

# UML Profile for MARTE: Time Model and CCSL

Frédéric Mallet<sup>1</sup>

Université Nice Sophia Antipolis, Aoste team INRIA/I3S, Sophia Antipolis, France

`Frederic.Mallet@unice.fr`

**Abstract.** This 90 minutes tutorial gives a basic introduction to the UML Profile for MARTE (Modeling and Analysis of Real-Time and Embedded systems) adopted by the Object Management Group. After a brief introduction to the UML profiling mechanism, we give a broad overview of the MARTE Profile. Then, the tutorial shall focus on the time model of MARTE and its companion language CCSL (Clock Constraint Specification Language).

**Keywords.** UML Profile, Real-Time, Embedded systems, MARTE, CCSL

**Key terms.** StandardizationProcess, UbiquitousComputing, ConcurrentComputation, ModelBasedSoftwareDevelopmentMethodology

## 1 Audience and focus

The targeted audience is academics or industrials interested in high-level modeling with UML and its application to the analysis of real-time and embedded systems.

The tutorial does not require any preliminary background as it will give a high-level and broad description of the UML Profile as well as a closer focus on its time model. To ensure that a large public can follow the presentation, we should start by a brief overview of UML light-weight extension mechanism, the so-called profiling mechanism.

## 2 Topic

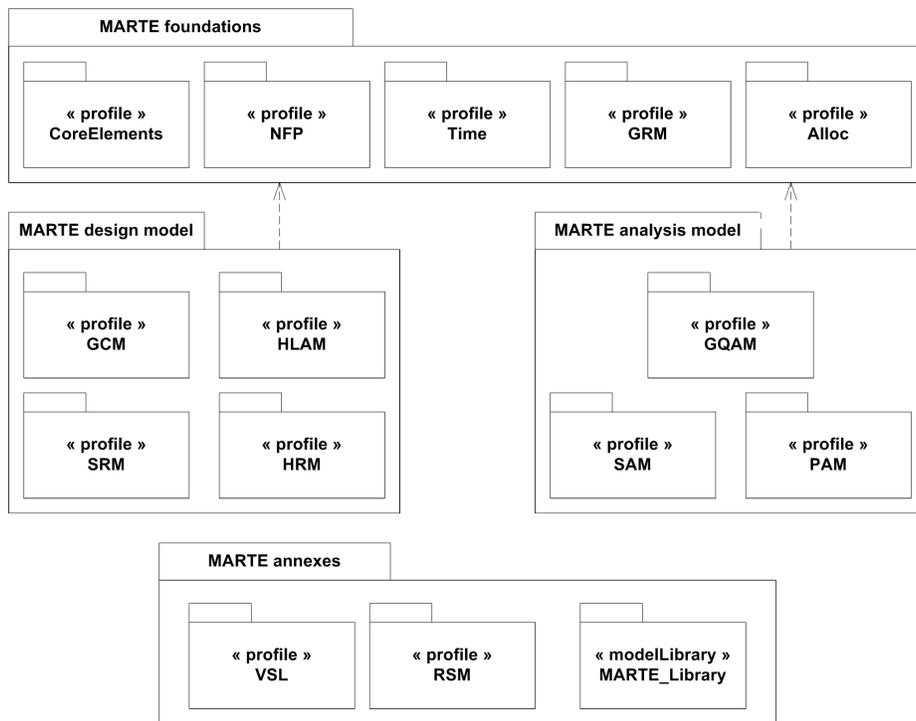
The UML profile for Modeling and Analysis of Real-Time and Embedded systems, referred to as MARTE [1], has been adopted by the OMG in November 2009 and revised in June 2011. It extends the Unified Modeling Language (UML) [2] with concepts required to model embedded systems.

This ninety minutes tutorial gives a basic introduction to the UML Profile for MARTE. After a broad view of the Profile, the tutorial shall focus on the time model of MARTE and its companion language CCSL (Clock Constraint Specification Language).

## 2.1 General Introduction to MARTE

Figure 1 shows the general structure of MARTE.

