Compositional Verification of Timing Constraints for Embedded Real-Time Systems

HUI GUO, WOO JIN LEE
School of Electrical Engineering and Computer Science
Kyungpook National University
1370 Sanyeok-dong, Buk-gu, Daegu
SOUTH KOREA

Abstract: - In real-time software, not only computation errors but also timing errors can cause system failures, which eventually result in significant physical damages or threats to human life. To efficiently guarantee the timely execution of expected functions, it is necessary to clearly specify and formally verify timing requirements before performing detailed system design. This paper proposes a compositional approach to specifying and verifying timing requirements for real-time systems in a systematic manner. We specify both requirements and specification of a system using Modular TER nets, an extension of TER nets, to support timing analysis in a compositional way. By incrementally composing the requirement model and the specification model, we can check timing anomalies in requirements and the specification model.

Key-Words: - Timing constraints, real-time systems, timing verification, compositional analysis

1 Introduction

Real-time systems are rapidly gaining influence in the contemporary world; they include cars, transport systems, military weapon systems, and medical devices [1]. In real-time systems a failure in the temporal aspect can be as critical as one in the functional aspect [2,3]. Therefore, for developing real-time systems, it is essential to specify the timing requirements as well as the functional ones and to apply some systematic development for satisfying the required timing constraints.

Unified modeling language (UML) has been used for describing real-time systems by extending time concepts [4,5,6,7]. VERTAF[4] is proposed fro describing and verifying a formal UML-based real-time system. In VERTAF, extended sequence diagram, timed statecharts, and class diagram are used for described design of real-time system and are transformed into Petri nets for checking scheduability. But they are focused on design phase and do not consider the environmental issues of real-time and embedded systems. UML-RT is used for modeling and simulating real-time systems [5] and for checking temporal consistency sequence diagrams and statecharts [6]. And Object Constraint Language (OCL) is extended for expressing state-related time-bounded constraints [7]. There are few works for specifying and analyzing timing constraints in the UML-based models at the earlier requirements phase.

This paper presents a compositional approach to specifying and verifying timing requirements for real-time systems with the support of scalability. First, we describe scenario-based requirements model and component-based specification models for specifying behaviors of real-time systems. To support compositional timing analysis, we propose Modular TER nets, an extension of Time ER net [8]. Conflicts among timing requirements can be incrementally checked by compositional analysis on the Modular TER nets. In addition, in our component-based approach, not only the specification models but also analysis results of components can be independently maintained and can be reused for developing another real-time system.

The remainder of the paper is organized as follows. Section 2 briefly describes the overall approach to verifying timing requirements. Section 3 discusses the specification of component-based real-time systems using Modular TER nets. An unmanned entrance gate system is illustrated as a case study throughout the paper. Section 4 presents compositional timing analysis from the Modular TER nets. Finally, in Section 5 we present conclusions.

2 Compositional Analysis Approach

Fig. 1 shows an overview of our compositional approach to the specification and verification of timing constraints.

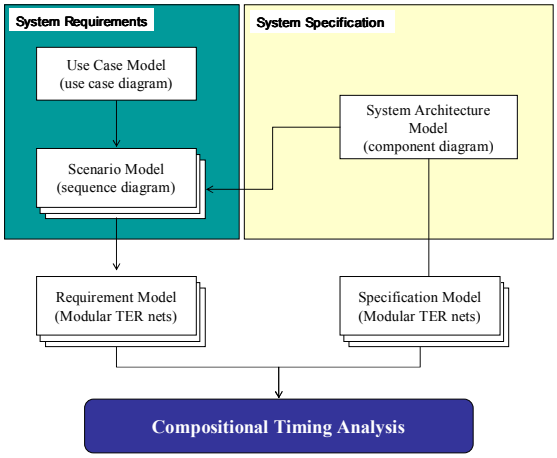


Fig.1 Overall approach to compositional timing analysis

Use case modeling technique has been widely adopted for requirement elicitation and description. To describe the functional requirements of a real-time system, we use the use case model, which is expressed in UML use case diagram. Each use case is realized by the sequence diagram later.

System architecture model describes the architectural view of the system including external environment. From the functional requirements, we determine S/W components in the system and identify external components in the environment, and describe relationships among those components. UML component diagram is used to describe the system architecture model.

