Formal Specification for Transportation Cyber Physical Systems

Lichen Zhang, Jifeng He and Wensheng Yu

Shanghai Key Laboratory of Trustworthy Computing
East China Normal University
Shanghai 200062, China
Zhanglichen1962@163.com

Abstract. Transportation cyber physical systems such as automotive, aviation, and rail involve interactions between software controllers, communication networks, and physical devices. These systems are among the most complex cyber physical systems being designed by humans, but added time and cost constraints make their development a significant technical challenge. Formal specification technologies are now indispensable for quickly developing safe and reliable transportation systems. In this paper, we propose a formal specification approach for Transportation cyber physical systems. The proposed formal framework is such a formwork. On the one hand, it can deal with continuous-time systems based on sets of ordinary differential equations. On the other hand, it can deal with discrete-event systems, without continuous variables or differential equations. We present a combination of the formal methods Timed-CSP, ZimOO and differential (algebraic) equations or differential logic. Each method can describe certain aspects of a transportation cyber physical system: CSP can describe communication, concurrent and real-time requirements; ZimOO expresses complex data operations; differential (algebraic) equations model the dynamics and control (DC) parts. A case study of train control system illustrates the specification process for Transportation cyber physical systems.

Keywords: Transportation Cyber Physical Systems, ZimOO, Timed-CSP, Differential Logic

1 Introduction

Transportation cyber physical systems[1] – automotive, aviation, and rail – involve interactions between software controllers, communication networks, and physical devices. These systems are among the most complex cyber physical systems being designed by humans, but added time and cost constraints make their development a significant technical challenge. Formal specification technologies are now indispensable for quickly developing safe and reliable transportation systems.

Transportation cyber physical systems consist of three parts: the dynamics and control (DC) parts, the communication part and computation part. The DC part is that

of a predominantly continuous-time system, which is modeled by means of differential (algebraic) equations, or by means of a set of trajectories. The evolution of a hybrid system in the continuous-time domain is considered as a set of piecewise continuous functions of time. The computation part is that of a predominantly discrete-event system. A well-known model is a (hybrid) automaton, but modeling of discrete-event systems is also based on, among others, Z,VDM, process algebras, Petri nets, and data flow languages. Clearly, cyber physical systems represent a domain where the DC, communication and computation aspects must be met, and we believe that a formalism that integrates the DC, communication and computation aspects is a valuable contribution towards integration of the DC, communication and computation methods, techniques, and tools [2].

In this paper, we provide some ideas for formal specification of transportation cyber physical systems and one well known case study to validate formal specification.

2 Formal Specification for Transportation Cyber Physical Systems

Transportation systems are complex systems and current formal specification technology does not scale to the sizes of these systems. These systems need to be analyzed at several levels of abstraction. It is unlikely that a single specification technique will suffice at every level.

CSP is suitable for showing the order of the occurrence of events but lack the ability to handle complex abstract data types and operations.[3] ZimOO [4] is based on Object-Z [5], an object-oriented extension of Z [6], ZimOO is an extended subset of Object-Z allowing descriptions of discrete and continuous features of a system in a common formalism ZimOO supports three different kinds of classes: discrete as in Object-Z, continuous and hybrid classes.

The differential dynamic logic (dL) [7] is a logic for specifying and verifying hybrid systems [17][15]. The logic dL can be used to specify correctness properties for hybrid systems given operationally as hybrid programs . The basic idea for dL formulas is to have formulas of the form $[\alpha]\varphi$ to specify that the hybrid system α always remains within region φ, i.e., all states reachable by following the transitions of hybrid system α statisfy the formula φ. Dually, the dL formula $<\alpha>\varphi$ expresses that the hybrid system α is able to reach region φ, i.e., there is a state reachable by following the transitions of hybrid system α that statisfies the formula φ.

Aspect-oriented approaches[8] use a separation of concern strategy, in which a set of simpler models, each built for a specific aspect of the system, are defined and analyzed. Each aspect model can be constructed and evolved relatively independently from other aspect models. Aspect-oriented specification is made by extending TCOZ [9] and ZIMOO notation with aspect notations. The schema for aspect specification in has the general form as shown in Fig.1.

