A New MARTE Extension to Address Adaptation Mechanisms in Scheduling View

Mohamed Naija(✉) and Samir Ben Ahmed(✉)

Laboratory of Computer for Industrial Systems, INSAT, Tunis, Tunisia
naija.mohamed@gmail.com, samir.benahmed@fst.rnu.tn

Abstract. The modeling of Real-Time Embedded Systems (RTES) is one of the biggest challenges facing designers of such systems. These systems are considered high-assurance since errors during execution could result in injury, loss of life, environmental impact, and financial loss. The addition of adaptability to RTES further hardens and delays their modeling and validating especially with the current lack of design models and tools for adaptive RTES. The profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE) defines a framework for annotating non-functional properties of embedded systems. In particular, the SAM (Schedulability Analysis Model) sub-profile offers stereotypes for annotating UML models with the needed information which will be extracted to fulfil a scheduling phase. However, SAM does not allow designers to specify data to be used in the context of adaptive systems development. It is in this context that we propose an extension for the MARTE profile, and especially the sub-profile Schedulability Analysis Modeling, to include adaptation mechanisms in scheduling view.

Keywords: Adaptability · Real-Time & Embedded Systems · MDE · MARTE · Scheduling analysis

1 Introduction

The modeling of Real-time & Embedded Systems (RTES) may be stated as a crucial problem in the software engineering domain. RTES are subject to a multitude of constraints (e.g., battery, temperature …) and real-time requirements. Thus, designers are encountering the challenge of resource limitations, time, highly variable environment, etc. The addition of adaptivity to such systems further hardens and delays their modelling and scheduling analysis especially with the current lack of design models and tools for adaptive RTES. Lightening the task of adaptive systems designers and reducing the development cost and time to market represent a major challenge in the field [1], which requires the use of high-level approaches such as MDE and MARTE [2].

MDE is a way to beat the growing complexity of real time systems and verifying their correctness. In particular, Unified Modeling Language (UML) profiles promote an adequate solution to support the whole lifecycle co-design of complex systems. In RTES domain, its adoption is seen promising for several purposes: requirements specification, behavioral and architectural modeling with their real time constraints and performance

L.A. Maciaszek and J. Filipe (Eds.): ENASE 2016, CCIS 703, pp. 27–43, 2016.
DOI: 10.1007/978-3-319-56390-9_2

issues. In this context, the profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE) fosters the building of models that support the specification of scheduling analysis problem. This profile has the capacity to model tasks, dependencies between them and events under shape a Workload Behavior of system. Subsequently, it promotes the validation of the system temporal accuracy. Unfortunately, MARTE does not define a clear semantics for modeling and analysis of the adaptation in RTES.

Thus, we propose in this paper the main changes to be made on MARTE profile for supporting adaptation mechanisms. These amendments affect mainly the stereotypes of MARTE/SAM (Scheduling Analysis Modeling) since it is the sub-profile intended to model the schedulability analysis.

Our contribution is to improve the meta-models of the existing annotations. We try to modify in the structure of existing annotations, by referring to their cardinality and by adding a new concept. This work is the result of previous investigations and published work. Starting from [3] a MARTE-based approach was proposed to concurrency model construction at early design stages. Moreover, we identify using Petri Nets formalism a new threading strategy in complex RTES. Particularly, we identified that operation (*saStep*) can be mapped in multiple Schedulable Resources. In [4], we have presented three technical reconfigurations of RTES to meet performance constraints. These software and hardware solutions are able to reduce the utilization factor of the processor by changing periods and deadlines of tasks, adjustment the frequency of the processors and software/hardware migration tasks. Notably, we perceive a MARTE semantics limitation in modeling level. Accordingly, we have identified the needed of the new version of the MARTE profile supporting adaptation mechanisms. In [1], we have proposed a new extension of MARTE to address adaptation in the scheduling view.

This work is to be integrated in a model-based approach to guide RTES designers for building and analysing adaptive RTES models. It facilitates complex systems modeling, reduces the development time and cost and improves software process quality. The above benefits have been illustrated through the application of our extensions to different examples of adaptive RTES [1].

The present paper is organized as follows. Section 2, introduces the concept of the adaptation. Section 3 surveys relevant related works in the adaptive RTES field. While a brief definition of the MDE paradigm, its MARTE profile and the SAM sub-profile is described in Sects. 4 and 5 present our research scope. Section 6 specifies our proposal. To better explain our contribution which is highlighted in Sect. 7, we rely on a case study. Finally, Sect. 8 concludes the paper and sketches some future work.

2 The Adaptability Concept

2.1 Definition

There are several definitions of adaptation in the literature. In [5], a software adaptation is defined as any software modification that changes the reliability or timeliness of the software without affecting other aspects of its functionality. Software adaptation encompasses many common software-tuning techniques. These include:

- resource reallocation, such as moving a software component from one processor to another,
- adjustments to processor schedules,
- modification of replication factors for N-modular-redundant software components,
- modification of retry limits or time-out periods for delivery of a service by a software component.

In [6, 7] adaptation means change in the system to accommodate change in its environment. More specifically, the adaptation of a software system (S) is caused by a change (Che) from an old environment (E) to a new environment (E'), and results in a new system (S') that ideally meets the needs of its new environment (E'). Formally, adaptation can be viewed as a function:

Adaptation: $E \times E' \times S \rightarrow S'$, where meet $(S', \text{need } (E'))$.

In [8] adaptive system is defined as a system that is able to change its structure or behavior at run-time in response to the execution context variations and according to adaptation engine decisions.

