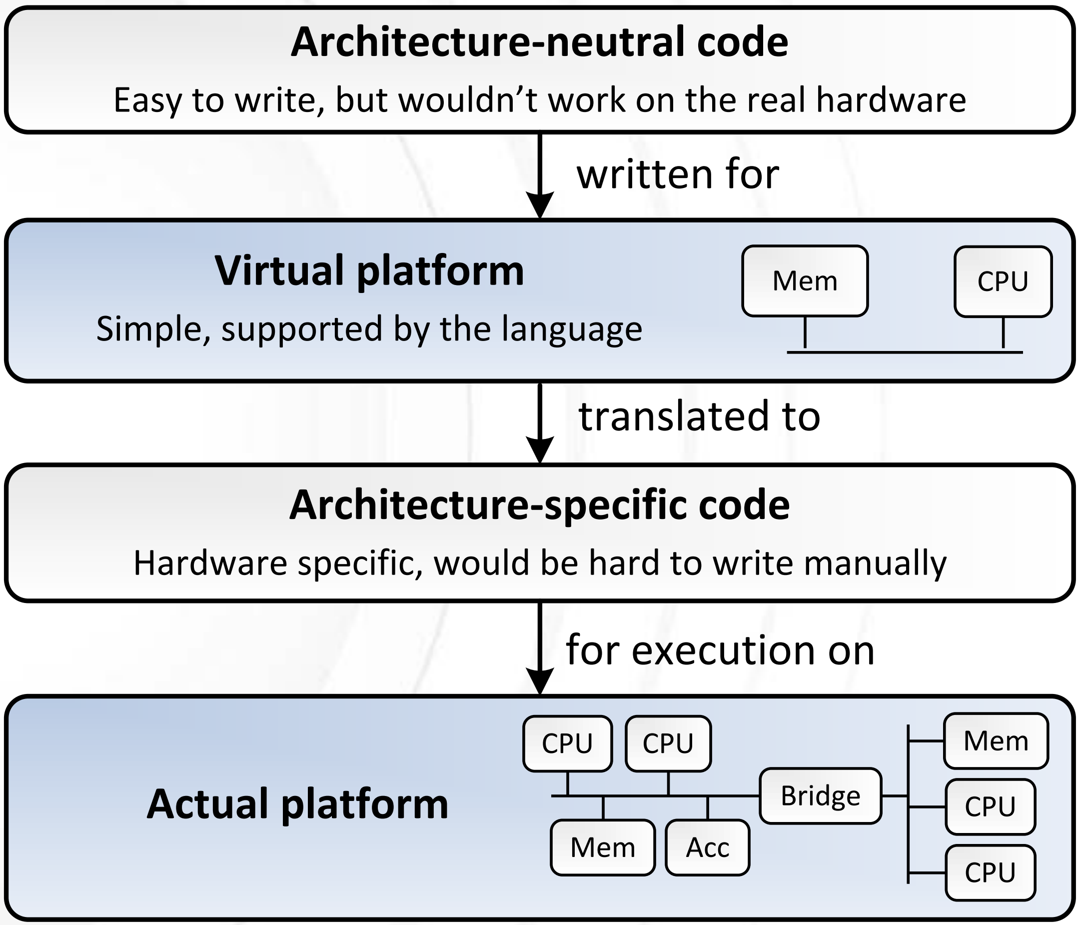
XMI



MHS



MADES: CCAS Software Generation

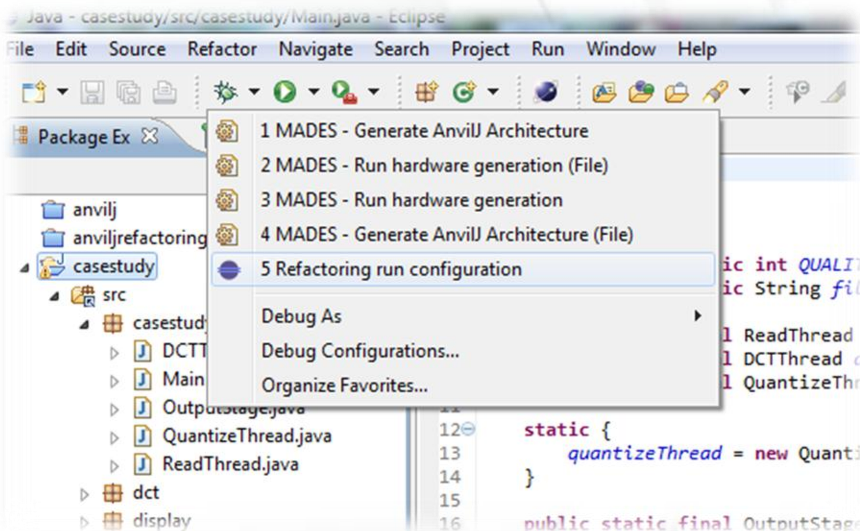


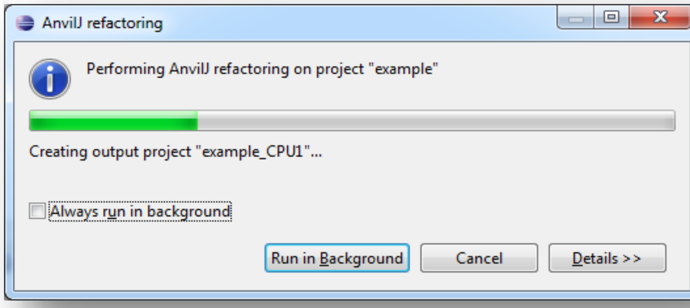
C / Java / etc. Assume this simple architecture

