A Formalism for
Remotely Interacting Processes
— Working Paper —

Ichiro Satoh* and Mario Tokoro**
Department of Computer Science, Keio University
3-14-1, Hiyoshi, Kohoku-ku, Yokohama, 223, Japan

Abstract. An earlier paper [21] introduced process calculi with notions of time suited to express concurrent and distributed real-time computations. However, they cannot sufficiently model asynchronous communication in distributed systems. In this paper we present a process calculus with the ability to express asynchronous message passing with location-dependent transmission delay. It allows us to describe temporal and behavioral properties of asynchronous interactions among remotely located processes. Based on the process calculus, we also develop a "speed-sensitive" order relation for distributed real-time processes. It is formulated based on the notion of bisimulation and can distinguish between behaviorally equivalent processes performing at different speeds.

1 Introduction

Distributed systems consist of multiple processors cooperating by sending messages over communication networks. These communication networks often have transmission delay: a physical and logical function of geographic distance, communication bandwidth, and communication protocol overhead. The delay makes it to be difficult to develop correct distributed systems. This is because it affects the arrival timings of messages and thus often leads these systems to failure. Also, a sender process in distributed systems must be blocked for at least the length of communication delay in synchronous communication settings. For the sake of efficiency, communication among remotely located processes is often realized in an asynchronous form. However, asynchronous communication often creates unpleasant non-determinism. Therefore, in order to construct correct programs for distributed real-time systems, we must take communication delay and its influences into consideration. The objective of this paper is to propose a framework to describe communication delay and to analyze the behavioral and temporal properties of asynchronous interactions among distributed real-time processes.

* Email: satoh@mt.cs.keio.ac.jp, Partially supported by JSPS Fellowships for Japanese Junior Scientists.
** Email: mario@mt.cs.keio.ac.jp, Also with Sony Computer Science Laboratory Inc. 3-14-13 Higashi-Gotanda, Shinagawa-ku, Tokyo, 141, Japan.
The framework is formulated by extending a timed process calculus [16, 21] with the ability to express communication delay and asynchronous message passing. We also develop a performance analysis for distributed real-time systems by means of a "speed-sensitive" order relation on processes. It is formulated on the basis of the bisimulation technique[11, 15] and can relate two distributed processes with respect to their speeds. If two processes are behaviorally equivalent and if the first process can execute faster than the second one, it asserts that the first process is faster than the second one.

There have indeed been several process calculi for distributed systems. They are extensions of existing process calculi with some features of distributed computing: causality between events (e.g., [6]), location information (e.g., [4]), port passing mechanism (e.g., [8, 11]), and local time (e.g., [17]), but none addresses the issue of communication delay. Also, there have been several temporal extensions of process calculi for synchronous communication, for example [7, 13, 14, 16, 20, 22]. Among these, Moller and Tofts in [13] studied a preorder relating timed processes with respect to speed. However, the order relation is inherently dependent on synchronous communication and thus cannot compare the performances of asynchronously communicating processes. On the other hand, there are many process calculi with the ability to express asynchronous communication (e.g., [1, 3, 5, 8, 9, 10]). However, there are only a few that deal with both delay and asynchrony in communication. Among them, in [2] Baeten and Bergstra proposed a process calculus with the ability to express asynchronous communication with delay and failure, based on a timed extended calculus of ACP [1]. It represents asynchronous message transmission as the creation of a process corresponding to the message like the methods developed in [8, 10] and like our approach it can represent communication delay by suspending the created process for the amount of the delay. However, it provides just a language to describe systems with asynchronous communication with delay and failure and it does not provide any kind of analysis for these systems.

The organization of this paper is as follows: In the next section we briefly present our basic ideas concerning the process calculus and then define its syntax and semantics. In Section 3 we define a timed pre-bisimulation that can relate two processes according to their speeds. In Section 4 we show some examples. The final section contains some concluding remarks.

2 Timed Calculus

Our framework is an extension of an existing process calculus, CCS [11], with the ability to express communication delay, delayed processing, and non-blocking message sending. Its syntax and semantics are essentially those of CCS, except for these extensions. Before formally defining the framework, we summarize our basic assumptions.