Specification of Cyber Physical Systems Based on Clock Theory

Bingqing Xu[1], Jifeng He[2], and Lichen Zhang[3]

[1,2,3]*Shanghai Key Laboratory of Trustworthy Computing*
East China Normal University
Shanghai 200062, China
[1]*xbqjoya@gmail.com*

Abstract

Cyber Physical Systems are integrations of computing, communication and control systems, and CPS make it possible to link the physical world and discrete world much more tightly and accurately. Though the appearance of CPS has triggered a great revolution to the electronic, information and internet technology, many problems still need to be solved when using the existing theories and technologies especially the description and analysis of time in Cyber Physical Systems. Clock theory puts forward the description of time and corresponding mechanism. This paper aims to apply the clock theory to the specification of Cyber Physical Systems.

Keywords: Cyber Physical Systems, continuous and discrete, clock, time analysis.

1 Introduction

Time[13] is a very important element especially in Cyber Physical Systems. Cyber Physical Systems are dynamic systems that exhibit both continuous and discrete[12] dynamic behavior. And the continuously bilateral interaction between discrete events and continuous time flow makes it hard to know the dynamic and real-time[10, 11] feature of the system, so keeping synchronization is really important work. We would prefer a unified clock to so many different time lines in the same system.

And when it comes to specification of Cyber Physical Systems, K. L. Man, and R. R. H. Schiffelers have done much constructive work that the thesis *Formal Specification and Analysis of Hybrid Systems*[8] by them introduces a hybrid process algebra named χ formalism. In the thesis, they not only elaborate the syntax and semantics of the χ specification, but also the examples and translation between other formalisms and χ. Bottle filling system is one of the cases in the thesis, but in χ, authors tend to make all the parameters and events together. Although every subsystem is clear, the interaction of them is not so good. *Link Continuous World with Discrete World*[9] by Jifeng He has a detailed specification on *time*, description of dynamic operations of continuous world, and every event is combined with a clock. This paper aims to use clock to specify the Cyber Physical Systems by cases–railroad crossing control system and bottle filling system[8].

2 Clock Theory

Clock theory[9] puts forward the possibility to describe the event in physical world by a much clearer *time* description, and can analyze, record the event by clock as well. To specify Cyber Physical Systems based on clock, the time description is clearer to every event and can link continuous world with discrete world more tightly. The definition and linking mechanism of clock theory is provided as below.

DEFINITION 1 *A clock* ***c*** *is an increasing sequence of real numbers,* ***c[1]*** *stands for the first element of* ***c****. We define its low and high rates by:*

$$\triangle(c) \ =_{df} \ \boldsymbol{inf}\{(c[i+1]-c[i])|i \in Nat\} \tag{1}$$

$$\bigtriangledown(c) \ =_{df} \ \boldsymbol{sup}\{(c[i+1]-c[i])|i \in Nat\} \tag{2}$$

DEFINITION 2 *If* ***c*** *is a healthy clock, it does not speed up infinitely. Then*

$$\triangle(c) > 0$$

DEFINITION 3 (Partial Order in Clock) *If* ***c*** *runs faster than* ***d****. For all* $i \in Nat$, $c[i] \leq d[i]$. *Then relation of* c *and* d *can be denoted by:*

$$c \preceq d \tag{3}$$

Lemma 1

$$c \preceq c'$$

DEFINITION 4 *Let* ***c*** *and* ***d*** *be clocks. We define the transition latency between the two clocks as*

$$\rho(c,d) =_{df} \ \boldsymbol{sup}\{|c[i]-d[i]||i \in \ Nat\} \tag{4}$$

Lemma 2

$$\rho(c,d) \geq 0$$

DEFINITION 5 (Local Clock and Global Clock) *Let* ***l*** *be a label denoting a location, and* ***c*** *a clock. Then* ***l:c*** *denotes clock* ***c*** *that locates at* ***l****.*

$$l : c \preceq l : d =_{df} \ (c \preceq d) \tag{5}$$

$$\rho(l : c, \ l : d) =_{df} \ \rho(c, \ d) \tag{6}$$

A global clock [l:c] is defined as an equivalent class of local clocks,

$$\rho([l_1 : c_1], [l_2 : c_2]) =_{df} \ \rho(l_1 : c_1, l_2 : c_2) \tag{7}$$

DEFINITION 6 *Some dynamic features of continuous variable can be described better as follows.*
climb(u,r) *is introduced to describe the time instants when the value of* u *rises up to* r.
drop(u,r) *is introduced to describe the time instants when the value of* u *falls below* r.

