Critical evaluation of two UML profiles for Distributed Embedded Real-Time Systems Design

Muhammad Waqar Aziz, Radziah Mohamad and Dayang N. A. Jawawi

Universiti Teknologi Malaysia
wamuhammad2@live.utm.my, radziahm@utm.my, dayang@utm.my

Abstract

The successful implementation of Service-Oriented Computing (SOC) for the development of Distributed Embedded Real-time Systems (DERTS) in the recent years has proved the importance of service-orientation over previous paradigms. However, efforts are still needed on systematic service-oriented design of DERTS, especially on the modeling of DERTS. Several UML profiles are proposed for modeling of embedded real-time system and SOC separately. However, a holistic modeling solution, covering both the domains, is still missing. Therefore, a thorough investigation of existing UML profiles is necessary before proposing the holistic modeling solution. In this regard, this paper investigates two UML profiles: MARTE and SoaML profiles from the view of service-oriented development of DERTS. The paper highlights the portions of the two profiles that can be adopted and provides some suggestions that can be used for service-oriented modeling of DERTS. The results of this investigation would be useful in developing a service-oriented based meta-model for DERTS.

Keywords: Distributed Embedded Real-Time Systems, UML profiles, MARTE, SoaML

1. Introduction

The increase in use of ubiquitous computing environments constantly demands new methods and techniques for software modeling and design. The concepts provided by Service Oriented Computing (SOC) such as reusability, interoperability and loose coupling allow its successful application in many domains including the development of Distributed Embedded Real-time Systems (DERTS) [1-4]. However, mostly SOC has been applied in an ad hoc manner or the focus is on the implementation and very less attention has been given to systematic design especially modeling.

The Unified Modeling Language (UML) [5], de facto standard for software specification, is too general to be used for domain-specific modeling. However, UML provides extension mechanisms - the Profile concept - to specialize its elements for particular domains. Several UML profiles are developed to extend it for domain-specific modeling; some of them are adopted by the Object Management Group (OMG) as standard. In this trend, several UML profiles are proposed for embedded systems, real-time systems and SOC, e.g., UML-MARTE [6], UML-RT [23] and UML SoaML [7]. However, the service-oriented development of DERTS requires the modeling of both the service-oriented and DERTS concepts; whereas the existing profiles provide the modeling facility that covers only a single domain, thus, can partially support the service-oriented modeling of DERTS. For example, the UML-MARTE profile allows the modeling and analysis of many concepts of embedded real-time systems such as soft/hardware resources, time, and Non-Functional Properties (NFP) [6]. However, it does not cover the modeling of SOC concepts. Similarly, UML-RT allows the designers to model complex, event-driven DERTS but lacks in providing service-oriented support. On the
other hand, SoaML profile provides a way to model Service-Oriented Architecture (SOA) using UML [7], but it does not deal with the modeling of embedded real-time systems.

Before proposing a holistic modeling mechanism that includes the embedded, real-time and service-oriented concepts, it is necessary to investigate the existing profiles to find out (i) which portion of the existing profiles can be adopted in the holistic solution and (ii) whether the multi-domain concepts can be modeled by integrating the existing profiles. To this end, two of the standard UML profiles: Modeling and Analysis of Real-Time and Embedded systems (MARTE) [6] and Service oriented architecture Modeling Language (SoaML) [7] are investigated in this paper. The purpose of this investigation is to reuse the concepts of these profiles and to propose an enhancement in the two profiles for service-oriented DERTS development.

This paper deals with the critical evaluation of Time profile of UML MARTE and of SoaML profile for the development of DERTS. In addition, the paper provides some suggestions about SoaML concepts to be used in DERTS modeling. The paper is organized as follows: Section 2 provides the methodology of selecting existing UML profiles from the literature, which is followed by the overview of existing UML profiles in Section 3. Section 4 and 5 provide the overview of UML MARTE and SoaML profiles respectively. Both the profiles are evaluated in Section 6. The discussion and recommendations are provided in Section 7 and the paper is concluded in Section 8.

