Timing Aspects Construction Using UML-MARTE Profile

Qingqing Sun, Xiaopu Huang, Jiangwei Li, and Tian Zhang*

State Key Lab. for Novel Software Technology, Nanjing University, Nanjing, China
Department of Computer Science and Technology, Nanjing University, Nanjing, China
{sunqq,hxp,ljw}@seg.nju.edu.cn, ztluck@nju.edu.cn

Abstract. Modern real time embedded systems are typically composed of multiple functional and nonfunctional concerns with nonfunctional concerns affect the former in many aspects. MARTE, an extension profile of UML2, aims to be the unified standard language for real time and embedded systems. Aspect-Oriented technology, as a complement to Object-Oriented technique, decomposes systems into distinct features by separating and modularizing crosscutting concerns. In this article, we illustrate how to use plenty of time modeling elements in MARTE profile to support comprehensive modeling of RTES. Similar to general crosscutting concerns, time concerns are often triggered at multiple concerns and tangled with other requirements. We try to deal with time as typically crosscutting concerns by AO technology. We practice these thoughts by means of examples and seek to explore an effective modeling mechanism using of both Aspect-Oriented methods and MARTE profile.

Keywords: MARTE, time, modeling, Aspect-Oriented technology, RTES.

1 Introduction

UML profile for MARTE (Modeling and Analysis of Real-Time and Embedded Systems) [1] is a new UML specification which adds capabilities for model-driven development of Real-Time Embedded Systems (RTES). This extension provides support for specification, design, and verification stages and intends to replace the existing UML profile for SPT. With plenty of time modeling elements, MARTE provides comprehensive support for modeling time of RTES.

The evolving cognitive complexity of large computer systems is a topic of utmost concern, and then the efforts on simplicity are required. In academic research, lots of specific approaches and architectures have already emerged, such as object oriented technology, design patterns and aspect-oriented techniques [2]. Each new methodology has presented new ways to decompose problems and concerns and allowed a more natural mapping from system requirements to programming constructs. But as complexity grew, either these methods had some limitations or raised new problems [3]. Theme approach is a typically aspect-oriented method. It performs well in modeling

* Corresponding author.

Y. Yuan, X. Wu, and Y. Lu (Eds.): ISCTCS 2012, CCIS 320, pp. 474–481, 2013.

the crosscutting concerns of RTES. In our opinions, modeling the time as an aspect with Theme approach and describing the details in MARTE will be a necessary step towards an appropriate way for modeling RTESs.

The rest of this article is structured as follows. Section 2 describes time relevant concepts in MARTE profile. Section 3 presents an aspect-oriented design approach: Theme approach. Section 4 illustrates how MARTE can aid time modeling how to model time concerns in embedded systems as aspects, while Section 5 concludes the article and outlines plans for future work.

2 The MARTE Time Structure

As a successor of SPT, MARTE must align with the new version UML2 which added many new meta-classes to represent time, duration and some forms of time constraints.

2.1 Time Structure and Accesses to Time

Time structure is well defined in the specification [1]. It can be viewed as the combination of a set of time bases and time structure relations. A time structure contains a tree of multiple time bases. A MultipleTimeBase consists of one or more time bases.

Clocks are usual devices to measure time in real world. In MARTE we adopt a more general point of view: a clock is a model element giving accesses to time, be it physical or logical. LogicalClock and ChronometricClock are two concrete subclasses of Clock. In the time domain view, the corresponding concepts in the UML view are ClockType and TimedDomain. Figure 1 shows their relationships.

2.2 Time Related Entities

TimeRelatedEntities package contains multiple entities such as TimedObservation and TimedProcessing. A TimedObservation is made in the runtime context of a (sub) system behavior execution. TimedProcessing is a generic concept for modeling activities that have known start and finish times, or a known duration, and whose instants and durations are explicitly bounded to Clocks.

3 Theme Approach

Aspect-Oriented Modeling (AOM) is an emerging solution for handling complexity of software models and application code [4] in which a system is viewed as a set of concerns. Recent work in AO design: [5], [6] and [7] have demonstrated the need to deploy this technology early in the software life cycle in order to utilize the technology to its full potential.

Theme approach is an approach for aspect-oriented analysis and design. It provides support for aspect-oriented development at two levels. At the requirements level, Theme/Doc provides views of requirements specification text. At the design level, Theme/UML allows to model features and aspects of a system, and specifies how they should be combined [8].

