Specification of Railway Transportation Cyber Physical Systems Using Formal Approach

Lichen Zhang*, Jifeng He and Wensheng Yu

Shanghai Key Laboratory of Trustworthy Computing
East China Normal University, Shanghai 200062, China

**Corresponding Author: 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 dynamic logic (DL). 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 dynamic logic (DL) 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] 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

There are three parts in Transportation cyber physical systems. First, the physical part is physical entities of a cyber physical system. It is simply that part of the systems that is not realized with computers or digital systems. It can include mechanical parts, biological or chemical processes, or human operators. Second, there are one or more computational plateforms, which consist of sensors, actuators, one or more computers, and (possible one or more operating systems). Third, there is a communication part, which provides the mechanisms for communications and physical parts to communicate. Thus, we can separate cyber physical systems into three parts: cyber, physical, and communication by aspect-oriented development methods [3-5] as shown in Figure 1.

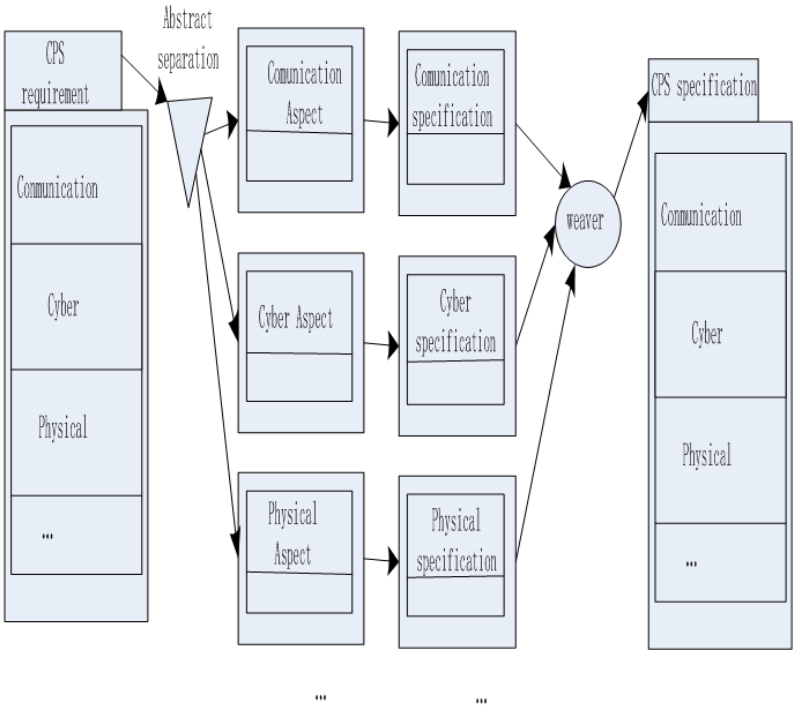

Figure 1. Specification of Three aspects of CPS

In this paper, we propose a formal approach that integrates CSP [6], ZIMOO [7] and differential dynamic logic [8] to specify cyber physical systems by aspect-oriented methods. The proposed approach allows us to present in a modular way the dynamic continuous, data changes, and timing aspects of the systems we want to verify. We use Communicating Sequential Processes (CSP) to specify the communication part, ZIMOO to describe the state space and its change, and the differential dynamic logic to model the dynamic continuous of transportation cyber physical systems. After the establishment of three sub-aspects models, then form a complete system model by integrating the three sub-aspects through aspect-oriented method.

Z [19] is a formal language used to define data types and to show the effect of operations on these types. It lacks, however, features to express the order in which the operations are executed. Process algebras [5], like CSP, on other hand, are suitable for showing the order of

the occurrence of events but lack the ability to handle complex abstract data types and operations. Finally, formalisms like differential dynamic logic on dynamic aspects [17].

ZimOO is an extended subset of Object-Z [20-21]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. Thus, the system can be structured better and the well-known suitable formalisms can be applied to describe, analyze, and refine the different parts of the system. The bridge between the continuous and the discrete world is built by hybrid classes.

The differential dynamic logic (dL) 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. For instance, the following formula expresses that for the state of a train controller train, the property $y \leq m$ always holds true when starting in a state where $v^2 \leq 2b(m-y)$ is true: $v^2 \leq 2b(m-y) \rightarrow [train]y \leq m$.

