Challenges On Coordination For Cyber-Physical Systems

Sun Meng
LMAM & Department of Informatics, School of Mathematical Sciences
Peking University
Beijing, China
sunmeng@math.pku.edu.cn

***Abstract*—Cyber-physical systems integrate computing and communication with monitoring and control of physical entities. The complex interaction with the physical world makes coordination models and languages very important for the analysis, design and development of cyber-physical systems. This paper discusses several new challenges on coordination models and languages in the context of the emerging phenomenon of cyber-physical systems.**

Keywords-coordination; cyber-physical systems; verification; QoS requirements; composition; self-adaptation

I. INTRODUCTION

The term "*cyber-physical systems*" (CPSs) refers to a new generation of systems that integrate computing and communication with monitoring and control of physical entities [5,14,15]. The complex interaction with the physical world through computation, communication, and control leads to the dynamic behavior of CPSs. Example CPSs include airplanes and space vehicles, hybrid gas-electric vehicles, power grids, oil refineries, medical devices, patient monitoring, defense systems, etc. The design of such systems requires understanding the joint dynamics of software, hardware, networks and physical processes. Coordination is playing a crucial role in the operation of such systems, and must be trustworthy, dependable, reliable, safe, secure, efficient and in real-time.

Researchers from different disciplines have recognized the challenges on coordination for cyber-physical systems [15]. A proper coordination model is very important for the design of cyber-physical systems. Although coordination models and languages have been investigated for decades, progress in this area still remains limited to relatively simple systems, especially software systems in which only discrete behavior is emphasized. When the continuous behavior for physical processes is taken into consideration, new trends appear with the possibilities to change the existing coordination mechanisms.

In the past years, computer scientists and software engineers have pioneered the development of methods and tools, such as new programming languages, visual modeling formalisms, different verification and validation techniques and tools, embedded systems architectures, and hardware/software co-design methods to ensure system trustworthy, reliability, safety, efficiency and fault-tolerance. Meanwhile, researchers in system and control sciences have also developed a variety of powerful methods and tools such as state space analysis, optimization, robust control, stochastic control, etc. Cyber-physical systems research aims to integrate methods and tools from different disciplines to develop new CPS science and engineering principles and apply them in the design and development of new cyber-physical systems.

To build large-scale cyber-physical systems, one often needs to select components from a set of functionally equivalent candidates, meaning, those that implement the same functionality but differ in their non-functional characteristics, i.e., Quality of Service (QoS) properties.

Cyber-physical systems go beyond traditional embedded and distributed systems, which are typically closed with respect to both physical locations and computational boundaries. Nowadays, most of the cyber-physical systems are shifting towards openness, which not only makes controlling the systems more convenient and flexible, but also introduces extra complexity into the systems heavily. Systems are becoming multi-scale and the abilities for self-adaptation and reconfiguration are becoming more important when it is possible for agents to join or leave systems at run-time, which also makes coordination of agents in cyber-physical systems a more difficult task [10].

This paper is organized as follows. First, Section II gives a brief introduction to the notion of coordination and the usage of coordination models and languages in software systems and service-oriented applications. Section III discusses several challenges on coordination models and languages in the context of CPSs. Finally, we conclude in Section IV.

II. COORDINATION IN SOFTWARE SYSTEMS AND SERVICE-ORIENTED APPLICATIONS

Coordination models and languages have been widely used in specification and implementation of interaction protocols for communicating software components [2,4,6,19]. The coordination mechanisms provide powerful glue code for describing the interaction protocols in software systems, and focus on the global behavior of the whole system by coordinating the behavior of separate components in a distributed environment. The use of coordination models and languages distinguishes the interaction among components from computing in single component explicitly. This can simplify the development process for complex systems and reasoning and verification of system properties.

With the growing complexity and size of modern cyber-physical systems, the need for models to support their

Published by Atlantis Press, Paris, France.

architectures, design their coordination, and manage their QoS at run time has become a critical issue. In recent years, Service Oriented Applications (SOAs) have emerged as viable approaches to tackle the complexity in distributed systems [20,21]. SOAs consist of services running on large-scale distributed platforms and are notoriously difficult to construct. Most service-oriented applications rely on the collaborative behavior of their constituent services / components, which implies complex, intricate coordination. Each component or service, in general, takes part in multiple scenarios. Consequently, not only the functionality of components, but also their interaction protocols are important in the development of such distributed systems. Therefore, construction of these applications crucially depends on deriving a correct coordination model that specifies the precise order and causality of the actions of their constituent services.

