Specification, Analyzing Challenges and Approaches for Cyber-Physical Systems (CPS)

Kaiyu Wan * K.L. Man† D. Hughes‡

Abstract— **Cyber-Physical Systems (CPS) integrate computation with physical processes. By merging computing and communication with physical processes CPS allows computer systems to monitor and interact with the physical world. However, today's computing and networking abstractions do not adequately reflect the properties of the physical world. This shortcoming necessitates the development of effective methods and tools for analyzing and designing CPS. This paper analyzes the limitations of the current tools and methods by illustrating a motivating example of health care systems and proposes a unified framework for designing, simulating, and verifying CPS.**

Keywords: Cyber-Physical Systems, specification, analyzing

1 Introduction

Cyber-Physical Systems(CPS) are integrations of computation with physical processes. Networked embedded computers are used to monitor and control physical processes based upon local (i.e. in-network) and remote (i.e. back end) computation [9]. CPS tend to feature a tight coupling between physical and software components. CPS may operate on different spatial and temporal scales and exhibiting multiple and distinct behavioral modalities. Furthermore, CPS are continuously interacting with the physical world, as a result the behavior of a CPS may change with the operational or environmental context.

Applications of CPS include: high confidence medical devices and systems, traffic control and safety, advanced automotive systems, process control, energy conservation, environmental control, avionics, instrumentation, critical infrastructure control, distributed robotics (telepresence, telemedicine), defense systems, manufacturing, and smart structures [6].

*Xi'an Jiaotong-Liverpool University (XJTLU), 111 Ren'ai Road, Suzhou, Jiangsu 215123, China. Email: kaiyu.wan@xjtlu.edu.cn. Tel: +86 512 8816 1506. Fax: +86 512 8816 1899.

†Xi'an Jiaotong-Liverpool University (XJTLU), 111 Ren'ai Road, Suzhou, Jiangsu 215123, China. E-mail: ka.man@xjtlu.edu.cn.

‡Xi'an Jiaotong-Liverpool University (XJTLU), 111 Ren'ai Road, Suzhou, Jiangsu 215123, China. E-mail: daniel.hughes@xjtlu.edu.cn.

The physical platforms which support CPS offer five capabilities: computation, communication, precise control, remote collaborative and autonomous capabilities. Some CPS are also required to perform complex computation. For example: a CPS installed in an automobile may run a traffic control algorithm and compute the best route according to the current traffic situation.

In the case of environmental monitoring, CPS may be distributed over geographically large and remote areas such as forests, rivers, and mountains where they are expected to operate without human intervention for long periods of time with a constrained power supply. In such an environment, collecting accurate and timely information over unreliable low power ad-hoc networks is a key challenge.

CPS for avionics, electric power control, water resource control and defense systems ask for precise and reliable control, which makes applying software methodologies to ensure the quality of software extremely important. In contrast to traditional embedded systems, CPS interface directly with the physical world. This makes detecting changes in the environment and adapting the system's behavior accordingly a key challenges of designing such systems.

Moore's law implies that the physical size of an embedded computer of fixed capability will halve every two years. Coupled with falling prices, this makes it ever more feasible add computational capability to physical systems. The growing pervasiveness of CPS is predicted to have broad impact on the economy and society [10]. By merging computing and communication with physical processes and mediating interaction with the physical world, CPS bring many benefits, including: making physical systems safer and more efficient; reducing the cost of building and operating physical systems; and allowing for individual machines to work together to form complex systems that provide new capabilities.

In the physical world, the passage of time is inexorable and concurrency is intrinsic. However today's computing and networking abstractions do not reflect either of these properties. Lee argued that the mismatch between these abstractions and properties of physical processes impede technical progress [10]. Therefore, technical approaches that can bridge the abstraction gap are urgently required.