A system is usually realized through collaborations of the components that provide some services for implementing requirements on the system. Scenario model captures and describes such collaboration information among components. In this paper, Modular TER nets are used to formally specify the functionality of each requirement.

In the specification model, a Modular TER nets is used to describe the behavior of each system component. For a component, each Modular TER nets shows how the component reacts to an event given from other system components or external component. The overall behavior of a system can be derived and analyzed by composing the Modular TER nets of all the system components. Such compositional analysis is an essential property for building large-scale systems.

The proposed analysis approach to timing requirements is based on the composition of requirements model and specification models. We analyze timing requirements by incrementally composing those models. The timing requirements analysis is based on the formalism Modular TER nets, which is used as a common formalism to support compositional timing analysis of multiple models.

3 Specification of Real-time Systems

An unmanned entrance gate system is illustrated as a case study in the remaining sections of the paper. Fig. 2 shows an overview of the unmanned entrance gate system. ID tags for the system will be issued to students and staffs working for the university. When a car with an ID tag approaches the gate, a RF reader recognizes its ID tag and sends it to a main controller. After checking whether it is valid or not, the gate is opened when the ID is valid.

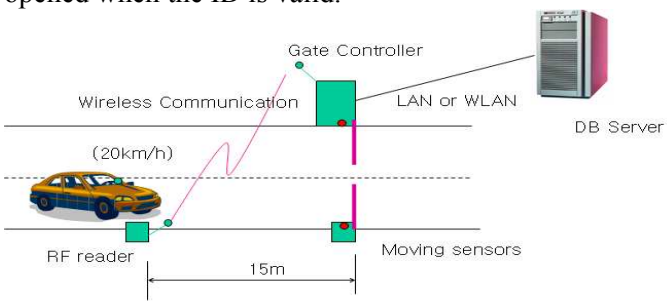


Fig.2 Overview of an example system

3.1 Modular TER nets

In requirements analysis phase, timing constraints are generally given in the form of absolute and global time manner, while relative time axis are used in describing components in design time. We chose Petri Nets as a base formalism since some Petri nets such as Time ER nets [8] and TCPN [9] can handle both relative and global timing viewpoints and have several analysis techniques for timely behaviors. And Petri nets have many advantages of representing the concurrency among models naturally. We choose TER nets as our base formalism and simplify its graphical notation.

Definition 1 For a character set Σ, a TER net is a 6-tuple TER net = (P, T, E, A, L, M0), where

- P, T, E, A and M0 are the same as those of a TER net, each of which stand for the sets of places, transitions, environments, actions and an initial marking respectively,
- $L : T \rightarrow \Sigma+$ is a label function that associates a distinct label taken from strings (Σ+) with each transition of T.

In TER nets, each token (called environments) contains a variable, called chronos, representing the timestamp of the token, and action associated with transition controls the timely behavior of token. To improve understandability and modeling simplicity we introduce an arc-based time representation notation, a time pair is associated with incoming arcs of a transition. In this case, the arc can control the time slot through which the token in input place could pass. Fig. 3 shows an example of TER nets.

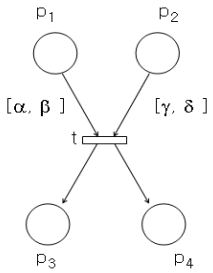


Fig. 3 Arc-based representation of time intervals

To compositionally analyze the behaviors of a system, the methodology should support the modular concept. Since TER nets do not support the modular concept, it is impossible to analyze the behaviors of target system in the compositional manner. Therefore, we propose Modular TER nets on the basis of TER nets to support compositional timing analysis.

Definition 2 Modular TER nets (MTER nets) are defined as a set of TER nets { TER net_i | i = 1…n} satisfying the following conditions:

- T_i should be disjoint for TER net_i,
- M0 for shared places should be the same.

We assume that the transitions shared among multiple TER nets are marked with shades. In MTER net models, two TER nets are parallel composed by unifying shared transitions while preserving incoming/outgoing arc information and timing constraints. When there are several pairs of corresponding shared transitions, they are unified independently as shown in Fig. 4. A unified transition is transformed to a local transition. Note that the number of places is preserved after the parallel composition.

