International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.6 (2014), pp.49-58
http://dx.doi.org/10.14257/ijmue.2014.9.6.06

Modeling and Verifying of CPS Component Services Based on Hybrid Automata

Jianning Zhang[1], Guanquan Zhang[1,2]*,Rongjie Yan[2],Yi Zhu[3] and Xingjun Qi[1]

[1]School of Computer Science & Technology,
Soochow University, Suzhou, 215006, China
[2]State Key Laboratory of Computer Science, Institute of Software,
Chinese Academy of Science, Beijing, 100190, China
[3]School of Computer Science and Technology,
Jiangsu Normal University, Xuzhou, 221116, China

[*]gqzhang@suda.edu.cn

Abstract

In recent years, the modeling and verifying of Cyber-Physical System (CPS) is now an important aspect of CPS researches. Because of the CPS' complex architecture, it may suffer from the state-space explosion problem when we verify CPS models by model checking methods. Therefore, we offer a method which models CPS with Component Services. The method treats the CPS components as a service provider, and models component services to further simplify the system's state-space. We verify the correctness of this model and solve the synchronous/asynchronous communication problems.

Keywords: *Cyber-physical System; component services; state-space explosion; model checking*

1. Introduction

Cyber-Physical Systems (CPSs) are integrations of the computation and physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa [1]. At present, applications of CPSs mainly include high confidence medical devices and systems, traffic control and safety, critical infrastructure control, advanced automotive and energy conservation [2-5]. Most of these systems are resource-constrained, and have a high requirement for real-time response and fault tolerance. The work in [6] names it as Performance Critical Systems (PCSs), and therefore the goal of this paper is to ensure the performance of CPS.

In model-based design (MBD) and model driven development (MDD), models play an important role in the design process, so we can analyze and verify the system's properties in the early time. The MBD and MDD methods can ensure the quality of the system, and efficiently reduce the time and cost of the system development. However, the quality of system is determined by the architecture of system [7], and hence applying the design and developing methods of the Model-based Architecture-driven to the CPS modeling can efficiently guarantee the system performance. Because it can automatically prove properties of systems, the Model Checking technology is widely used in the MBD and MDD. As the increase of the number of states of the system, the model checking method will suffer from the so called state-space exploration problem, which restricts its application a lot.

ISSN: 1975-0080 IJMUE

Compositional verification techniques are used to cope with this problem in concurrent systems. The idea is to apply divide-and-conquer approaches to infer global properties of complex systems from properties of their components, and separate verification of components limits state explosion [8].

Since CPS is a kind of distributed embedded systems, and its components distributed in different physical environments. If we model all system components, the architecture of system will be too large to analyze, therefore, we combine compositional verification techniques with service-oriented architecture and propose a definition of "Component-Service", which means the components of CPS are the service provider and register the components in system according to the its service model, and the "request / response" operation mode is adopted. As a part of component service model, the physical environment of components is used to describe constrains of the service, while for the upper system it used for being shield from the physical environment, and thus can simplify the architecture of the system.

In Section 2, we introduce the related works. In Section 3, we give the concepts that will be used through the rest of this paper and model CPS component service based on hybrid automata. In Section 4, we demonstrate the CPS component service with an intelligent traffic control system. Finally, Section 5 makes a conclusion for the present study, and also gives some suggestions for the further studies.

2. Related Work

In recent years, many scholars have paid close attention to the research of the Service-oriented Architecture (SOA) and the component-based architecture. The work in [9] proposes the Service Component Architecture Specification to describe the system's model with Service-Oriented Architecture. The work in [10] proposes leverage existing and emerging standards from both the embedded-device and IT domains within a Service-Oriented Device Architecture (SODA) to eliminate much of the complexity and cost associated with integrating devices into highly distributed enterprise systems. Based on DPWS (Device Profile for Web Service), the work in [11] proposes a concept of Service Gateway, which serves as a middleware used for communication and translation between Web services and embedded systems. The work in [12] proposes a framework to analyze and verify the compositional properties of the Web service. In this framework, the BPEL process is used to describe Web service, transformed to automata model and described with Promela language so it can be verified by SPIN. The work in [13] proposes a CP-nets-based design and verification framework for Web services composition, which is used to create and verify the BPEL process. These researches mainly focus on the creation of integrated framework or the description of Web services, and have no concern about the physical world while the physical world is an important aspect of CPS.

