The UML Profile for MARTE: modelling predictable real-time systems with UML

Julio Medina
Universidad de Cantabria, Spain
julio.medina[at]unican.es

(Based on extracts from the MARTE Tutorial:
http://www.omgmarте.org/sites/default/files/TutorialMARТЕ-Final_version1_1_0.pdf)
General Requirements of the MARTE RFP

- Proposals shall support Modeling and Analysis of Real-Time and Embedded (in short MARTE) systems including its software and hardware aspects.
  - The Proposals will define a metamodel and its underlying UML profile.
  - It shall be possible to use independently software and hardware parts of the profile.
  - It shall comply with existing standards (UML2)
  - It shall update the SPT profile 1.1

Ref. of MARTE RFP (realtime/05-02-06):
http://www.omg.org/cgi-bin/doc?realtime/05-02-06
The ProMARTE Team

- **Partners**
  - **Industrials**
    - Alcatel*
    - Lockheed Martin*
    - Thales*
  - **Tool vendors**
    - ARTISAN Software Tools*
    - International Business Machines*
    - Mentor Graphics Corporation*
    - Softeam*
    - Telelogic AB (I-Logix*)
    - Tri-Pacific Software
    - France Telecom
    - No Magic
    - Mathworks
  - **Academics**
    - Carleton University
    - Commissariat à l’Energie Atomique
    - ESEO
    - ENSIETA
    - INRIA
    - INSA from Lyon
    - Software Engineering Institute (Carnegie Mellon University)
    - Universidad de Cantabria

* Submitter to OMG UML Profile for MARTE RFP
Design Pattern Adopted for the MARTE Profile

- **Stage 1** → Description of MARTE domain models
  - Purpose: Formal description of the concepts required for MARTE
  - Techniques: Meta-modeling

- **Stage 2** → Mapping of MARTE domain models towards UML2
  - Purpose: MARTE domain models design as a UML2 extensions
  - Techniques: UML2 profile
MARTE Overview

Foundations for RT/E systems modeling and analysis:
- CoreElements
- NFPs
- Time
- Generic resource modeling
- Allocation

Specialization of MARTE foundations for modeling purpose (specification, design...):
- Generic component model
- High-level application modeling
- Software resource modeling
- Hardware resource modeling

Specialization of foundations for annotating model for analysis purpose:
- Generic quantitative analysis
- Schedulability analysis
- Performance analysis
How to read, use, & implement MARTE
(see Section 2)

Extension Units

- NFP  Non-Functional Properties  Section 8
- Time  Enhanced Time Modeling  Section 9
- GRM  Generic Resource Modeling  Section 10
- Alloc  Allocation Modeling  Section 11
- GCM  Generic Component Model  Section 12
- HLAM  High-Level Application Modeling  Section 13
- SRM  Software Resource Modeling  Section 14.1
- HRM  Hardware Resource Modeling  Section 14.2
- RTM  Real-Time objects Modeling  Section 13
- GQAM  Generic quantitative Analysis  Section 15
- SAM  Schedulability Analysis  Section 16
- PAM  Performance Analysis  Section 17
- VSL  Value Specification Language  Annex B
- CHF  Clock Handling Facilities  Annex C
- RSM  Repetitive Structure Modeling  Annex E
- AADL  AADL models with UML  Section A.2
## Usage & Compliance Cases vs. Extension Units

### Table 7.2 - Extension Units that must be supported in each Compliance Case

<table>
<thead>
<tr>
<th>CASE</th>
<th>Level</th>
<th>GRM</th>
<th>NFP</th>
<th>VSL</th>
<th>Time</th>
<th>CHF</th>
<th>SRM</th>
<th>HRM</th>
<th>GCM</th>
<th>Alloc</th>
<th>HLAM</th>
<th>GQAM</th>
<th>PAM</th>
<th>SAM</th>
<th>RSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software</td>
<td>Base</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td>Base</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>System</td>
<td>Base</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Base</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schedulability</td>
<td>Base</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Infrastructure</td>
<td>Base</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Methodologist</td>
<td>Base</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Julio Medina
Needs for Model-Based RTES Analysis

Design-Oriented Models  
(code generation,...)  