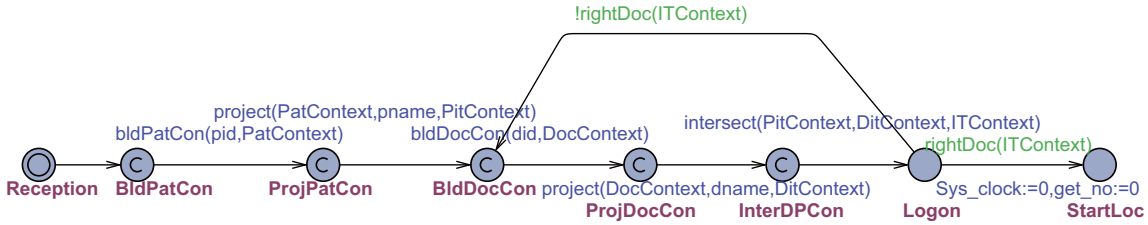

Figure 1: Initialization Mechanism

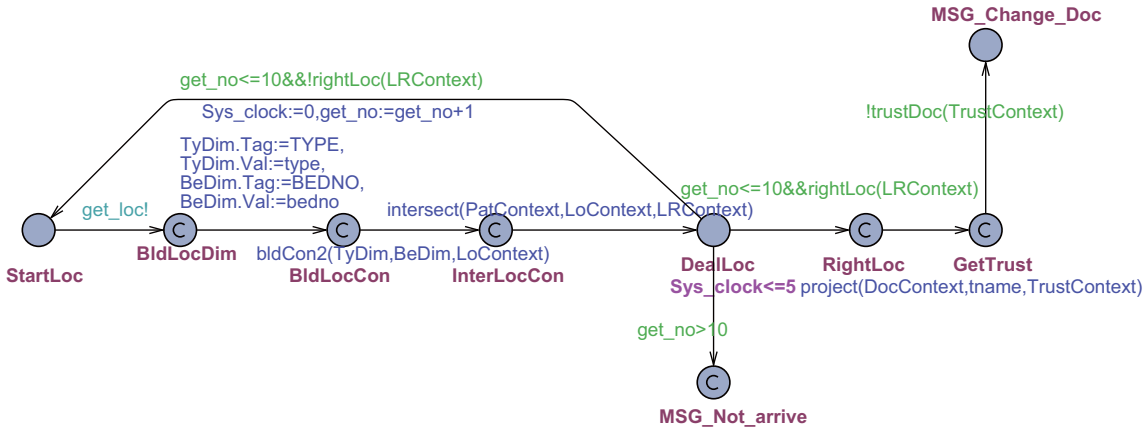

Figure 2: Location Mechanism

CPS products are in general heterogeneous systems comprised of multiple types of physical systems and multiple models of computation and communication. Although there are effective methods and tools for designing both computational and physical systems, the design of CPS is much more than the union of those two fields. Therefore new methods and tools must be investigated, which will allow system designers to apply the principles of CPS to new industries and applications in a reliable and economically efficient way.

The goal of this paper is to investigate tools for analyzing and designing CPS and propose a unified framework for CPS. The structure of the paper is as follows: In section two, the motivating example of an in-home health care system is introduced. In section three, the background of CPS is reviewed. In section four, the limitations of current work are discussed. In section five, popular candidates for modeling CPS are described. In section six, we propose a unified framework for designing, simulating, and verifying CPS. We conclude this paper with discussing our future research directions in section seven.

2 Motivating Example

As the national population ages, we will need to make more efficient use of our health care systems, including facilities, medical data and information. Currently many elders need assistance in physical mobility so they have to move into nursing homes. Besides, some elders with cognitive impairment need daily supervision of medication and health-condition monitoring. With CPS facilities and infrastructure, those people can stay at home. In addition, physiological parameters critical to the medical maintenance of health can be monitored remotely. With in-home health care, people can maintain their independence without loss of privacy and save on nursing expense at the same time.

In ubiquitous health care systems, medical devices and medical information systems will be connected through wired and wireless network to form a secured, reliable, and privacy-preserving health care. The ubiquitous health care system aims to provide treatment accurately, reduce expense effectively and guarantee health-care services exist anytime and everywhere. It receives digital signals from each device, observes every patient's condition, analyzes context information, communicates with the doctor or the nurse remotely and gets information about the next treatment.