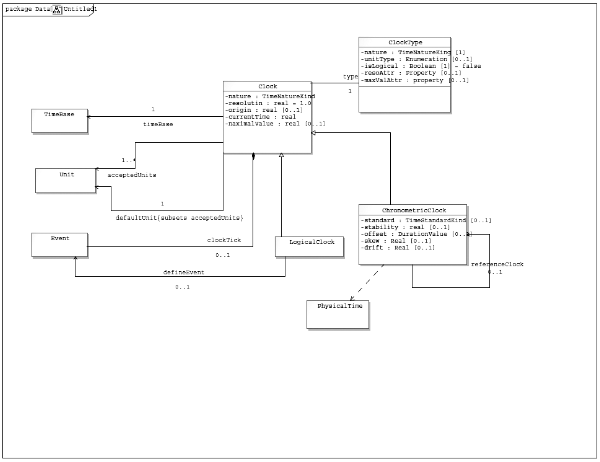

Fig. 1. The ClockType

3.1 Theme/Doc

Theme/Doc deals with the requirements specification in the following steps:

- To identify main themes in the system
- Ascertain whether a theme is an aspect or not

When ascertaining the aspects, there are many features that need to be considered such as whether we can split the requirements and estimate the triggers.

3.2 Theme/UML

For every identified theme, Theme/UML allows the developer to design them respectively in UML style. Base themes are designed in traditional UML style and aspect themes are designed by Theme/UML method. Besides, Theme/UML provides a template and binding paradigm to illustrate their relationships. We present an example of log behavior in figure 2 and figure 3.

Log file is a special in many systems, and log actions are trigged by many behaviors. Those triggers are modeled as LoggedClasses and they cooperate with log file. Figure 3 illustrates the binding mechanism.

4 MARTE Modeling with Timing Aspects

Theme approach handles system evolution which is common but important in software engineering effectively by the definition of themes. Traditional Object Oriented can hardly manage it properly. A theme is designed closed focusing only on its directly related entities. Then the system evolution is viewed just as new themes composing to the primary themes. These compositions are easy to achieve with respect to the unknown relationships brought by uncertain new entities and behaviors in OO technique.

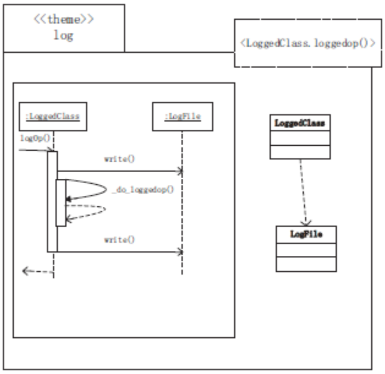

Fig. 2. Thelog theme

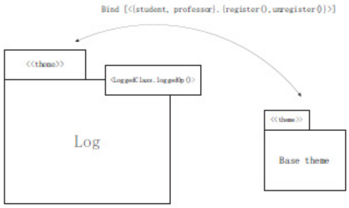

Fig. 3. The Aspect Theme

In RTESs, time concerns are common but also the key points. AO usually handles crosscutting requirements as aspect themes separating them at point cuts and weaving at join points. Time related requirements can also be treated like this. Time related entities are modeled as TimedClass and behaviors modeled as trigger actions of Timer class in Theme approach. After implementing separately, designers weave these themes to form systems.

MARTE offers the possibility to precisely describe RTES by means of its different and useful components [10]. Moreover, MARTE profile provides multiple time elements to make up existed UML2 profile. To model time concerns, we must first model time accesses: Clock. According to the system requirements, we set different precision clock using IdealClock and LogicalClock elements. We present the proposing phases in figure 4.

4.1 MARTE Clock Modeling

MARTE aims to be the unified standard language for RTES and provides support for specification, design and verification/validation stages for it. It extends the semantics of UML model by several kernel concepts of ClockType, Clock and TimeBase. Among these concepts, MultipleTimeBase are introduced to depict multiple types of clock and Clock to use to define synchronization and priority.

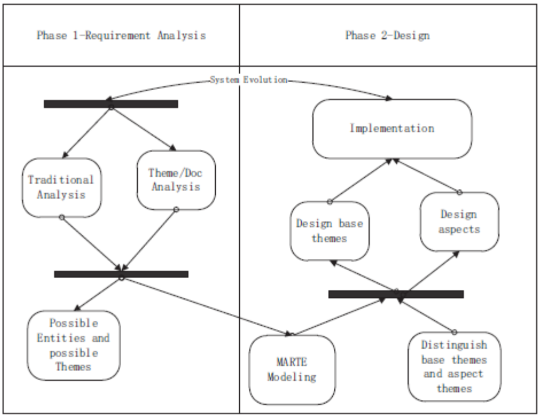

Fig. 4. Phases in MARTE Modeling with Theme process

4.1.1 Modeling Examples

Train-Bridge system, a typical real time and embedded system, is a classic example used in UPPAAL which is a model checker tool for Timed Automata. The bridge in this system likes a critical region in operating system. It has multiple pathways but only one bridge allowing a single train to get through at the same time. The requirements are illustrated in figure 5.