3.2 System Requirements

There are two main scenarios for the proposed unmanned entrance gate system. One is a scenario which is also being used for the existing ticket issuing system, which is to be used for cars without ID tags or guests' cars. The other scenario is new for the unmanned entrance gate system. A scenario for passing registered cars is described in terms of a sequence of actions in natural language as follows.

- A car's RF module detects, connects, and sends its ID to the RF module of the gate.
- The gate system checks the validity of the ID.
- If the ID is valid, the gate will be opened.
- After the car passed, the gate is closed.

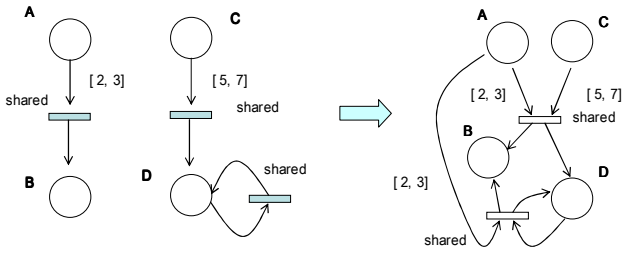


Fig.4 Composing rules of shared transitions

There are three timing constraints for the example system:

- Timing constraints 1 (TC1): A registered car can pass the gate at the speed of 20km/h or less. That is, a car can arrive at the gate from RF reader in 2.7 seconds.
- Timing constraints 2 (TC2): The gate should be opened in 1 second before a car passes it.
- Timing constraints 3 (TC3): The gate should begin closing within 0.1 second after the car passes.

A scenario model represents a sequence of message interactions among components in both the system and the environment. Since the concrete interactions naturally depend on a specific requirement, a scenario model is constructed for each scenario. For example, Fig. 5 shows component interactions for the scenario of passing registered cars. We simply use the sequence diagram to concisely describe the component interactions. The sequence diagram represents message exchanges among the components according to the scenario. And timing constraints can be explicitly described by time intervals between events. Timing constraints TC1, TC2, and TC3 are explicitly described by a time interval, [0,2.7s] between 'RFdetect' and 'passed', [1s, ∞], and [0, 0.1s], respectively.

Fig. 6 shows a TER net for the scenario of passing registered cars. The behavior of this TER net conforms to that of the sequence diagram. As shown in Fig. 6, TER net for this scenario starts and ends with transitions 'scenario_start' and 'scenario_end',

respectively. Each transition can be sequentially mapped to an event of the sequence diagram.

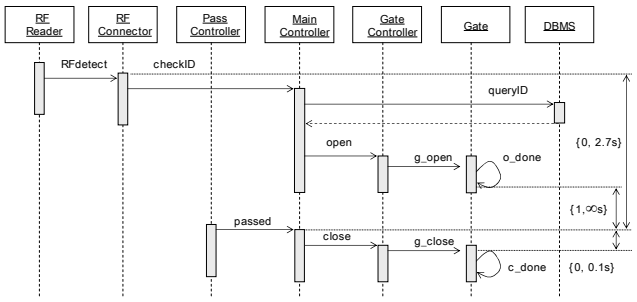


Fig.5 Component interactions of the scenario for passing registered cars

Additionally, timing constraints TC1, TC2 and TC3 are transformed to places T1, T2 and T3 and their outgoing arcs with the time intervals. For example, TC1 is represented as arcs from transition 'RFdetect' to transition 'o_done' through place T1 with time interval [0, 2.7].