In our previous work [9], we define adaptation as any modification in the structure, behavior or architecture of the system to accommodate external or internal change of their operating environment or context and according to predefined adaptation plan and rules.

2.2 Axes of Adaptation

Several adaptation techniques are defined to manage reconfiguration in the software development lifecycle. In model-based approaches for RTES, these adaptation techniques can affect [9]: (i) the functional model in the case of a change in the behavior of the system, (ii) the platform model if there is an adjustment in performance of material resources or unavailability of a resource for a certain period at run-time, and (iii) the implementation model when a task migration between resources is required.

Possible changes in the system can be caused by external variation, such as change in the operational environment (e.g., airplane mode in smart phone) or internal variation (e.g., new requirement) [9]. The adaptation can be either static or dynamic. Since static adaptation requires stopping the system and restarting it with a new configuration. The dynamic adaptation is based on a set of predefined adaptation behavior, statically designed and verified at design time. At runtime, the system can select an alternative from the available ones in accordance with adaptation rules and context. The change undergone by the system can be qualified as partial or full adaptation. Full adaptation completely changes the initial configuration of the system, while partial adaptation concerns only one level of the system configuration while the remaining portions continue their normal execution.

Thus any software system could be reconfigurable in one or more axes of adaptation [9].

3 Related Work

The design of adaptive RTES presents many challenges due to the complexity of the problem it handles [10]. In the present paper, we limit our study to research works particularly tackling adaptive RTES using the MARTE profile.

Many researchers have benefited from the MARTE profile for the design and verification of adaptive RTES from high-level models. In [11] authors have benefited from MARTE to model reconfigurable architectures such as FPGAs based Systems-on-Chip (SOC). They extended the MARTE profile with some semantics and Xilinx specific concepts, which limits their applicability for diverse systems, to support Dynamic and Partial Reconfiguration (DPR) of FPGA. Unlike this contribution, we aim to propose a new extension to support adaptation which is independent from any specific platform.

In [12] the authors give a classification of 13 publications that have dealt with the subject of adaptation in the design approach. Following this classification, the authors illustrate using an avionic example the need for the validation of adaptation rules at design-time according to the real-time features of the system. In this context of verification approaches, they have proposed in [13] an MDE approach for modeling and offline validation of application timing constraints. In fact, this article uses state machine to represent the application configurations and transitions between them to represent adaptation rules. This work is based on the generation of all possible configurations of a system before running, in order to validate timing constraints. The number of configurations varies from one system to another and it can be very large, this combinatorial explosion makes the timing analysis inapplicable. Furthermore the proposed approach considers only periodic tasks and cannot be applicable to aperiodic and sporadic tasks.

Two major scheduling approaches are available in the literature: the partitioned and the global approaches. Originally, MARTE supports only the modeling of the systems to be scheduled according to the partitioned approach. In [14] the authors have proposed various updates for MARTE meta-models of specialization and generalization stereotype in order to support global scheduling approaches, allowing task migrations. Those changes allow a schedulable resource to be executed on different computing resources in the same period [15]. Unfortunately, extensions proposed in MARTE profile do not allow assessing the gain in time of an adaptation operation (task migration in this case).

In [10], five patterns have been proposed to model and evaluate adaptation. These design patterns are presented in a static form through class diagrams and stereotyped MARTE profile. In this work, adaptation is considered as a dynamic and partial change of the operating mode, without taking into account the platform adaptation which is essential in the verification of time constraints.

All the previously mentioned works are beneficial since they facilitate the design of adaptive real-time systems. However, they present some weaknesses. These research studies [10] are not sufficiently generic since they tackle a specific adaptation problem, which consequently compromises their reusability as well as their ability to adapt to new system requirements and constraints. Additionally, most of them only focus on the software side adaptation while ignoring the hardware and implementation adaptation which are essential in the design and analysis of complex systems.

Table 1 presents a classification of works around modeling and verification of adaptive real-time systems according to the level of adaptability considered and timing verification supported. As we can see in this classification, to the best of our knowledge, there is no work that deals with all axes of adaptability.

Table 1. State of the art classification.

Criteria	Related Work			
	[10]	[12]	[14]	[9]
MDE approach	+	+	+	+
Modeling software adaptability	−	+	−	+
Modeling hardware adaptability	+	−	−	−
Modeling implementation adaptability	−	−	+	−
Scheduling Analysis supported	−	+	−	−
Adaptation Type	Internal	Internal	Internal	External
	Dynamic	Static	Static	Dynamic
	Partial	Partial	Full	Full

4 MDE and RTES Development

The Model Driven Engineering (MDE) is a software development methodology aiming to increase the level of development and overcome the growing complexity challenge. It covers the entire systems lifecycle, simplifies the design process by using the concept of models and offers independency between different steps of development flow.

In the context of the schedulability analysis, MDE is mainly used in the modeling step and the transformation of scheduling analysis models to the models of the chosen scheduling analysis tool [15]. MDE uses the UML profile and especially MARTE.

4.1 MARTE Capabilities for RTES Modeling

MARTE is an extension of UML profile providing support for specification, modeling and verification step of real time and embedded systems. MARTE supports the modeling of software and hardware features at a high-level of abstraction. In addition, it offers a rich set of annotations for modeling schedulability analysis. This profile encompasses a lot of sub-profiles such as: SRM (Software Resource Modeling), HRM (Hardware Re-source Modeling), GQAM (Generic Quantitative Analysis Modeling), SAM (Schedulability

Analysis Modeling), PAM (Performance analysis Modeling), etc. In this paper, we will focus especially on the SAM sub-profile since it is the package affected by our proposal.

