一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
ASN2017-102 (2018-01)

Simple and Proximate Time Model Framework of Cyber-Physical Systems

Yuan FANG† ††　Cheng LI††　Yuto LIM††　Yasuo TAN††

†Dalian Polytechnic University, Qinggongyuan 1#, Dalian, Liaoning, China

††Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi City, Ishikawa Prefecture, 923-1292 Japan

E-mail: †fangy@dlpu.edu.cn, ††{s1510216, ylim, ytan}@jaist.ac.jp

Abstract　With the rapid development technologies of computing, communication and intelligent control, the emergence of Cyber-Physical systems (CPS) is becoming increasingly popular these days. Many CPS applications that are used in safety and mission critical scenarios, such as healthcare system in nature should be verified to meet the requirements of safe and operational before their deployment. Therefore, the modeling and analysis play an important part of the safety and mission critical system of systems (SoS) development in CPS. In this paper, a simple and proximate time model (SPTimo) framework is proposed for non-real-time critical CPS applications to meet the requirements of modeling and analysis with minimum cost and time. The feasibility of the time modeling method for iHouse that is based on the SPTimo framework and UPPAAL tool is validated and discussed.

Keywords　Proximate Time Model, System of Systems, Cyber-Physical Systems, Time Automata, iHouse

1. INTRODUCTION

Cyber-Physical systems (CPS) is becoming increasingly popular, with the rapid development technologies of computing, communication and intelligent control [1]. The modeling and analysis play an important part of the safety and mission critical system of systems (SoS) development in CPS. Some of the modeling of the discrete and continuous behavior of heterogeneous systems has been developed recently, such as Ptolemy II [2] and Programming Temporally Integrated Distributed Embedded Systems (PTIDES), the procedure and initial process of the modeling are still complex and highly complicated.

CPS applications are not only the non-time-critical systems, e.g., building thermal comfort control system, but also the time-critical systems, e.g., intelligent transportation system. The existing modeling method (i.e., PTIDES model) deals with both time-critical and non-time critical systems but it leads to a huge time consuming and high resource cost.

Furthermore, time is very consuming because it requires a full and in-depth understanding of the details of the physical environments. In this paper based on timed automata, a Simple and Proximate Time Model (SPTimo) framework is proposed for non-real-time critical CPS applications to meet the requirements of modeling and analysis with minimum cost and time.

To achieve fast, simple and proximate modeling purpose method for non-time-critical systems. First, to derive a new time model based on an a proximity method, which is used to model each components of the entire time model into sensing, fabric network, and actuating as sensor time model, network time model, and actuator time model, respectively. Second, the SPTimo framework is modeled as individual timed automata reflecting their distributed and real-time characteristics. Then, all these time automata come into a network that represents concurrent communication and collaboration of the whole CPS. Finally, some properties (including time constraints) with the CPS are presented, and the correctness of the whole system is automatically verified by using the model-checking UPPAAL (Uppsala University and Aalborg University) tool.

In this paper, the objective is to examine the models for non-real-time critical CPS applications to achieve fast, simple and proximate modeling. The main contribution is achieved. This paper presents novel SPTimo model method for iHouse experiment environment, and this model is verified by UPPAAL tool based on the simple and proximate time automata.

The rest of this paper is organized as follows. Section 2 introduces the background on modeling of CPS, time automata, iHouse and UPPAAL. In Section 3, we describe the models of the SPTimo model of iHouse. UPPAAL tool verifications and discussions are presented in Section 4. Some relevant conclusions and future works are drawn in Section 5.

2. BACKGROUND

2.1. Modeling of CPS

CPS modeling requires portrayal of how interactions between process and physical processes are calculated and how they behave when they are merged [3]. Model-based analysis provides a better understanding of CPS behavior, and model-driven design can improve design automation and reduce errors in refinement. Edward Lee proposed CPS is an integrated computing power and physical process of the system [4], which uses embedded computers and networks to monitor the physical processing process, and with feedback loop, physical process and computation process affect each other. According to S.Shanker Sastry, CPS is a system that integrates computation, communication and the storage capabilities to monitor or control physical and engineering systems, which is operated in a safe, secure, efficient and real-time manner [5].