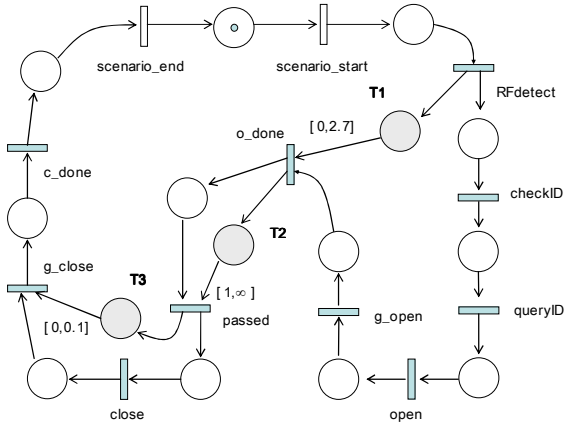


Fig.6 A TER net of the scenario for passing registered cars

3.3 Specification of Software Components and Environmental Components

For each component both within a system and in the environment, a TER nets is defined to specify the behavior of the component. For example, a total of 9 TER nets are constructed; MovingSensors, Car, Gate, RFReader, DBMS for external components and MainController, GateController, PassController, RFConnector for software components. And dependencies among components can be described by shared events. Fig. 7 shows TER net models of external component Gate, and system components GateController and MainController. As shown in Fig. 7 (a), the gate is repeatedly lifting and lowering the crossing gate. We assume that the complete time of moving up ranges from 2 to 2.5 seconds while the complete time of moving down from 2 to 2.3 seconds.

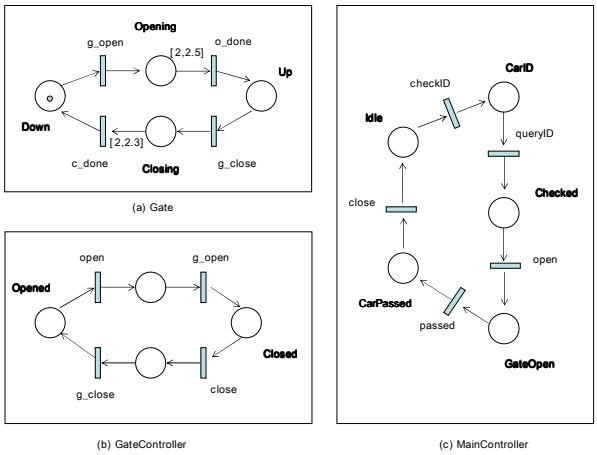


Fig.7 MTER net models for the components

4 Compositional Timing Analysis

Compositional analysis has a scalable analysis method. After independently behavior analysis of each component, the analysis results can be incrementally merged in the order of compositional hierarchy [10]. Fig. 8 shows the steps of compositional timing analysis. First, scenario models and component behavior models are independently analyzed. Next, according to coupling between components, they are merged incrementally to make a composition hierarchy. For efficient analysis, composition is performed in such way that local transitions are reduced.

Our compositional timing analysis can be used for checking conflicts among timing requirement and also for checking whether time intervals in components are consistent with timing requirements in other components. We focus on analyzing conflicts among timing requirements which are described on requirement documents or implicitly and explicitly given to environment components.

4.1 Reduction Rules

For minimizing analysis state space, we perform reduction of intermediately composed models. We use 6 reduction rules of Murata [11], which are not related with timing constraints. In addition, we propose two more reduction rules concerning timing constraints: timing abstraction rule and timing domination rule. Timing abstraction rule states that two subsequent timing constraints can be merged into a single timing constraint. In other words, timing constraints [t1, t2]

and [t3, t4] can be merged to [t1+t3, t2+t4]. Figure 9(a) shows an example of a timing abstraction rule. Timing domination rule states that a dominated timing constraint can be abstracted by a dominating timing constraint in the special case of control flows which start at a single transition and end at a single transition. The following definition and theorem describe the timing domination rule in detail.

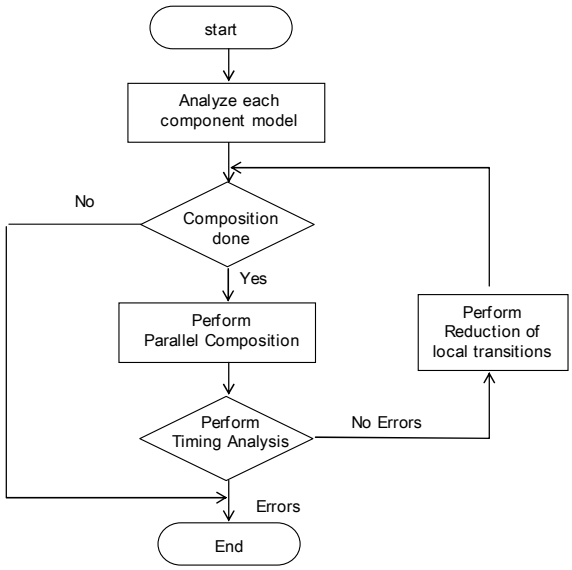


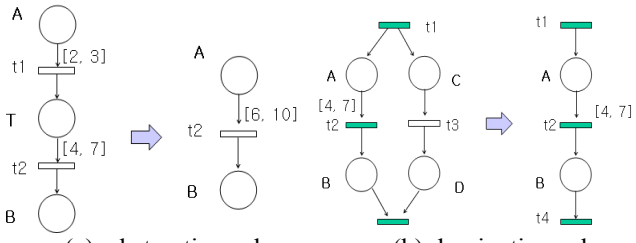
Fig.8 A flowchart for compositional timing analysis