4.2 SAM

In order to establish an early validation of the system's temporal behavior a well-formed analyzable model, called SAM, is defined. It offers a variety of annotations related to temporal features. Thus this profile has the capacity to model tasks, dependencies between them and events. It has the capacity to predict if all tasks meet their time constraints by defining the workload behavior. This is a chain of operation activations representing executions scenarios for the application.

5 Scope of the Work in Relation to MARTE

Our research scope concerns mainly the usability of the UML/MARTE profile for modeling adaptive systems. In this section, we will discuss the need of a new version of MARTE on the three levels of modeling identified in Sect. 2.2: functional model, platform model and implementation model.

5.1 Adaptability in the Functional Model

To model the functional model of static systems, we use MARTE/SAM (Schedulability Analysis Model) capabilities which offer a variety of stereotypes for annotating models with real-time features. This profile has the capacity to model tasks, dependencies between them and events under shape a Workload Behavior of system. Subsequently, it promotes the validation of the system temporal accuracy by the construction of the end-to-end computation. The end-to-end computation represents the processing load of the system. It represents the different steps executed in the system and triggered by one or more external stimulus. «*saStep*» is a stereotype annotating an action/operation. A set of steps specify the so-called Schedulable Resources. This concurrency model is independent from any particular Real-Time Operating System (RTOS) in order to fulfill the MDA principals.

In adaptive system, additional information has to be modelled such as adaptation rules, transitional modes and conditions. Thus, we need to model all alternatives and possible variations of the system elements in order to validate the non-functional properties. Unfortunately, the designer is not able to specify all these properties with the actual version of MARTE.

5.2 Adaptability in the Platform Model

SAM platform is a package providing sufficient concepts to model a general platform, at a high-level of abstraction, for executing the functional model. It is a specialization of the sub-profile Generic Resource Modeling (GRM), which provides mechanisms to manage access to different execution resource. Originally it does not support modeling

of unavailability of resource for a certain period at run-time. This uncertainty is a main factor that can influence the effectiveness of the configuration and affecting its performance considerably [9].

Moreover in literature a popular alternative to static power management in RTES is to allow the speed factor to adjust dynamically to the number of requests in the system. Using the MARTE/SAM, the designer is able to specify these properties. But, for each adjustment, he must repeat the modeling of the same resource to specify the new features this is due to the multiplicity of the concerned attributes [9].

5.3 Adaptability in the Implementation Model

To this end, at this level, functional model (event, end-to-end flow, shared resources) and platform model (execution resources) are specified. To be executed, a software resource must obviously be allocated on processors or busses. This allocation model, called implementation model, is needed to have an estimation of execution time for tasks. Consequently, a schedulability analysis test can be carried out on this model. The task migration is considered as an adaptive technique that allows improving application performance and achieves optimality. Currently, MARTE do not support this dynamic allocation technique. Thus, task that can be across multiple processors for different periods of time is not permitted in MARTE.

5.4 Example

In this section we illustrate the kind of problem we want to solve. The following Fig. 1 shows a description of a basic execution scenario. This is an example of adaptation mechanism that cannot be modeled using the actual version of the MARTE profile.

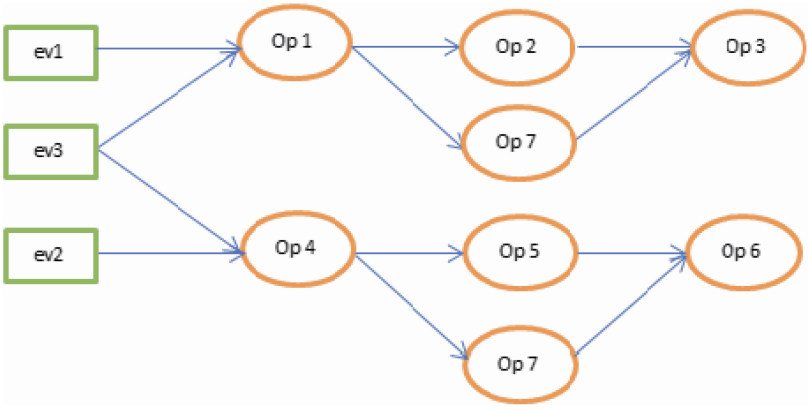

Fig. 1. Example of adaptation scenario.

The Workload Behavior of the system is activated by the both external events e1 and e2, initially. The event e1 (e.g., timers, internal event and external occurrences) triggers

the first behavior scenario of a system and precedes all the operations (Op1, Op2 and Op3). The event e2 triggers the second behavior scenario of the system composed of the related operations (Op4, Op5 and Op6). Afterwards, the adaptive event e3 (e.g., new requirement) triggers the both execution scenarios of the system, which is impossible to modeled with MARTE. Additional, the operation Op7 appears in response to the need of an adaptation. When occurs, Op7 affects the both execution scenarios, this is also not possible to do with originally version of MARTE.

6 Our Proposal: Sam Extension

The scheduling analysis modeling is performed through the MARTE/SAM profile.

The idea of performing scheduling analysis based on MARTE models assumes that all the information that is needed for the analysis is already part of the MARTE model [3]. In fact, SAM meta-model supports the modeling of different systems as it models all the temporal features needed in the scheduling step except those used to model adaptation constraints. Consequently, we seek to improve SAM meta-model in order to support modeling and early analysis of adaptation process. The amendments to be done on the SAM sub-profile affects also the GQAM sub-profile since some classes of the sub-profile SAM inherit from GQAM sub-profile.

6.1 Amendments in the Functional Model

In this level, we propose to modify in the classes Event, Step and EndToEndFlow.