For a system modeling, simulation and verification, computational science and control science have different ways. The discrete event model in computational science usually ignores the computational time. However, in the control science, the research on the physical world is often based on time, abstracting the system as a continuous time model, and time is the most important factor in the model this will result in collisions and unpredictable failures in the interaction between the computational unit and the physical entity model.

The formal modeling method is to abstract the system based on mathematics theory, extract the system attribute, simulate the system behavior, establish the system model by clearly defining the state and the operation, and analyze and verify the system behavior based on the model. At present, the main formal modeling methods mainly include, Formal Inference, Petri, Time Automata, can be extended or customized new modeling language and tools to model.

In [6] extended the spatial and temporal properties of event models and devised a layered, time-space event model to formally analyze the close association between computational and physical worlds in CPS. In [7] extends the traditional service-oriented architecture (SOA) and proposes a service-oriented model based on Physical-Entity (PE), including PE-Ontology model and PE-SOA service specification to finish the CPS model. In [8] proposed a framework for modeling and verifying CPS. It decomposes CPS into various concurrent processes and uses extended logic programming to describe the communication behavior and numerical constraints between these processes. Finally, the CPS is verified by means of inquiries system related properties. However, it uses the logic programming method is more complex, and not intuitive, not easy for developers to implement.

2.2. iHouse

iHouse stands for ishikawa, internetted, inspiring, intelligent house. iHouse that is based on Standard House Design by Architectural Institute of Japan is an advanced experimental environment for future smart homes. iHouse is located at Nomi city, Ishikawa prefecture that consists of sensors, home appliances, and electronic devices are connected using ECHONET Lite version 1.1 and ECHONET version 3.6.

There are more than 50 sensors in iHouse for temperature, humidity, pyrheliometer, illumination, wind direction, pyroelectric and other sensing work. There are different kinds of controllers here, for example, fire alarm, hot water valve, awning, proximity switch. There are more than 100 controlled physical objects, such as electric curtains, lighting fixtures, air conditioners, etc. All of these are connected to IoT system by IP address, and are communicated by the ECHONET protocol. Simplified schematic diagram of SPTimo shown in **Fig.1**.

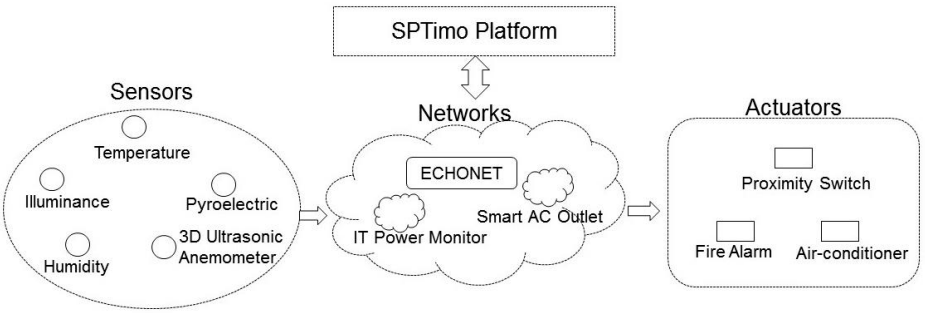

Fig. 1: Schematic diagram of SPTimo Framework.

2.3. Time Automata

The time automaton proposed by Alur and Dill [9] is a formalized model for describing and analyzing real-time system behavior.