Aspect-oriented approaches 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 and ZIMOO notation with aspect notations. The schema for aspect specification in has the general form as shown in Figure 2, Figure 3, Figure 4, and Figure 5.

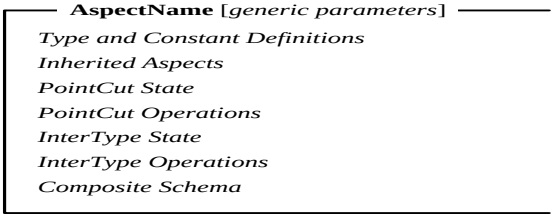

Figure 2. Aspects of Model Structure

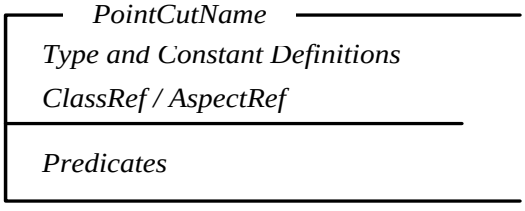

Figure 3. PointCut Operation Schema of Structure

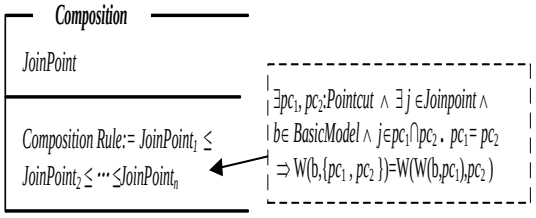

Figure 4. Composition Schema of Structure

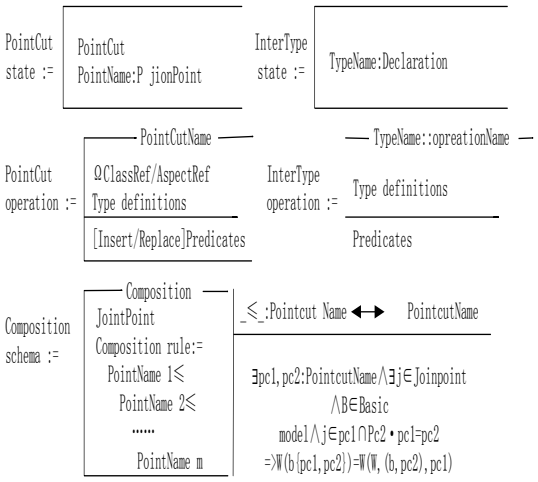

Figure 5. Specification of Operations and Schema

3. Case Study: Formal Specification of Train Control Systems

Figure 6 surveys the controller architecture we want to specify in this case study [9-12]. In the centre of the diagram 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 keeps track of the speed and position of the train. The position is measured by various means. 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.

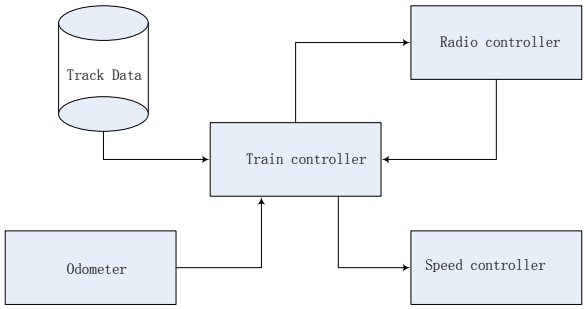

Figure 6. Logic Structure of Train Dispatching

Safety and security are very are important in our life [18, 22], we should use formal technique to specify train control systems. The first aspect is communication. Formal methods is very power to make strict specification [13]. Most of the communications are initiated by the train controller itself, *e.g.*, the train controller decides when it is time to secure a track element. But there are also communications initiated externally, *e.g.*, the signal that is sent when a crossing affirms that it is safe. These communications can be naturally modelled with CSP [14]. As an example we can model the loop supervising the speed in CSP by the following recursive equation:

$$Radio\_com \overset{c}{=} SuperviseTrain1 \,|||\, SuperviseTrain2 \cdots$$

$$SuperviseTrain1 \overset{c}{=} getSpd \rightarrow getPos \rightarrow calcMaxSpd \rightarrow setMaxSpd \rightarrow SuperviseTrain1$$

$$SuperviseTrain2 \overset{c}{=} \cdots$$

…

There are two communication operation for train (controler)part: one is to report the situation of current train , another is to supervise the situation of train movement. The CSP model is as follows:

$$Train\_com \overset{c}{=} \mathrm{Re}\, port \,|||\, Supervise$$

$$\mathrm{Re}\, port \overset{c}{=} reportInfo \rightarrow \mathrm{Re}\, port$$

$$Supervise \overset{c}{=} \mathrm{sup}\, erviseInfo \rightarrow Supervise$$

The communication model of the whole transport systems is expressed by CSP as follows:

$$Main \underline{\overset{c}{=}} radio\_com \,||\, train\_com \,||\, OtherParal\, lel\, \mathrm{Pr}\, ocess$$

The purpose of the train control system is to ensure that train cannot crash into other trains or pass open gates. Its secondary objective is to maximize throughput and velocity without endangering safety. Permission to move is granted dynamically by decentralized Radio Block Controllers (RBCs) depending on the current track situation and movement of the other traffic agents within the region of responsibility of the RBC as shown in Figure 7.

We assume 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) [15-16] can specify the following crucial safety property of the train control system, which we state as a DL formula as shown in Figure 8. It expresses that a train always remains with its MA.

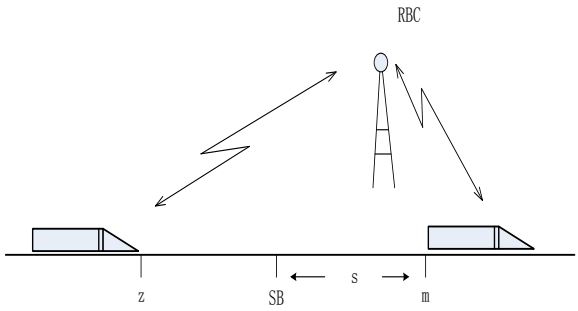

Figure 7. Train Coordination Protocol

$$\psi \rightarrow \left[(control ; dirve)^* \right] z \leq \mathrm{m}$$

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

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

Figure 8. Safety Property of the Train Control System

A track element has a unique identifier id and can be either a crossing or a point. The associated danger position is stored in pos as shown in Figure 9.

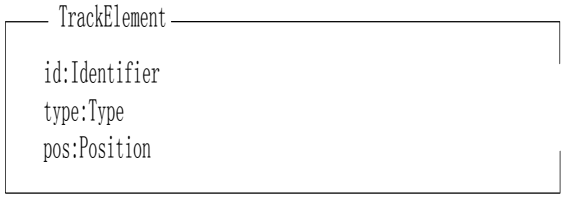

Figure 9. Specification of Track Element

The track atlas contains information about track elements, current speed, current position, direction and the maximum speed for each track segment. It is also represented by a Z schema as shown in Figure 10.

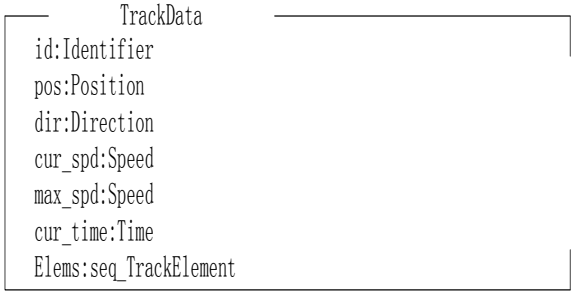

Figure 10. Specification of Track Data

The train control is decomposed into different sub-aspects so that complex application is decomposed into simple small aspects. For train control, 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. Figure 11 gives out the model of the train control.