Semantic mismatch  

Analysis-Oriented Models  
(formal models)  

Model Editors  

Data-flow, Components, Classes, State-Charts,...  

Schedulability Analysis  

Power Analysis  

Holistic models, RMA-based, queuing theory...  

How to specify unambiguous non-functional Information?  

Trade-offs between different parameters?
MARTE Features for Quantitative Analysis

- Improvements w.r.t. SPT
  - Extend implementation and scheduling models
    - e.g. distributed systems, hierarchical scheduling
  - Extend the set of analysis techniques supported
    - e.g. offset-based techniques
  - Extend timing annotations expressiveness
    - Overheads (e.g. messages passing)
    - Response times (e.g. BCET & ACET)
    - Timing requirements (e.g. miss ratios and max. jitters)
- New features w.r.t. SPT
  - Support for sensitivity analysis
  - Improve modeling reuse and component-based design.
  - Support of the “Y-chart” approach: application vs. platform models
GQAM Profile factorizes common constructs and NFPs
- Stereotypes define “analysis” abstractions
  - workload events, scenarios,…
  - schedulable entities, shared resources, processing nodes, schedulers…
- Stereotype attributes define pre-defined NFPs
  - e.g. event arrival patterns, end-to-end deadlines, wcet-bcet-acet,…

The analysis sub-profiles define model well-formedness rules
- It includes “constraints” to construct “analyzable” models, w.r.t…
- ”Analysis Model Viewpoints” (e.g., schedulability analysis viewpoint)
- Specialized constraints must be refined by technique-specific approaches
GQAM: Dependencies and Architecture

- General NFP types
- Processing & Scheduling model
- Timed processing model
- Performance analysis (non-deterministic performance)

Diagram:
- NFPs
- Time
- GRM
- "modelLibrary" -> MARTE_Library
- GQAM
- GQAM_Workload
- GQAM_Resources
- GQAM_Observers
- SAM
- PAM

- Schedulability analysis (timeliness)
GQAM: Analysis Modeling Structure
Processing schema for model-based analysis

UML2 + Marte

« profile »
MARTE

UML2 editor

Annotated model

Results/Diagnostic

Model converter

Analysis specific framework

Analysis model

Analysis tool

Analysis results

Results converter
Key reasoning for the use of UML in the scheduling analysis of RTE Systems

- UML is a standard semi-visual language for conceptual modeling that enables the usage of a Model Based approach for software and system engineering.

- MDD and UML have been broadly introduced and used in principle by the software engineering community, and have reached a significant number of practitioners and tool support.

- To take benefit of this in the RTE domain, they need to be capable of supporting the necessary (at least timing) verifications.

- This leads to the necessity of model based scheduling analysis techniques.

- As well as the necessity to have the modeling elements to describe the platform, the interacting environment and the timing requirements.
A Simple Example (Classical Scheduling Theory)

- Typically analyzed with RMA:
  - Critical instant calculation
  - Utilization bound test or Response time calculation for the first deadline.
More general approach

Scenario(instance) based, distributed, control-flow dependencies
SAM: Integration Different Approaches

Other Sched. Analysis tools: Livedevices’ Real-Time Architect, CoMET from VaST, Vector’s CANAlyzer…
General Procedure to Use the SAM Profile

1. Design Models
2. Annotated Behavior Models
3. Workload Behavior Models (PIM)
4. Specify Parameterized Analysis Context Model
5. Resources Platform Models (PDM)
6. Non-functional values for specific analysis contexts

Analysis Tools
Analysis File
Determine Desired NFPs of Interest (given and predicted parameters)
SAM: The Notion of End-To-End Flow

An “End-To-End Flow” is the basic workload unit to be evaluated by schedulability analysis tools.

→ An end-to-end flow refers to the entire causal set of steps triggered by one or more external workload events.

Step: basic behavioral unit (e.g., execution actions, call actions, messages, ...)