Let's take a close look at an in-home health care system as an example of a CPS. In our case-study, a general in-home health care system consists of the following components : $WristSensor$, $PillContainer$, $InfusionPump$, $Controller$, $Communicator$, $DailyCheckPDA$ and $CPSDataBase$. The functionality of each component is as follows:

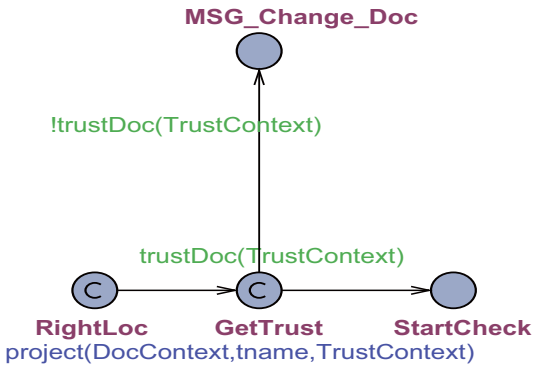

Figure 3: Trustworthy Mechanism : Trust Judging

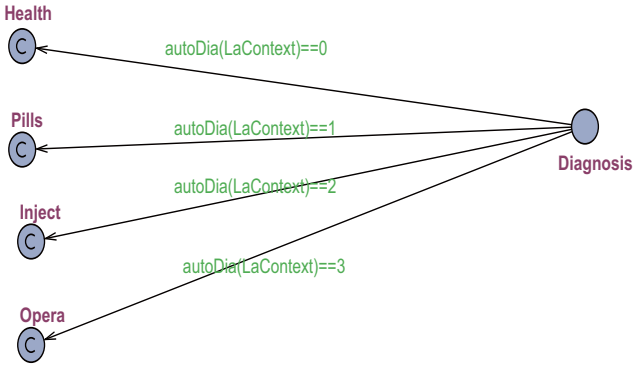

Figure 4: Diagnosis Mechanism

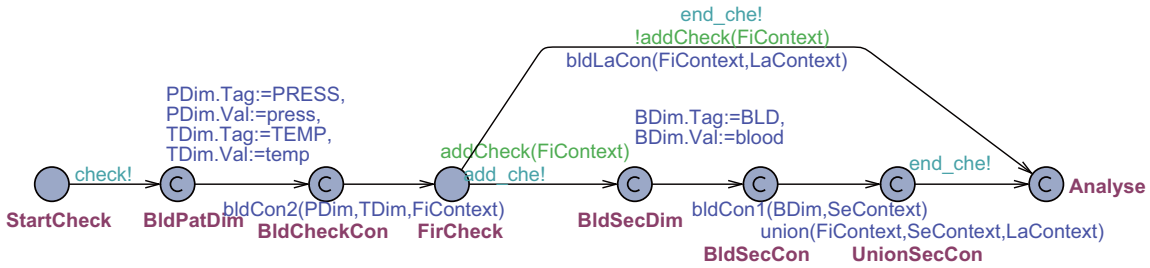

Figure 5: Daily Check Mechanism

- The elders are equipped with a $WristSensor$ which can check the elders' blood pressure, sugar level, heart beat and temperature on a regular base everyday. These sensors may vary according to the elders' specific situation though. If the collected data is out of normal range, the $WristSensor$ can send warning messages to the $Controller$.
- The elders can be equipped with the $PillContainer$ so that the elders can have medication from the pill box regularly everyday.
- The elders may be equipped with the $InfusionPump$ so that they can be treated at home. Once the liquid runs out the $InfusionPump$ can send warning messages to the $Controller$.
- The $Controller$ collects the information, analyzes it and decides whether the information should be sent to the hospital or local community through the $Communicator$ component.
- The $Communicator$ is in charge of communicating through wired and wireless networks.
- The $CPSDataBase$ stores various information including elders' preference on the doctors, elders' medical history, and elders' condition etc.
- The $DailyCheckPDA$ may be used to help doctors to do daily checking, verify the identity of doctors, sense the patient's location, assess if the doctor is trustable, and rebuild the trust degree of the doctor, get the information of elders, and execute the corresponding treatment.

Based on the above requirements, we modeled and verified the $DailyCheckPDA$ component with UPPAAL, which is a toolkit for simulation and verification of real-time systems [2].