The General Component Modeling (GCM) and Repetitive Structure Modeling (RSM) packages offer a support to capture the application functionality. GCM defines basic concepts such as data flow ports, components and connectors. RSM provides concepts for expressing repetitive structures and regular interconnections. It is essential for the expression of parallelism, in both application modeling and execution platform modeling; and for the allocation of applications onto execution platforms.



**Fig. 1.** Structure of MARTE specification

The Hardware Resource Modeling (HRM) package, which specializes the concepts of GCM into hardware devices such as processor, memory or buses, allows the modeling of the execution platforms in MARTE. The Allocation (Alloc) package allows the modeling of the space-time allocation of application functionality on an execution platform. Both the HRM and Alloc packages can be used with the RSM package for a compact modeling of repetitive hardware (e.g., grids of processing elements) and data and computation distributions of a parallel application onto such a repetitive hardware.

The models described with the previous packages can be refined with temporal properties specified within the Time package [3]. Such properties are typically clock constraints denoting some activation rate constraints about considered components. The concepts of the Time package are often used with the Clock Constraint Specification Language (CCSL) [4, 5], which was initially introduced as a non-normative annex of MARTE.

## 2.2 MARTE Time Model

In MARTE, a *clock*  $c$  is a totally ordered set of *instants*,  $\mathcal{I}_c$ . In the following,  $i$  and  $j$  are instants. A *time structure* is a set of clocks  $\mathcal{C}$  and a set of relations on instants  $\mathcal{I} = \bigcup_{c \in \mathcal{C}} \mathcal{I}_c$ . CCSL considers two kinds of relations: *causal* and *temporal* ones. The basic causal relation is *causality/dependency*, a binary relation on  $\mathcal{I}$ :  $\preceq \subset \mathcal{I} \times \mathcal{I}$ .  $i \preceq j$  means  $i$  causes  $j$  or  $j$  depends on  $i$ .  $\preceq$  is a pre-order on  $\mathcal{I}$ , *i.e.*, it is reflexive and transitive. The basic temporal relations are *precedence* ( $\prec$ ), *coincidence* ( $\equiv$ ), and *exclusion* ( $\#$ ), three binary relations on  $\mathcal{I}$ . For any pair of instants  $(i, j) \in \mathcal{I} \times \mathcal{I}$  in a time structure,  $i \prec j$  means that the only acceptable execution traces are those where  $i$  occurs strictly before  $j$  ( $i$  precedes  $j$ ).  $\prec$  is transitive and asymmetric (reflexive and antisymmetric).  $i \equiv j$  imposes instants  $i$  and  $j$  to be coincident, *i.e.*, they must occur at the same execution step, both of them or none of them.  $\equiv$  is an equivalence relation, *i.e.*, it is reflexive, symmetric and transitive.  $i \# j$  forbids the coincidence of the two instants, *i.e.*, they cannot occur at the same execution step.  $\#$  is irreflexive and symmetric. A consistency rule is enforced between causal and temporal relations.  $i \preceq j$  can be refined either as  $i \prec j$  or  $i \equiv j$ , but  $j$  can never precede  $i$ .

In this paper, we consider discrete sets of instants only, so that the instants of a clock can be indexed by natural numbers. For a clock  $c \in \mathcal{C}$ , and for any  $k \in \mathbb{N}^*$ ,  $c[k]$  denotes the  $k^{\text{th}}$  instant of  $c$ .

Specifying a full time structure using only instant relations is not realistic since clocks are usually infinite sets of instants. Thus, an enumerative specification of instant relations is forbidden. The Clock Constraint Specification Language (CCSL) defines a set of time patterns between clocks that apply to infinitely many instant relations [4].