Definition 3 Suppose that there are two or more timing flows which are branched at one transition and are joined at another transition. If a timing constraint of one flow is completely included by those of some others, the former flow dominates the other flows in timing perspective.

For example, consider two timing constraint [2, 3] and [1, 4]. Since timing constraint [2, 3] is completely included by [1, 4], [2, 3] dominates [1, 4]. Basically, a flow with [n, m] dominates any flows without timing constraints. Figure 9 (b) shows an example of timing dominating rule. Since the flow t1, t2 and t4 dominates the flow t1, t3 and t4, the latter flow can be excluded. A dominated flow including shared transitions can not be reduced since it can be changed by merging other shared transitions.

4.2 Compositional Hierarchy

Compositional hierarchy shows the order of composing component models. Fig. 10 shows a compositional hierarchy of the unmanned entrance gate system. Each MTER nets of the scenario model, component behavior models and environment behavior model are listed at the bottom. A pair of numbers over a component represents the numbers of places and transitions in its MTER nets, respectively.



(a) abstraction rule (b) domination rule

Fig.9 Reduction rules concerning timing constraints

In addition, the second pair indicates the size of the reduced MTER nets after applying the reduction rules including timing abstraction rule and timing dominating rule. Compositional hierarchy shows the order of composing component models. The composition order can be determined by the intuition of analyzers or according to coupling among components. Coupling between components can be defined to be the ratio of the number of coexisting shared transitions to the number of all shared transitions.

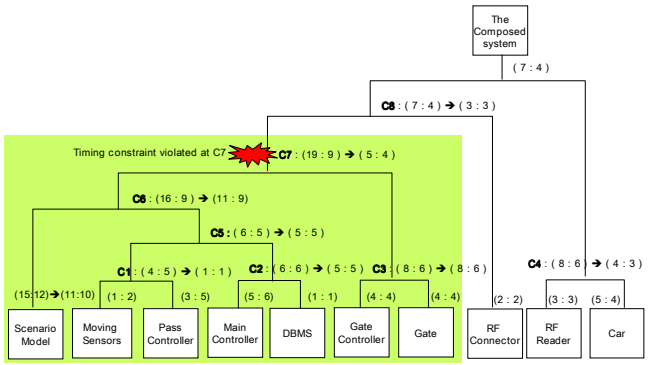


Fig.10 Compositional hierarchy of the unmanned entrance gate system

Fig.11 shows the composition steps where C7 is obtained by composing C3 and C6. Fig. 11 (a) represents the first composed model of C6 over C3 model, where the highlighted parts with thick line indicate C6 model. The gray parts of the composed model show the result of applying the timing dominating rule. For example, a flow "open → o_done" is dominated by a flow "open → g_open → o_done". Fig. 11 (b) shows the reduced model by repeatedly applying the timing abstraction rule. As shown in Fig. 11 (b), we can easily detect that the transition 'passed' cannot be enabled since there is no available common interval between two branched flows which start at the transition 'RFdetect'. This deadlock situation originates from conflicts in timing

constraints given in Car and Gate models. That is, a car cannot pass the entrance gate at the speed of 20 km/h. To solve this problem, we can choose one of the following solutions and revise the system models.

- The RF reader should be installed farther away from the gate.
- The gate should be replaced by a faster moving gate.
- The requirements for the speed of cars should be weakened.