- $DailyCheckPDA$ consists of following mechanisms: *sensing, initialization, location, trust assessment, daily check and diagnosis mechanism.* We construct all the mechanisms except *sensing* as the main template named PDA. The sensing mechanism is split into *elders' condition sensing mechanism* and *location sensing mechanism*, while the former is constructed as the $CheckSensor$ template and the latter as the $LocationSensor$ template.
- The scenario for $DailyCheckPDA$ is as follows : If the daily check is not executed on time, the PDA sends a warning message to the reception. Within the time limit, if the trust degree of the doctor satisfies the requirement, the PDA starts to check functions, otherwise, the PDA would send a "doctor replacement message" to the reception. The PDA checks elder's condition including blood pressure,

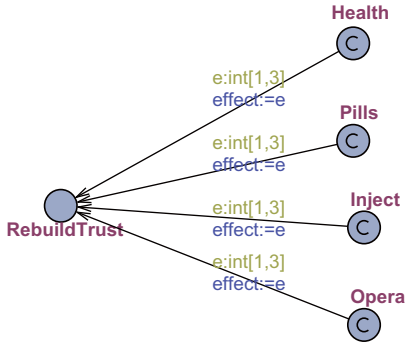

Figure 6: Trustworthy Mechanism : Trust Rebuilding

body temperature, and heart rate etc. The *PDA* may add the checked items according to elder's condition. Based on the collected elder's data, the doctor makes diagnosis and decide treatment methods.

Below is a scenario of a successful diagnosis process. Due to the space limitation, we split the modeling figure into six parts and present them separately as follows:

- A doctor, whose number is 1, wants to make daily check of a patient whose number is 1. In this *PDA* system, information contexts of doctor and patient are built in initialization mechanism. Through computing these contexts, system judges whether the doctor is a proper doctor to make daily check of this patient by function *rightDoc* shown in Figure 1.
- Then the doctor needs to locate the specific patient. *PDA* system senses current location by location sensor(*LocSensor*), then builds and processes current location context. Function *rightLoc* taking the location context and patient context as parameters and judges whether the doctor has reached this patient. If the doctor hasn't located the patient within 10 time units after sensing the location sensing, *PDA* will send a message to reception. The scenario is described in Figure 2.
- Now the doctor has reached the patient. This patient has the right to decide if the doctor is trustworthy. If the doctor is not trustworthy, the patient can ask the doctor to leave and send a message to the reception. This mechanism is realized by function *trustDoc*. The scenario is described in Figure 3.
- Then the daily check begins. *PDA* gets patient's blood pressure and body temperature by the checking sensor(*CheckSensor*). Then *PDA* builds patient's condition context(which is named as *FiContext*) by the sensed information. The function *addCheck* judges whether this patient needs to be checked more items. If so, *PDA* would get blood analysis condition by the checking sensor(*CheckSensor*). The patient's condition is rebuilt by combining blood analysis context and *FiContext*. No matter whether the patient needs to be checked more, the patient's condition is transferred to context *LaContext* shown in Figure 5.
- The *PDA* system makes diagnosis and provides proper treatment for the patient according to the patient's condition context. This functionality is realized by Function *autoDia* shown in Figure 4.
- After diagnosis is done, the trust degree of the doctor is rebuilt. Rebuilding of trust degree is realized by function *reBuildTrust* shown in Figure 6.
- Now one round of the daily check is successfully done. There are a lot of other scenarios in this System. For example, a variant of the scenario is that doctor is not a proper doctor to do the daily check, or the doctor does't find his patient, or the patient does't trust this doctor, and so on. This scenario is only the most successful one.
- The *PDA* system runs safely and reliably after several rounds of simulation.

We have shown our modeling in [19]. However, in that model we didn't express and verify the trustworthy properties, which is quite important for the confidential health care systems. Thus we made the following extension :

- We make the division of different components more reasonable. For example, sensors (*LocSensor* and *CheckSensor*) only have the right to sense the required information but have no right to build information. Only *PDA* template has the right to build information.
- We make some proper change in gaining doctor's identity and the elder's identity. We set two parameters of *PDA* template: *pid* and *did*.
- We add the trustworthy judging and rebuilding mechanisms.

Based on the model, we check the *Accessibility* of the model as follows:

- E⟨⟩Pda.MSG_Not_arrive : *PDA* can send a message to the reception warning that the doctor hasn't arrived.
- E⟨⟩Pda.MSG_Change_Doc: *PDA* can send a message to the reception if the elder doesn't trust the current doctor.

- E⟨⟩Pda.BldLocDim&&LocSensor.Sense&& CheSensor.Idle: When the PDA needs the location information, the Location sensor is activated and the Check sensor is idle.
- E⟨⟩Pda.Diagnosis&&LocSensor.Idle&&CheSensor.Idle: When the PDA is sensing elder's condition, the Location sensor and the Check sensor are idle.