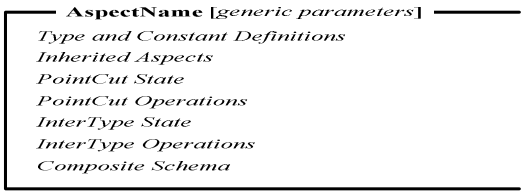

Fig.1. Aspects of Model Structure

3 Case Study: Formal Specification of Train Control Systems

Train control systems contain several components connected by communication channels. One important component is the train controller whose purposes are to limit the speed of the train, decide when it is time to switch points and secure crossings, and make sure that the train does not enter them too early. The odometer component keeps track of the speed and position of the train. The speed controller supervises the speed and makes sure that it does not exceed the limit set by the train controller, otherwise it automatically slows down the train. When the speed limit is set to zero, the train will break until it comes to a safe halt. The communication with crossings is done by the radio controller. As said above, the communication medium is radio based. Special care has to be taken, because radio transmissions are inherently unsafe. The safety must still be established under the assumption that no message can be transferred [10][11]

For specification of train control using formal methods, First, the communication channels of the class are declared. Every channel has a type which restricts the values that it can communicate. There are also local channels that are visible only inside the class and that are used by the CSP, ZIMOO, and differential dynamic logic (dL) parts for interaction. Second, the CSP part follows; it is given by a system of (recursive) process equations. Third, the Z part is given which itself consists of the state space, the Init schema and communication schemas. For each communication event a corresponding communication schema specifies in which way the state should be changed when the event occurs. Finally, below a horizontal line the differential dynamic logic (dL) part is stated. Classes can be combined into larger specifications by CSP operators like parallel composition, hiding and renaming.

The first aspect is communication. These communications can be naturally modelled with CSP. As an example we can model the loop supervising the speed in CSP by the following recursive equation:

$$Radio\_com \overset{c}{=} SuperviseTrain\,1 \,|||\, SuperviseTrain\,2 \cdots$$

$$SuperviseTrain\,1 \overset{c}{=} getSpd \rightarrow getPos \rightarrow calcMaxSpd \rightarrow setMaxSpd \rightarrow SuperviseTrain\,1$$

$$SuperviseTrain\,2 \overset{c}{=} \cdots$$

$$\cdots$$

It is assumed that an MA (movement authority) has been granted up to some track position, which we can call in, and the train is located at position z, heading with current speed v towards m. We represent the point SB as the safety distance s relative

to the end m of the MA (i.e., m-s=SB). In this situation, differential dynamic logic (dL) can specify the following crucial safety property of the train control system, which we state as a DL formula which expresses that a train always remains with its MA [12].

$$\psi \rightarrow [(\ control \quad ; dirve \quad)^*]z \le \mathrm{m}$$

$$where \quad control \quad \equiv (?\ m - z \le \mathrm{s}\ ; a := -b\ ) \cup (?\ \ \mathrm{m} - \mathrm{z} \ge \mathrm{s}\ ; a := A\ ),$$

$$drive \quad \equiv \tau := 0\ ; (\ z' = v\ , v' = a\ , \tau' = 1\ \&\ v \ge 0 \wedge \tau \le \varepsilon\ ).$$

A train controller limits the speed of the train, decides when it is time to switch points and secure crossings, and makes sure that the train does not enter them too early [13].There are main variables and operations in the train movement process: speed, time, reportinfo, update log, receive command, brake, supervise. This paper uses ZIMOO to specify state space. Fig.2 gives out the model of the train control.

