Schedulability analysis of Ravenscar systems with MAST+

Marco Panunzio and Tullio Vardanega

Department of Pure and Applied Mathematics
University of Padova, Italy
{panunzio, tullio.vardanega}@math.unipd.it

ArtistDesign Workshop on
Real-Time System Models for Schedulability Analysis

University of Cantabria - Santander, Spain

February 7-8, 2011

Outline

- **Overview on MAST+**
  - ❑ Use of MAST+ in the ASSERT project
- **Overview on the Ravenscar Profile**
- **Ravenscar-aware schedulability analysis**
- **Implementation in MAST+**
- **Conclusions**
  - ❑ MAST+
  - ❑ Evolution of MAST

Overview on MAST+

- **Developed in the scope of the ASSERT project**
  - FP6 [2004-2008]
  - Model-based process for the development of on-board software
- **Support for**
  - Modeling of systems abiding by the priority-band architecture
    - Hierarchical architecture of interest for the project
  - Modeling of Ravenscar systems
  - Ravenscar-aware scheduling analysis
    - Holistic analysis
  - XML output [missing in MAST at that time]
- **Realized as an extension to MAST 1.3.6**
  - Developed in the period August 2006-December 2007

Use of MAST+ in the ASSERT project

1. Construct a PIM, which represent a problem solution independent of any particular implementation

2. Complement the PIM with information on the target platform and the deployment plan

3. The design environment automatically generates a PSM via model transformation

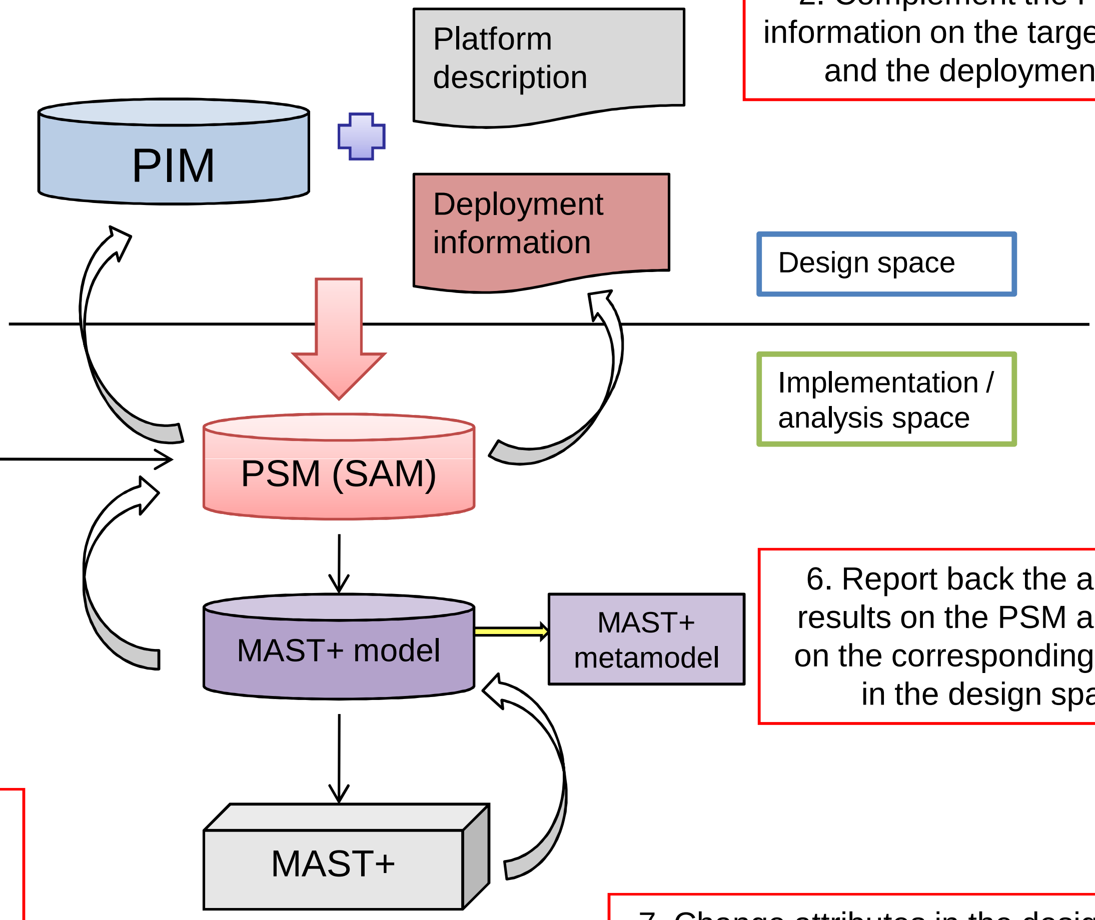
Equivalent to the abstraction level of MARTE