3 Characteristics of Cyber-Physical Systems

Providing accurate models of CPS is complicated by a number of factors including heterogeneity, unreliable network communication, mobility and a tight coupling with the physical environment. In combination, these characteristics introduce a level of uncertainty that is difficult to capture using traditional formal modeling techniques.

- *Heterogeneity*: CPS demonstrate a high level of heterogeneity. This may include:
  - sensor nodes with a small amount of RAM and flash memory connected via low-bandwidth unreliable wireless networks,
  - mobile devices such as smart-phones operating over GSM-based technologies, and
  - high end workstations and servers connected via reliable wired networks. This level of complexity demands rich support for the modeling of underlying technologies.
- *Unreliable networking*: in modern CPS, application elements interact in a distributed fashion over a network. Furthermore, many cyber-physical applications operate over ad-hoc wireless networks and in power constrained environments. For those CPS that incorporate Wireless Sensor Network (WSN) technologies, the use of low-power wireless network technologies such as 802.15.4 [8] and Nordic [12] leads to a high rate of packet loss, making distributed interactions unpredictable. To be successful, any modeling approach for networked embedded systems must therefore respect this uncertainty.
- *Mobility*: CPS that incorporate mobile devices present additional complexity. In mobile systems, devices may interact opportunistically. For example: in a vehicular network, interactions may be possible only when two vehicles come within range of each other. Furthermore, the movement of mobile nodes may be determined by unpredictable factors such as user behavior.
- *Tight Environmental Coupling*: even in statically deployed CPS, a tight coupling with the environment means that system externalities require greater consideration than in traditional distributed systems. For example, in a wireless system, external interference sources may have a greater effect upon the reliability of distributed interactions than any of the internal system components. It is therefore necessary to provide good support for modeling of these externalities.

4 Design, Analysis and Tools for Cyber Physical Systems (CPS)

The modeling and verification of CPS is complicated by their heterogeneous nature as well as their complexity. A cyber physical system can be modeled by either a structural/architectural specification (how their components: sensors, actuators and processors work; and how they are interconnected together) or behavioral specification (showing the response of each component to an internal or external event).

Existing modeling techniques for CPS rely upon semantics to represent the relationship between the cyber and physical features of a CPS, which is necessary for accurate modeling of any system.

Generally speaking, mathematical formalisms (e.g. hybrid automata [7] and process algebras [13]) and description languages (e.g. Labeled Hybrid Petri Net [1]) are popular candidates for modeling CPS.

Although hybrid systems are a very versatile tool for the specification and analysis of CPS models, they do not consider some fundamental concepts, which are intrinsic or essential in CPS models such as networking, services and support for performance & functional analysis. Furthermore, modeling or analyzing a CPS with all of its details always results in state explosion. Nevertheless, over the years, various techniques, algorithms, specification logic and software tools have been developed (e.g. [1, 18]) for simplifying CPS models to achieve certain verification goals.

The remainder of this section provides a number of case study examples of CPS modeling from the literature.

4.1 Natural Gas Transport System

Security Process Algebra (SPA) [3] is an extension of the Calculus of Communicating Systems (CCS) [4]. Using SPA, a system is modeled as a composition of processes in which the composition can be in a sequential, alternative and/or parallel fashion.

In [3], SPA was applied to model a natural gas transport system. Such a transport system is a critical infrastructure consisting of a networks of pipes. This CPS model (the natural gas transport system) contains a rich interaction of physical flows, physical actions and cyber actions.

Then the model checker CoPS [3] was used to validate

several information flow security properties in the system which are important properties for CPS because of the inherent composition of various cyber and physical elements. The work presented in [3] provides several research directions for model checking the information flow security properties of CPS models.

4.2 Cooling System for a Nuclear Reactor

A Labeled Hybrid Petri Net (LHPN) is a Petri net model [14] originally developed for representing analog and mixed-signal (AMS) circuits; and it is highly inspired by both Petri net and hybrid automata.

The CPS considered in this example was the cooling system for a nuclear reactor. The temperature of the nuclear reactor is monitored. When the temperature is too high, one of two control rods is put to cool the reactor core.