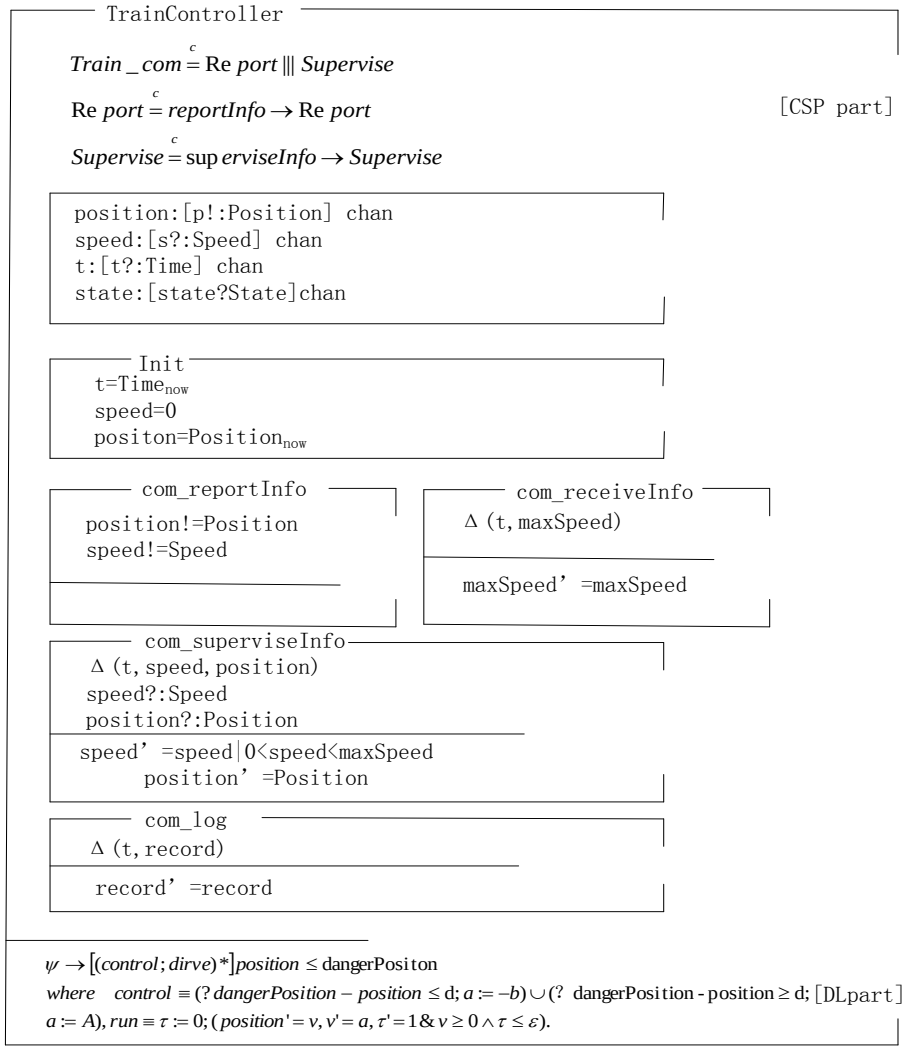

Figure 11. Modeling Train Controller by the Integration of CSP, ZIMOO and Differential Dynamic Logic (dL)

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). 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, ZIMOO, and differential dynamic logic (dL) as shown in Figure 12 and Figure 13.

train:seq Position
speed:seq Speed
maxSpeed:Speed
emergencyTrain:N
d:Position
a:R
n:N
brakingDist:Speed→Position

$n = \#train = \#speed$

$0 < d = brakingDist(\max Speed)$

$\forall s, s' : Speed \mid s \le s' \bullet brakingDist(s) \le brakingDist(s')$

Figure 12. State Space of Radio Controller

RadioController

$Radio\_com = SuperviseTrain1 \,|||\, SuperviseTrain2 \cdots$

$SuperviseTrain1 \overset{c}{=} getSpd \rightarrow getPos \rightarrow calcMaxSpd \rightarrow setMaxSpd \rightarrow SuperviseTrain1$

$SuperviseTrain2 \overset{c}{=} \cdots$

...

train:seq Position
speed:seq Speed
maxSpeed:Speed
emergencyTrain:N
d:Position
a:R
n:N
brakingDist:Speed→Position

$n = \#train = \#speed$

$0 < d = brakingDist(\max Speed)$

$\forall s, s' : Speed \mid s \le s' \bullet brakingDist(s) \le brakingDist(s')$

Init

$emergencyTrain > n$

$\forall i : dom\ speed \bullet 0 \le speed(i) \le \max Speed$

$\forall i : dom\ train \mid i \ne 1 \bullet trian(i) < train(i-1) - brakingDist(speed(i))$

com_superviseTrain

$\Delta(speed, emergencyTrain)$

$newEmergencyTrain? : N$

$newEmergencyTrain? \le n$

$emergencyTrain' = \min\{newEmergencyTrain?, emergencyTrain\}$

$speed'(emergencyTrain') = 0$

$\forall i \in N \mid i \ne emergencyTrain' \bullet speed'(i) = speed(i)$

Figure 13. Specifiction of Radio Block Controller

In the schema CrossController, train represents the particular finite set of potential Trains embodied in the systems. All of the trains need not actually exist at any particular time, in this sense they are like Trains with a unique segments being reserved for all possible Train segments instantiations during the lifetime of the system. The *importance* function gives a user-assigned priority to each train and is chosen according to how critical thr Trains is to the safe and correct function of the system. The schema CrossController describes the information necessary to schedule processes in a running systems as shown in Figure 14.