2. Methodology

Although several UML profiles are developed for embedded real-time systems and SOC in the past decade, some of the profiles were selected for investigation in this work. The UML profiles were selected based on the inclusion and exclusion criteria defined in the guidelines for performing systematic literature review [20]. The criteria are defined in Table 1.

<table>
<thead>
<tr>
<th>Included</th>
<th>Excluded</th>
</tr>
</thead>
<tbody>
<tr>
<td>The conference papers, journal papers and other electronic articles</td>
<td>Non-English papers</td>
</tr>
<tr>
<td>A UML profile for Service-Oriented, embedded real-time system modeling</td>
<td>Duplicate publication on same modeling</td>
</tr>
<tr>
<td>The material published from 2003 to date</td>
<td>Letters and editorials</td>
</tr>
</tbody>
</table>

3. Overview of existing profiles

3.1. UML profiles for Embedded Real-Time Systems

In this section, an overview of selected UML profiles for embedded real-time systems is provided. The Real-Time UML (RT-UML) profile allows the application of UML to model embedded and real-time systems. However, RT-UML does not allow users to define new annotations, or other analysis techniques [21]. Embedded UML [22] is a profile for specification, design and verification of embedded real-time systems. It uses the best concepts of UML, Real-Time UML, functional-architecture co-design and platform based design. The profile, however, is concerned with concurrency, communication and implementation issues
and lacks the representation of service-oriented concepts. UML-RT profile [23] is a modeling language of its own that allows the designers to model complex, event-driven DERTS, rather than just annotating the design models with quantitative information. However, the profile does not support time and time constraints modeling [24], is difficult to learn [25] and has limited modeling capabilities for performance and architecture. Hence, it should be used complementary to UML SPT profile [26].

The UML Profile for Schedulability, Performance and Time (SPT) [9], provides concepts of time, model-based performance and schedulability analysis, and notations for building real-time systems models. However, it has shortcomings in expressive power and flexibility [10]; it is extended from UML 1.4 and does not specify a full methodology. That’s why UML SPT is replaced by the UML Modeling and Analysis of Real-Time and Embedded systems (MARTE) profile [6]. MARTE profile is used for modeling and analysis of real-time and embedded systems. It extends UML 2.0 to cover the aspects of time, hardware and software resources and NFP; however, it does not deal with the service-oriented concepts and cannot be used for modeling of dynamic composition as it provides the static and pre-defined set of configuration [27].

3.2. Service-oriented UML profiles

The selected UML profiles for SOA are over viewed in this section. A UML profile for SOA concepts representation is proposed in [28] that considers Model Driven Architecture (MDA) principles. This profile provides a big picture of SOA by presenting only the major elements such as provider, requester and registry. However, the details and representation of other important service concepts such as service description and message is missing. López-Sanz, et. al., [30] proposed another UML profile in MDA (Platform Independent Modeling-level) and its corresponding meta-model for SOA modeling, however, in the profile the service interface and NFP are not discussed. The UML 2.0 profile for software services [31] is a generic model representing a small number of SOA concepts. UML profile for SOA [32] is very close to implementation and is not suitable for generic SOA platform independent development. An SOA representation of service, messages and connectors is presented in [33]. The solution is more focused on modeling the NFP. Baresi, et al., [34] proposed a UML model to represent SOA static concepts. Since this model considers the service providers, requesters and discovery agencies necessary for SOA development, it cannot be generalized to other execution platforms. The UML profile for SOA (UML4SOA) [8] shows how to model services and service composition in UML. The activity diagram is used for behavioral modeling in UML4SOA. SoaML [7] is a recent effort by the OMG that is used to provide modeling solutions for SOA using UML. It covers all the basic concepts of SOC but lacks in the representation of embedded real-time systems.

In this paper, two profiles, i.e., UML-MARTE and UML-SoaML were selected for investigation from all of the above mentioned UML profiles. The reasons for their selection are as follows:

i) These are the state-of-the-art UML profiles in their respective domains.

ii) These profiles are standardized by OMG.

iii) MARTE is widely accepted in industry for DERTS development.

iv) MARTE can be applied in the whole software development life cycle.

v) SoaML is a standardized meta-model for describing and formalizing SOA.
vi) Although available as a beta version, SoaML is becoming increasingly accepted and employed. A brief overview of both of these profiles is provided in the next sections.