A methodology for automatically abstracting the CPS model of the cooling system for a nuclear reactor described as a LHPN was proposed in [1]. This methodology leads to significant simplifications of LHPN models (in general) and it was implemented in the LEMA verification tool [1]. Applying the LEMA verification tool (for checking properties of LHPN models), a relevant safety property (the reactor never shuts down) of the system was verified successfully. Overall, this work presents an efficient method for reachability analysis of CPS systems described as LHPN models.

4.3 Temperature Control System for Two Zones

In [11], an architectural modeling approach was used to model a CPS deployed as a temperature control system for two zones. The architecture of this system was generated by AcmeStudio [16].

However, the architectural elements describe only the structure information about a system. In order to perform a formal analysis on the system behavior, the architecture must be annotated with behavioral information.

Such an architecture generated by AcmeStudio was annotated to Linear Hybrid Automata (LHA) [15] and further analyzed using PHAVer [15]. An acceptable range of timeout periods was identified given the minimum and maximum ranges of rates of change of temperature while heating and cooling.

5 Limitations of current work

As you may see from Section 2, there are certain limitations in our modeling. For example, the security properties shown below can not be verified yet.

- A[] not deadlock: The system has no deadlock.
- A[]Pda.Health+Pda.Pills+Pda.Inject+Pda.Opera <=1: The PDA can only decide one diagnosis method.
- A[]Pda.RightLoc imply get_no<=10: The PDA must goto the right location within 10 time units after the location sensor's detection.
- A[]Pda.StartCheck imply TrustDoc(TrustContext): Only the trusted doctor has the right to do the daily check
- A[]Pda.Health+Pda.BldSecCondition<=1: The PDA can decide automatically that the healthy person doesn't need the second round of check.

Although these properties should be satisfied in theory, with UPPAAL we can't verify them. We argue the reason might be that the computation ability of UPPAAL server is quite limited, and verification of our example is beyond the ability of the tool. Another limitation of our modeling is that we have not modeled and simulated the communication among the components due to the limitation of the tool.

If we put the health care systems into practice, we can see developing tools for specifying and analyzing systems actually raises many challenges.

- In reality, people have different levels of clinical criticality. Therefore the information collected from the devices have different levels of importance. However these devices are communicated through the shared infrastructure. How to specify the levels of clinical criticality, how to combine the levels of importance of devices information with the network resources, and develop a on-demand low-cost deployment is one of challenges for designing tools for plug and play medical devices in the future [17].
- The information collected from the devices are generally continuous and realtime streams. Therefore we need tools with clear semantics to represent an on-demand, reliable realtime streaming of critical medical information in a wired or wireless network.
- With the devices plug and play at run time, it is very important to make sure the integrated system remain reliable and its behavior predictable. Therefore we need tools and method to integrate model-based, precise, and predictable systems from components.
- Elders' medical information and history are confidential. Therefore we need tools and method to check and guarantee if the systems remain secure when malicious attacks from either the cyber or physical domains.

6 A Unified Framework for Specification and Analysis of Cyber Physical Systems

In this section, we discuss the features, benefits and applicability of a unified framework (an enhanced version of the framework proposed in [5]), which enables the design and analysis of large -scale and heterogeneous CPS in an unified manner.

The key features required for the modeling, simulation and verification of CPS are enumerated below:

- *Heterogenous application support*: a CPS usually consists of various physical devices and hence any modeling approach should be able to simulate heterogeneous logics simultaneously.
- *Physical modeling*: the physical modeling environment should support mathematical expressions and incorporate specification logic which facilitates formal verification (e.g. model checking).
- *Scalability support*: a unified modeling approach sound provide support for the development, simulation and verification of both small-scale and large-scale CPS.
- *Mobility support*: to provide support for the modeling of CPS, mobility must be considered including the provision of suitable abstractions to model the movement of mobile devices.
- *Integration with existing simulation and verification tools*: easy-to-use support for connecting the modeling environment to existing simulation and verification tools is required.

Together, the features of the above mentioned unified framework will facilitate the development of CPS by providing various levels of modeling, simulation and verification in an integrated environment. Such a system can reduce development efforts by enabling the reuse of existing simulation and verification tools.It is worth highlighting that today's models and methodologies for the design and analysis of CPS typically separate the cyber and physical features of the system design. Due to this separation, it becomes difficult to assess the impacts and tradeoffs of design decisions that cut across the boundaries between these domains. The further development of the unified framework will particularly focus on the relationship between the modeling and analysis of cyber and physical features.