The work in [14] proposes models and their relationship to realizations of CPSs. However, the models they design are not structure models but primarily about dynamics, the evolution of a system's state in time, so they can not represent static information about the construction of a system. The work in [15] proposes a compositional method for the verification of component-based systems described in a subset of the BIP language encompassing multi-party interactions. However, this method doesn't consider the physical environment which is an important part of CPS.

Therefore, based on these researches, we introduce a component service model which can reduce the complexity of CPS model.

3. CPS Component Service Model

In this section, we present a basic model of the CPS component service. The CPS combines the communication between discrete and continuous processes, and the hybrid automata includes discrete and continuous state, so we can use it to model CPS component services. Here's the brief introduction about the hybrid automata.

Definition 1 [Hybrid Automata] A hybrid automata HA[16] is defined by $HA = (Q, X, Init, f, Inv, Jump)$ where:

- Q is a set of finite discrete states, which uses to describe the cyber properties;
- X is a set of finite continuous states, which uses to describe the physical properties;
- $Init$ is a set of initial state and $Init \subset Q \times X$;
- $f : Q \times X \rightarrow X$ is a set of continuous dynamic functions;
- $Inv : Q \rightarrow 2^X$ is an invariant whose free variables are from $q \in Q$;
- $Jump : Q \times X \rightarrow 2^{Q \times X}$ is a set of jump functions.

Since the CPS components are mostly distributed at natural environments, and usually are resource-constrained, so the services offered by CPS components are mostly atomic services. An atomic service is a kind of service which can fulfill a function but the service itself cannot be further divided into two or more services. Every atomic service has the following properties:

1) an unique id to be differentiated from other services;
2) a set of variables which contain discrete and continuous state;
3) a set of functions such as numerical calculation, devices control, etc.;
4) a set of ports which can be used to communicate with other services.

Therefore, the formal definition of the atomic service as shown below:

Definition 2 [Atomic Service] An atomic service can be defined as $S =< Sid, port, HA >$, where Sid is the service id, $port$ is a set of service port which is used to communication with other services, $HA = (Q, X, Init, f, Inv, Jump)$ is a hybrid automata which is used to describe properties and behavior of CPS component services.

Example 1. Figure 1 presents a simplified temperature-control system using CPS component services model. There are only 2 services in this model: a temperature-sensing service S_1 and a temperature-changing service S_2. S_1 has two states: low-temperature l_1 and high-temperature l_2, and two ports: p_1 and q_1. S_2 has two states: open-state l_3 and close-state l_4, and two ports: p_2 and q_2. Compared with the classical component-based system model which needs a lot of sensors and actuators, the CPS component service model is much more simplified.

Based on the definition 2, we can define the CPS component. A CPS component is a set of services, and the following are the formal definition.

Definition 3 [connector] Given a set of CPS atomic services $S_1, S_2, \cdots, S_n$, a connector is defined by $Con \subseteq \bigcup_{i=1}^{n} S_i.port$ .where $S_i.port$ is the port of S_i, and for $\forall i = 1, \cdots, n$, we have $\left| Con \bigcap S_i.port \right| \le 1$, i.e., each connector has at most one port per service.

In Figure 1, as an example, the set $\{p_1, p_2\}$ is a connector between S_1 and S_2. This connector describes a synchronization between different services by ports p_1 and p_2 .

Lemma 1. Let $f(s_i, p_i) = s_j$, $f(s_i, p_j) = s_t$, if $p_i = p_j$, then $s_j = s_t$, i.e., the transfer function is determined by the transfer function f.

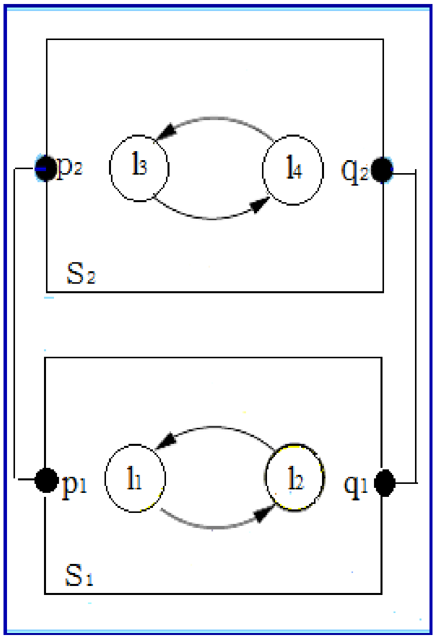

Figure 1. A Simplified Temperature-control System

Definition 4 [executable procedure] Given an atomic service $S =< Sid, port, HA >$, and $s_i, i = 1, 2, \cdots, n$ is a set of states of S, $p_i, i = 1, 2, \cdots, n$ is a set of ports of S, an executable procedure is defined by $\lambda(S) = s_1 p_1 s_2 p_2 \cdots s_n p_n$, where $s_{i-1} \xrightarrow{p_i} s_i, i = 1, 2, \cdots, n$.