Changes to be Done for the Workload Event. The workload behavior of the system [2] is characterized by their workload events and behavior scenarios. Workload events annotating *UML AcceptEventActions* introduce the semantic of event sequence arrivals for the execution of each *callBehaviorAction*. Originally event triggers only one behavior scenario. When adaptation is required, an event triggers all the behavior scenarios of the system workload. For example, let us imagine that for an adaptive event that denoted a low level of battery, this can affect more than one behavior scenario. Thus, the designer is not able to specify this property. Hence, we propose to modify in the association linking the two classes *WorkloadEvent* and *BehaviorScenario* of the package GQAM Resources (Figs. 2 and 3, reproduced from [1]). The multiplicity [1..*] denotes that an event can affect one or more behaviour scenarios.

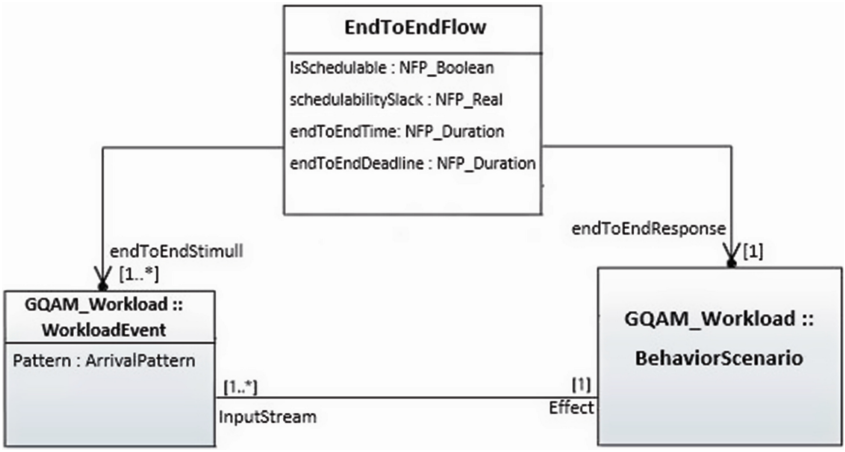

Fig. 2. The old meta-model of the GQAM package [1].

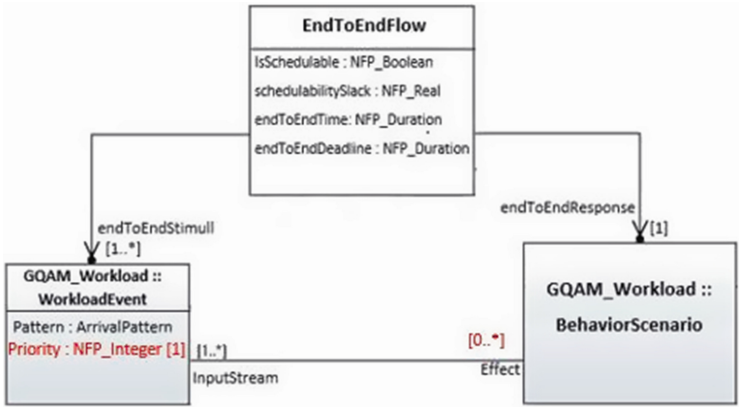

Fig. 3. The new meta-model of the GQAM package [1].

At run time, multiple events can occur simultaneously and must to be managed. Consequently, we propose to add the attribute ≪priority≫, which indicates the priority of the event. This additional real-time features, allows concurrency management. Indeed, GQAM is a generalization of the package SAM. So, this change will be inherited by SAM.

Changes to be Done for the Step. The class *Step* may represent a small segment of code execution [2]. It contains a lot of attributes specifying the temporal features of software resources. In UML MARTE model step can be part of only one behavior. Otherwise, in the case of adaptation process a new *Step* can appear in multiple behavior scenarios to manage adaptation. Thus, MARTE/SAM doesn't allow this specification. Thereby, we propose to change in the association linking the two classes *BehaviorScenario* and *Step* (Fig. 5).

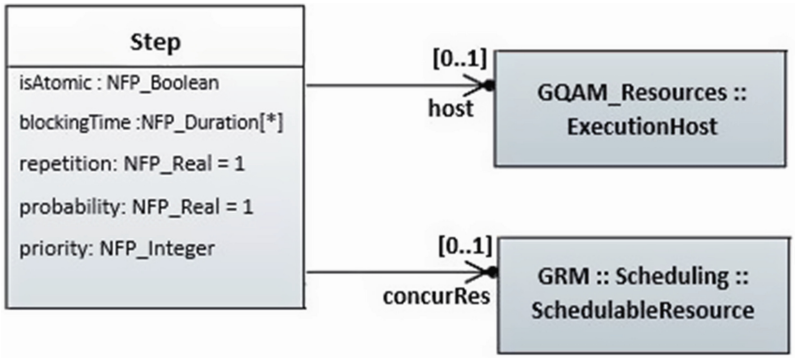

Fig. 4. The old meta-model of the SAM package [1].

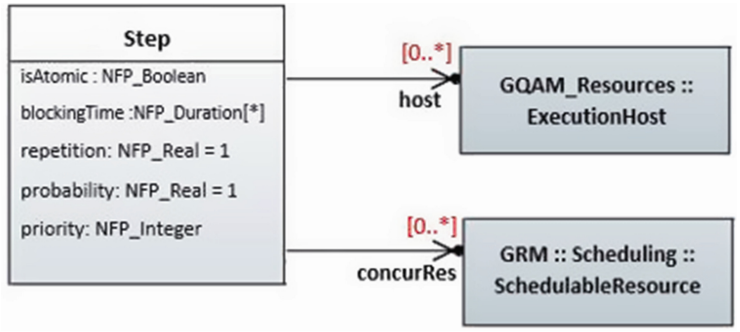

Fig. 5. The new meta-model of the SAM package [1].

