UML/MARTE modelling for Mixed-Criticality Systems

Fernando Herrera
University of Cantabria

Funded by the EC under Grant Agreement 611146
System components and their associated requirements have different “importance”.

Modelling and Design has to be Mixed-Criticality aware!
3 Modelling Methodology: Relevant Characteristics

- Single Source
- Relying on Standards
- Separation-of-Concerns
- Incremental development
- Component-based
- Support of system-level design activities: DSE, SLS
4 Mixed-Criticality

- Criticality: annotation that can be associated to
  - Application (PIM) Components
  - Platform Resources
  - Extra-Functional Requirements
  - Value annotations

- Generic concept and flexible interpretation that enables adapting the methodology to different domains

<table>
<thead>
<tr>
<th>(CONTREX) Criticality</th>
<th>IEC 61508 SIL</th>
<th>EASA DAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>SIL4</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>SIL3</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>SIL2</td>
<td>C</td>
</tr>
<tr>
<td>1</td>
<td>SIL1</td>
<td>D</td>
</tr>
<tr>
<td>0</td>
<td>SIL0</td>
<td>E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(CONTREX) Criticality</th>
<th>ISO2626 ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xD</td>
<td>D</td>
</tr>
<tr>
<td>0xC</td>
<td>C</td>
</tr>
<tr>
<td>0xB</td>
<td>B</td>
</tr>
<tr>
<td>0xA</td>
<td>A</td>
</tr>
</tbody>
</table>
Proposed minor MARTE extension

Enables two basic modelling techniques:
- Criticality constraint associated to modelling element
- Criticality associated to value
6 Associating criticalities to Modelling Elements (in PIM)

+ flight_alg : FlightAlgorithmC
  structure

«NfpConstraint_Contrex»
criticality=[3]
safety_critical

{ }

+ datamining : DataMiningC
  structure

+ telemetry : RCTelemetryC
  structure
7 Associating criticalities to Modelling Elements (in HW resources)

+ cpu1 : ARM_Cortex_A9
  structure

+ cpu2 : ARM_Cortex_A9
  structure

«nfpConstraint_Contrex»
«NfpConstraint_Contrex»
criticality=[2]
mission_critical_resources
{}
8 Associating criticalities to Performance Requirements

- VSL expression with criticality value:
- \( \text{relDL} = \{2,\text{ms}, \text{criticality}=3\} \)
NFP constraint with

- **criticality annotation**
- `<<Expression Context>>`: performance requirement

```plaintext
«Component»
quadcopter_system
structure

«nfpConstraint_Contrex»
«expressionContext»

«nfpConstraint_Contrex»
criticality=[3]

Power
{out$cpu1.power(W,est)+out$cpu2.power(W,est)+out$cpu3.power(W,est) +out$cpu4.power(W,est)+out$axi1.power(W,est)+out$axi2.power(W,est) +out$axi3.power(W,est) < 15W}

+ cpu1
structure

+ cpu2
structure

+ cpu3
structure

+ cpu4
structure

axi64 : ...
structure

+ axi1
structure

+ axi3
structure

«nfpConstraint_Contrex, expressionContext»
«NfpConstraint_Contrex»
criticality=[2]

throughput
{out$frame_sending_throughput(Hz,est) >= (30,Hertz)}
```
10  Associating criticalities to EFP annotations
Mixed-Criticality Information can be used along the design flow at different phases, e.g.

- At modelling
- At verification
- At analysis (e.g., schedulability, performance)
- At the design space exploration phase
- At the implementation
## Mixed-Criticality aware modelling rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Criticality Assignation</strong></td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>A criticality shall be assigned to all RtUnit Instances</td>
</tr>
<tr>
<td>R2</td>
<td>A criticality shall be assigned to all PpUnit Instances</td>
</tr>
<tr>
<td>R3</td>
<td>A criticality shall be assigned to all HwProcessors</td>
</tr>
<tr>
<td><strong>Allocation (Segregation of components with different criticalities)</strong></td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>There cannot be several application component instances with different criticalities allocated to the same memory space</td>
</tr>
<tr>
<td>R5</td>
<td>A memory space with the highest criticality level (or a given criticality level threshold) and a less critical memory spaced shall not be allocated to the same RTOS.</td>
</tr>
<tr>
<td>R6</td>
<td>Two or more component instances with different associated criticalities cannot be allocated to the same resource</td>
</tr>
</tbody>
</table>
## Mixed-Criticality aware modelling rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Coherent Mapping</strong></td>
<td></td>
</tr>
<tr>
<td>R7</td>
<td>A PIM component instance of a given criticality shall not be mapped, either directly or indirectly, to a processing resource of a lower criticality</td>
</tr>
<tr>
<td>R8</td>
<td>A component instance of a given criticality cannot be mapped, either directly or indirectly, to a resource of a lower criticality</td>
</tr>
</tbody>
</table>
Validation: Implementation Example

- Model Validation tool
- Mixed-Criticality aware Model Validation
- Identifying and Fixing a criticality-related modelling error
- Identifying more tricky criticality-related modelling errors
- Fixing the criticality-related modelling errors and warnings
16 Validator Implementation

- OML Model-To-Tex (MTL)
  - Queries (OCL): Model navigation and querying
  - Templates: Text generation

- OMG MTL Model-to-Text
  - Standard Description
  - Portable
  - Easy to Maintain and extend

- MTL for Validation:
  - Queries (OCL): Same as the ones for code generation
  - Templates: Report to the Eclipse “Error log” and dump a Model Validation Log File
17 Conclusions

► Mixed-Criticality: A novel and mandatory aspect to consider in complex embedded system design

► Mixed-Criticality Modelling techniques

► Extension of a Single-Source Modelling Methodology

► Mixed-Criticality: Information used along the design process (modelling, verification, DSE, implementation)
More Information

- www.essyn.com
- D2.1.1: CONTREX System meta-model
- D2.2.2: CONTREX System modelling methodology (final)
- D2.3.2: System Modelling, Analysis and Validation tools (final)
- Fernando Herrera, Pablo Peñil, Eugenio Villar
  "A model-based, single-source approach to design-space exploration and synthesis of mixed-criticality systems"
  18th International Workshop on Software and Compilers for Embedded Systems, SCoPES 2015, ACM. 2015
- Fernando Herrera, Pablo Peñil, Eugenio Villar
  "UML/MARTE Modelling for Design Space Exploration of Mixed-Criticality Systems on top of Time-Predictable HW/SW Platforms"
  Jornadas de Computación Empotrada (JCE15). 2015-09