Definition 5 [reachable state set] Given an atomic service $S =< Sid, port, HA >$, a reachable state set of S is defined by $rss(S) = \{s_i \mid s_0 \xrightarrow{\lambda} s_i, i = 1, 2, \cdots, n\}$, where s_0 is the initial state of S , and $\lambda = s_1 p_1 s_2 p_2 \cdots s_n p_n$ is an executable procedure of S .

Definition 6 [correctness] Given an atomic service $S =< Sid, port, HA >$, if for every $s_i \in S$, $s_i \in rss(S)$, we say the service S is correct.

The correctness of a service means that we can get a state of the service after a list of executable procedures.

Definition 7 [composite service] A composite service is defined by $CS = \gamma(S_1, S_2, \cdots, S_n)$, where

- $CS.port = \bigcup_{i=1}^{n} Connector(S_i)$, i.e., the compositional service 'ports is a union of all services 'ports.
- $CS.Init = \bigcap_{i=1}^{n} S_i.Init$, i.e., the initial state of compositional service is intersection of the initial state of all services.
- $CS.Q = \sum_{i=1}^{n} S_i.Q$ is a set of cyber properties of all services, and $CS.X = \sum_{i=1}^{n} S_i.X$ is a set of physical properties of all services.
- $CS.f = \sum_{i=1}^{n} S_i.f$ is a transfer function of CS.

Lemma 2. Let $\gamma = \bigcup_{i=1}^{n} S_i.port$ be a connector of a composite service $CS = \gamma(S_1, S_2, \cdots, S_n)$, for $\forall i \in n, S_i \in CS \wedge S_i.port \notin \gamma$, the states of S_i keep constant, i.e., the communication among some services don't influence others.

Theorem 1. Given a composite service $CS = \gamma(S_1, S_2, \cdots, S_n)$, the transfer function $CS.f$ is unique iff (1) for $\forall p_i \in S_i$, if $CS.f(s_i, p_i) = s_j$, we have $S_i.f(s_i, p_i) = s_j$, and (2) the states which are not in $CS.f$ keep constant.

Proof. ($\rightarrow$) Suppose $\exists p_i \in S_i$ so that $CS.f(s_i, p_i) = s_j$ and $S_i.f(s_i, p_j) \neq s_j$. Just let $S_i.f(s_i, p_i) = s_k$, where s_i, s_j, s_k are states of S_i. Depending on the Definition 7, we can know that $S_i.f \subset CS.f$, which is contradictory to that $CS.f$ is unique. So we have $S_i.f(s_i, p_j) = s_j$, and based on Lemma 2, we know that the states which are not in $CS.f$ keep constant.

($\leftarrow$) Suppose there are 2 transfer functions $CS.f_1, CS.f_2$, then for $\forall s_i, s_j, p_i$, if $CS.f_1(s_i, p_i) = s_j$, we have $S_i.f(s_i, p_i) = s_j$, and the states which are not in $CS.f$ keep constant, so we can know that $CS.f_2(s_i, p_i) = s_j$ which is contradictory to the Lemma 1. So the transfer function is unique.

Lemma 3. Given a composite service $CS = \gamma(S_1, S_2, \cdots, S_n)$, if the transfer function $CS.f$ is unique, the service model is correct.

Definition 8 [system] A system can be defined as $Sys =< CS, Init >$, where CS is defined by Definition 7, and $Init$ is an initial state of the system.

As the temperature-control system shown in figure 1, $Sset = < S1, S2 >$ and $Init = l_1 \wedge l_3$ *i.e.*, at initial state, the temperature-control system is at low-temperature state and the temperature-changing devices are also closed.

4. Case Study

In this chapter, we will use a smart-traffic system as an example, and specify the application of CPS component service model which is defined previously.

For the purpose of simplicity, we suppose that there are only five different kinds of services: the GPS service can get location information of every components repeatedly over a time period; the traffic lights service can control the vehicles' state; the parking service can show the information of the nearest car park; the information service can send the real-time road's state to system's control center; and the warning service can send unpredictable circumstances to the system.