Workload event: basic stimuli unit (e.g., timers, external occurrences, internal events, ...)

processing times (worst and best case)
End-to-end flows (from MAST)
Execution and communication steps may be causally related by one of the following precedence relations:
SAM: Workload Domain Metamodel (end-to-end)

- **End-to-end response and deadline times**
  - Predictions provided by analysis tools
  - **SAM_Observers:: TimingObserver**
  - **EndToEndFlow**
    - workload \(1..^*\)
    - **GQAM_Workload:: WorkloadBehavior**
      - isScheduled: NFP_Boolean
      - schedulabilitySlack: NFP_Real
      - endTime: NFP_Duration
      - endToEndTime: NFP_Duration
      - endToEndDeadline: NFP_Duration
  - **GQAM_Workload:: WorkloadEvent**
    - pattern: ArrivalPattern
    - inputStream
    - effect
    - **Stimuli information**
## SAM: Example of Stereotype Extensions Usage

<table>
<thead>
<tr>
<th>SAM Domain Model</th>
<th>SAM Stereotype</th>
<th>UML Metaclasses</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>WorkloadBehavior</td>
<td>GaWorkloadBehavior</td>
<td>UML::Interactions::Fragments::CombinedFragments</td>
<td>Modeled in a high-level interaction</td>
</tr>
<tr>
<td>EndToEndFlow</td>
<td>SaEnd2EndFlow</td>
<td>UML::Interactions::Fragments::InteractionOperand</td>
<td>Modeled in a high-level interaction</td>
</tr>
<tr>
<td>WorkloadEvent</td>
<td>GaWorkloadEvent</td>
<td>UML::Interactions::BasicInteractions::Message</td>
<td>Modeled in a high-level interaction</td>
</tr>
<tr>
<td>BehaviorScenario</td>
<td>GaScenario</td>
<td>UML::Interactions::BasicInteractions::Interaction</td>
<td>Modeled as a low-level interaction nested within a higher-level interaction</td>
</tr>
<tr>
<td>Step</td>
<td>SaStep</td>
<td>UML::Interactions::BasicInteractions::Message</td>
<td>Messages in low-level interactions</td>
</tr>
<tr>
<td>CommunicationStep</td>
<td>SaCommStep</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReleaseStep</td>
<td>GaRelStep</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AcquireStep</td>
<td>GaAcqStep</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SAM: Examples of Behavior Annotations

- async. message transmission
- behavior scenario
- lock and unlock of a shared resources
- syncr. execution message
SAM: Example of Workload Annotations

end-to-end flow

concurrent fragments

User

triggering events (async. messages)

use of an interaction

Julio Medina 26
SAM: Resources Concepts

- Provide additional (analysis-specific) annotations to annotate resources platform models

![Diagram showing domain, processing resources (execution and communication), shared resources, and schedulable resources with annotations for execution host, communication host, shared resource, scheduler, and schedulable resource.]
Resources (from GRM)

- **GRM::ResourceCore::Resource**
  - resMult: Integer
  - isProtected: Boolean
  - isActive: Boolean

- **StorageResource**
  - SynchResource
  - ConcurrencyResource

- **CommunicationResource**
  - ComputingResource

- **TimingResource**
  - DeviceResource
Domain model for Schedulable resources
Shared resources
Resource Usage

**ResourceUsages**

- **MARTE::CoreElements::Causality::CommonBehavior::Event**
  - 0..1 *+event

- **UsageDemand**
  - 0..* workload

- **ResourceUsage**
  - usage
  - 0..*

- **StaticUsage**

- **DynamicUsage**

- **GRM::ResourceCore::Resource**
  - usedResources 1..*
  - requiredAmount 1..*

- **UsageTypedAmount**
  - execTime: NFP_Duration [*]
  - msgSize: NFP_DataSize [*]
  - allocatedMemory: NFP_DataSize [*]
  - usedMemory: NFP_DataSize [*]
  - powerPeak: NFP_Power [*]
  - energy: NFP_Energy [*]

- **MARTE::CoreElements::Causality::CommonBehavior::Behavior**
1. If the list usedResources is empty the list subUsages should not be empty and viceversa.
2. If the list usedResources has only one element, all the optional lists of attributes refer to this unique Resource and at least one of them must be present.
3. If the list usedResources has more than one element, all of the optional lists of attributes that are present, must have that number of elements, and they will be considered to match one to one.
4. If the list subUsages is not empty, and any of the optional lists of attributes is present, then more than one annotation for the same resource and kind of usage may be expressed. In this case, if the annotations have also the same source and statistical qualifiers they will be considered in conflict, and hence the ResourceUsage inconsistent.
SAM: Resources Domain Metamodell
## SAM: Examples of the Stereotypes Usage