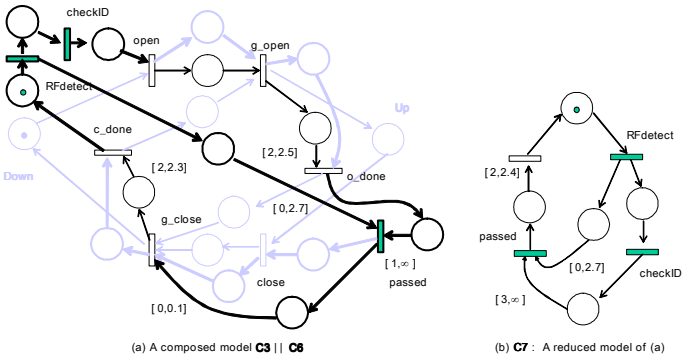


Fig.11 Steps of composing and reducing C3 and C6

Table 1 shows the size of states (or places) and transitions in the component models and the final model from both approaches. In the final model, the generated analysis space of the compositional approach is outstandingly smaller than that of state diagram approach.

Table 1: Analysis spaces of MTER nets

	S1	C1	C2	C3	C4	C5	P1	P2	P3	P4	F
St(SD)	9	5	2	2	1	1	4	2	2	1	80
Tr (SD)	8	6	2	4	1	1	4	2	2	2	219
Pl(MTER)	15	5	4	3	2	1	5	4	3	1	7
Tr(MTER)	12	6	4	5	2	1	4	4	3	2	4

Legends: S1: Scenario model 1, P1: Car, P2: Gate, P3: RFReader, P4: MovingSensors, C1: Main Controller, C2: Gate Controller, C3: Pass Controller, C4: RF connector, C5: DBMS

5 Conclusions

In real-time systems, timing requirements are one of the most important aspects to be considered for building dependable software systems. In addition, it is beneficial to analyze timing constraints of real-time systems as early as possible. We described a timing analysis based on Modular TER nets. Timing analysis can be efficiently performed in a compositional way to check conflicts among timing constraints by incrementally composing requirement model and specification model. With the proposed approach to component-based specification and verification for real-time systems, we can achieve the expected benefits such as analysis results reusability of components and compositional timing analysis.

References:

[1] B. Selic and L. Motus, Using Models in Real-time Software Design, *IEEE Control Systems*, Vol. 23, No. 3, June, 2003 pp. 31-42

[2] H. Kopetz, Software Engineering for Real-Time: A Roadmap, in *"The Future of Software Engineering"* , Anthony Finkelstein (Ed.), ACM Press 2000

[3] N. Wirth, Toward a discipline of real-time programming, *Communication of the ACM*, Vol. 20, No. 8, Aug. 1977

[4] P.A. Hsiung, S.W. Lin, C.H. Tseng, T.Y. Lee, J.M. Fu, and W.B. See. VERTAF: An Application Framework for the Design and Verification of Embedded Real-time Software, *IEEE Trans. on Software Engineering*, Vol. 30, No. 10, Oct. 2004

[5] J.B. Michael, M.T. Shing, M.H. Miklaski, and J.D. Babbitt. Modeling and Simulation of System-of-Systems Timing Constraints with UML-RT and OMNeT++, *Proc. of the 15th IEEE International Workshop on Rapid System Prototyping*, 2004

[6] J.M. Küster and J. Stroop. Consistent Design of Embedded Real-time systems with UML-RT, *Proc. of the 4th IEEE Symposium on Object-Oriented Real-Time Distributed Computing*, 2001

[7] Stephan Flake. Real-Time Constraints with the OCL, *Proc. of the 5th IEEE Symposium on Object-Oriented Real-Time Distributed Computing*, 2002

[8] C. Ghezzi, D. Mandrioli, S. Morasca, and M. Pezze, A Unified High-Level Petri net Formalism for Time-Critical systems, *IEEE Trans. on Soft. Eng.*, Vol. 17, No. 2, Feb. 1991, pp. 160-172

[9] Jeffrey J. P. Tsai, Steve Jehnwa Yang, and Yao-Hsiung Chang, Timing Constraint Petri Nets and Their Application to Schedulability Analysis of Real-Time System Specifications, *IEEE Trans. on Soft. Eng.*, Vol. 21, No. 1, Jan. 1995

[10] Wei Jen Yeh, *Controlling State Explosion in Reachability*, PhD. Thesis, Purdue University, Dec. 1993

[11] T. Murata, "Petri nets: Properties, Analysis and Applications", *Proceedings of The IEEE*, Vol. 77, No. 4, April 1989