DEFINITION 7 (Linking Mechanism) *Here* $\boldsymbol{c}$ *is a clock with* $c[1] > 0$*, and* x_0 *is an initial value. In this equation, we assign the value of continuous variable* $\boldsymbol{u}$ *to discrete variable* $\boldsymbol{x}$ *at the every instant of clock c.*

$$x = u \ every \ c \ init \ x_0 \tag{8}$$

In differential equation, $\boldsymbol{u}$ *is a continuous variable,* u_0 *is an initial value, and* f *is an expression.* $(\dot{u} = f) \wedge (u(0) = u_0)$ *is the relation between* u *and* f*,then*

$$\dot{u} = f \ init \ u_0 \tag{9}$$

Let **e** be an event. **clock**(e) denotes the clock that records the time instants when **e** occurs. And **clock**(**event**(c)) denotes the time instants that the event takes place at every time instant c[i].

3 Case Study 1: Railroad crossing control system

As Figure 1 shows below, four train stations named Station 1, Station 2, Station 3, Station 4 are connected by two orthogonal railways. Train a goes through between Station 1 and Station 2, while train b goes through between Station 3 and Station 4. Four sensors named A, B, C, D are installed several meters away from the crossing area to detect whether a train approaches or not. A radio transmission center(Figure 2) can permit the train to go through to guarantee a safe traffic. Suppose that train a and train b set out from Station 1 and Station 4 separately when $t = 0$. When $t = t_a$, train a reaches sensor A, and train a is permitted to go through since the railway is safe. And when $t = t_b$, b arrives and asks for entering, the request is declined. All the permission, refusal and waiting time should be finished in $[t_1, t_2]$. v_a, v_b represents the speed of train a and train b.

We should promise that the system will never deadlock; two trains will never collide together or enter the crossing area at the same time only when it is permitted.

We use continuous variable **ua** and **ub** to describe the running time for a and waiting time for b. And control unit use event **pass** and event **wait** to control the condition of trains.

To train a:

$$climb(ua, t_2 - \varepsilon) \preceq clock(pass)$$
$$\rho(climb(ua, t_2 - \varepsilon), clock(pass)) \leq \varepsilon$$
$$drop(ua, t_1 + \varepsilon) \preceq clock(wait)$$
$$\rho(climb(ua, t_2 - \varepsilon), clock(wait)) \leq \varepsilon$$
$$\varepsilon \leq min(t_2 - t_a, t_a - t_1)/2$$

To train b:

$$climb(ub, t_2 - \zeta) \preceq clock(pass)$$
$$\rho(climb(ub, t_2 - \zeta), clock(pass)) \leq \zeta$$
$$drop(ub, t_1 + \zeta) \preceq clock(wait)$$