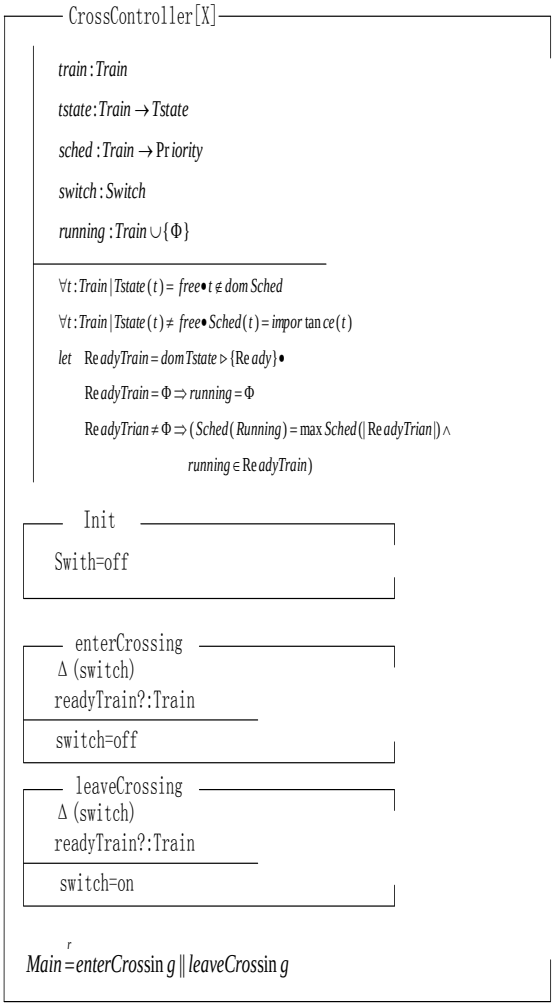

Figure 14. Specification of Crossing Control

Logs monitor system behavior. Whether you are running train control systems, whether each event happens, all of these have some sort of event logging inbuilt into them. Any time your system reacts to anything an event log is generated as shown in Figure 15.

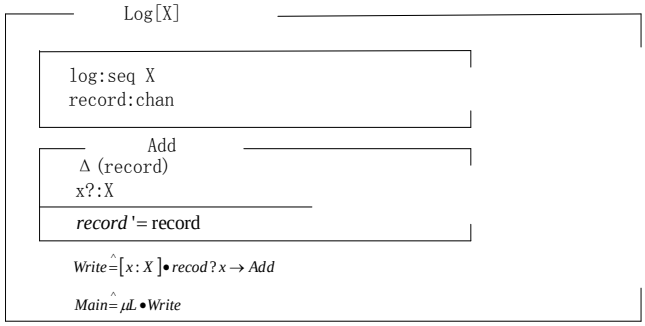

Figure 15. Specification of Log

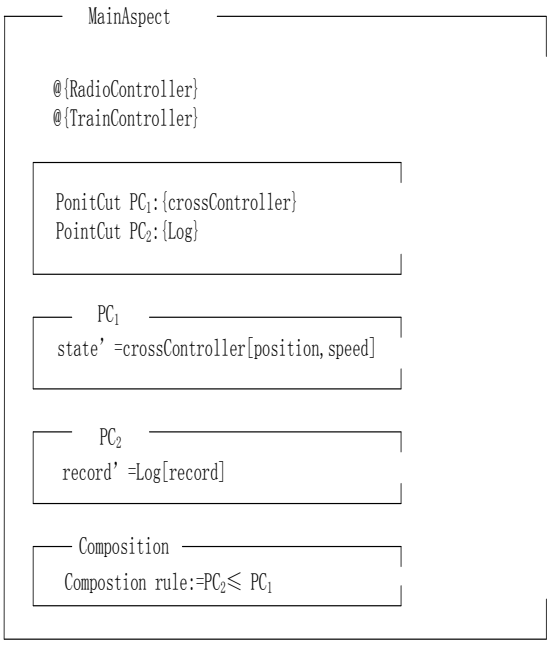

Figure 16. Composition of the System

4. Conclusion

In this paper we proposed to use formal specification for transportation cyber physical systems based on the combination of the formal methods Timed-CSP, ZimOO and differential dynamic logic. A case study of train control system was used to illustrate the specification process of formal specification for Transportation cyber physical systems.