The UML Profile for MARTE proposes several specific stereotypes in the Time chapter to capture CCSL specifications. Figure 2 briefly describes the three main stereotypes. Boxes with the annotation «metaclass» denote the UML concepts on which our profile relies, so-called metaclasses. Boxes with stereotype are the concepts introduced by MARTE, *i.e.*, the stereotypes. Arrows with a filled head represent extensions, whereas normal arrows indicate properties of the introduced stereotypes. **Clock** extends UML Events to spot those events that can be used as time bases to express temporal or logical properties. **ClockConstraint** extends UML Constraints to make an explicit reference to the constrained clocks. **TimedProcessing** extends **Action** to make explicit their start and finishing events. When those events are clocks, then a **ClockConstraint** can constrain the underlying action to start or finish its execution as defined in a CCSL specification.

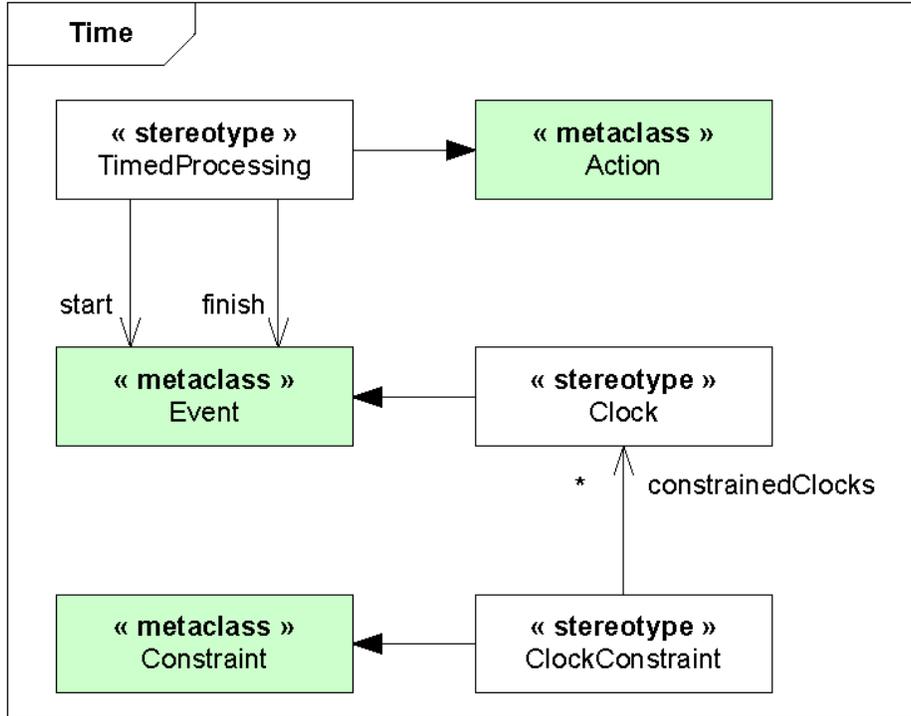


Fig. 2. Excerpt of the MARTE Time Profile

### 2.3 The Clock Constraint Specification Language

On top of MARTE clocks, the Clock Constraint Specification Language defines a set of operators (relations and expressions) [4]. As an example, consider the clock relation *precedence* (denoted  $\prec$ ), a transitive asymmetric binary relation on  $\mathcal{C}$ :  $\prec \subset \mathcal{C} \times \mathcal{C}$ . If *left* and *right* are two clocks,  $left \prec right$ , read 'left precedes right', specifies that the  $k^{\text{th}}$  instant of clock *left* precedes the  $k^{\text{th}}$  instant of clock *right*, for all  $k$ . More formally: For a pair of clocks  $(left, right) \in \mathcal{C} \times \mathcal{C}$ ,  $left \prec right$  means  $\forall k \in \mathbb{N}^*, left[k] \prec right[k]$ . Similarly, let us consider the transitive and reflexive binary relation on  $\mathcal{C}$  called *isSubclockOf* and denoted  $\sqsubset$ .  $left \sqsubset right$  (read *left* is a sub clock of *right*) means that for all  $k$ , the instant  $left[k]$  of *left* coincides with exactly one instant of *right*. More formally:  $left \sqsubset right$  means  $\forall k \in \mathbb{N}^*, \exists n \in \mathbb{N}^* | left[k] \equiv right[n]$ . The relation  $\sqsubset$  is order-preserving. All the coincidence-based relations are based on *isSubclockOf*. When both  $left \sqsubset right$  and  $right \sqsubset left$  then we say that *left* and *right* are synchronous:  $left \equiv right$ .