4. Extract information from the PSM / SAM to create a MAST model (conforming to the MAST+ metamodel)

5. Generate the input to feed MAST+ and perform schedulability analysis

6. Report back the analysis results on the PSM and then on the corresponding entities in the design space

7. Change attributes in the design space and iterate the analysis until the system is satisfactory in the various functional and extra-functional dimensions



[PaVa07]
[BoPaVa08]

Overview on the Ravenscar profile

- **Restricted tasking model for the Ada programming language**
  - ❑ Removes all sources of non-determinism and unbounded execution cost
  - ❑ Can be implemented in a small and efficient real-time kernel
  - ❑ Can be conceptually mirrored in other languages (e.g. RTSJ)
- **Ravenscar programs are amenable to static analysis**
  - ❑ Static existence model
    - ❑ Fixed set of tasks and interrupts, fixed priorities, no task termination
  - ❑ Static synchronization and communication model
    - ❑ No task synchronization (rendezvous)
    - ❑ Asynchronous one-way communication mediated by protected objects
  - ❑ Deterministic execution model
    - ❑ Max 1 PO Entry, Max 1 Task per PO Entry, No Relative Delay, No Asynchronous Control, use of high-precision notion of time (e.g. Ada.Real_Time)
  - ❑ Deterministic memory usage
    - ❑ No_Implicit_Heap_Allocations

[BuDoVa03]

Ravenscar-aware schedulability analysis

Classical uniprocessor response time equation

$$w_i^n = B_i + C_i + \sum_{j\in hp(i)} \left\lceil \frac{w_i^{n-1} + J_i}{T_j} \right\rceil C_j$$

Validity condition $D_i \le T_i$

Termination condition $w_i^{n+1} = w_i^n$

Ravenscar uniprocessor equations

$$w_i^n = \max(\underbrace{B_{\ker,}}_{\text{Kernel blocking time}} B_i) + \underbrace{CS1}_{\text{“In” Context switch}} + C_i + \sum_{j\in hp(i)} \left\lceil \frac{w_i^{n-1} + J_i}{T_j} \right\rceil (\underbrace{CS1}_{\text{“In” Context switch}} + C_j + \underbrace{TS}_{\text{Suspension overhead}} + \underbrace{CS2}_{\text{“Out” Context switch}}) + \underbrace{I_{clock}^{w_i^{n-1}}}_{\text{Clock overhead}} + \underbrace{I_{extint}^{w_i^{n-1}}}_{\text{Interrupt management}}$$

$$I_{clock}^{w_i^n} = I_{periodic}^{w_i^n} + I_{demanded}^{w_i^n} = \left\lceil \frac{w_i^n}{T_{periodic}} \right\rceil CH_{periodic} + \sum_{j\in hp_{periodic}(i)} \left\lceil \frac{w_i^n}{T_j} \right\rceil CH_{demanded} + \sum_{k\in lp_{periodic}(i)} CH_{demanded}$$

