Microprocessors and Microsystems 39 (2015) 1296–1304

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

An integrated framework of formal methods for interaction behaviors among industrial equipments ☆

Pan Deng [a,b,*], Gang Ren [a,b], Wei Yuan [a,b], Feng Chen [a,b], Qingsong Hua [c]

[a] Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[b] University of Chinese Academy of Sciences, Beijing 100190, China
[c] Qingdao University, Qingdao 266071, China

ARTICLE INFO

Article history:
Received 20 June 2015
Revised 22 July 2015
Accepted 25 July 2015
Available online 28 August 2015

Keywords:
Formal methods
Formal verification
Model checking
Interaction behaviors among industrial equipments

ABSTRACT

With the rapid advancement of Internet of Things, interaction behaviors among their industrial equipments have been complex dramatically whereas they have been becoming a kind of safety–critical systems and high requirements for safety have been urgent unprecedentedly. Therefore, it has been a great challenge for practicing engineers to ensure temporal correctness and reliability of interaction behaviors among industrial equipments. Nowadays, π-calculus, a process algebra and NuSMV, a symbolic model checker, have been widely applied to address this posed challenge respectively. However, they are always used separately. Because different formal methods focus on different aspects of systems, only one single method is still difficult to cope very well with the posed challenge. Therefore in this paper, an integrated framework of formal methods, which combines π-calculus with NuSMV, is constructed. π-Calculus can definitely specify equipment interaction, and NuSMV can automate verification process. Especially counterexamples fed back by NuSMV can help practicing engineers to trace temporal violations. Furthermore, a cooperative traffic lights control strategy is illuminated to show how the framework works.

1. Introduction

With the rapid advancement of Internet of Things, interaction behaviors among their industrial equipments have drawn much attention in recent years [1–3]. Because there exist a variety of industrial equipments and also these equipments always need directly or indirectly communicate and cooperate with each other to reach information sensing and automatic control goals, their complexity has been increasing dramatically.

On one hand, interaction behaviors among industrial equipments have become complex dramatically. on the other hand, they have become a kind of safety–critical systems [4,5] and high requirements for safety are urgent unprecedentedly. A subtle temporal flaw may cause catastrophic loss of money, time and even human lives. Therefore it has been a great challenge for practicing engineers to ensure correctness and reliability of temporal properties of interaction behaviors among industrial equipments.

It is well known that traditional techniques such as testing or simulation do not cope very well with the posed challenges. For example, testing is not suitable due to high monetary costs and time overheads. Likewise, simulation is not competent for verification of temporal properties because it doesn't consider all the possible system states. Therefore, formal methods [6–10], which rest on mathematical logic, draw much attention. Generally speaking, formal methods comprise formal specification and formal verification. Formal specification can definitely specify interaction behaviors among industrial equipments and formal verification can theoretically account for every possible system state. In contrast to testing and simulation, they are more complete and reliable for analysis and verification of interaction behaviors among industrial equipments.

Nowadays, there have been many different formal methods to specify and verify interaction behaviors among industrial equipments [11–17]. Among these methods, there are two formal methods used widely. One is π-calculus [7], which is a progress algebra for specifying concurrent systems. The other one is NuSMV [10], which is a symbolic model checker for verifying the model of a finite state system against its desired properties expressed in

☆ Project supported by National Nature Science Foundation of China (No. 61100066).

* Corresponding author at: Institute of Software, Chinese Academy of Sciences, Box 8718, 4 Zhongguancun South 4 Street, Beijing 100190, China.
E-mail addresses: dengpan@iscas.ac.cn (P. Deng), rengang2013@iscas.ac.cn (G. Ren), futureyuan628@gmail.com (W. Yuan), chenfeng@iscas.ac.cn (F. Chen), 8988596@qq.com (Q. Hua).

http://dx.doi.org/10.1016/j.micpro.2015.07.015
0141-9331/

temporal logic. Generally speaking, the two formal methods facilitate analysis and verification of interaction behaviors among industrial equipments and ensure correctness and reliability to some extent.

However, the two methods are usually used separately. Because different formal methods focus on different aspects of systems, only one single method is still difficult to cope very well with the posed challenge. For example, π-calculus is only good at system specification, not at system verification, and NuSMV is just the opposite. Therefore, in this paper an integrated framework of formal methods is constructed, which combines π-calculus with NuSMV and takes their respective advantages to cooperate with each other. π-Calculus can definitely specify interaction behaviors among industrial equipments, and NuSMV can automate verification process.

