MARTE Based Modeling Tools
Usage Scenarios in Avionics Software Development Workflows

Alessandra Bagnato, Stefano Genolini
Txt e-solutions
FMCO 2010, Graz, 29 November 2010
Overview

• MADES Project and MADES Modeling Tools
• TXT avionic software development
  – Tools
  – Software Development Workflow phases
  – Modeling Tools
• MADES Methodology in avionic development in TXT
  – MADES Methodology phases in TXT Avionic software development workflow
  – MADES Modeling Tools
  – MADES new diagrams
• Benefits and future work
MADES Project

• MADES is developing new model-driven technologies to improve current practices in the development of embedded systems for the avionics and surveillance embedded systems industries
  – Integrated design, validation, simulation, and code generation environment
  – Better support for component reuse
MADES Modeling Tools

- Modelling tools **Modelio** from Softeam and **Open Source Modeler** from University of York will provide specific support for new MADES modelling annotations.

- A web-based **MADES Component Repository** enabling greater component reuse will contain the full knowledge gathered about components during each development phase.
SDLC @ TXT Avionic Unit

Current status

Typical Tools:
- DOORS
- Artisal RT Studio, Modelio, ...
- HP Quality center, Cantata, Vector cast
Tools @ TXT Avionic Unit

**Analysis**
- Doors
- MatLab/Simulink

**Design**
- Artisan RealTime Studio

**Coding**
- Green Hills AdaMulti - Ada95GNAT
- Artisan RealTime Studio

**Testing**
- HP Quality Center
- Vector Cast Cover / Cantata

**Configuration and Traceability**
- HP Quality Center - Source Safe
- Serena Dimensions

**Other/Planning**
- MS Project
- Proprietary Metric Toolset
Improved SDLC @ TXT

Requirement analysis and design phase enhancement

Phases not taken into account by this presentation are in grey
Modelling Tool usage @ TXT Avionic Unit

- **Requirements** are provided by Avionic system application experts, modeled in natural language and Simulink.

- **Model requirements in UML with SysML and use cases**
  - Define derived requirements
  - Define system decomposition in parts
  - Requirements drive the development process: Traceability

- **Modeling of the system in UML**
  - Identification of classes and relationships
MADES General methodology

High-level specification phase describes the functional and behavioural aspect of the systems. It uses diagram with the higher abstraction.

- Allocation and Partitioning phase describe the task allocation and details the hardware architecture. Hardware, Software and Allocation diagrams are the most relevant.

Synthesis phase the models defined with Hardware, Software and Allocation diagrams will be used to generate the code.

Scheduling phase specify scheduling aspect, refines previously created diagrams detailing specific elements like MARTE time element.
<table>
<thead>
<tr>
<th>MADES Diagram</th>
<th>Purpose</th>
<th>TXT workflow phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>Based on the SysML requirement diagram, exposes system requirements and the relations existing between them or to other model elements.</td>
<td>Requirement Phase – derivative requirements</td>
</tr>
<tr>
<td>diagram</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High-Level</td>
<td>High-Level Structure diagram, based on the UML composite structure diagram, is created in order to depict the system structure in terms of high-level component.</td>
<td>Design Phase – structure decomposition</td>
</tr>
<tr>
<td>Structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>diagram</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td>Hardware diagram based on the UML composite structure diagram, will be used for exposing system hardware part concepts.</td>
<td>Design Phase – structure decomposition</td>
</tr>
<tr>
<td>diagram,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td>Software diagram is the counterpart of the Hardware diagram and is dedicated to highlighting software part concepts.</td>
<td>Design Phase– structure decomposition</td>
</tr>
<tr>
<td>diagram</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allocation</td>
<td>Allocation diagram allows users to map functional elements onto the available resources.</td>
<td>Design Phase- space allocation</td>
</tr>
<tr>
<td>diagram</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time diagram</td>
<td>Time diagram will be used in order to model the time environment (creation of Clock, ClockConstraint, etc.) necessary for specify real-time aspect.</td>
<td>Design Phase – time allocation</td>
</tr>
</tbody>
</table>
Introduction of MADES phases in avionic development in TXT

- High-level specification phase
- Allocation and Partitioning phase,
- Scheduling phase
- Synthesis phase
Introduction of MADES phases in avionic development in TXT

• During requirements modeling:
  – Capture of the functionality
    • Functional requirement: what the system shall do (input, behavior, output)
    • Non functional requirements: impose constraints on the design (performance, security, timing, power consumption, heat dissipation, reliability, etc)