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

Acknowledgements

This work is supported by Shanghai 085 Project for Municipal Universities and the Innovation Program of Shanghai Municipal Education Commission under grant No. ZF1213, 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, No.91118008), doctoral program foundation of institutions of higher education of China (No. 20120076130003), 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] G. Kiczales, "Aspect-Oriented Programming", Proc. of the 11th European Conf. on Object-Oriented Programming, **(1997)** June.
[4] X. Wen and H. Yu, "Real-Time Systems Modeling and Verification with Aspect-Oriented Timed Statecharts", International Journal of Hybrid Information Technology, vol. 5, no. 1, **(2012)** January, pp. 193-198.
[5] J. Y. Liu, L. C. Zhang, Y. Zhong and Y. Chen, "Middleware-based Distributed Systems Software Process", International Journal of Advanced Science and Technology, vol. 13, **(2009)** December, pp. 27-50.
[6] J. Davies and S. Schneider, "A Brief History of Timed CSP", Theoretical Computer Science, vol. 138, no. **(1995)**, pp. 243-271.
[7] V. 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.
[8] A. Platzer, "Differential dynamic logic for verifying parametric hybrid systems", LNCS 4548, Springer, **(2007)**, pp. 216-232.
[9] J. 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.
[10] J. Faber, S. Jacobs and V. Sofronie-Stokkermans, "Verifying CSP-OZ-DC Specifications with Complex Data Typesand Timing Parameters", Integrated Formal Methods, **(2007)**, July 3rd.
[11] P. Derler, E. A. Lee and A. Sangiovanni-Vincentelli, "Modeling Cyber-Physical Systems", Proceedings of the IEEE special issue on CPS, **(2011)** December.
[12] J. Hoenicke, "Combination of Processes, Data, and Time", PhD thesis, University of Oldenburg, **(2006)** July.
[13] C. F. Ngolah and Y. Wang, "Tool Support for Software Development Based on Formal Specifications in RTPA", International Journal of Software Engineering and Its Applications, vol. 3, no. 3, **(2009)** July, pp. 71-88.
[14] A. Sherif, A. Cavalcanti, J. He and A. Sampaio, "A process algebraic framework for specification and validation of real-time systems", Formal Asp. Comput., vol. 22, no. 2, **(2010),** pp. 153-191.
[15] A. Platzer, "Differential dynamic logic for hybrid systems", Journal of Automated Reasoning, vol. 41, no. 2, **(2008)**, pp. 143-189.
[16] A. Platzer, "Differential dynamic logic for verifying parametric hybrid systems", LNCS 4548, Springer, **(2007)**, pp. 216-232.
[17] S. Rekhis, N. Bouassida, R. Bouaziz, C. Duvallet and B. Sadeg, "Modeling Real-Time applications with Reusable Design Patterns", International Journal of Advanced Science and Technology, vol. 22, September **(2010)**, pp. 71-86.
[18] M. B. Swarup and P. S. Ramaiah, "A Software Safety Model for Safety Critical Applications", International Journal of Software Engineering and Its Applications, vol. 3, no. 4, **(2009)** October, pp. 21-32.
[19] J. Spivey, "The Z Notation: A Refernce Manual (2rd Edition)", Prentice Hall, UK, **(1992)**.
[20] G. Smith, "The Object-Z Specification Language", Software Verification Research Centre University of Queensland, **(2000)**.
[21] B. P. Mahony and J. S. Dong, "Blending Object-Z and Timed CSP: An introduction to TCOZ", ICSE'98, **(1998)** April.
[22] F. Y. Sattarova and T. -h. Kim, "IT Security Review: Privacy, Protection, Access Control, Assurance and System Security", International Journal of Multimedia and Ubiquitous Engineering, vol. 2, no. 2, **(2007),** April, pp. 17-32.