An Approach to Constructing Timing Diagrams from UML/MARTE Behavioral Models for Guidance and Control Unit Software

Jinho Choi[1,2], Doo-Hwan Bae[1]

[1] Dept. of Computer Science, College of Information Science and Technology,
Korea Advanced Institute of Science and Technology (KAIST)
[2] Agency for Defense Development (ADD)
Daejeon, Republic of Korea
{jhchoi, bae}@se.kaist.ac.kr

Abstract. Timing-related issues need to be managed from early design phase for successful development of GCU (Guidance and Control Unit) software. UML/MARTE behavioral models can specify timing information in the multiple viewpoints. Among UML behavioral models, UML timing diagrams are useful to show timing information intuitively. We propose an approach to constructing timing diagrams with MARTE annotations from the state machine and sequence diagrams with MARTE annotations. The proposed approach consists of the consistency checking step to get well-formed UML/MARTE models and the model transformation step to construct timing diagrams.

1 Introduction

GCU (Guidance and Control Unit) software used in military avionics systems is rapidly growing in complexity and size. A GCU is a safety-critical and real-time embedded system, and GCU software controls GCU resources and communicates with other subsystems and also executes flight-related functions [1]. Furthermore, diverse experts in the fields of aerospace, electronics, mechanics and computer science attend to develop a GCU. To develop GCU software successfully, timing-related issues such as timing constraints and execution scenarios should be specified and analyzed from early design phase [1][4].

The UML (Unified Modeling Language) is a general purpose modeling language for visualization and understanding of software structures and behaviors [2]. However, UML is hard to specify timing characteristics of RTES. To solve limitations of UML, MARTE (Modeling and Analysis of Real-Time Embedded systems) profile is adopted [3]. MARTE provides predefined stereotypes and tagged values for real-time embedded software.

UML sequence diagrams and UML state machine diagrams are frequently used for describing interactions between objects and behavior of objects, respectively. Since UML timing diagrams show intuitive specifications for timing constraints and have been used widely in electronic domain [4], timing diagrams are useful for providing a common understanding and effective communications to stakeholders attended in the development of GCU software.

We observe that timing diagrams can be constructed from sequence diagrams and state machine diagrams because sequence diagrams and state machine diagrams have relevant information to timing diagrams such as timing ruler values, lifelines, states, events and durations. We propose an approach to constructing timing diagrams with MARTE annotations (TDs/MARTE) from sequence diagrams with MARTE annotations (SDs/MARTE) and state machine diagrams with MARTE annotations (SMDs/MARTE). With the proposed approach, we can save modeling time for TDs/MARTE and can easily understand and analyze timing behavior of RTES. This research is extended with our previous work [5].

2 An Approach to Constructing Timing Diagrams

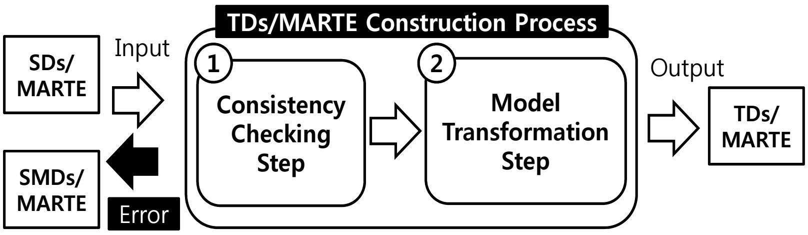
TDs/MARTE Construction Process
SDs/ MARTE
Input
SMDs/ MARTE
Error
1
Consistency Checking Step
2
Model Transformation Step
Output
TDs/ MARTE

Figure 1 Overall approach

Figure 1 shows the overall approach for constructing timing diagrams. SDs/MARTE and SMDs/MARTE are input to the TDs/MARTE construction process. The TDs/MARTE construction process consists of two steps such as the consistency checking and the model transformation. We explain UML/MARTE behavioral modeling, consistency checking and model transformation as follows:

UML/MARTE Behavioral Modeling for RTES

We propose guidelines for UML/MARTE behavioral modeling to describe behavior of RTES for event-driven or timing-triggered systems (e.g., a GCU). Since UML is informal, guidelines are necessary to use UML/MARTE in RTES domain. We assume that temporal behaviors are performed under the synchrony hypothesis. Figures 2 and 3 show an example of SD/MARTE and SMDs/MARTE for Counter and Displayer. The SD/MARTE in Figure 2 shows message interchange between Counter and Displayer every 50 milliseconds. Lifelines, messages, time observations, execution specifications and MARTE annotations are used to specify SDs/MARTE. SMDs/MARTE in Figure 3 describe the overall behavior of Counter and Displayer. In SMDs/MARTE modeling, states, events, actions, and MARTE annotation are used.

Consistency Checking for SDs/MARTE and SMDs/MARTE

UML/MARTE inconsistency exists naturally between an UML model and MARTE annotations (intra-model), and between SDs/MARTE and SMDs/MARTE (inter-model). In the consistency checking step, we check UML/MARTE consistency using a rule-based method to get well-formed UML/MARTE behavioral models. To this end, we defined 20 Rules and developed the UMCA (Uml/Marte Consistency Analyzer) tool to detect the inconsistency points automatically. Figures 2 and 3 do not have inconsistency points.