Once the workload behavior is performed, it is necessary to identify the so-called *schedulable resources* (called tasks in scheduling literature). Schedulable resources are defined by mapping the execution of the end-to-end computations to them, in order to generate the task model. Different types of mapping exist in the literature [3, 16, 17]. As explained previously, a software resource can be mapped into more than one thread. Accordingly, the cardinality of the association between *Step* and *SchedulableResource* must be [0..*] (Fig. 4 reproduced from [1]).

Changes to be Done for the End-To-EndFlow. In adaptive systems, we model all alternatives and possible variations of the system elements. Moreover, the modeling step is based on the concept of mode (end-to-end Flow) which is a subset of system features: when the system is in a given mode, it provides this subset of features. We need to build a model for the source mode and a model for the target mode. The source and target models should not include information about each other, or about the adaptation. In addition, event signals are used between models of source and target to define transitional modes (Fig. 6).

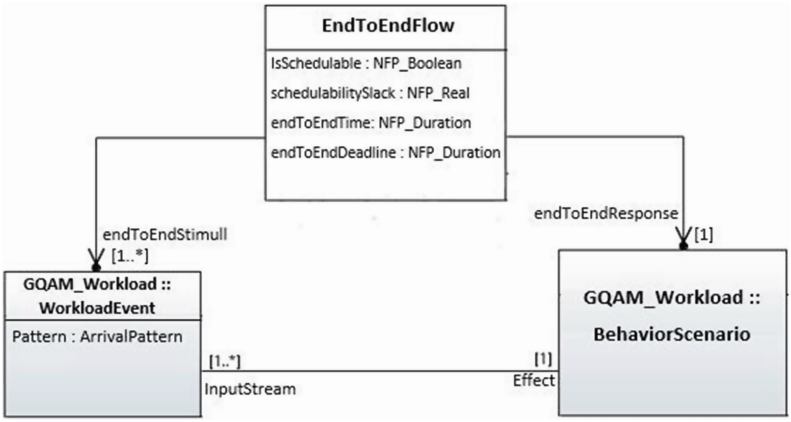

Fig. 6. The old meta-model of the GQAM package.

After identifying modes, it is necessary to specify adaptation rules. These are conditions that should be respected during and after adaptation step. In this context, we propose to extend the meta-model of SAM by the class *Rules*. This extension allows the modeling of the conditions that trigger modes and limit changes (Fig. 6).

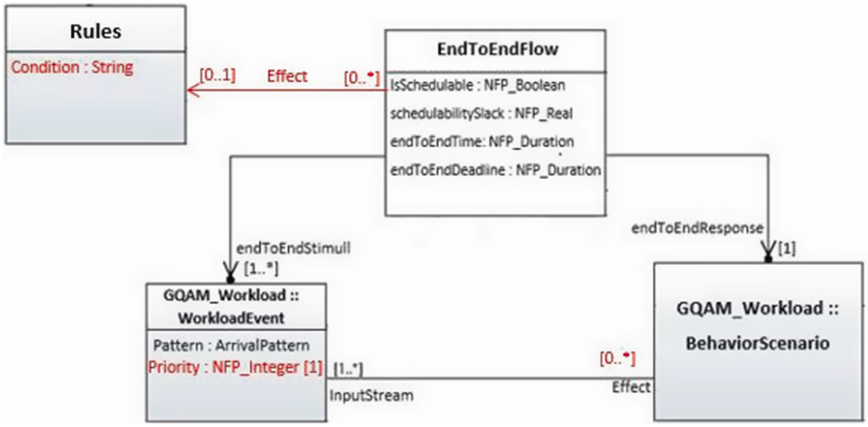

Fig. 7. The new meta-model of the GQAM package.

6.2 Amendments in the Platform Model

To this end an abstracted view of the execution platform resources is assumed to have execution time estimation for steps. Thus, the processor resources are represented as components with the «ExecutionHost» stereotype. To be executed, a software resource must obviously be mapped on processors or busses. Involved shared resources should also be described.

At run time, an execution resource can be unavailability for a certain period. This uncertainty is a main factor that can influence the effectiveness of the configuration and affecting its performance considerably. Originally this constraint is not support in MARTE. So, we propose to modify in the «ExecutionHost» stereotype by adding the attribute *unavailabilit*y. This allows specifying the duration of resource unavailability (Fig. 8).

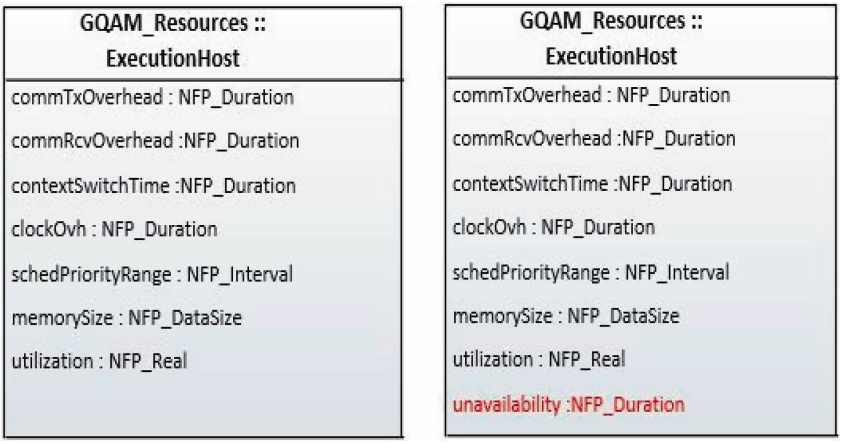

Fig. 8. Amendments in the meta-model of the GQAM package.