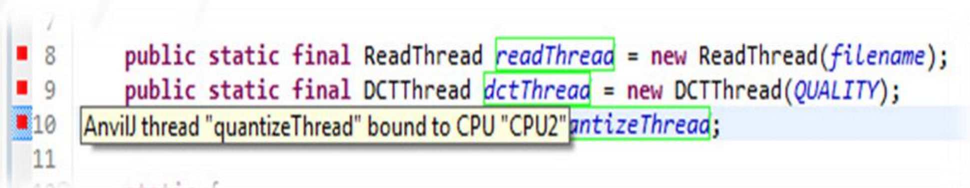
Many different actual platforms may be targeted

MADES: CCAS Software Generation

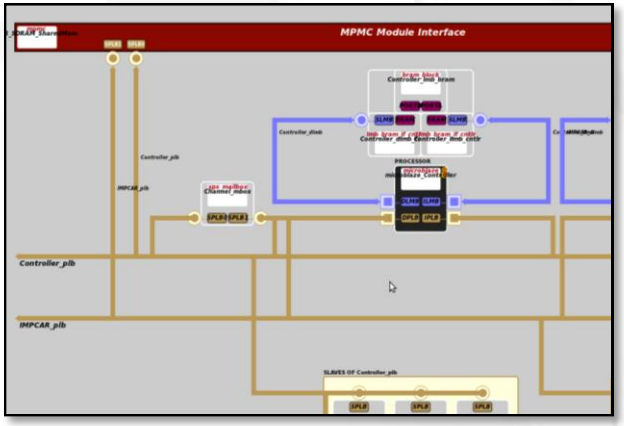
- Integrated into the Eclipse development environment