With the growth of interest in service-oriented applications in recent years, a key aspect of aggregating business processes and web services is the coordination among services that are autonomous, platform-independent computational entities running on large-scale distributed systems. Services can be invoked by other services or simply interact with each other in order to carry on a task. Compositional coordination models have a key role in such applications to support the development of loosely coupled, large-scale distributed applications [17].

Wegner took coordination as constrained interaction in [23]. Later in coordination models like Reo [2], interactions are taken as constraints on communication actions, and formally as mathematical relations, which are non-directional and specify what must hold in a declarative way.

III. Research Challenges

Cyber-physical systems have always been held to a higher reliability and predictability standard than general-purpose computing facilities. The increasing complexity in both hardware and software components pose many challenges to realizing the promising prospects of CPSs. Particularly, general theories are needed for designing and reasoning both physical and behavioral interactions among different hardware/software components. Furthermore, such theories need powerful tool supports to make them usable in analysis and synthesis of coordination models. In the following subsections, we briefly discuss some challenges on coordination for cyber-physical systems that will be faced by researchers.

A. Compositional Modeling of Interactions in Hybrid Contexts in CPSs

Compositional modeling means that a complex system that consists of a network of interacting subsystems is modeled by modeling the individual subsystems and the interactions among them respectively. Nowadays cyber-physical systems often have complex architectures, which makes it difficult or even impossible to model such systems without compositional modeling methods. Thus one of the most challenging aspects of cyber-physical systems is compositional modeling of interactions and the related properties in the hybrid contexts of cyber-physical systems. Meeting such a challenge requires systematic application and adaptation of state-of-the-art models and techniques based on existing methods, and is likely to reveal the need for the development of new models and techniques.

The behavior of cyber-physical systems consists of a combination of both continuous and discrete dynamics, which are in general connected to each other. Different compositional modeling approaches have been developed for continuous and discrete systems respectively. Composition of continuous systems can be determined as the intersection of the trajectories of all the subsystems being composed, while for discrete systems, several approaches like process algebra (CSP, CCS, etc.) [9,18], Petri Nets [22] and timed automata [1] have been proposed and successfully used, in which interactions can be expressed via synchronized transitions. Regardless of the notations used (process algebra, Petri Nets or timed automata), most of the existing approaches for discrete systems focus only on state based models of separate components in systems.

Exogenous coordination models such as Reo offer a powerful glue-code to express interaction protocols. They provide a proper approach that focuses on the interaction aspects in distributed applications, instead of the behavior models for individual components. In such coordination models, connectors provide the actual protocols used for communication among different components, and components do not need to contain any protocol information. However, in cyber-physical systems, not only software components, but also physical components are coordinated as well. It is critical to express the coordination mechanism in such hybrid contexts. For example,

- What kinds of interaction protocols are needed to describe the communication among physical and computational components?
- What kinds of coordination primitives are needed to describe the interaction protocols among physical and computational components?
- What kinds of logics are needed to specify properties of coordination in cyber-physical systems that are related to both continuous and discrete behavior?

B. Compositional Reasoning of QoS Properties

QoS refers to the non-functional requirements that users may have on the service they receive from a system [24]. In cyber-physical systems, QoS refers to a variety of quality metrics, such as availability, security, reliability, performance (response time, for instance), and so on. Over the past few decades, quantitative and stochastic methods like queueing theory methods [11], interactive Markov Chains [7], quantitative extension of process algebra [8], and stochastic Petri Nets [16] have been proposed to study such metrics and have proved themselves as powerful means to address and solve QoS problems in a range of areas, such as telecommunication networks and software systems. However, current cyber-physical systems always involve large-scale distributed applications that cross intra- and inter-organizational borders frequently. In such environments it is a challenge to achieve better understanding and provision of end-to-end QoS for cyber-physical systems.