6.3 Amendments in the Implementation Model

In literature, three scheduling approaches are presented: the partitioned, the semi-partitioned and the global approaches [18]. Regarding the partitioned approach, it affects each task to be executed on one processor. Accordingly, tasks are not allowed to migrate between processors [14]. CPU utilization is therefore not optimal. As for the global approach and semi-partitioned, they enable a tasks migration such that schedulable resource may be allocated, not simultaneously, on different computing resources.

The task migration is considered as an adaptive technique that allows improving application performance and achieves optimality. Currently, MARTE/SAM supports only the partitioned approach. Thus, task that can be across multiple processors for different periods of time is not permitted in SAM. Subsequently, the multiplicity of the attribute corresponding to the execution resource (ExecutionHost) on which a task (schedulableResource) is allocated must be [0..*] instead of [0..1] (Figs. 9 and 10). This extension is adopted from the research work [19].

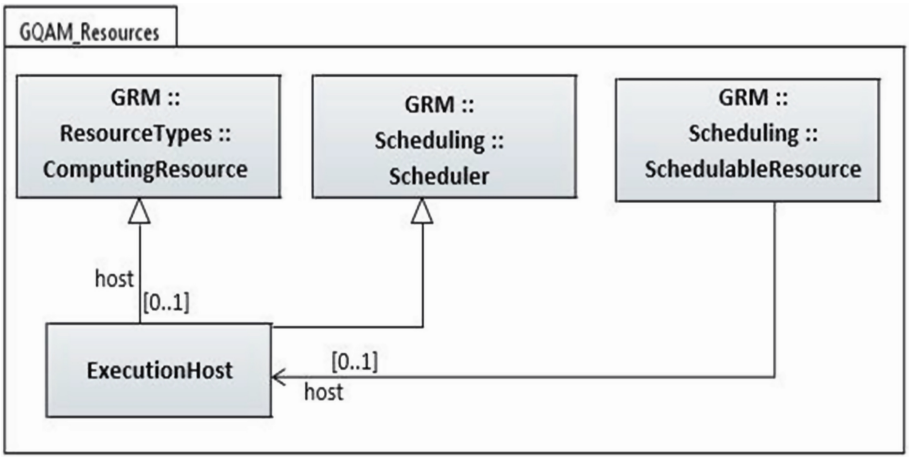

Fig. 9. Meta-model of the GQAM package [1].

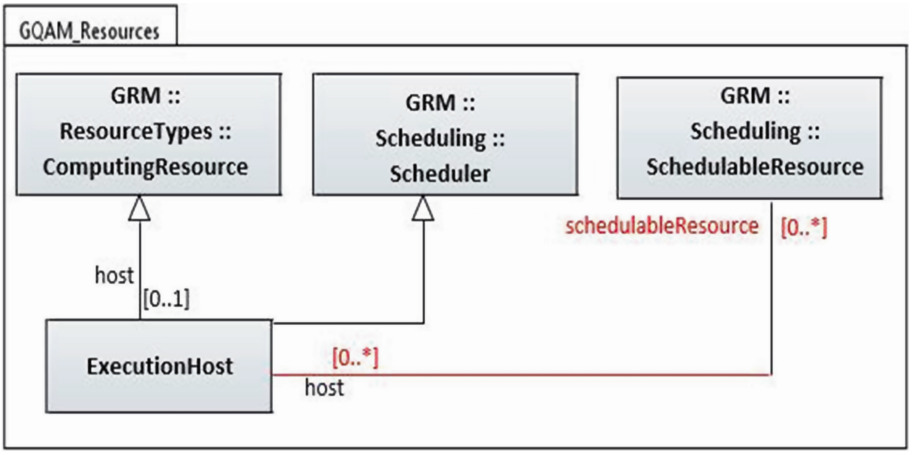

Fig. 10. Meta-model of the GQAM package with amendments [1].

While migrating from one processor to another, the execution time of a task is not the same, then the attribute «deadline» of the stereotype *SaStep* should have a multiplicity of [0..*]. In the same vein, a task can be interrupted several times during one period. Consequently the attribute «preemptT», which refers to the length of time that the step is preempted, must have a multiplicity [0..*] instead of [0..1]. Similarly for the attribute «readyT» which indicate length of time since the beginning of a period. Hence, this attribute must have a multiplicity of [0..*]. The set of values for the attributes «deadline», «preemptT» and «readyT» must be ordered (Fig. 11). This extension is adopted from the research work [15].

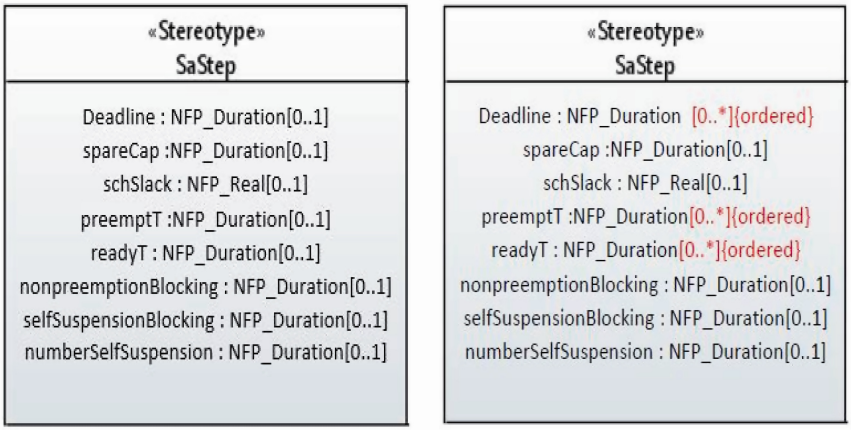

Fig. 11. Amendments in the meta-model of the GQAM package [1].