- A train must notice the controller in advance when it is approaching the bridge.
- Controller decides whether to let it go through or wait.
- Receiving the let go signal, a train must pass the bridge in 10 min and inform the controller.
- Receiving wait signal, a train must stop in 5 min.
- Other requirements about response time constraints of hardware.

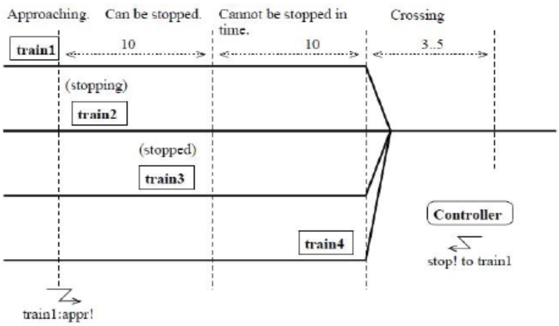

Fig. 5. Train – Bridge Requirements

Considering the accuracy of the system demands, we set two chronometric clocks clk1 and clk2 using the Clock stereotype in MARTE. clk1 is to model the time constraints of passing the bridge and stopping the train, and clk2 is used to model strict time constraints of hardware. These clocks are instances of the Chronometric which applies the ClockType stereotype. Using MARTE time modeling elements, figure 6 shows the final time model of the system.

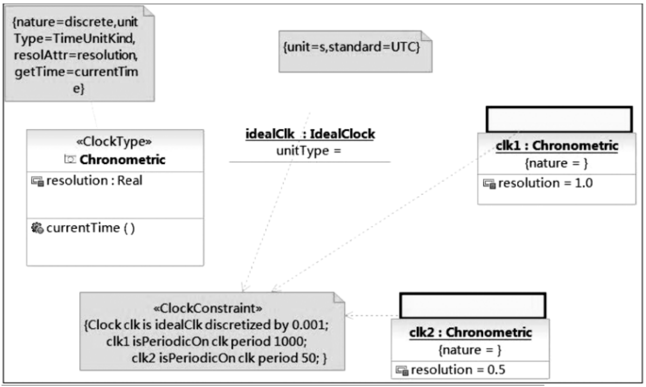

Fig. 6. Time model of Train – Bridge System

Clock clk is an idealClock discretized by 0.001. Clock clk1 is periodic on clk period 1000 which means this clock is triggered one time every minute and it will be used to measure train's approach and leave behaviors. clk2 is periodic on clk period 50 as noted in the ClockConstraints. It means clk2 ticks twice every one second and it will be used as reference of response time constraints. With these clocks, modeling of time related behaviors is possible.

4.2 Timing Aspects

Aspect-Oriented software development is a separation of concerns technique that decomposes systems into distinct features with minimal overlap [9]. Focusing on separating of crosscutting concerns and uncoupling base modules with core modules leads to high reliability and maintenance of a system and make evolution simple to deal with. Time is the decisive concern in RTES and it will be an effective way to manage it as crosscutting concerns and aspect themes.

Train-Bridge system contains several time related classes such as the controller, train and signal sender. Their behaviors are time constrained. With this resemblance, they are abstracted as TimedClass in figure 7 and the corresponding operations are just expressed as an abstract operation op (). When this operation comes, firstly it will invoke the timer or check its time constraints. After this, it returns to its own executing thread. At the end of the operation, it will cancel the timer. Then, all the time related behaviors are modeled as cooperation with Timer (fig.8).