$GPS\_Service = < GSid, port1, (period, longitude, latitude) >$, where GSid is an id of this service, port1 is the only port of this service, there are three parameters in $GPS\_Service$: $period$ is the period of $GPS\_Service$ and $longitude, latitude$ is the location information of $GPS\_Service$. In the initial state, $port1 = off$, and when $time = period$, we have $port1 = on$.

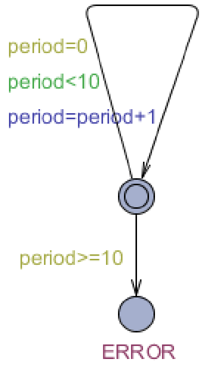

Figure 2. Hybrid Automata Model for GPS Service

The traffic lights service $L\_S = < LSid, port1, (per, red, yel, blue, f) >$, where $LSid$ is an id of this service, $port1$ is the only port of this service, there are four parameters in L_S: per is the period of L_S, red, yellow, blue are light information, and $f : red \times period \rightarrow blue, blue \times period \rightarrow yel, yel \times period \rightarrow red$ is a transition function of L_S.

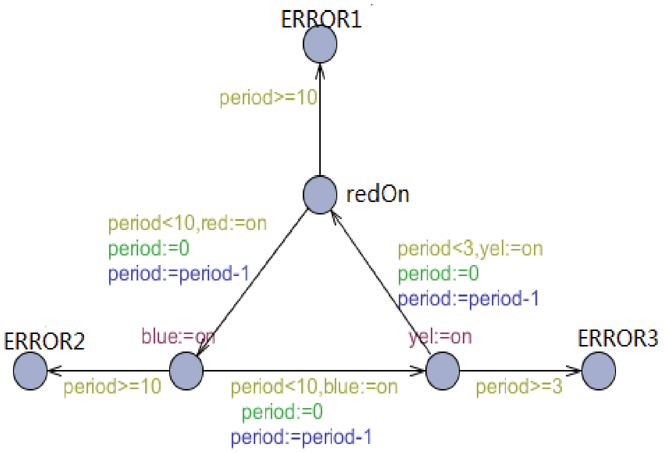

Figure 3. Hybrid Automata Model for L_S

Parking service $P\_S =< PSid, port1, (total, rest, f) >$, where $PSid$ is an id of P_S, $port1$ is the port of this service, there are two parameters in P_S: $total$ is the parking spaces and $rest$ is the rest parking spaces of the car parking, $f: rest \to rest+1, rest \to rest-1$ is the transition function.

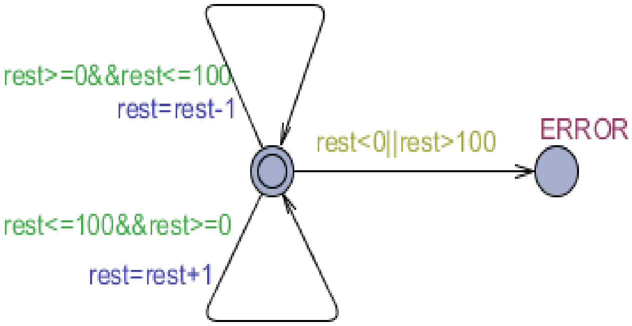

Figure 4. Hybrid Automata Model for P_S

Information service $I\_S =< ISid, port, (stat, isbusy, wea) >$, where $ISid$ is the id of this service, $port$ is the port of this service, there are three parameters in I_S: $stat$ presents the real-time state of roads, $isbusy$ is a Boolean value which is used to check whether the road is blocked, wea is used to offer weather information.

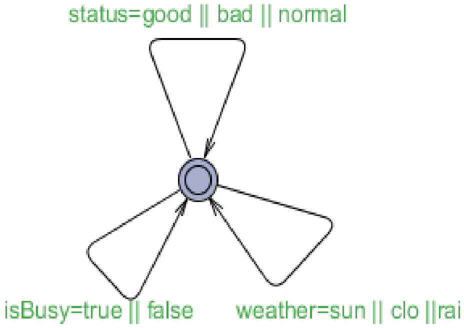

Figure 5. Hybrid Automata Model for I_S

The warning service $W\_S=<WSid, port1,(warning, guard)>$, where $WSid$ is an id of this service, $port1$ is the only port of this service, there are two parameters in W_S: warning and guard. In the initial state, $port1=off$, $warning=false$ while $guard=true$, $port1=on, warning=true$.