New models and tools are needed to provide QoS guarantees for the coordination among heterogeneous components that interact in a complex, multi-scale environment. Some of the challenges include:

- What kind of compositional coordination model will allow us to specify QoS at different layers (computational, communicational and physical) and to reason about the relation of QoS between different layers?
- How to decompose coarse system-level QoS requirements into QoS requirements on basic connectors to utilize existing QoS mechanisms in networks and operating systems?
- How to assess and monitor the end-to-end QoS of cyber-physical systems dynamically and take proper actions when deteriorating QoS is detected?
- How to gain the full control over the cyber-physical environment by adapting the implementation of connectors to maintain QoS when possible?

C. Verification and Validation

The connectors in cyber-physical systems must be highly dependable, adaptable, reconfigurable and certifiable. Such connectors must possess a high trustworthiness at all scales. This creates new challenges for developing highly trustworthy connectors. New theories, algorithms and tools that incorporate existing verification and validation approaches for both hardware and software are needed.

Model checking is an automatic verification technique that has been extensively used in the hardware industry, and has become feasible for verifying many types of software as well [3]. Several quantitative extensions of existing models and logics have been proposed as well as corresponding model checking algorithms for verifying QoS properties. Most of the existing techniques rely on general automata models and existing model checking tools such as UPPAAL [13] and PRISM [12] offer high-level modeling languages with just simple features to specify interactions, like shared variables or simple synchronization. For cyber-physical systems, such encoding would require technical changes of the system specifications resulting in models that rely on non-intuitive representations of the systems that usually do not preserve the structures of the systems and the elegance of exogenous coordination, and thus make compositional verification very difficult.

Current verification and validation methods do not align well with changing goals in CPSs and this creates new challenges for developing highly trustworthy cyber-physical systems. For example:

- How to ensure that the dynamic changes in the running cyber-physical systems are performed correctly?
- What existing V&V techniques can help dealing with the V&V issues of coordination in cyber-physical systems? And how can these techniques be applied in this context?
- What are the differences between V&V done during design and runtime?
- How the new techniques for V&V performed at runtime modify the assumptions of current techniques?

D. Self-Adaptation and Reconfiguration

In general, a cyber-physical system should be able to modify its behavior according to changes in its environment, errors happening at runtime, and changes to its requirement, i.e., the system should be self-adaptive. A self-adaptive system has the potential to improve its performance or its other QoS parameters, by tailoring the configuration of the system at runtime so as to match the varying requirements of the system to the changing pool of resources that may be available to support those requirements. Further, a self-adaptive system may reduce deployment and maintenance costs by adjusting its own configuration, based on the required QoS, the actual rather than predicted load on the system, and the actual rather than predicted behavior of the application in a given environment. As such, a self-adaptive system must continuously monitor changes in its context and react accordingly. Its reaction consists of: (1) assessing if an adaptation is required and if so, (2) which adaptation is the most suitable answer to a detected change, and finally (3) carrying out the adaptation while the system is running. It is critical that in spite of unexpected changes, systems are required to operate correctly.

Therefore, it is an important task to develop a generic and agnostic approach to express, compare, catalogue and reason about adaptation patterns for coordination in CPSs. An adaptation pattern refers to a reusable abstraction of adaptation strategies, which enables rigorous analysis and formal certification essential to the development of highly trustworthy, self-adaptable cyber-physical systems. Technically, such adaptation patterns will be developed as transformations between connector designs embodying specific adaptation strategies. For the development of adaptation patterns for connectors, the following challenges are recognized:

- What kind of adaptation patterns for basic connectors and complex connector deployments are needed for CPSs?
- How to develop a calculus of QoS-aware connector designs for cyber-physical systems with self-adaptability, building on the reconfigurable system architectures?
- How to develop proper theory of adaptation patterns for connectors, with appropriate notions of equivalence and refinement?

IV. CONCLUDING REMARKS

As an emerging phenomenon, cyber-physical systems are expected to play a vital role in the design and development of next-generation systems in different domains with higher reliability, security, and trustworthiness. Coordination is a crucial part in the behavior of such systems, which are both software intensive, and integrated with physical processes.

This leads to many new challenges for the analysis, design and development of new coordination models and languages. In this paper, we discussed a variety of such challenges. Advances in CPS research can be accelerated if we can solve the challenges for coordination that appear in the context of CPS.