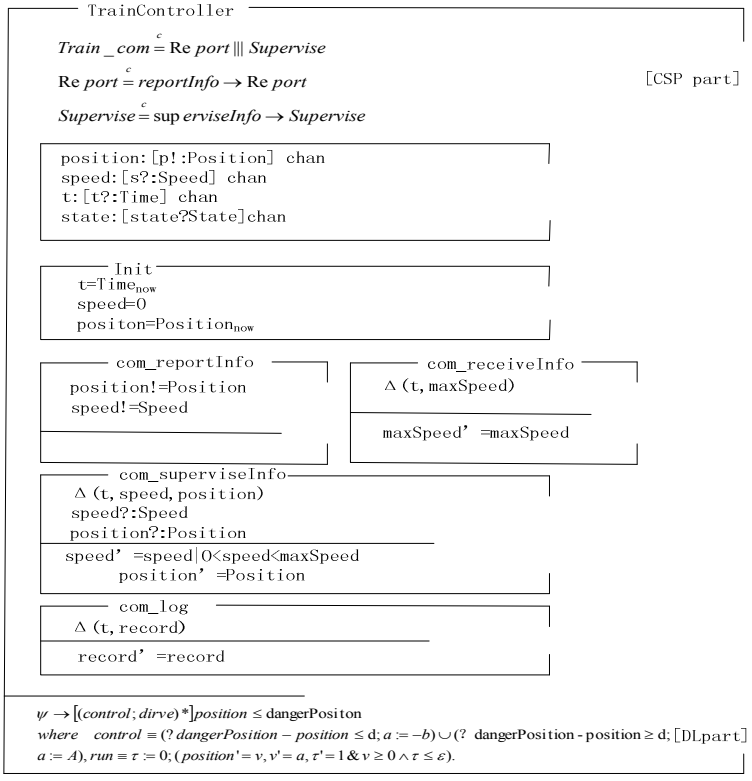

Fig. 2. Modeling Train Controller by the integration of CSP, ZIMOO, and differential dynamic logic (dL)

The movement permissions of trains are neither known beforehand nor fixed statically. They are determined based on the current track situation by a Radio Block Controller (RBC) [14] [15] [16]. Trains are only allowed to move within their current

movement authority (MA), which can be updated by the RBC using wireless communication. Hence the train controller needs to regulate the movement of a train locally such that it always remains within its MA . The RBC is modeled by CSP[17], ZIMOO, and differential dynamic logic (dL) [18] as shown in Fig.3.