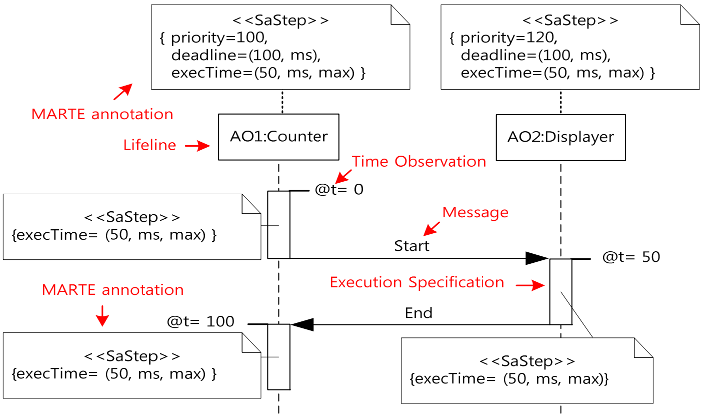

Figure 2 Example of SD/MARTE

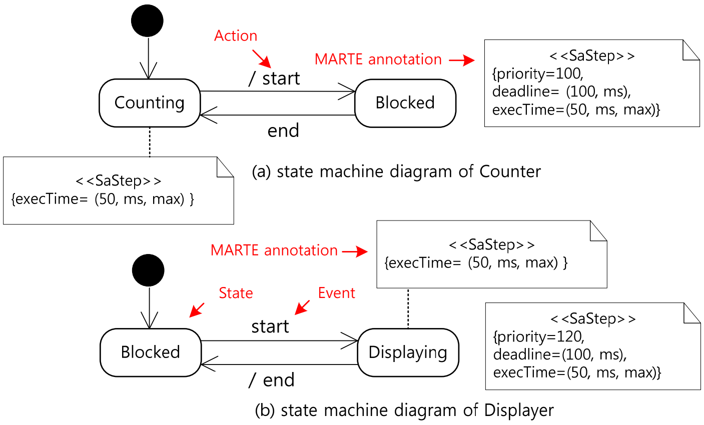

Figure 3 Example of SMD/MARTE

Model Transformation for TDs/MARTE

We construct TDs/MARTE using four transformation rules. Rule 1 is for construction of timing ruler. Time unit is determined from MARTE annotations. In Figures 2 and 3, time unit is milliseconds. In Rule 2, lifelines of TDs/MARTE are extracted from SDs/MARTE. Counter and Displayer are selected from Figure 2. Rule 3 is for construction of states. States of TDs/MARTE are extracted from SMDs/MARTE of the lifelines selected in Rule2. Counting, Blocked, Displaying states are extracted from Figure 3. In Rule 4, durations and events are constructed from SDs/MARE and SMDs/MARTE. In previous our work [5], we proposed the algorithm to specify durations and events in TDs/MARTE. After applying four transformation rules, we can construct the TD/MARTE as shown in Figure 4.

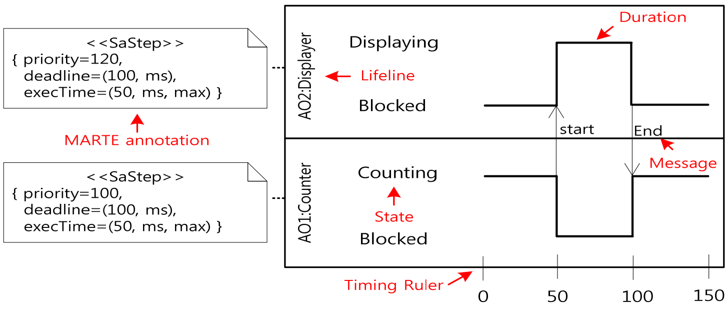

Figure 4 TD/MARTE constructed from Figures 2 and 3

3 Conclusion

We presented an approach to constructing TDs/MARTE from SDs/MARTE and SMDs/MARTE for GCU software in military avionics systems. UML/MARTE modeling guidelines are presented to specify UML/MARTE models in GCU software domain. The consistency checking step makes consistent SDs/MARTE and SMDs/MARTE to construct error-free TDs/MARTE. The model transformation step constructs TDs/MARTE from the consistency-checked SDs/MARTE and SMDs/MARTE. We have three plans. First, we will apply the proposed approach in GCU software domain. Second, we will refine and extend guidelines for UML/MARTE behavioral modeling. Last, we will develop an automated tool for constructing TDs/MARTE.

Acknowledgments. This research was sponsored by the Agency for Defense Development under the grant UD100031CD.

References

1. J. Choi, E. Jee, H.-J. Kim, D.-H. Bae, “A case study on timing constraints verification for a safety-critical, time-triggered embedded software,” Journal of KIISE: Software and Applications, Vol. 38, No. 12, pages 647-656, December 2011. (in Korean)
2. Unified Modeling Language: Superstructure, version 2.4.1 (ptc/2011-08-06), OMG, 2011, available from : http://www.omg.org.
3. UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems, version 1.1 (formal/2011-06-02), OMG, 2011, available from : http://www.omg.org.
4. M. Fowler, Uml Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd edn., Addison-Wesley, 2004.
5. J. Choi, E. Jee, D.-H. Bae, “Toward Systematic Construction of Timing Diagrams from UML/MARTE Behavioral Models for Time-Triggered Embedded Software,” The 6th IEEE International Conference on Software Security and Reliability, pp. 118-127, June 2012.