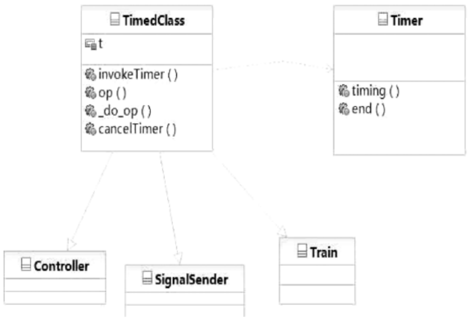

Fig. 7. Timed Class

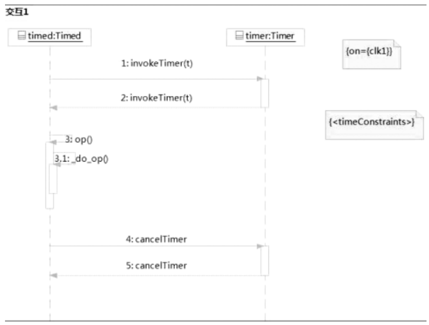

Fig. 8. Aspect Theme

Train-Bridge system contains several time related classes such as the controller, train and signal sender. Their behaviors are time constrained. With this resemblance, they are abstracted as TimedClass in figure 7 and the corresponding operations are just expressed as an abstract operation op (). When this operation comes, firstly it will invoke the timer or check its time constraints. After this, it returns to its own executing thread. At the end of the operation, it will cancel the timer. Then, all the time related behaviors are modeled as cooperation with Timer (fig.8).

4.3 System Modeling

Modeling necessary clocks with MARTE profile after traditional requirement analysis provides basic access to system time. Design processes consists base theme design and aspect theme design inside which modeling time as aspect theme. When system evolves or requirement changes, designers may back to first process to check new themes. System modeling in this way helps resolve code tangling and code scattering problems and explores an effective modeling mechanism making use of Aspect-Oriented methods and MARTE profile.

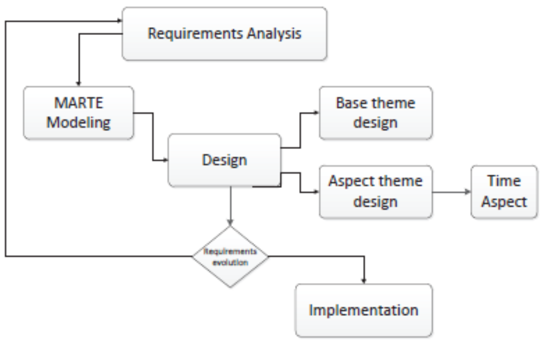

Fig. 9. Process Of system modeling

5 Conclusion and Future Work

In this article, we have presented our thoughts about Timing Aspects Construction using UML-MARTE profile. We illustrated the outline of aspect-oriented technique and useful timed elements in MARTE. We followed that by describing how MARTE facilitates the time modeling with Theme approach. In a Train-Bridge system, we have demonstrated how to describe the time by MARTE profile and model them as aspect themes. System modeling is then regulated as requirement analysis, MARTE modeling and design processes.

However, the current version of this method has some limitations, some of which have already been identified throughout this article. Our method does not capture all real-time and embedded systems concerns and provides little support for synchronization. We are currently investigating a more mature framework about this thought by experimenting on more complex RTES and dealing with more complicated time constraints. Besides, we are exploring a useful and efficient tool for both Theme approach and MARTE profile.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No.61003025, 91118002), and by the Jiangsu Province Research Foundation (BK2010170).

References

1. OMG, Inc. UML Profile for MARTE: Modeling and Analysis of Real-Time and Embedded systems, Tech. rep. formal/2009-11-02, Object Management Group (2009)
2. Laddad, R.: I want my AOP! Java World (2002), http://www.javaworld.com/javaworld/jw-01-2002/jw-0118-aspect.html
3. Clarke, S., Baniassas, E.: Aspect-Oriented Analysis and Design: The Theme Approach
4. Nouh, M., Ziarati, R., Mouhed, D., Alhadidi, D., Debbabi, M., Wang, L., Pourzandi, M.: Aspect Weaver: a model transformation approach for UML Models
5. Clarket, S., Walker, R.J.: Composition Patterns: An Approach to Designing Reusable Aspects
6. Gray, J., Bapty, T., Neema, S., Tuck, J.: Handling Crosscutting Constraints in Domain-Specific Modeling
7. Elrad, T., Aldawud, O., Bader, A.: Aspect-Oriented Modeling: Bridging the Gap between Implementation and Design. Computer Science, 189–201 (2002)
8. Baniassas, E., Clarke, S.: Theme: An Approach for Aspect-Oriented Analysis and Design. In: Proceedings of the 26th International Conference on Software Engineering, ICSE 2004 (2004)
9. Driver, C., Reilly, S., Linahan, É, Cahill, V.: Managing Embedded Systems Complexity with Aspect-Oriented Model-Driven Engineering
10. Marcello, M., Lius, G.M., Mauro, P.: Model-based Design Space Exploration for RTES with SysML and MARTE