<table>
<thead>
<tr>
<th>SAM Domain Model</th>
<th>SAM Stereotype</th>
<th>UML Metaclasses</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResourcesPlatform</td>
<td>GaResourcesPlatform</td>
<td>UML::StructuredClasses::SctructuredClass</td>
<td>Main container of resources</td>
</tr>
<tr>
<td>SaExecutionHost</td>
<td>SaExecHost</td>
<td>UML:: StructuredClasses::Property</td>
<td>Parts of the resources platform</td>
</tr>
<tr>
<td>SaCommunicationHost</td>
<td>SaCommHost</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRM::Scheduler</td>
<td>Scheduler</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRM::SchedulableResource</td>
<td>SchedulableRes</td>
<td>UML:: StructuredClasses::Property</td>
<td>Parts of processing resources</td>
</tr>
<tr>
<td>SaCommChannel</td>
<td>SaCommChannel</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SAM: Example of Resources Stereotype Usage

Resources platform under analysis

Processing Resources as UML parts

Concurrency resources as nested parts

Scheduler as UML part

User

« gaResourcesPlatform » RP1

« saCommHost »

commH1

« saCommChannel »

ch1: CH

« saCommChannel »

ch2: CH

« saExecHost »

execH1

« schedulableRes »

schR1: SCHR

« schedulableRes »

schR2: SCHR

« scheduler »

sch1: SCH
An analysis context is the root concept used to collect relevant quantitative information for performing a specific analysis scenario.

An analysis context integrates workload behavior models and resources platform models.
SAM: Analysis Context Domain Metamodel

- **SAM**
  - **GQAM:: AnalysisContext**
    - **GQAM_Workload:: WorkloadBehavior**
      - 1 workloadBehavior
    - **GQAM_Resources:: ResourcesPlatform**
      - 1:* resourcesPlatform
    - **SaAnalysisContext**
      - isScheduled: NFP_Boolean
      - optimalityCriterion: optimalityCriterionKind
    - **« enumeration » OptimallityCriterionKind**
      - meetHardDeadlines
      - minimizeMissedDeadlines
      - minimizeMeanTardiness
      - undef
      - other

Global analysis annotations
SAM: Example of Analysis Context Stereotype Applic.
Example: A Teleoperated Robot

ClassesView_TeleoperatedRobot

DisplayData
- data: Integer [*]
- read(): Data
- write(D: Data)

DisplayRefresher
- updateDisplay()
- updateGraphics()

CommandInterpreter
- processEvent()
- planTrajectory()

StationCommunication
- sendCommand(C: Command)
- awaitStatus(): Status

ControllerCommunication
- sendStatus(S: Status)
- awaitCommand(): Command

Reporter
- report()

ServosData
- Data: Integer [*]
- get(): Data
- set(D: Data)

ServosController
- controlServos()
- controlAlgorithms()
- doControl()

DeploymentView_TeleoperatedRobot

Station

CAN_Bus

Controller

VM defence RobotArm
Example of Annotated Scenario with SAM

Step (exec. time)

« gaScenario » Report

Report

ContrClock

« saSharedRes »

ServosData

{ protectKind= priorityCeiling, priority=10 }

« saStep » report()

{ execT= (1.1, 1.22, ms) }

« gaAcqStep »

lock()

« saStep » get()

{ execT= (0.004, 0.02, ms) }

« gaRelStep » unlock()

« saCommStep »

transmit()

{ msgSize= (100, kB) }

« saCommStep »

transmitCommand()

{ execT= (2.56, 2.56, ms) }

« saStep »

sendStatus()

{ execT= (0.031, 0.031, ms) }

« saStep »

awaitStatus()

{ execT= (0.022, 0.022, ms) }

« saCommStep »

transmit()

{ msgSize= (100, kB) }

« saStep »

updateDisplay()

{ execT= (0.24, 0.4, ms) }

« saStep »

read()

« saExecStep » write()

« saStep »

updatesGraphics()

{ execT= (5, 10, ms) }

Shared Resource

User
Example of Annotated Resources Model with SAM

Threads owned by the processing resource
Example of Analysis Context Model

- End To End Flows (end2end deadlines and predicted times)
- Workload Events (arrival patterns)

User

Workload Behavior

Scenario (response times, hosts utilization...)

