Towards MARTE++: an enhanced UML-based language to Model and Analyse Real-Time and Embedded Systems for the IoT age

Julio L. Medina
julio.medina@unican.es

Eugenio Villar
eugenio.villarb@unican.es

Universidad de Cantabria. Santander, Spain

Modern Platforms and Mixed Criticalities

MARTE should support the specification of hardware platforms that include multiple cores. This extent must include not only the hardware description but also the modern scheduling algorithms for multicore, including the capacity of tasks to migrate from one core to another as well as the mechanisms to do the core assignment (affinity) to tasks.

Criticality is a designation of the level of assurance against failure needed for a system component. A mixed criticality system is one that has two or more distinct levels (consider for example safety critical, mission critical and non-critical). Reviewing the standards in the field (IEC 61508, DO-178B, DO-254 and ISO 26262 standards) they propose to use up to five levels. Then, in general an integer value represented with NFP_Integer is sufficient to annotate the criticality level for a value or a constraint.

General purpose and ad-hoc networking

One of the outstanding opportunities but also biggest challenges for the traditional embedded systems design techniques is to accommodate for their interconnection through general purpose and/or unreliable networks. Eventually, traditional resource account oriented distributed systems may scale well for this purpose, but only in very controlled industrial environments. The Internet of Things and the emergent behavior in cyber-physical systems of systems are design and validation environments on which MARTE needs a boost. Some initial efforts have been produced to enhance its suitability to model combined situations, but many more distributed modelling use cases need to be envisioned, formalized and assessed.

In particular the exploitation of cloud-based services and platforms for reconfiguration, and pre-analyzed environment identification, or even predictability analysis can be categorized.

A conceptual formalization of operational stages for devices, motes, components, nodes, sub-systems, systems, and larger composites may well help practitioners to specify their models at the right level of abstraction and envisioned functionality.

Methodological aspects

One of the main criticisms of MARTE was its large size and the various ways in which a concept can be mapped into UML elements. These characteristics of MARTE responded to the various dimensions in which UML (and its profiles) can be used along the specification, development and analysis processes. A way to minimize potential inconsistencies, clarify mappings, and eventually fix the semantics of the language and its elements is defining very precise and decoupled modeling scopes and associating to them exclusive modeling elements that may conform complementing dialects of MARTE.

Proposals might be forced to provide the needed dialects in the form of inter-related domain specific languages. All of them would be supported by a complete common meta-model whose semantics must be fixed in accordance to the UML meta-model and the executable UML related specifications. The idea here is similar to promoting as normative a coherent version of all the domain models of the current MARTE.

The normative or non-normative quality of this meta-model needs to be discussed and clearly specified in the requirements of the RFP. Alternatively, stricter conformance cases (see clause 6 or MARTE) may be used instead to define the dialects to be used in that strict normative way. The contents of each conformance case would need to be very clearly specified and promoted along the spec through the examples and the model libraries provided in it.

This effort looks for those requirements that would be needed in the community in an enhanced version of the UML Profile for MARTE, the current standard of the OMG for the modelling and analysis of real-time embedded systems.

Since its adoption by the OMG in 2009 and after the various additions along recent years, MARTE has been essayed in a number of application domains and validation approaches. We make here an initial review of these various efforts describing extensions, additional functionality, and modeling needs that may serve as inputs for the preparation of a formal Request For Proposals (RFP) at the OMG.

Aspects that have been found useful to have in it include modern platforms like Multi-core, Many-core and GPUs, networking for broader domains like the Internet of Things, federation of all modelling artifacts involved in the development process, including tracing mechanisms embedded in the language to link design and run-time artifacts, and more elaborated kinds of quantitative analyses and extra functional properties, like energy and memory consumption, heat dissipation, and temperature distribution.

Also methodological aspects like its specification as a profile and/or as a meta-model will need to be discussed. Finally, the standard needs to be reviewed against the new executable UML related specifications; particularly to be in alignment with those that fix the semantics of state machines and composite structures.

Managing views and tracing mechanisms along the development process and iterative validation strategies.

It is clear that most of the possible solutions to manage views, traces along the development process, and tagging of models for their usage in any particular way are severely influenced by the methodological guidelines followed by the modeler. Hence, it is discouraged to try to get them as part of the language.

Nevertheless, any generic strategy that may help to link process oriented data with models will be of extreme value for mastering the complexity of model driven artifacts explosion.

So far, in the current version of MARTE, the only help to address these kinds of concerns are the concepts of "mode" and "configuration". An enhanced version of MARTE should help practitioners to link models with specific phases of the development process followed. General categories described in SPEM may be linked to models so that stakeholders can select and view the appropriate information at the appropriate time.

Hardware and Software synthesis

Considering the effects of the end of Moore's Law, semiconductor technology could become much more accessible to companies with smaller volumes. In the near future, the design of integrated circuits will be accessible to any company in which the value added by silicon compensates the higher non-recurrent engineering costs. This implies that the synthesis of high level models or the combined transformations of functional models with the intermediate models of computation, could be implemented on both, hardware and software.

This additional freedom implies an extended design space which may need additional constraints and extra-functional properties to be fully explored. Even the concepts of allocation and assignment managed in MARTE might need to be revisited.

Links to hardware description languages, and again richer kinds of allocation will have to be explored.

High-level application modelling semantics for analyzability and xUML conformance

One of the main risks for real-time systems designed with UML is the absolute freedom of the designer to use constructs and patterns that can easily lead to actual systems whose timing predictability is completely impossible to analyze or predict. The HLAM clause in MARTE was meant to help in this concern. Unfortunately, the semantics of the way the behaviors in RtUnits and PpUnits are activated is not explicitly included in MARTE. Several methodologies are proposed to keep in synch the executable generated code with the corresponding analysis models. It is very important that whichever constrained design environment is defined for MARTE , its semantics gets aligned with the new standards that complement the current trend towards an executable UML (fUML, PSCS, and PSSM).

This is critical to unsure that any prediction done with analysis models, that claim to be faithful to the high-level design models semantics, can be truly respected by the semantics implemented in the code properly generated from such design models. Other specific models of computation might also be defined as valid variants for MARTE, but at least the one in the current executable UML should be supported.

Write YOUR requirements here:

Additional analysis and validation domains

From its inception, the generic quantitative analysis modeling (GQAM) sub-profile in MARTE was meant to accommodate more and various kinds of behavioral based quantitative analyses. Some efforts have been produced regarding dependability, but many other related topics can require specific modelling needs. Profiling cache usage for worst-case execution time calculation and bounding stack utilization for example are among those analyses partly supported in MARTE that might be formally extended with specific analysis sub-profiles.

Other extra-functional properties, like power consumption, temperature space distribution, heat dissipation, cost, inner complexity, criticality compatibility at allocation, security impact, etc. may have specific sub-profiles specializations of GQAM.

Finally, contract based design, and its assessment may benefit from an extension of GQAM to non-behavioral but static structural model validation, both for deployment (allocation) assessment and cloud-based remote operational discovery and running

ACKNOWLEDGEMENTS

This work receives funding from the Spanish Government under grant number TIN2014-56158-C4-2-P (M2C2), and from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737494 (MegaM@RT2). This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Sweden, France, Spain, Italy, Finland, Czech Republic.

We thank the anonymous reviewers for their insights and proposals of improvements.