Definition: A time automaton P can be represented by seven-elements as formulation 1 shown.

$$P = (S, s_0, Inv, A, X, V, E) \quad (1)$$

where S is the finite set, s_0 is the initial state, A is the behavior set, X is the clock variables set, V is the data variables set, E is the directed graph set as follow:

$$E \subseteq S \times A \times D(X) \times \phi(V) \times S \quad (1a)$$

where $\phi(v)$ is the assignment sequence, Inv is the initialization set which is the mapping that specified a constraint in the $D(X)$ for each state S in as follow:

$$Inv : S \to D(X) \quad (1b)$$

CPS is often modeled as timed automata networks, which are parallel compositions of timed automata. Multiple automations in the network share some clock variables and data variables, but each has its own state. Configuration is used to describe the concept of the state of all automata running in the network. One configuration can be described as a set of states, values, and clocks.

The time model formal modeling [10] method presented the asynchronous time process is defined as follows. In this paper, T is time task. Assumptions, there are different time task $T_1, T_2 \ldots T_i$, where $i=1,2,3,\ldots n$, there n is total number of time task in the whole system.

$$T_i^k = P_i^k \quad (2)$$

where P is time automata defined by (1). k is the set of time model, there are ss is means sensor time model, aa is means actuator time model, and nn means network time model.

$$k = \{ss, aa, nn\} \quad (2a)$$

Synchronization of different time automata is using the shared channel, instead of the value of communication. So, in this paper, shared variables are used to achieve the synchronization of different time automata. A shared variable is assigned by the output side, and then can be accessed by the input side that can obtain the value directly.

2.4. UPPAAL Tool

The UPPAAL tool is developed in collaboration between the Department of Information Technology at Uppsala University, Sweden and the Department of Computer Science at Aalborg University in Denmark. It is an integrated tool environment for modeling, validation and verification of real-time systems modeled as networks of timed automata, extended with data types (bounded integers, arrays, etc.). It is appropriate for systems that can be modeled as a collection of non-deterministic processes with finite control structure and real-valued clocks, communicating through channels or shared variables. Typical application areas include real-time controllers and communication protocols in particular, those where timing aspects are critical.

3. SPTimo Framework

There are various interactive entities involved in CPS system. It can be a natural environment, building, machine, a physical device, etc., of course, including human beings. The environment can be real-time sensing part (such as the physical entity) can also be controlled. Moreover, the function of CPS is based on the real-time interaction.

There are two requirements in the CPS of time modeling.

1) Time modeling must be based on functional requirements and time must be prime factor..

2) Time should be a variety of forms. In addition, due to the CPS functional performance for their perception of interactive environment and control, thus in the requirements model choice, its functional requirements must be related to the change of the physical entity.

The time model of the SPTimo Frameworks as shown in **Fig. 2**, which include Sensor Time Model, Networks Time Model, Actuator Time Model, and Time task of these three model.

Sensor Time Model is abstract by sensors node. Moreover, the sensor nodes and their time task (T_{ss}). Networks Time Model includes fabric network nodes and their time task (TP_{nn}). Actuator Time Model includes the actuator node and their time task (TP_{aa}). In addition, task in these three time models is abstract by T_i.