CS1=`Ready+Select+Switch`

CS2=`Select+Switch`

$TS_{periodic}$=`Delay_until`

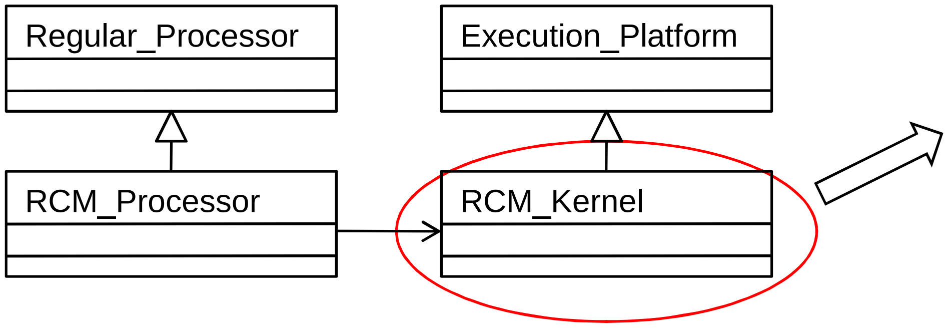
$TS_{sporadic}$=`Delay_until+Wait(Enter)`

Knowledge of *which* kernel mechanisms (and *how many times*) are used in a busy interval

[VaZaPu05]

Implementation in MAST+

1) System modeling



Regular_Processor

Execution_Platform

RCM_Processor

RCM_Kernel

GNATforLEON_2.1.0

Ready	10µs
Select	5µs
Switch	5µs
Wait	15µs
Delay_Until	3µs
CH_demanded	12µs
...	

Introduced the concept of **execution platform**, to model the cost of RTOS / Kernel mechanisms

Partly redundant with other MAST entities

Ideally, the cost shall be in processor cycles, later scaled according to the processor frequency

2) Analysis tools

- **Implemented a Ravenscar-aware holistic analysis**
  - ❑ The analysis tool can access the kernel metrics and use them as terms in the equations
  - ❑ Considerable as a “feasibility study” for the implementation of Ravenscar-aware offset-based analysis

Conclusions

- **MAST+**
  - ❑ Created as part of a MDE process
    - ❑ Supported also by the “follow-up” of the ASSERT project, named CHESS
  - ❑ Responds to important modeling and analysis needs
    - ❑ Ravenscar systems
    - ❑ The implementation was not optimal
      - ❑ Constrained by other project- and maintenance-related aspects
  - ❑ Support is discontinued
    - ❑ Too costly to backport the changes of newer versions of MAST
      - ❑ Easier to re-start directly from MAST 1.3.8

Evolution of MAST

- **Relationship with the UML MARTE profile**
  - ❑ Development of standard converters to and from MAST

- **Easy of extensibility to support additional analysis tools and platforms**

- **How to promote a third-party extension to the mainstream version of MAST?**

Bibliography

- [BoPaVa08] M. Bordin, M. Panunzio, T. Vardanega: "*Fitting Schedulability Analysis Theory into Model-driven Engineering*". ECRTS 2008. MDE approach to model-based schedulability analysis.
- [BuDoVa03] A. Burns, B. Dobbing, and T. Vardanega. Guide for the use of the Ada Ravenscar Profile in high integrity systems. Technical Report YCS-2003-348, University of York, 2003. Ravenscar Profile for the Ada language.
- [PaVa07] M. Panunzio, T. Vardanega: "*A Metamodel-driven Process Featuring Advanced Model-based Timing Analysis*". Reliable Software Technologies - Ada-Europe 2007*.* Detailed ASSERT approach to model-based schedulability analysis.
- [VaZaPu05] T. Vardanega, J. Zamorano and J. A. de la Puente: "*On the Dynamic Semantics and the Timing Behavior of Ravenscar Kernels*". Real-Time Systems 29(1), 2005. Uniprocessor response time analysis for Ravenscar systems.