Julio Medina
Example of Parametric Analysis Context

Instance of a WorkloadBehavior model

Context-specific variables

Simple Schedulability Analysis context

Sensitivity Analysis context

User

<table>
<thead>
<tr>
<th>Simple Schedulability Analysis context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedulability: TeleoperatedRobotSAM</td>
</tr>
<tr>
<td>isSched_System= (true, $v0, calc)</td>
</tr>
<tr>
<td>wcet_Report= (5, ms, determ)</td>
</tr>
<tr>
<td>procRate_CAN= (1, determ)</td>
</tr>
<tr>
<td>period_Report= (30, ms, determ)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sensitivity Analysis context</th>
</tr>
</thead>
<tbody>
<tr>
<td>SensitivityAnalysis: TeleoperatedRobotSAM</td>
</tr>
<tr>
<td>isSched_System= (true, req)</td>
</tr>
<tr>
<td>wcet_Report= (50, $v1, ms, max, calc)</td>
</tr>
<tr>
<td>procRate_CAN= (0.2, $v2, min, calc)</td>
</tr>
<tr>
<td>period_Report= (10, $v3, ms, min, calc)</td>
</tr>
</tbody>
</table>

Context under Analysis

TeleoperatedRobotSAM

«saAnalysisContext»
{ isSched= ($isSchSys) }

var { dir= inout } isSched_System: NFP_Boolean= isSchSys
var { dir= inout } wcet_Report: NFP_Duration= wcet1
var { dir= inout } procRate_CAN: NFP_Real= prCAN
var { dir= inout } period_Report: NFP_Duration= pR

« gaWorkloadBehavior » : NormalMode
« GaResourcesPlatform » : TeleoperatedRobot_Platform
MARTE Tooling

- Current Implementations supporting MARTE
  - Full MARTE Profile & Libraries for Eclipse UML2
  - VSL edition assistant and type checker as a Eclipse plug-in for the UML Papyrus tool and RSA 7.0
  - Implemented by RSA, Magicdraw, Rhapsody, PapyrusUML,…

- Links to other tools:
  - Eclipse plug-ins to transform UML models annotated with the SAM profile to input files of MAST, SymTA/S, Cheddar and RapidRMA
  - TimeSquare: handling of Clock Constraint Specification Language

The official OMG web page holds links to all these tools:
[www.omgmarте.org](http://www.omgmarте.org)

MARTE Open Source Eclipse based implementation in UML Papyrus: [www.papyrusuml.org](http://www.papyrusuml.org)
Conclusions on analysis capabilities of MARTE

- **Industrial Use of V&V can benefits from MDE**
  - Analysis task must be cohesively integrated with Design tasks
  - Application of individual analysis techniques should be regarded as an essential part of an integrated V&V methodology

- **Methodological support is still under way:**
  - Complex analysis scenarios for Interface-Based Design, Multiobjective Design Space Exploration…
  - Means to manage NFP measurement models
  - Methods to map/transform MoCCs into analysis models
MARTE Frontiers and Challenges

- MARTE defines the language constructs only!
  - Common patterns, base building blocks, standard NFP annotations
  - Generic constraints that do not force specific execution models, analysis techniques or implementation technologies

- It does not cover methodologies aspects:
  - Interface-Based Design, Design Space Exploration
  - Means to manage refinement of NFP measurement models
  - Concrete processes to storage, bind, and display NFP context models
  - Mapping to transform HLAM’s into analysis models

MARTE is to the RTES domain as UML to the System & Software domain: a family of well known and open specification formalisms!
Thanks for your attention..!

Questions?

- www.omgmar-te.org
  - The MARTE web site

- www.papyrusuml.org
  - On open source Eclipse plug-in for UML2 graphical modeling