7 Case Study

To better explain our proposal, we use a FESTO [20] production system as an intact running application in this paper. It is a well-documented laboratory system used by many universities for research and education purposes.

The working process of FESTO is composed of three units: the distribution unit, the test unit, and the processing unit. The distribution unit consists of two steps: a pneumatic feeder and a converter. It forwards cylindrical workpieces from a stack to the testing unit. The test unit consists of three steps: the detector, the tester, and the evacuator. It performs the checking of workpieces for their height, material type, and color. Workpieces that pass the test unit successfully are forwarded to the rotating disk of the processing unit, where the drilling of workpieces is done. The result of the drilling operation is next checked by a checker and finally the finished product is removed from the system by an evacuator.

Note that in this work two drilling machines Drill1 and Drill2 are used to drill workpieces. Drill1 is used in case of medium production. When high production is required, Drill2 is recommended. According to user requirements, the system FESTO is able to reconfigure automatically at run-time in response to any changed working environment caused by errors or new requirements to improve system performance without a halt. The workload behavior in Fig. 9, reproduced from [1], represents the processing load of the system, founded on our proposal.

After identifying the behavior model of the system, it is necessary to specify the so-called *schedulable Resources*. For sake of simplicity, we use in this paper the scenario-based mapping [16] which is also one of the most used. The idea is to regroup all the operations executed at the same rate and belonging to the same linear end-to-end computation to the same task. In our FESTO system, we obtain three different threads namely task1 (*pieceEjection*, *Convert*, *Test* and *Evacuate*), task2 (*pieceEjection*,

Convert, *Test*, *Elevate, Rotate*, *Drill1*, *Checker* and *Evacuate*) and task3 (*pieceEjection*, *Convert*, *Test*, *Elevate*, *Rotate*, *Drill2*, *Checker* and *Evacuate*) (Fig. 12).

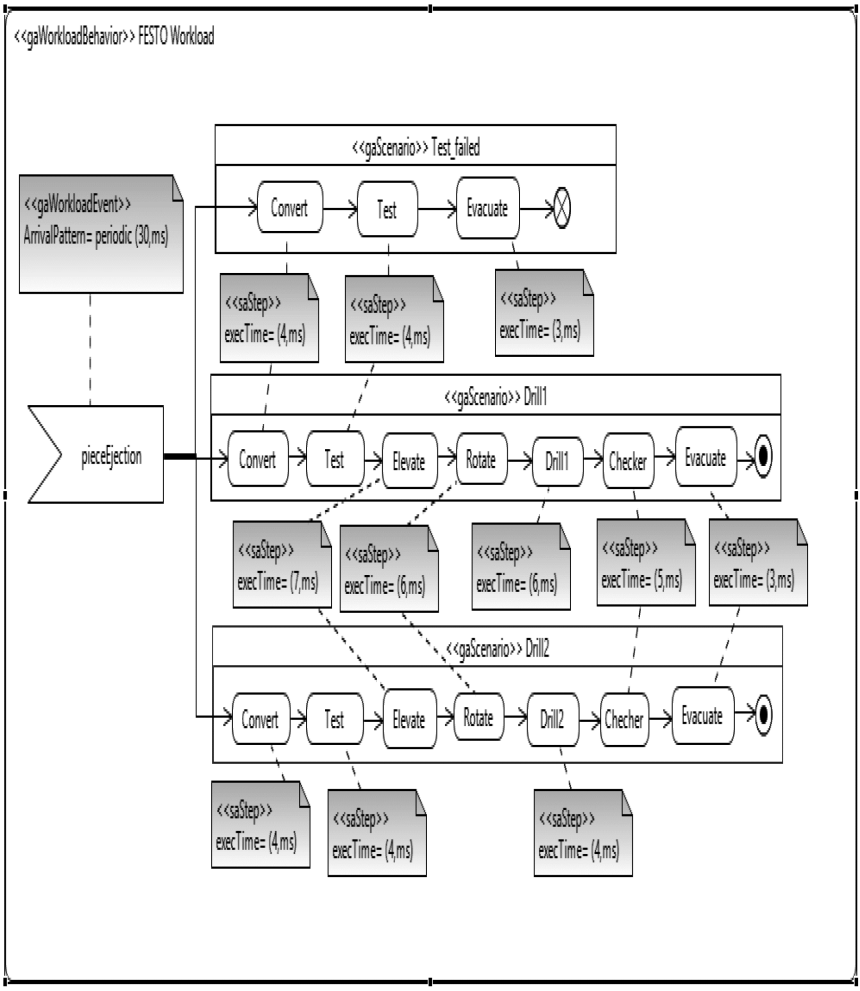

Fig. 12. The workload behavior of FESTO system.

Following this case study, founded on our proposal. We illustrate that an event (e.g., *epieceEjection*) triggers all tasks. The steps *Convert* and *Test* are part of three tasks. Similar for steps *Elevate*, *Rotate*, *Checker* and *Evacuate*, those participate for the execution of both schedulable resources Task2 and Task3. Compared to original version of MARTE, specifying these properties is not permitted. Note that, tasks can have dynamic properties (*readyT*, *preemptT* and *deadlines*) due to the concept of task migration, but we use the same corresponding values to facilitate our example. Anyway, we can add the different values and they will be ordered to perform scheduling analysis. After

scheduling all tasks, we can specify in our SAM view the used allocations through the attributes *Host: GaExecHost* and the corresponding attribute *ExecT: NFP_Duration*.