7 Future Research Directions

In this paper, we present research directions for a unified framework that supports the specification and analysis of CPS. However, the proposed framework is at a prototypical stage. In reality, there are several factors that complicate the modeling of distributed CPS. In the near future, we shall investigate tools that can be used to model and analyze the communication and interaction in distributed CPS. Modeling and verifying the security of network protocols is our another research direction. Each of these individual subsystems plays an important role in providing a unified framework for specification and analysis of CPS.

References

[1] Robert A. Thacker, Kevin R. Jones, Chris J. Myers, Automatic Abstraction for Verification of. Cyber-Physical Systems, Proceedings of the 1st International Conference on Cyber-Physical Systems, 2010, Stockholm. Sweden.

[2] Gerd Bejramm, Alexamdre Davod, and Kim G.Larsen, A Tutorial on UPPAAL, November 2004, http://www.uppaal.com/.

[3] Ravi Akella, Bruce M. McMillin, Model-Checking BNDC Properties in Cyber-Physical Systems, Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference, 2009, Washington, USA.

[4] Robin Milner, A Calculus of Communicating Systems, Springer Verlag, 1980.

[5] Ji Eun Kim, Daniel Mosse, Generic Framework for Design, Modeling and Simulation of Cyber Physical Systems, ACM SIGBED Review - Special issue on the RTSS forum on deeply embedded real-time computing, V5, N1, 2008, pp. 1-2.

[6] Cyber-Physical Systems: Executive Summary, CPS Steering Group, 2008.

[7] Thomas A. Henzinger, The Theory of Hybrid Automata, Proceedings of the 11th Annual Symposium on Logic in Computer Science, IEEE Computer Society Press, 1996, pp. 278-292.

[8] IEEE Standard 802.15.4d-2009, Low-power Wireless Network technology, IEEE Computer Society, April 17, 2009.

[9] Edward A. Lee, Cyber-Physical Systems - Are Computing Foundations Adequate?, Position Paper for NSF Workshop On Cyber-Physical Systems : Research Motivation, Techniques and Roadmap, October 16 - 17, 2006 Austin, TX.

[10] Edward A. Lee, Cyber Physical Systems: Design Challenges, International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), May, 2008.

[11] Akshay Rajhans, Shang-Wen Cheng, Bradley Schmerl, David Garlan, Bruce H. Krogh, Clarence Agbi, Ajinkya Bhave, An Architectural Approach to the Design and Analysis of Cyber-Physical Systems, Proceedings of the 3rd International Workshop on Multi-Paradigm Modeling, Denver, USA, 2009.

[12] Nordic, Low-power Wireless Network Technology, http://www.nordicsemi.com/index.cfm?obj=product &act=display&pro=97.

[13] J.C.M. Baeten, W.P. Weijland, Process Algebra, Cambridge University Press, 1990.

[14] A Petri Net Model, http://www.informatik.uni-hamburg.de/TGI/PetriNets.

[15] G. Frehse, PHAVer : Algorithmic Verification of Hybrid Systems past HyTech, Lecture Notes in Computer Science 3414, Proceedings of the 5th International Workshop on Hybrid Systems: Computation and Control (HSCC), 2005, pp. 258-273.

[16] B. Schmerl, D. Garlan, AcmeStudio : Supporting Style-Centered Architecture Development, Proceedings of the 26th International Conference on Software Engineering, 2004, Edingurgh, Scotland.

[17] Lui Sha, Sathish Gopalakrishnan, Xue Liu, and Qixin Wang, Cyber-Physical Systems : A New Frontier, Machine Learning in Cyber Trust (ISBN 978-0-387-88734-0), Springer US, April 2009.

[18] Marius C. Bujorianu, Manuela L. Bujorianu, Howard Barringer, A Unifying Specification Logic for Cyber-Physical Systems, Proceedings of the 17th Mediterranean Conference on Control and Automation, 2009, Tessaloniki, Greece.

[19] Shujun Zou, Kaiyu Wan, and Zongyuan Yang. Modelling and Verifying of Medical Diagnosis System Based on Context-awareness Framework, FCST 2010.