• During Design Phase
  – Definition of HW and SW architectures including
    • Definition of the spatial allocation (where execute)
    • Definition of temporal allocation (when execute)
  – Definition of scheduling policies supporting several timing mechanisms to satisfy the time constraints imposed by non functional requirements
Introduction of MADES phases in avionic development in TXT

- High-level specification phase
- Allocation and Partitioning phase,
- Scheduling phase
- Synthesis phase
MADES Modeling Tools

- Modeler
  - Softeam Modelio
  - University of York Open Source Modeler

- Web based MADES Component Repository
MADES Modeling Tools in TXT Development

- Non functional requirements:
  - Timing:
    - Functions shall be scheduled on time (periodically)
    - Functions shall terminate within a given period
    - Reaction time on events within given amount of time

- Component Library
  - Reusable components
    - Design diagrams
      - Its reusability depends on component complexity and size
    - Allocation and timing are specific of the application, only attributes and constraints are stored, diagrams are to be used as training set
The case study is related to the User Interface development.
MADES Case study: Main features

Antenna and management system:
- Convoy emitted power on the target,
- Follow aircraft movements (yaw, pitch, roll).

Data processor:
- Filter and transform analogical data,
- Digitalize and send map information,
- Receive and forward commands to antenna.

I/O manager:
- Share information with FMS
- Share information on AHRS

User interface module:
- Acquire map information
- Analyze data
- Display map data on Display
- Send command to antenna

MADES PROJECT – FP7 248864
MADES Case study: Radar equipment
MADES Case study: Target Platform

UI: Hardware Intel Proc
Windows XP Embedded
.NET Framework
C++ I/O Manager
C# User Interface

The whole system is not critical in term of flying, the radar system is developed according to the DO178B civil avionic standard level D.
Modeling of time behavior

Communication between board and antenna 100ms

- Command request: 100ms
- Status answer: 100ms
- Map information: 1s

Communications between board and FMS/AHRS:

- Greenwich Mean Time: 1s
- Radar altitude: 100ms
- Barometric altitude: 100ms
- True air speed: 100ms
- Static air temperature: 100ms
- Date: 1s
- Waypoint/Station Type: 100ms
- Nav/Waypoint/Airport Name (1&2): 100ms
- Nav/Waypoint/Airport Latitude: 100ms
- Nav/Waypoint/Airport Longitude: 100ms
- Point Latitude: 100ms
- Point Longitude: 100ms
- Ground speed: 100ms
- True heading: 20ms
- Magnetic heading: 20ms
- Pitch angle: 10ms
- Roll angle: 10ms
- Acceleration: 20ms
Time model

Time constrain:

\[ T_{mf} \geq \sum_i (T_i + T_{cc}) + T_{rtos} \]

\[ T_{Mf} = N \cdot T_{mf} \]
Time allocation

Process barometric altitude each 100ms

Process pitch and roll each 10ms

Process true heading each 20ms
MADES Component Repository

Components, from MARTE Time Subprofile

TXT-ClockType1
- resolution: Real [1] = 0.001
- offset: Real [1]
- Property_0: <Undefined> [1]
- timeunitkind: TimeUnitKind [1] = ms
- eventkind: EventKind [1] = consume
- name: String [1]

currentTime: Real()
getTime: Real()
setTime(in Real)

MyClock-Instance : IdealClock, NF...
- resolution = 1
- duration = 20
- clock = MyClock
- unit = ms

MARTE Components
e.g. TXTClockType1 with Coordinated Universal Time (UTC), clock instance of type ClockType1
To be considered as training set
The MADES Modeling Formalism benefits

- **MADES Methodology potential benefits**
  - To be evaluated with presented case study
  - Means to trace elements that are part of models which reside at different levels of abstraction
  - Support to dedicated view, allowing developers to separate concerns and to focus on one specific aspect of the system.
  - Support all modeling phases with a unique environment able to generate complete documentation
  - Support reuse of previously built components
Benefit from complete MADES Tool set

- **Traceable:** Each step has to be traced on the previous and the next.
- **Configurable:** Each change should be explained, traced and be connected to a specific version of the software.
- **Reusable:** The experiences matured on a specific baseline should be ready for the next.
- **Coordinate:** Each tool has to help coordination of large teams and allows concurrent access to information.
- **Unique:** As much functionality as possible should be reliable in the same software tool to offer synergic result.
Thanks!

Alessandra Bagnato  
Corporate Research Division - TXT e-solutions S.p.A  
alessandra.bagnato@txt.it

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

TXT Web Site:  
http://www.txt.it/

TXT A&D Web Site:  

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.