8 Conclusion

This paper focused on the modeling of adaptability requirements of RTES, which is judged a hard engineering task, using high-level approaches. In the same context, MARTE facilitates the modeling of RTES thanks to the set of stereotypes that it offers. This profile does not allow modeling and analysis of adaptive systems. To solve this issue, we proposed an extension for MARTE profile and especially for SAM sub-profile to makes MARTE able to stand adaptability. The benefit of our approach is the ability to model adaptive properties which will be extracted, to serve during the scheduling step. Our proposal has already been performed on the papyrus tool, which is an editor of MARTE-based modeling, and validated through a case study.

As future work, we will investigate in exploiting these extensions in a new design pattern providing support for modeling adaptive RTES.

References

1. Naija, M., Ahmed, B.S.: Extending UML/MARTE-SAM for integrating adaptation mechanisms in scheduling view. In: 11th International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE 2016, pp. 84–90 (2016)
2. OMG Object Management Group: A UML Profile for MARTE: Modeling and Anal sis of Real-Time Embedded systems, Beta2, Object Management Group (2008)
3. Naija, M., Ahmed, B.S., Bruel, J.-M.: New schedulability analysis for real-time systems based on MDE and petri nets model at early design stages. In: 10th International Conference on Software Engineering and Applications, ICSOFT 2015, pp. 330–338 (2015)
4. Naija, M., Ahmed, B.S.: Aid to design and reconfiguration of the MPSOC architectures. In: IEEE Computer Society 45th International Conference on Computers & Industrial Engineering, CIE 2015 (2015)
5. Bihari, T.-E., Schwan, K.: Dynamic adaptation of real-time software. ACM Trans. Comput. Syst. **9**, 143–174 (1991)
6. Lehman, M., Ramil, J.: Towards a theory of software evolution – and its practical impact (working paper). In: Invited Talk, Proceedings International Symposium on Principles of Software Evolution, pp. 2–11 (2000)
7. Subramanian, N., Chung, L.: Architecture – driven embedded systems adaptation for supporting vocabulary evolution. In: Proceedings of International Symposium Principles of Software Evolution International (2000)
8. Oreizy, P., Gorlick, M.M., Taylor, R.N., et al.: An architecture based approach to self-adaptive software. IEEE Intell. Syst. Appl. **14**(3), 54–62 (1999)
9. Naija, M., Bruel, J.-M., Ahmed, B.S.: Towards a MARTE extension to address adaptation mechanisms. In: 17th IEEE International Symposium on High Assurance Systems Engineering, HASE 2016, pp. 240–243 (2016)
10. Said, M., Kacem, Y.M., Kerboeuf, M., Amor, N.B., Abid, M.: Design patterns for self-adaptive RTE systems specification. Int. J. Reconfigurable Comput. **8** (2014)

11. Cherif, S., Quadri, I.R., Meftali, S., Dekeyser, J.-L.: Modeling reconfigurable Sytems-on-Chips with UML MARTE profile: an exploratory analysis. In: Proceedings of the 13th Euromicro Conference on Digital System Design, pp. 706–713 (2010)
12. Boukhanoufa, M.-L., Radermacher, A., Terrier, F.: Towards a model-driven engineering approach for developing adaptive real-time embedded systems. In: New Technologies of Distributed Systems, pp. 261–266 (2010)
13. Boukhanoufa, M.-L., Radermacher, A., Terrier, F.: Offline validation of real-time application constraints considering adaptation rules. In: International Conference on Trust, Security and Privacy in Computing and Communications, pp. 974–980 (2011)
14. Magdich, A., Kacem, Y.H., Kerboeuf, M.: A UML/MARTE-based design pattern for semi-partitioned scheduling analysis. In: International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, pp. 300–305 (2014)
15. Magdich, A., Kacem, Y.H., Mahfoudhi, A., Abid, M.: A MARTE extension for global scheduling analysis of multiprocessor systems. In: International Symposium on Software Reliability Engineering, pp. 371–379 (2012)
16. Masse, J., Kim, S., Hong, S.: Tool set implementation for scenario-based multithreading of UML-RT models and experimental validation. In: Proceedings of the 9th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2003. IEEE Computer Society (2003)
17. Mraidha, C., Tucci-Piergiovanni, S., Gerard, S.: Optimum: a marte-based methodology for schedulability analysis at early design stages. ACM SIGSOFT Softw. Eng. Notes **36**(1), 1–8 (2011)
18. Muhammad, K.B., Cécile, B., Michel, A.: Two level hierarchical scheduling algorithm for real-time multiprocessor systems. J. Softw. **6**(11), 2308–2320 (2011)
19. Magdich, A., Kacem, Y.H., Mahfoudhi, A.: Extending UML/MARTE-GRM for integrating tasks migrations in class diagrams. In: International Conference on Software Engineering Research, Management and Applications, pp. 73–84 (2013)
20. Khalgui, M., Hanisch, H.M.: Automatic NCES-based specification and SESA-based verification of feasible control components in benchmark production systems. Int. J. Model. Ident. Control **12**(3), 223–243 (2011)

http://www.springer.com/978-3-319-56389-3

Evaluation of Novel Approaches to Software Engineering
11th International Conference, ENASE 2016, Rome, Italy, April 27–28, 2016, Revised Selected Papers
Maciaszek, L.A.; Filipe, J. (Eds.)
2016, XII, 245 p. 91 illus., Softcover
ISBN: 978-3-319-56389-3