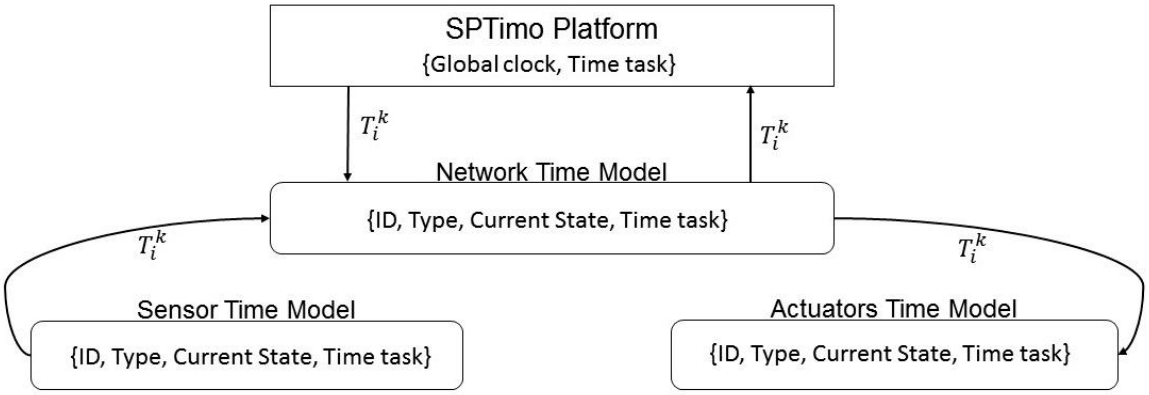

Fig. 2: Time model of the SPTimo Freamworks.

3.1. Sensor Time Model

The function of the sensor is to perceive the state of the physical world. Each sensor corresponds only to the same physical phenomenon. They measure data at a certain time frequency or send sensing data. Therefore, when modeling sensing behavior, the behavior of obtaining data is represented as an input action, while reading the value of the specified shared variable（sensing data）.The act of sending data is represented as an output action, while the output variable is given a value. In addition, the sensor model requires a clock parameter to control the data measurement or transmission. Sensor node is abstract by one set is given by

$$SS = \{ID_i, Q_i, S_c, T_i^{ss}\} \quad (3)$$

where ID_i is sensor's uniquely identifies. $i=1,2,3,\ldots a$. Q_i is the type of sensor. S_c is the current state of sensor. T_i^{ss} is the time task of sensor. Assumption $n \geq a+b+c$.

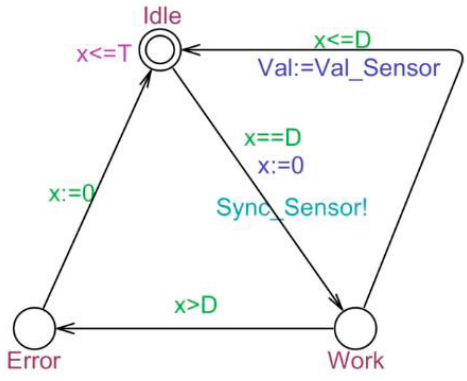

Fig. 3: Sensor timed automata

Sensor timed automata as shown in **Fig.3**, which includes three states **{Idle, Work, Error}**. **T** is the data measurement cycle; **D** is the maximum time limit for correctly collecting data in the working state. If the clock of sensor value is more over the **D** that the sensor data measurement is error. **Sync_Sensor!** in the conversion system means that the sensor sends synchronization information to the system when it enters the working state, which is used to synchronize with other devices. In the actual system, there need a data measurement synchronization signal or a uniform clock be used for timing. The variable **Val** will get the data that can be used by other devices when the sensor correctly collected data.

3.2. Actuators Time Model

The function of an actuator is to change the state of the physical world. Likewise, each actuator only refers to the same type of physical object. It performs the corresponding action according to the control instructions sent by the computation (control) unit. Actuator is abstract by one set is given by

$$AA = \{ID, Q_i, S_c, T_i^{aa}\} \quad (4)$$

where ID_i is actuator's uniquely identifies. $i=1,2,3,\ldots b$. Q_i is the type of actuator. S_c is the current state of actuator. T_i^{aa} is the time task of actuator. Assumption $n \geq a+b+c$.

In this paper, iHouse actuator timed automata is generally in standby mode, waiting to receive information from the central control system, such as start and stop instructions, the implementation of equipment, time automaton model shown in **Fig.4**. It includes two states {**Idle**, **Work**}. In iHouse application, such as lighting, ventilation and other equipment, only one switch signal can work, most of environment conditioning equipment are similar to this state. In this paper, these two states are used as modeling model abstraction, but the complex equipment is not involved.