4. MARTE: The UML profile

The UML MARTE profile allows the modeling and analysis of embedded real-time systems. The modeling part covers the specification and detail design of embedded and real-time characteristics of a system while the analysis part allows for performing any kind of analysis especially performance and schedulability on these models [6]. MARTE provides support in the specification, design and verification phases [6]. MARTE is the replacement of the UML-SPT profile [9]. Although, UML-SPT provided concepts for time, model-based performance and schedulability analysis, it is less expressive and flexible [10] and is extended from UML 1.4.

The key features of MARTE are the modeling of NFP, time and resources, the definition of concepts for soft/hardware modeling and their allocation and the support for performing quantitative analysis [11]. To provide these capabilities the MARTE specification includes four main packages: foundations, design, analysis and annexes [10].

The foundations package allows for NFP Modeling, Time Modeling, Generic Resource Modeling (GRM) and Allocation Modeling (Alloc). The design package consists of Generic Component Model (GCM), High-Level Application Modeling (HLAM) and Detailed Resource Modeling (DRM) sub-profiles, which further contains Software Recourse Modeling (SRM) and Hardware Recourse Modeling (HRM) sub-profiles. The analysis package includes Generic Quantitative Analysis Modeling (GQAM), Schedulability Analysis Modeling (SAM) and Performance Analysis Modeling (PAM). The annexes package contains Value Specification Language (VSL) as an expression language and Repetitive Structure Modeling (RSM) sub-profiles. For the details of each sub-profile, the reader is referred to [6]. The main benefits of MARTE are the modeling of both software and hardware of real-time embedded systems; improve communication between developers and construction of models for quantitative predictions [12].

5. UML-SoaML profile

UML-SoaML provides a standard way to architect and model SOA using UML [7]. It uses the basic UML mechanisms for modeling composite structures enhanced with its stereotypes to model structural service. The goals of SoaML are to support service modeling and design activities while following a model-driven development approach. Moreover, this profile supports the specification of services, service interfaces and service implementations for SOAs [7]. SoaML supports both the “Service Contract” and the “Service Interface” based approaches for specifying services [7]. With SoaML, SOAs can be specified at two levels of granularity: Community-level and Participant-level architecture. Community-level architecture is a top-level view of how participant collaborates, while the Participant-level architecture shows the collaboration of internal parts of a participant [13].

The main benefits of SoaML are to provide a higher level of abstraction, to enable interoperability and integration at the model level, to enable SOA both on and among existing platforms and integration with existing OMG standards [14]. However, SoaML only specify a language, i.e., a meta-model and UML profile, for service design and does not provide a methodology for developing services [13].
6. Profiles Evaluation

Besides the promising solutions provided by both MARTE and SoaML, the suitability of these profiles for service-oriented DERTS development need to be further evaluated. In this section, both MARTE and SoaML are evaluated to check the degree of their suitability for being used in service-oriented DERTS development.

6.1. The evaluation criteria

For service-oriented DERTS development, a modeling mechanism is needed that supports the concepts of both embedded real-time systems and service-oriented domains. To evaluate the modeling capabilities of both profiles that can be used in service-oriented DERTS development, the evaluation criteria, shown in Table 2, were used. The criteria were designed to cover different aspects such as model-related criteria and supportive features criteria. It is important to note that each criterion is accompanied with an evaluation question.

Table 2. Evaluation criteria

<table>
<thead>
<tr>
<th></th>
<th>Ease of understanding: Are the concepts of the profile easy to understand?</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>Adaptability: In which application domain the profile can be used?</td>
</tr>
<tr>
<td>3.</td>
<td>Design expressiveness: How well the profile can express the concepts and the relationships between them?</td>
</tr>
<tr>
<td>4.</td>
<td>Real-time concepts support: Is the profile support the real-time concepts?</td>
</tr>
<tr>
<td>5.</td>
<td>Service-oriented concepts support: Is the profile support the service-oriented concepts?</td>
</tr>
</tbody>
</table>

Although the evaluation performed is qualitative, it can be used for comparative analysis in a quantitative manner. In this regard, the following four-point scale is used to formulate the above mentioned evaluation criteria:

- 0: no support is provided
- 1: some concepts need more elaboration, modification or omission
- 2: partial support is provided, where some concepts are missing
- 3: support is provided

6.2. MARTE evaluation

MARTE defines many concepts and elements that require much time to learn. The feasible approach is to select a subset of MARTE elements for a particular problem [15]. For example, the use of MARTE framework for annotating NFP of embedded systems to service-oriented models. One of the key features of MARTE is its support for modeling of NFP; therefore, MARTE profile was investigated to find the possible representation of some of the NFP. The NFP included deadline, period, response time and Worst Case Execution Time (WCET). These properties were selected as they cover the real-time (deadline, period and WCET) and performance (response time) requirements of DERTS. Moreover, deadline, period and WCET are needed to perform scheduling analysis [18].

MARTE profile is enriched with time modeling as it provides a broadly expressive time structure and representation of multiform time (chronometric, logical, discrete) that allows the modeling of timing requirements of DERTS. The two main time-related concepts provided by MARTE are TimedProcessing, to represent the duration of a behavior [19], and TimedEvent,
that represents an explicitly time bounded event [36]. In fact, MARTE Time Model is a set of clocks, i.e., time is modeled with the help of clocks. Both of TimedProcessing and TimedEvent concepts are bound to clocks and are represented by <<timedProcessing>> and <<timedEvent>> stereotypes to specify duration of behaviors and relating time event with clocks respectively [19]. It means that time events such as period, deadline and duration can be represented by <<timedEvent>>, whereas the duration of behavior can be represented by <<timedProcessing>> as shown in Figure 1. In other words, time modeling using MARTE allows a concrete representation of clocks in the UML model, thus, the duration of behaviors and the occurrences of events can be specified and deadline and period can be measured [18].

Figure 1. UML state machine annotated with MARTE information [19]

6.3. SoaML evaluation

SoaML describes the contents of service design [17], which are the participating element, required / provided service interface elements and service description, however if used for DERTS development, these concepts may have a different semantic and need to be evaluated. Therefore, the following concepts of SoaML were investigated in this paper: Service, Service Provider, Service Interface, Service Description and Service Composition.

In SoaML, the participant is the basic unit of functionality implementation. A port is a feature of a participant through which participant provides or consume services. Each port is typed with a service interface that specifies provided or required operations at that port. Service description explains the interaction among the participants. The service interaction can be specified as UML Interface, ServiceContract and ServiceInterface [7].

6.4. Comparative evaluation

Based on the evaluation criteria discussed in Section 6.1, the comparative analysis of both profiles for service-oriented DERTS development was performed. The results of comparative
analysis are provided in Table 3. With regards to *ease of understanding*, the representation of different types of clocks in MARTE makes the design models difficult to develop and understand. Furthermore, the use of the same <<timedEvent>> stereotype to represent *period* and *deadline* make it difficult to differentiate. For example, it is hard to tell that the <<timedEvent>>, used in Figure 1, is representing *period* or a *deadline*. On the other side, the concept of participant in SoaML is so general to represent a particular device as it has different scale representation ranging from people, to large organization to small components. It is also hard to differentiate between a participant internal and external to the organization. In addition, the specification of a service into three different types at the interface and the encapsulation of service description in the service specification at ports make the overall structure difficult to understand.

**Table 3. Results of comparative analysis**

<table>
<thead>
<tr>
<th>Evaluation Criteria</th>
<th>MARTE</th>
<th>SoaML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of understanding</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Adaptability</td>
<td>Embedded Real-Time Systems</td>
<td>Service-oriented development</td>
</tr>
<tr>
<td>Design expressiveness</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Real-time concepts support</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Service-oriented concepts support</td>
<td>0</td>
<td>2 (structural only)</td>
</tr>
</tbody>
</table>

With regards to the *design expressiveness*, SoaML does not illustrate the timing requirements of service and aspects related to embedded system and communication. Similarly, the idea to handle a miss deadline or error condition is not supported; the concept of representing the composite services, along with their atomic services, at the ports is not specified and the modeling of embedded system properties such as power consumption and memory allocation is not provided. Furthermore, the representation of service interface in terms of operations and their processing during the matchmaking process would consume more resources.