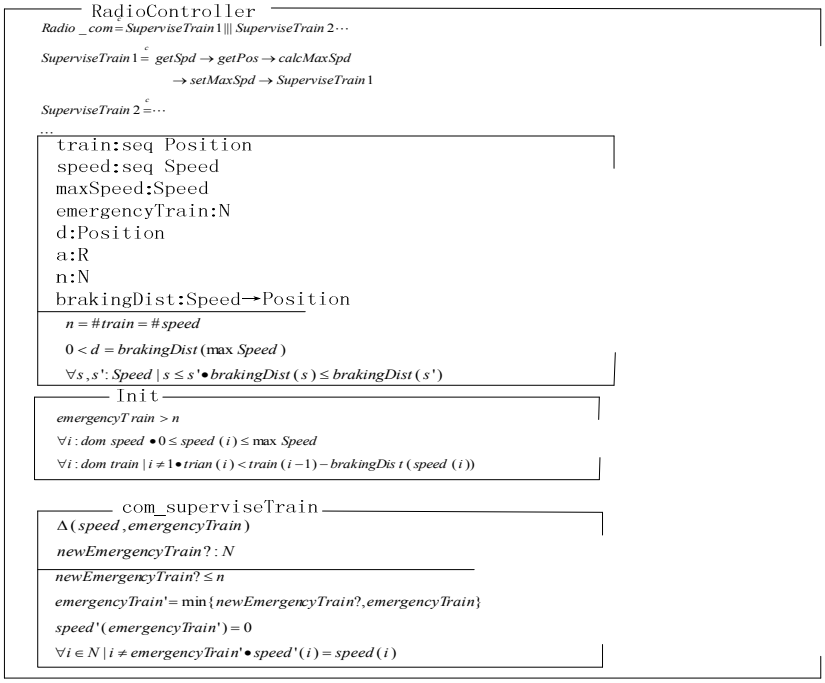

Fig.3. Specification of Radio Block Controller

4 Conclusion

In this paper, we proposed a formal specification approach for Transportation cyber physical systems. The proposed formal framework is such a formwork. On the one hand, it can deal with continuous-time systems based on sets of ordinary differential equations. On the other hand, it can deal with discrete-event systems, without continuous variables or differential equations. We presented a combination of the formal methods Timed-CSP, ZimOO and differential (algebraic) equations or differential logic. Each method can describe certain aspects of a transportation cyber physical system: CSP can describe communication, concurrent and real-time requirements; ZimOO expresses complex data operations; differential (algebraic) equations model the dynamics and control (DC) parts. A case study of train control system illustrates the specification process for Transportation cyber physical systems. system was used to illustrate the specification process of formal specification for cyber physical systems.

The further work is devoted to integrated formal specification with AADL further.

Acknowledgments. This work is supported by national high technology research and development program of China (No.2011AA010101), national basic research program of China (No.2011CB302904), the national science foundation of China under grant No.61173046, No.61021004, No.61061130541), doctoral program foundation of institutions of higher education of China (No. 200802690018), national; science foundation of Guangdong province under grant No.S2011010004905.

References

1. Grand Challenges for transportation Cyber-Physical Systems www.ee.washington.edu/.../GregSullivan-20081014102113
2. E. A. Lee and S. A. Seshia, Introduction to Embedded Systems – A Cyber-Physical Systems Approach, Berkeley, CA: LeeSeshia.org, 2011
3. Adnan Sherif, Ana Cavalcanti, Jifeng He, Augusto Sampaio: A process algebraic framework for specification and validation of real-time systems. Formal Asp. Comput. 22(2): 153-191 (2010)
4. Viktor Friesen. An Exercise in Hybrid System Specification Using an Extension of Z. citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.2010&rep.
5. Graeme Smith. The Object-Z Specification Language[M]. Software Verification Research Centre University of Queensland. 2000
6. J. Spivey： The Z Notation: A Refernce Manual (2rd Edition). Prentice Hall, UK, 1992
7. André Platzer. Differential dynamic logic for hybrid systems. Journal of Automated Reasoning, 41(2), pp 143-189, 2008
8. Kiczales G,et al.Aspect-Oriented Programming. In:proc.of the 11th European Conf. on Object-Oriented Programming, June 1997.
9. B. P. Mahony and J.S. Dong. Blending Object-Z and Timed CSP: An introduction to TCOZ. ICSE'98, April 1998.
10. Jochen Hoenicke. *Combination of Processes, Data, and Time*. PhD thesis, University of Oldenburg, July 2006
11. Jochen Hoenicke：Specification of Radio Based Railway Crossings with the Combination of CSP, OZ, and DC. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.4394.
12. A. Platzer. Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics. Springer, 2010. 426 p. ISBN 978-3-642-14508
13. Johannes Faber, Swen Jacobs, Viorica Sofronie-Stokkermans： Verifying CSP-OZ-DC Specifications with Complex Data Types and Timing Parameters. Integrated Formal Methods 2007, July 3rd.
14. André Platzer. Differential dynamic logic for verifying parametric hybrid systems. LNCS 4548, pp 216-232. Springer, 2007.
15. Jochen Hoenicke and Patrick Maier. Model-checking of specifications integrating processes, data and time. In J.S. Fitzgerald, I.J. Hayes, and A. Tarlecki, editors, *FM 2005*, volume 3582 of *LNCS*, pages 465-480. Springer, 2005.
16. J. Hoenicke and E.-R. Olderog. CSP-OZ-DC: A combination of specification techniques for processes, data and time. *Nordic Journal of Computing*, 9(4):301-334, 2002.
17. Davies J, Schneider S. A Brief History of Timed CSP[J]. Theoretical Computer Science, 1995, 138(1):243-271
18. A. Platzer. A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems. *Logical Methods in Computer Science*, 42 pages. Special issue for selected papers from CSL'10