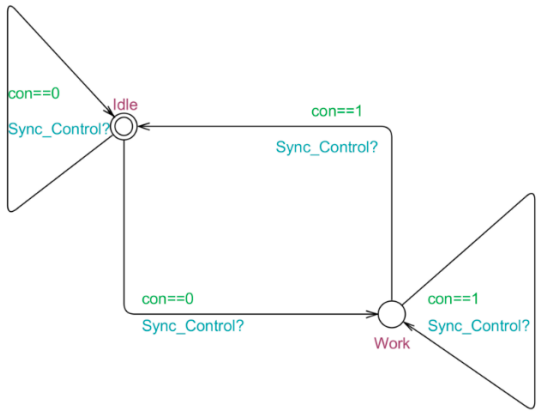

Fig. 4: Actuator timed automata

The actuator timed automatat shown in **Fig.4**. Ordinary equipment usually work in standby mode, the model to **`Idle`** for the initial state. As passive control of environmental control equipment, the actuator is no time information, it only based on the control signal to work, and is no time constraints. The **`Sync_Control?`** variable is used for the device controlled. The **`con`** variable is the only input signal of actuator , that is the control arrival flag signal of the control information.

3.3. Networks Time Model

iHouse systems is representative system of IoT. IOT gateway device as the bridge between the physical layer and the application layer is the necessary communication device for the IoT architecture. The IoT gateway uses proximity communication to collect the sensor data and forwards it to the Internet and control system. In the meantime, it can convey the control order to the execution equipment. There are two aspects of its' operation mode, one is the data transmission function, another is to send the control instruction. After the sensor collects the environmental information, it sends the data to the gateway of IoT. The gateway of IoT receives the port data and then processes the data. After the data is processed, the data are forwarded to the field control system and the Internet separately to complete a data transmission function. Similarly, when the control device sends an execution control instruction to the executing device, the IoT gateway processes the received control message and then sends it to the execution device through the corresponding port to complete a control operation.

In this paper, the gateway device working in the network layer is chosen for network time model. For simple and proximate aims, other hardware devices, network communication protocols, such as Internet protocol, ECHONET, common Internet devices, other devices are transparent treatment.

IOT gateway time automaton model is abstract by one set is given by

$$NN = \{ID, Q_i, S_c, T_i^{nn}\} \quad (5)$$

where ID_i is network's uniquely identifies. $i=1,2,3,\ldots c$. Q_i is the type of network. S_c is the current state of network. T_i^{nn} is the time task of network. Assumption $n \geq a+b+c$.

Network timed automata as shown in **Fig.5**, which includes four states {**`Idle`**, **`Up`**, **`Down`**, **`Overtime`**}. Among them, **`Idle`**, **`Up`**, **`Down`**, and **`Overtime`** are respectively expressed data information upload status, control instruction download status, standby status, and timeout status. The **`t`** is the system timeout limit, the system state transition as shown in **Fig.5**.

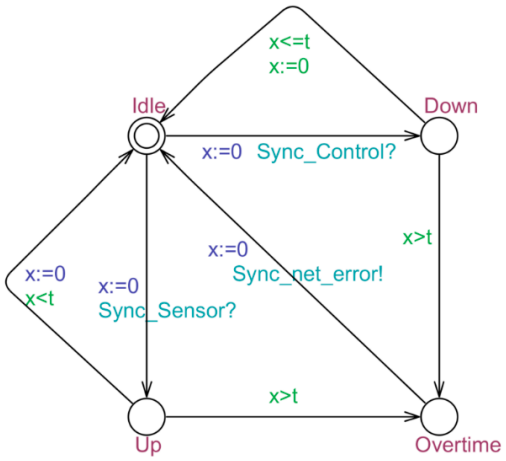

Fig. 5: Network timed automata

In **Fig. 5**, when the data transmission time out, the time automata out of the network error synchronization signal, the system receives the data transmission and command control signal synchronization, respectively, the data information and control instructions sent to the corresponding network. Gateway in the iHouse system as a data transmission device itself does not control the data storage and other operations, data only from the gateway through the gateway control of the input only time limit **`t`** , within a specified period of time can send and receive data successfully. There is only one clock **`x`**. Once the system sends a timeout, a data timeout signal, **`Sync_net_error!`** will be generated and sent to the entire system via the synchronization signal. For this timeout processing in this model will only reset the time, the specific implementation may use replacement

equipment, gateway system automatically reset operation.