As already mentioned, MARTE allows modeling and analysis of many concepts of embedded real-time systems and it is suitable for designing of such systems, however, it does not cover the modeling of SOC concepts. Similarly, SoaML can be used in service-oriented application development as it allows modeling of SOA using UML, but it does not deal with the modelling of embedded real-time systems concepts. For example, SoaML does not provide the facility to model the timing requirements and other embedded real-time system properties. Moreover, SoaML only allows the structural modeling of software services and it does not cover the behavioral aspect. It means SoaML lacks in the ability to model service composition, which is a key aspect of service-orientation and the details of a composite service. Therefore, it cannot be used for the specification of real-time systems for which composition is a key aspect.

**7. Discussion and recommendation**

The enriched time model provided by MARTE makes it suitable for DERTS modeling, thus service-oriented design models for DERTS can be annotated with MARTE to represent the NFP especially the timing requirements. However, some concepts of MARTE need more elaboration, modification or omission, *e.g.*, Clocks and <<TimedEvent>>. We
suggest an explicit representation of NFP within the model elements to avoid the use of clocks and the ambiguity it creates. An example of such a representation is shown in Figure 2.

![Figure 2. Example of suggested representation](image)

Just like SoaML, we also suggest the use of UML to represent the concepts in service-oriented DERTS development. It is to gain the willingness of software designers to use the new approach. However, some of the concepts of SoaML such as participant, service specification need some elaboration and modification. It would be better to represent the service as UML component instead of at the port, as in SoaML, because service is a component or unit of functionality. Consequently, it would be better to represent the service provider at a higher level than component. UML package would be a better choice to represent service providers containing several components (services). A simple interface only representing the inputs and outputs of service internal operations would be better for DERTS. However, the modeling of service interface concept provided by SoaML can be adopted for service-oriented DERTS development.

We believe that the integration of UML MARTE and SoaML would still not be beneficial as these profiles are constructed for different domains with different purposes and design rationales. In addition, it is possible that different profiles when combined may be mutually inconsistent and may overlap in un-document ed ways [35].

8. Conclusion

Service-oriented DERTS development requires the modelling capabilities that cover both service-oriented and DERTS domains. UML MARTE profile not only provides support for modelling embedded real-time systems, but can also be used to annotate UML and service-oriented models. This paper evaluates MARTE in terms of its time modeling for DERTS and SoaML if used for DERTS development. A number of suggestions are presented for modelling of service-oriented concepts in DERTS development. The suggestions provided by evaluating both profiles can be used in developing a meta-model for service-oriented DERTS development.

Acknowledgements

We would like to thank the Malaysia Ministry of Higher Education for sponsoring the research and Universiti Teknologi Malaysia for providing the facilities and support for the research.

References


Authors

Muhammad Waqar Aziz has recently completed Doctor of Philosophy (Computer Science) from Universiti Teknologi Malaysia. He received MS-Software Engineering degree in 2009. He has been lecturer at University of Peshawar for more than eight years. His areas of interest are Service-Oriented Computing and Embedded Real-time software development.

Radziah Binti Mohamad is a senior lecturer at the Software Engineering Department, Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia. She received her Bachelor of Science (Hons) in Software Engineering from Sheffield Hallam University, Master of Science (Computer Science) and Doctor of Philosophy (Computer Science) from Universiti Teknologi Malaysia. She has been a lecturer at Software Engineering Department for more than ten years. Her research interests include service-oriented design, pattern-oriented design, formal methods, component-based software engineering and agent-oriented software engineering.

Dayang Norhayati Abang Jawawi is an associate professor at the Department of Software Engineering, Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia (UTM). She received her B.Sc. Degree in Software Engineering from Sheffield Hallam University, UK and M.Sc. and Ph.D research in software engineering from Universiti Teknologi Malaysia. Her research interests are software reuse, component-based reuse, software quality and embedded real-time software. Currently she is Head of Software Engineering Department, Faculty of Computer Science and Information Systems, UTM; and member of Software Engineering Research Group (SERG), K-Economy, UTM.