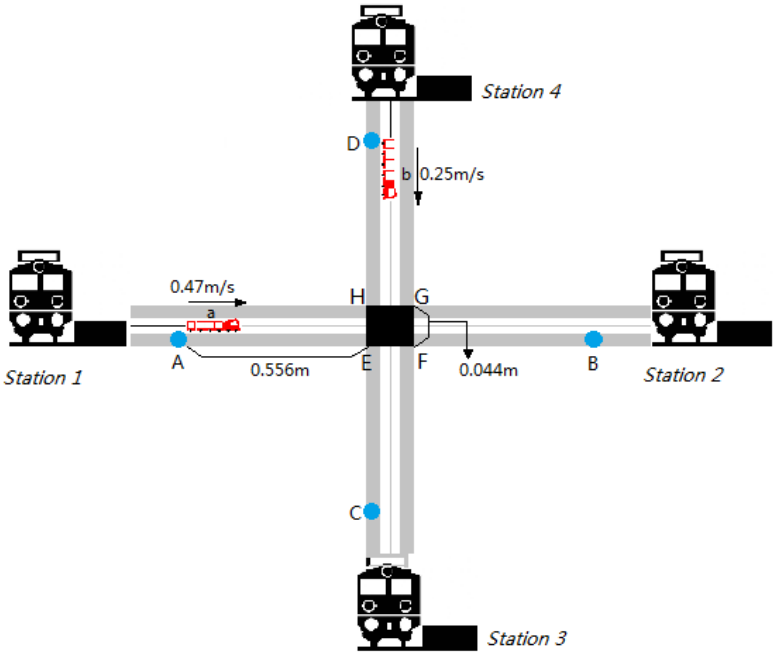

Figure 1. Railroad crossing control system

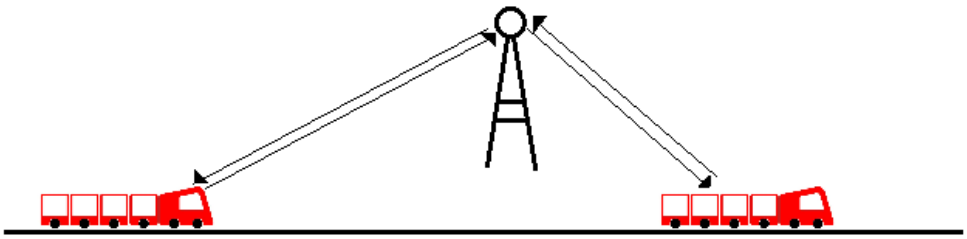

Figure 2. Radio transmission center

$$\rho(climb(ub, t_2 - \zeta), clock(wait)) \leq \zeta$$

$$\zeta \leq min(t_2 - t_b, t_b - t_1)/2$$

And the two events have noninterference.

$$clock(pass)[1] > 0$$

$$clock(pass) \preceq clock(wait) \preceq clock(pass)'$$

$$clock(pass) \wedge clock(wait) = \emptyset$$

Consider the condition of sensors, the sensors detect event **arrive** and **leave**. Here **sa** represents the response time of sensor A and B:

$$climb(sa, t_2 - \varepsilon) \preceq clock(arrive)$$

$$\rho(climb(sa, t_2 - \varepsilon), clock(arrive)) \leq \varepsilon/2$$

$$drop(sa, t_1 + \varepsilon) \preceq clock(leave)$$

$$\rho(climb(sa, t_2 - \varepsilon), clock(leave)) \leq \varepsilon/2$$

Here **sb** represents the response time of sensor C and D:

$$climb(sb, t_2 - \varepsilon) \preceq clock(arrive)$$

$$\rho(climb(sb, t_2 - \varepsilon), clock(arrive)) \leq \zeta/2$$

$$drop(sb, t_1 + \varepsilon) \preceq clock(leave)$$

$$\rho(climb(sb, t_2 - \varepsilon), clock(leave)) \leq \zeta/2$$

Since clock(a),clock(b),clock(sa),clock(sb),clock(sc), and clock(sd) have deviation from each other. We define that latency between train a and sensor A,B is δ ms, and the latency between train b and sensor C,D is σ ms.

$$|clock(a) - clock(sa)| < \delta$$

$$|clock(a) - clock(sb)| < \delta$$

$$|clock(sa) - clock(sb)| < \delta$$

$$|clock(b) - clock(sc)| < \sigma$$

$$|clock(b) - clock(sd)| < \sigma$$

$$|clock(sc) - clock(sd)| < \sigma$$

For train a, when its request is permitted:

$$clock(arrive) \preceq clock(pass)$$

$$\rho(clock(arrive), clock(pass)) \leq \varepsilon/2$$

when its request is declined:

$$clock(arrive) \preceq clock(wait)$$

$$\rho(clock(arrive), clock(wait)) \leq \varepsilon/2$$

For train b, when its request is permitted:

$$clock(arrive) \preceq clock(pass)$$

$$\rho(clock(arrive), clock(pass)) \leq \zeta/2$$

when its request is declined:

$$clock(arrive) \preceq clock(wait)$$

$$\rho(clock(arrive), clock(wait)) \leq \zeta/2$$

When train has gone through the crossing, control unit needs to inform the other train to go through.

If train a leaves, then:

$$clock(leave) \preceq clock(pass)$$

$$\rho(clock(leave), clock(pass)) \leq \varepsilon/2$$

If train b leaves, then:

$$clock(leave) \preceq clock(pass)$$

$$\rho(clock(leave), clock(pass)) \leq \zeta/2$$

4 Case Study 2: Bottle filling system

As Figure 3 shows below, the bottle filling system[8] consists of a liquid storage tank and two identical filling lines. The two incoming flows are liquid and acid, and the two outgoing flows are to fill the bottles on the assembly-line. A control system keeps the volume of the liquid storage tank between 2 and 10, and pH level of the liquid in the storage tank between 7 and 7.1. When liquid in tank becomes less acidic, the incoming flow of acid will be dribbled down to correct the situation. The volume of each bottle is 2, when a bottle is filled, the filling lines should stop and wait a period t_{tr} to transport a new bottle, and when the volume of the storage tank is under 0.7, it can not afford to support the filling lines and force them to stop. Since the change of molar quantity, molar concentration and storage tank volume is continuous, the incoming and outgoing flows are discrete, and we can specify the system by clock.