4. Specification Verification

After completing the modeling of the various components, they are combined into a whole system, the model detection tool UPPAAL to verify the correctness of CPS attributes. As shown in Table 1.

Table 1: Specification verification of system.

Item	BNF	Result
Deadlock	A[]not deadlock	Verified
Sensor work is normal	E<>Sensor. Idle or Sensor.Work or Sensor.Error	Verified
Actuator work is normal	E<>Actuator. Idle or Actuator.Work	Verified
Network work is normal	E<>Network. Idle or Network.Up or Network.Down or Network.Overtime	Verified

The security of the system is verified that means there is no deadlock of the entire system. This property is represented as a temporary logic description: **`A [] not deadlock`**. In addition, let the **`t>15`** or **`t>10`** for verifying the network time automata. The UPPAAL gives a qualified result with inputting this description. The result indicate that the CPS described in the model above is working properly.

5. Summary and Future Works

Through the analysis of CPS concept and research status, this paper presents a simple and proximate CPS system modeling and verification method for non-real-time critical applications. This method uses the time automata model to depict the various components of the CSP architecture. Moreover, automatically validates the system's related attributes through the UPPAAL model-checking tool. The SPTimo method development shows two advantages: first, it better simulates the integration of the physical world and information computing process, deepens the understanding of the nature of CPS; on the other hand, it has the intuitive and simplicity, ease of understanding and mastery of developers provides a framework for modeling CPS.

The future work is to optimize the state based on of this paper to improve the verification efficiency, and to implement the actual data validation function of the time model of SPTimo Framework.

References

[1] J.F. He, “Cyber-Physical Systems”.*Commun. China Computer Federation,* vol.6, no.1, pp.25-29, 2010.

[2] E. A. Lee, “The Past, Present and Future of Cyber-Physical Systems: A Focus on Models,” *Sensors (Basel, Switzerland)* vol.15, no.3, pp. 4837-4869. PMC. Web, 2017.

[3] P. Derler, E. A. Lee and A. S. Vincentelli, "Modeling Cyber–Physical Systems," in *Proceedings of the IEEE*, vol. 100, no. 1, pp. 13-28, 2012.

[4] J. C. Eidson, E. A. Lee, S. Matic, S. A. Seshia and J. Zou, "Distributed Real-Time Software for Cyber–Physical Systems," in *Proceedings of the IEEE*, vol. 100, no. 1, pp. 45-59, Jan. 2012.

[5] B. Sean, S. Shankar, A. John, S. Roundtable, “Reliability of embedded and cyber-physical systems”. *IEEE Security and Privacy*, vol.8,no.5 : pp27-32, 2010.

[6] Y. Tan, M. C. Vuran and S. Goddard, "Spatio-Temporal Event Model for Cyber-Physical Systems," *2009 29th IEEE International Conference on Distributed Computing Systems Workshops*, Montreal, QC, pp. 44-50,2009.

[7] J. Huang *et al*., "Extending service model to build an effective service composition framework for cyber-physical systems," *2009 IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, Taipei, pp. 1-8, 2009.

[8] S. Neda, G. Gopal, “A methodology for modeling and verification of cyber-physical systems based on logic programming”. *SIGBED Rev.* 13, pp:34-42, 2016.

[9] A. Rajeev, L.D. David, “A theory of timed automata”, Theoretical *Computer Science*, Vol.126, no.2, pp.183-235, 1994.

[10] R. Alur, *Principles of Cyber-Physical Systems*, eTextbook: MIT Progress. 2015

[11] X.H. Chen and J. Liu, “Modeling Software Timing Requirements: An Environment Based Approach,” *Chinese Journal of Computer*. Vol.36, No.1, pp. 88-102, Jan. 2013.