ACKNOWLEDGMENT

The work was partially supported by the National Natural Science Foundation of China under grant no. 61202069 and Research Fund for the Doctoral Program of Higher Education of China under grant no. 20120001120103.

REFERENCES

[1] R. Alur and D. L. Dill, “A theory of timed automata”, in Theoretical Computer Science, 126(2), pp. 183-235, 1994.

[2] F. Arbab, “Reo: a channel-based coordination model for component composition”, in Mathematical Structures in Computer Science, 14(3), pp. 329-366, 2004.

[3] C. Baier and J-P. Katoen, “Principles of Model Checking”, MIT, 2008.

[4] S. Bliudze and J. Sifakis, “The Algebra of Connectors – Structuring Interaction in BIP”, in IEEE Transactions on Computers, 57(10), pp. 1315-1330, 2008.

[5] P. Derler, E. A. Lee and A. Sangiovanni Vincentelli, “Modeling Cyber-Physical Systems”, in Proceedings of the IEEE, 100(1), pp. 13-28, 2012.

[6] L. P. J. Groenewegen and E. P. de Vink, “Evolution on-the-fly with Paradigm”, In Proceedings of COORDINATION 2006, LNCS 4038, pp. 97-112, Springer, 2006.

[7] H. Hermanns, “Interactive Markov Chains And the Quest for Quantified Quality”. Springer, 2002.

[8] J. Hillston, “A Compositional Approach to Performance Modelling”. Cambridge University Press, 1996.

[9] C. A. R. Hoare, “Communicating Sequential Processes”. Prentice Hall International, 1985.

[10] S. Kell, “A survey of practical software adaptation techniques”. Journal of Universal Computer Science, 14(13):2110–2157, 2008.

[11] L. Kleinrock, “Queueing Systems”. Wiley, New York. 1976.

[12] M. Kwiatkowska, G. Norman, and D. Parker, “Probabilistic symbolic model checking with PRISM: A hybrid approach”. International Journal on Software Tools for Technology Transfer, 6(2):128–142, 2004.

[13] K. G. Larsen, P. Pettersson, and W. Yi, “UPPAAL: Status & developments”. In Proceedings of CAV’97, volume 1254 of LNCS, pages 456–459. Springer Verlag, June 1997.

[14] E. A. Lee, “CPS Foundations”. In Proceedings of the 47th Design Automation Conference (DAC), pages 737-742, ACM, 2010.

[15] E. A. Lee, “Computing Foundations and Practice for Cyber-Physical Systems: A Preliminary Report”, Technical Report UCB/EECS-2007-72, Department of Electrical Engineering and Computer Sciences, UC Berkeley, 2007.

[16] M. A. Marsan, G. Conte and G. Balbo, “A class of generzlied stochastic Petri nets for the performance analysis of multiprocessor systems”. ACM Transactions on Computer Systems 2(2), 93-122. 1984.

[17] Sun Meng and F. Arbab, “Web Services Choreography and Orchestration in Reo and Constraint Automata”, In Proceedings of SAC'07, ACM, pages 346-353, 2007.

[18] R. Milner, “Communication and Concurrency”, Prentice Hall, 1989.

[19] J. Misra and W. R. Cook, “Computation orchestration: A basis for wide-area computing”, in .Software and System Modeling, 6(1), pp. 83-110, 2007.

[20] M. P. Papazoglou and D. Georgakopoulos, “Service Oriented Computing”, Communications of the ACM, 46(10):25–28, 2003.

[21] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, “Service oriented computing: a research roadmap”, Int. J. Cooperative Inf. Syst., 17(2):223–255, 2008.

[22] W. Reisig, “Petri Nets: An Introduction”, volume 4 of Monographs in Theoretical Computer Science. An EATCS Series. Springer, 1985.

[23] P. Wegner, “Coordination as constrained interaction (extended abstract)”, in Proceedings of COORDINATION 1996, LNCS 1061, pp. 28-33, Springer, 1996.

[24] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, “QoS-Aware Middleware for Web Services Composition”, IEEE Transactions on Software Engineering, 30(5):311–327, 2004.