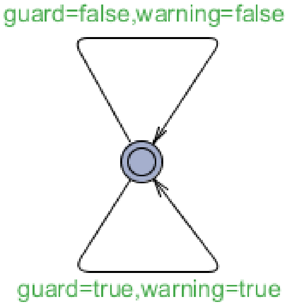

Figure 6. Hybrid Automata Model for W_S

5. Conclusion and Future Work

In this paper, we propose a model of CPS based on component services, verify the correctness of the model, and ensure the synchronous/asynchronous communication among services by connectors. We also give an example to demonstrate the application of CPS component service model. In the future, we will give a further validation on deadlock free models.

Acknowledgements

The work of this paper is supported by the Natural Science Foundation of Jiangsu, China (Grant No. BK2011281) and the Applied Fundamental Research Program of Suzhou, China (Grant No. SYG201241).

References

[1] E. A. Lee, "Cyber physical systems, Design challenges", Proceedings of ISORC, Piscataway, NJ, IEEE, (**2008**), pp. 363-369

[2] J. Hatcliff, A. King, A. MacDonald, A. Fernando, M. Robkin, E. Vasserman, S. Wininger, and J. M. Golderman, “Rationale and Architecture Principles for Medical Application Platforms”, Proceedings of the 3rd IEEE/ACM ICCPS,(**2012**) April 17-19, Beijing, China.

[3] P. Park and C. Tomlin, “Investigating Communication Infrastructure of Next Generation Air Traffic Management”, Proceedings of the 3rd IEEE/ACM ICCPS, (**2012**) April 17-19, Beijing, China.

[4] J. Kim, K. Lakshman, R. Rajkumar and R. Tasks, “A New Task Model with Continually Varying Periods for Cyber-Physical Systems”, Proceeding of the 3rd IEEE/ACM ICCPS, (**2012**) April 17-19, Beijing, China.

[5] C. L. Fok, M. Hanna, S. Gee, T. C. Au, P. Stone, C. Julien and Sriram Wishvanashi, “A Platform for Evaluating Autonomous Intersection Management Policies”, Proceedings of the 3rd IEEE/ACM ICCPS, (**2012**) April 17-19, Beijing, China.

[6] P. H. Feiler, B. A. Lewis and S. Vestal, “The SAE architecture analysis and design language (AADL)– A standard for engineering performance critical systems”, Proceedings of IEEE Computer Aided Control Systems Design, (**2006**) October 4-6, Munich, Germany.

[7] C. Atkinson and T. Kuhne, “Model-driven development: A metamodeling foundation”, IEEE Software, vol. 20, no. 5, (**2003**).

[8] M. Beisiegel, H. Blohm and D. Booz, “Service Component Architecture (SCA), version 1.0. Billerica: Organization for the Advancement of Structured Information Standards (OASIS)”, (**2007).**

[9] S. Deugd, K. Kelly, B. Market and J. Ricker, “SODA: Service oriented device architecture”, IEEE Pervasive Computing, vol. 5, no. 3, (**2006**).

[10] C. Buckl and S. Sommer, “Generating a Tailored Middleware for Wireless Sensor Network Application”, Proceedings of the IEEE SUTC, (**2008**) June 11-13, Taichung, Taiwan.

[11] L. Souza, P. Spiess, D. Guinard, M. Kohler, S. Karnoskov and D. Savio, “SOCRADES: A Web service based shop floor integration infrastructure”, Proceedings of the Internet of Things, (**2008**) March 26-28, Zurich, Switzerland.

[12] C. Hein, T.Ritter and M.Wagner, “Mode l-Driven tool integration with Model Bus”, In Workshop Future Trends of Model-Driven Development, (**2009**).

[13] J. Huang, “A SOA Model of Cyber Physical Systems”, https://utd.edu/~i lyencourse/service/project/jian.pdf.

[14] K. Bae, P. C. Olverzky, T. H. Feng and E. A. Lee, “Verifying hierarchical Ptolemy II discrete-event models using Real-Time Maude”, Science of Computer Programming, vol. 77, (**2012**).

[15] S. Bensalem, M. Bozga, T.H. Nguyen and J. Sifakis, “Compositional verification for component-based systems and application”, IET Software, vol. 4, no. 3, (**2010**).

[16] H. A. Henzinger, “The Theory of Hybrid Automata”, Proceedings of LICS, (**1996**) July 27-30, New Jersey, USA.