The remainders of this paper will proceed as follows: Section 2 constructs an integrated framework of formal methods for equipment interaction. Section 3 first examines a conceptual model of interaction behaviors among industrial equipments, and then based on this conceptual model, defines a collection of rules of formal specification using π-calculus. Section 4 translates π-calculus expressions into a NuSMV program. In Section 5 a case study is presented to show how the formal framework works. At last, Section 6 draws some conclusions on this paper.

2. The proposed integrated framework

The proposed formal framework consists of two layers. The first is specification layer, which adopts π-calculus as specification tool. The second is verification layer, which accommodates NuSMV as model checker.

π-Calculus is a process algebra for describing and analyzing a concurrent mobile system. It has been widely applied to modeling and reasoning of various concurrent and distributed systems [14,18,19]. Due to its excellent ability for modeling concurrent systems, it is adopted as modeling tool in this paper.

Another advantage of using π-calculus is that its presentations improve framework flexibility. The expressions of π-calculus can be analyzed by either the existed tools for π-calculus, (for example, HAL [9] or MWB [20]) or model checker (for example, NuSMV) by translating them into a corresponding program.

However, π-calculus is only a language for modeling system, not an automated tool for system verification. Therefore, NuSMV is incorporated into this framework as a model checker. NuSMV can automate to model checking temporal properties and it addresses state explosion problem in an efficient way by supporting both binary-decision diagram-based and propositional satisfiability-based model checking.

It is worth mentioning that although NuSMV can also be used in modeling equipment interaction directly, it is just a programming language rather than a mathematical notation. It is suitable only for model checking system not modeling system.

The two different formal methods are incorporated into one framework and collaborate with each other efficiently. As Fig. 1 shows, there are 6 steps in this framework in total. Each step is introduced as follows:

In Step 1, interaction behaviors among industrial equipments are specified in π-calculus. The method of specification will be presented in Section 3.

In Step 2, π-calculus expressions are translated into a NuSMV program. The rules of translation will be presented in Section 4 in detail.

In Step 3, the temporal properties of interaction behaviors among industrial equipments are defined in the context of specific applications.

In Step 4, the temporal properties of interaction behaviors among industrial equipments are specified in Computation Tree Logic (CTL) [6], which is a branching temporal logic and extends propositional logic by incorporating path quantifiers and temporal operators.

In Step 5, NuSMV program and CTL formulas are together input to NuSMV to verify whether or not system complies with the desired temporal properties. The procedures of verification will be introduced in Section 5 in detail.

In Step 6, counterexample, which includes a set of traces which can help practicing engineers to trace temporal violation, will be fed back to trace temporal violations by NuSMV if the system violates the desired temporal properties.

3. Formal specification using π-calculus

This section focuses on the step 2 of the framework. A conceptual model of equipment interaction is first proposed. Then the syntax and semantics of π-calculus are summarized. Afterwards based on this model, a set of specification rules are defined. Finally the related work on formal specification is discussed.

3.1. A conceptual model of equipment interaction

In order to help practicing engineers to model equipment interaction, a conceptual model is presented, as shown in Fig. 2, In this model, equipments are divided into two kinds: terminal equipment such as Device1, Device3 or Device4 and intermediate equipment such as Device2. Interaction behaviors are divided into three kinds: request operation, forwarding operation and response operation. Terminal equipment can launch both request operation and response operation and intermediate equipment can launch forwarding operation. In addition there exists message channels between equipments such as a, b and c. Each equipment has a status channel with the same name of equipment such as $device1$ or $device2$.

3.2. The specification rules in π-calculus

The syntax and semantics of π-calculus are based on countable sets of names representing communication channels and data and ranged over by lowercase letters $x, y, z, \ldots$. Processes are denoted with uppercase letters $A, B, C, \ldots$. The intuitive meanings of constructs and prefixes are listed as follows:

(1) Input Action $a(x).[x = msg]P$: means that a message is received along the channel of a, if the message is equal to msg, the process will continue as P.
(2) Output Action $\overline{a}\langle x\rangle.P$: means that the message x is sent along the channel of a and thereafter the process continues as P.
(3) Silent Action τ: means that τ is an inner action which can't be observed outsides.
(4) Choice Composition $P + Q$: means that process can enact either P or Q.
(5) Parallel Composition $P|Q$: means that the concurrent processes P and Q are executing in parallel.

According to the syntax and semantics of π-calculus, in the following lists, a set of rules are defined to specify interaction behaviors among industrial equipments.

Rule 1. A terminal equipment is specified as a process in π-calculus; an intermediate equipment is specified as a parallel composition comprising one synchronization sub-process and some forwarding sub-processes.