Available online at www.sciencedirect.com

ScienceDirect

IERI Procedia 10 (2014) 160 – 168

Procedia IERI

www.elsevier.com/locate/procedia

2014 International Conference on Future Information Engineering

Formalization of Signaling System by Process Calculus

Yasuaki Ibayashi, Shin-ya Nishizaki*

Department of Computer Science, Tokyo Institute of Technology
2-12-1-W8-69, Ookayama,Meguro-ku, Tokyo 152-8552, Japan

Abstract

Software systems play an important role in social infrastructures, and in the public and private sectors, for example in core banking, enterprise resource planning systems, electronic medical record systems, telephone exchange systems, etc. In such fundamental systems, we desire not only correctness but also robustness. Formal methods, such as software verification, have focused on correctness rather than robustness. The Spice calculus, which was proposed by Nishizaki et al., is a calculus process which enables us to formalize and analyze the resistance of communication protocols against Denial-of-Service attacks. The key idea of the analysis is analyzing costs of initiators and responders in communication protocols. In this paper, we extend the target area of systems to be analyzed by the Spice calculus and describe a new analysis methodology for distributed systems. We demonstrate analysis of a telephone exchange system, as an example of our methodology. In this paper, we extend the target area of systems to be analyzed by the Spice calculus and show a new analysis methodology for distributed systems. We demonstrate analysis of a telephone exchange system, as an example of our methodology. We set up ISUP of Signaling System No. 7 as a target and formalized a simple telephone network based on it in the framework of the Spice calculus. We then analyzed its connection establishing process and computation cost consumed the connection establishment.

Keywords: Process Calculus, Communication Protocol, Telephone Exchange System, Cost Analysis

* Corresponding author. Tel.: +81-3-5734-2772; fax: +81-3-5734-2772.
E-mail address: nisizaki@cs.titech.ac.jp.

doi:10.1016/j.ieri.2014.09.071

1. Introduction

Software systems play an important role in social infrastructures, and in the public and private sectors, for example in core banking, enterprise resource planning systems, electronic medical record systems, telephone exchange systems, etc. In such fundamental systems, we desire not only correctness with respect to specification requirements but also robustness to cope with unintended situations. Formal methods, such as software verification, have focused on correctness rather than robustness.

Many failures have occurred in such IT systems. For example, the core banking system of Mizuho Financial Group, which is the largest banking company in Japan, failed on March 14th 2011 and did not recover until the 23rd, owing to an overload caused by numerous payments of contributions for the Tohoku Earthquake[7]. Such lengthy failures of computer systems can be mitigated by improvements in system robustness.

Formal methods, such as software verification, have focused on correctness rather than robustness. Spice calculus [9], which was proposed by Nishizaki et al., is a calculus process which enables us to formalize and analyze the resistance of communication protocols against Denial-of-Service attacks, based on Milner's pi-calculus [6]. The key idea of the analysis is analyzing costs of initiators and responders in communication protocols. Vulnerability to Denial-of-Service attacks is caused by lack of balance between an initiator and a responder. Cost analysis is important not only for resistance against Denial-of-Service attacks but also for improving a system's robustness to withstanding various unintended situations. In this paper, we therefore apply Spice calculus to analysis of robustness of distributed systems. We conduct a case study using a telephone exchange system, as an example.

2. Spice calculus

We introduce Spice calculus [9] in this section. In the calculus, there are two categories of expression: one is a set of *terms* and the other is a set of *processes*. A term denotes data and a process denotes a series of computational operations which can be executed concurrently. *Terms* of spice-calculus are defined inductively as follows:

$M, N ::=$		term
	n	name
$\mid$	x	variable
$\mid$	i	integer
$\mid$	$(M_1, \ldots, M_n)$	pair

Processes of Spice calculus are defined inductively as follows:

$P, Q, R ::=$		
	$\mathrm{out}\ M\ \langle N \rangle; P$	message sending
$\mid$	$\mathrm{inp}\ M\ (x); P$	message receiving
$\mid$	$(P \mid Q)$	parallel composition
$\mid$	stop	stopping
$\mid$	end	termination
$\mid$	$\mathrm{store}\ x = M; P$	memory allocation
$\mid$	$\mathrm{free}\ x; P$	memory deallocation
$\mid$	$\mathrm{match}\ M\ \mathrm{is}\ N\ \mathsf{err}\{R\}; P$	matching
$\mid$	$\mathrm{split}\ [x_1, \ldots, x_n]\ \mathrm{is}\ M\ \mathsf{err}\{R\}; P$	pair decomposition
$\mid$	$(P + Q)$	selection