Some necessary parameters for the system are in Table 1.

First, we consider the volume control system; **u** is the continuous variable which denotes the volume of incoming liquid; **inflow_off** and **inflow_on** are events which control the incoming flow of liquid.

$$climb(u, V_{TH} - \epsilon) \preceq clock(inflow\_off)$$

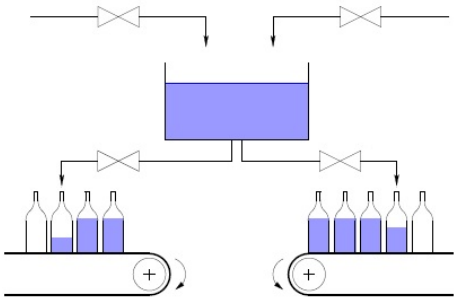

Figure 3. Bottle filling system

Table 1. Parameter list for Bottle filling system

Parameter	Value
a	speed of volume change in storage tank
b	speed of pH change in storage tank
d	speed of volume change in storage tank
I_1	current volume in storage tank
I_2	current pH level in storage tank
V_{TH}	10
V_{TL}	2
pH_H	7.1
pH_L	7
V_H	0.7
V_L	0-0.5

$$\rho(climb(u, V_{TH} - \epsilon), clock(inflow\_off)) \leq \epsilon/a$$

$$drop(u, V_{TL} + \epsilon) \preceq clock(inflow\_on)$$

$$\rho(drop(u, V_{TL} + \epsilon), clock(inflow\_on)) \leq \epsilon/a$$

$$\epsilon \leq min(V_{TH} - I_1, I_1 - V_{TL})/2$$

And the two events have noninterference.

$$clock(inflow\_off)[1] > 0$$

$$clock(inflow\_off) \preceq clock(inflow\_on) \preceq clock(inflow\_off)'$$

$$clock(inflow\_off) \wedge clock(inflow\_on) = \emptyset$$

Then, we consider the pH control system; **w** is the continuous variable which denotes the pH change of storage tank; **acid_off** and **acid_on** are events which control the incoming flow of acid.

$$climb(w, pH_H - \xi) \preceq clock(acid\_on)$$

$$\rho(climb(w, pH_H - \xi), clock(acid\_on)) \leq \xi/b$$

$$drop(w, pH_L + \xi) \preceq clock(acid\_off)$$

$$\rho(drop(w, pH_L + \xi), clock(acid\_off)) \leq \xi / b$$
$$\xi \leq min(pH_H - I_2, I_2 - pH_L)/2$$

And the two events have noninterference.

$$clock(acid\_off)[1] > 0$$
$$clock(acid\_off) \preceq clock(acid\_on) \preceq clock(acid\_off)'$$
$$clock(acid\_off) \wedge clock(acid\_on) = \emptyset$$

Next, we should consider the filling control system. v_T is the continuous variable which denotes the volume change of storage tank, and $high_{TANK}$, low_{TANK} is one of the decisive events which control the filling lines. $high_{TANK}$ means that storage tank can fill bottles now, while low_{TANK} means that storage tank has little water and can not fill bottles contemporarily.

$$climb(v_T, V_H - \lambda) \preceq clock(high_{TANK})$$
$$\rho(climb(v_T, V_H - \lambda), clock(high_{TANK})) \leq \lambda / d$$
$$drop(v_T, V_L + \lambda) \preceq clock(low_{TANK})$$
$$\rho(drop(v_T, V_L + \lambda), clock(low_{TANK})) \leq \lambda / d$$
$$\lambda \leq min(V_H - I_1, I_1 - V_L)/2$$

While the other important element influencing the filling lines is the bottle, when a bottle is filled, t_{tr}_start event starts and after the period, a new bottle comes and fills it until the $high_{BOTTLE}$ event starts or there is a low_{TANK}.

$$\begin{aligned} clock(t_{tr}\_start) \preceq clock(t_{tr}\_over) &\preceq (clock(new_{BOTTLE}) \wedge clock(high_{TANK})) \\ &\preceq clock(filling\_on) \preceq (clock(high_{BOTTLE}) \vee clock(low_{TANK})) \\ &\preceq clock(filling\_off) \preceq clock(t_{tr}\_start)' \end{aligned}$$