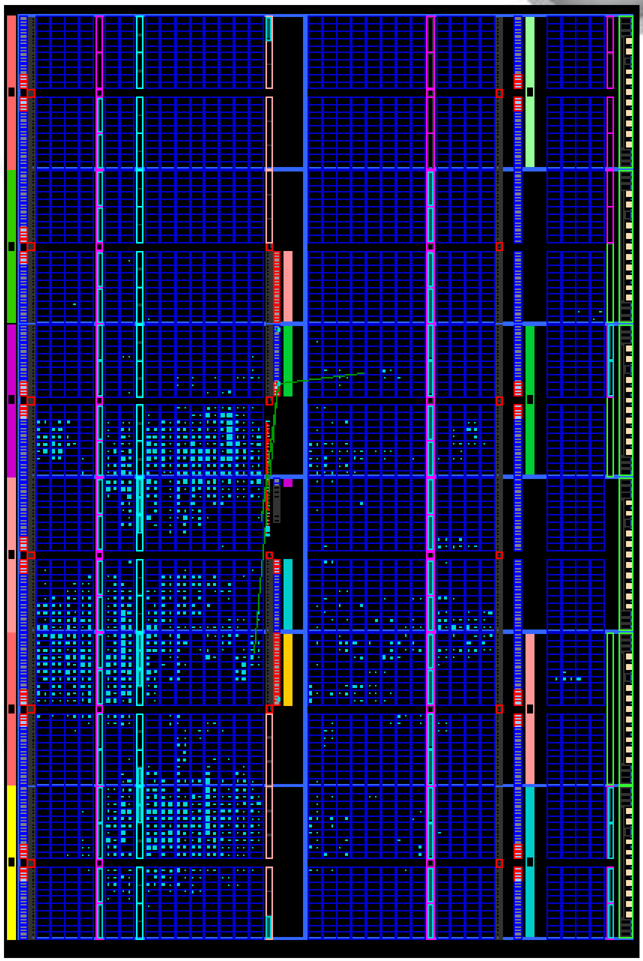




MADES: Synthesis on Execution Platform



Synthesis
Place & Route



Programming

END USER FEED BACKS

TXT Case Study

- 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

TXT Case Study

Modeling of RTES using MARTE and SysML through MADES Language subset,

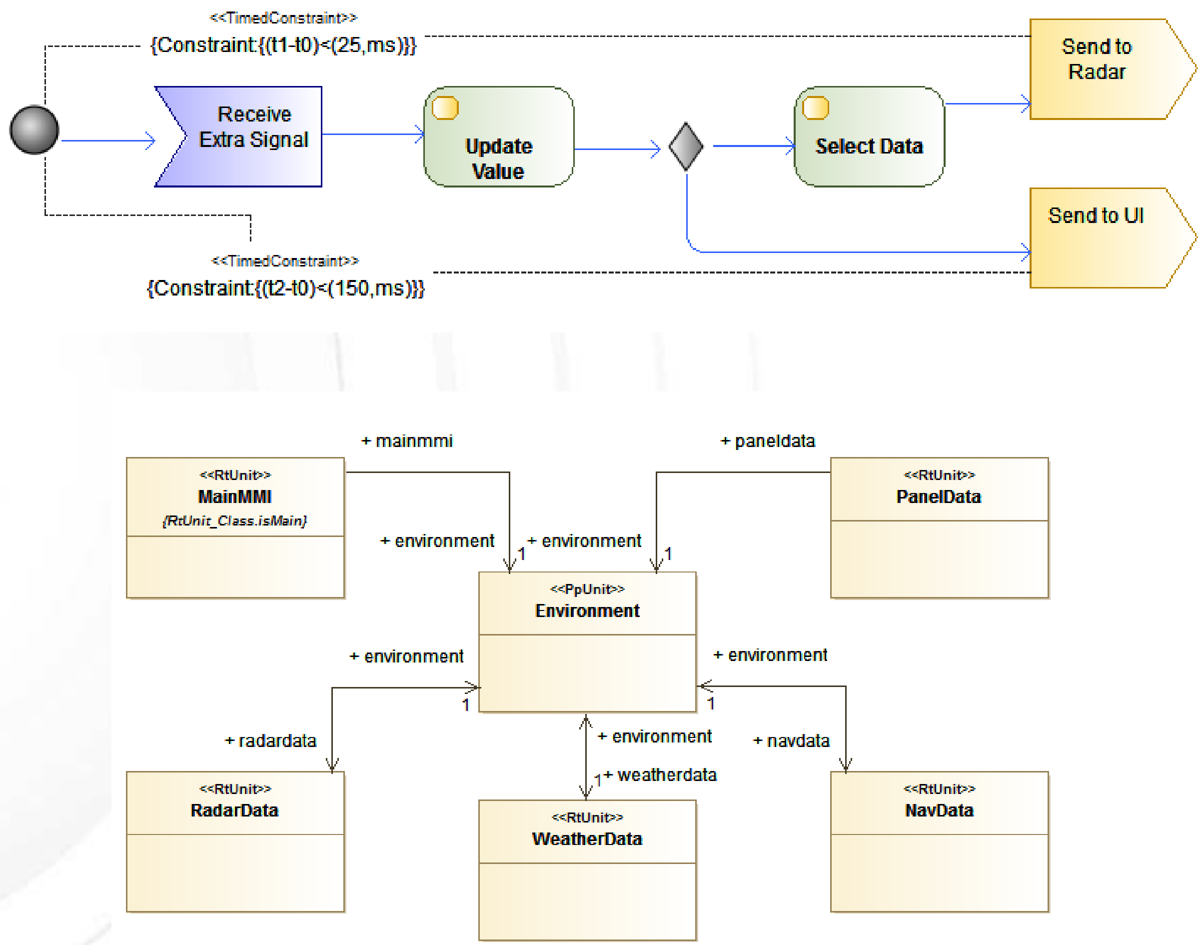
Component modeling and storage features,

Non-functional properties specification and verification,

Usage of a specific set of unique diagrams for expressing different aspects related to a system,

First experimentations with MADES Integrated environment

TXT Case Study Extract



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.