A CCSL specification consists of clock declarations and conjunctions of *clock relations* between *clock expressions*. A clock expression defines a set of new clocks from existing ones, most expressions deterministically define one single clock. An

example of clock expression is *delay* (denoted  $\$: \mathcal{C} \times \mathbb{N}^* \rightarrow \mathcal{C}$ ).  $c \$ n$  specifies that a new clock is created and is the exact image of  $c$  delayed for  $n$  instants:  $o = c \$ n$  defines a clock  $o \in \mathcal{C}$  such that  $\forall k \in \mathbb{N}^*, o[k] \equiv c[k + n]$ .

By combining primitive relations and expressions, we derive a very useful clock relation that denotes a bounded precedence.  $left \boxed{\prec_n} right$  is equivalent to the conjunction of  $left \boxed{\prec} right$  and  $right \boxed{\prec} (left \$ n)$ . The special case, when  $n$  is equal to 1 is called *alternation* and is denoted  $left \boxed{\sim} right$  (reads *left alternates with right*).

### 3 Conclusion

The UML Profile for MARTE is dedicated to the modeling and analysis of real-time and embedded systems. Its time model relies on a notion of clock borrowed from the synchronous languages [6] and their polychronous extensions [7]. Those clocks can be logical or physical. The time model also provides a support to build causal and temporal constraints to force the clocks to tick according to predefined patterns. Thus, the evolution of the clocks imposes an execution semantics on the underlying UML MARTE model. Whereas the MARTE time model provides the notions of clocks and constraints, its companion language, the Clock Constraint Specification Language provides a syntax to define the constraints themselves. This brief tutorial introduces the main concepts of MARTE time model and gives an overview of CCSL.

### Acknowledgments

This work has been partially funded by the PRESTO Project (ARTEMIS-2010-1-269362).

### References

1. OMG: UML Profile for MARTE, v1.1. Object Management Group. (June 2011) formal/2011-06-02.
2. OMG: UML Superstructure, v2.4.1. Object Management Group. (August 2011) formal/2011-08-06.
3. André, C., Mallet, F., de Simone, R.: Modeling time(s). In: 10th Int. Conf. on Model Driven Engineering Languages and Systems (MODELS '07). Number 4735 in LNCS, Nashville, TN, USA, ACM-IEEE, Springer (September 2007) 559–573
4. André, C.: Syntax and semantics of the Clock Constraint Specification Language (CCSL). Research Report 6925, INRIA (May 2009)
5. Mallet, F., André, C., de Simone, R.: CCSL: specifying clock constraints with UML/Marte. *Innovations in Systems and Software Engineering* 4(3) (2008) 309–314
6. Benveniste, A., Caspi, P., Edwards, S.A., Halbwachs, N., Le Guernic, P., de Simone, R.: The synchronous languages 12 years later. *Proc. of the IEEE* 91(1) (2003) 64–83

7. Le Guernic, P., Talpin, J.P., Le Lann, J.C.: Polychrony for system design. *Journal of Circuits, Systems, and Computers* **12**(3) (2003) 261–304
8. Mallet, F.: Logical Time in Model Driven Engineering. *Habilitation à diriger des recherches*, Université Nice Sophia-Antipolis (November 2010)

## Biography

Frédéric Mallet is an associate professor at Université Nice Sophia Antipolis. He is a permanent member of the Aoste team-project, a joint team between INRIA and I3S laboratory. He received a PhD in Computer Science in 2000 and his habilitation degree [8] in 2010. Since 2007, he has been heavily involved in the definition, finalization and revision of the UML Profile for MARTE<sup>1</sup> and has been a voting member of the successive finalization and revision task forces for MARTE at the OMG. His main research interests include the definition of models for the specification of functional and non-functional properties of real-time and embedded systems. He also develops tools and techniques for the validation and verification of such models.

---

<sup>1</sup> <http://www.omgmarte.org>