And the two events have noninterference.

$$clock(filling\_off)[1] > 0$$
$$clock(filling\_off) \preceq clock(filling\_on) \preceq clock(filling\_off)'$$
$$clock(filling\_off) \wedge clock(filling\_on) = \emptyset$$

In the equations, **u**, **w**, v_T are continuous variables, and **a**, **b**, **d** are discrete variables.

$$\dot{u} = a \ init \ a_0$$
$$\dot{w} = b \ init \ b_0$$
$$\dot{v_T} = d \ init \ d_0$$

$$a = u \ every \ c \ init \ a_0$$
$$b = w \ every \ c \ init \ b_0$$
$$d = v_T \ every \ c \ init \ d_0$$

5 Conclusion

Speaking of specification of Cyber Physical Systems, dealing with the continuous and discrete behavior dynamically is the tough work we have been trying to solve better. And the synchronization of all the subsystems in the whole system is also a difficulty.

Using clock to specify Cyber Physical Systems can give a more detailed description of every subsystem and give a much more considerate observation of the time line and sequence of every event.

It is brilliant to connect the event with clock, while it is difficult to link so many local clocks with the global clock. In my point of view, it is always hard work to make local clocks keeping consistent with the global clock, and the verification of the security and accuracy of the synchronization is very complicated, and we need more ideas on this tough work.

Acknowledgements

This work is supported by Shanghai Knowledge Service Platform Project (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] K. H. Kim, Desirable Advances in Cyber-Physical System Software Engineering. 2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing,(2010),pp. 2–4.

[2] K. H. Kim, Challenges and Future Directions of Cyber-Physical System Software. Proceedings of the 2010 34th Annual IEEE Computer Software and Applications Conference,(2010),pp. 10–13.

[3] Xiaohong Chen, Jing Liu, *Frédéric* Mallet and Zhi Jin, Modeling Timing Requirements in Problem Frames Using CCSL. 2011 18th Asia-Pacific Software Engineering Conference,(2011),pp. 381–388.

[4] Patricia Derler, Edward A. Lee and Alberto Sangiovanni Vincentelli, Modeling Cyber-Physical Systems. Proceedings of the IEEE, vol. 100, No. 1,(2012),pp. 13–28.

[5] Karl Henrik Johansson, John Lygeros, and Shankar Sastry, Modeling of Hybrid Systems. CONTROL SYSTEMS, ROBOTICS AND AUTOMATION, Vol. XV, (2002).

[6] D. A. van Beek, J. E. Rooda, R. R. H. Schiffelers, K. L. Man, and M. A. Reniers, Relating hybrid Chi to other formalisms. Electronic Notes in Theoretical Computer Science, vol. 191,(2007),pp. 85–113.

[7] R. R. H. Schiffelers, D. A. van Beek, K. L. Man, M. A. Reniers, and J. E. Rooda, Preliminary Report: Syntax and Formal Semantics of Hybrid Chi(2004).

[8] K. L. Man, and R. R. H. Schiffelers, Formal Specification and Analysis of Hybrid Systems. Universiteitsdrukkerij Technische Universiteit Eindhoven(2006).

[9] Jifeng He, Link Continuous World with Discrete World. Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China(2012).

[10] Xinxiu Wen and Huiqun Yu, Real-Time Systems Modeling and Verification with Aspect-Oriented Timed Statecharts. International Journal of Hybrid Information Technology, vol.5, No.2,(2012),pp. 193–198.

[11] Longbo Ran and Hai Jin, Real-time and Flexible Management of Storage Service Provider in Distributed Storage. International Journal of Hybrid Information Technology, vol.5, No.2,(2012),pp. 219-224.

[12] Ridha Aloui and Naceur Benhadj Braiek, On the Optimal State Observer Synthesis for Discrete-time Linear Systems. International Journal of Control and Automation, vol.5, No.3,(2012),pp. 65–78.

[13] Samer Charifa and Muammer Kalyon, Designing Continuous Proximate Time-Optimal Control System for a Class of Third-Order Systems. International Journal of Control and Automation, vol.4, No.3,(2011),pp. 107-122.

Authors

Bingqing Xu
Bingqing Xu is a PhD student in Software Engineering Institute from East China Normal University, China.