The Journal of Logic and Algebraic Programming 80 (2011) 377–391

Contents lists available at ScienceDirect

The Journal of Logic and Algebraic Programming

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

Timed Mobility in process algebra and Petri nets

Gabriel Ciobanu [a], Maciej Koutny [b,*]

[a] Faculty of Computer Science, A.I. Cuza University of Iasi, 700483 Iasi, Romania
[b] School of Computing Science, Newcastle University, Newcastle upon Tyne NE1 7RU, UK

ARTICLE INFO

Article history:
Available online 12 June 2011

Keywords:
Mobility
Local clocks
Process algebra
High level Petri nets
Syntax driven translation
Behavioural equivalence

ABSTRACT

We present a process algebra called TiMo in which timeouts of interactions and adaptable migrations in a distributed environment with explicit locations can be specified. Timing constraints allow to control the communication between co-located mobile processes, and a migration action with variable destination supports flexible movement from one location to another. The model of time is based on local clocks rather than a global clock.
We provide a structural translation of TiMo into behaviourally equivalent high level timed Petri nets. As a result, we obtain a formal net semantics for timed interaction and migration which is both structural and allows one to deal directly with concurrency and causality.

1. Introduction

The ever increasing complexity of mobile applications requires their effective analysis and verification. Our aim here is to explore formal modelling of mobile distributed systems including also time-related aspects of process migration.

In this paper, we first introduce the TiMo (Timed Mobility) model which is a process algebra for mobile systems where – in addition to process mobility and interaction – it is possible to add timers to the basic actions. We provide the syntax and operational semantics of TiMo which is a time semantics where each location runs according to its own local clock which is invisible to processes outside this location. Processes are equipped with input and output capabilities which are active up to a predefined time deadline. If such a capability is not taken, an alternative continuation for the process is followed. Another timing constraint allows one to specify the latest time for moving a process from one location to another. The timeout of such a migration action corresponds to the network time limit for that action, similar to TTL in TCP/IP network protocol.

The time model defined for TiMo generalises the one considered in the theory of structured timed Petri nets [20] because it supports local clocks. In the second part of the paper, we outline a structural translation of the finite process algebra terms into behaviourally equivalent finite high level timed Petri nets with local clocks, similar to that presented in [12]. Such a dual development yields a formal semantics for explicit mobility and time which is structural and, at the same time, allows one to deal directly with concurrency and causality which can be captured in the Petri net domain. The Petri net representation should also be useful for automatically verifying behavioural properties using suitable model-checking techniques and tools.

The paper generalises and extends ideas first described in [9], but here each location has its local clock which determines the timing of actions executed at that location. The paper is self-contained, although it would be an advantage for the reader to be familiar with some basic concepts of process algebras [19], high level Petri nets [8,17], and timed Petri nets [21].

* Corresponding author.
E-mail addresses: gabriel@iit.tuiasi.ro (G. Ciobanu), maciej.koutny@ncl.ac.u (M. Koutny).

1567-8326/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jlap.2011.05.002

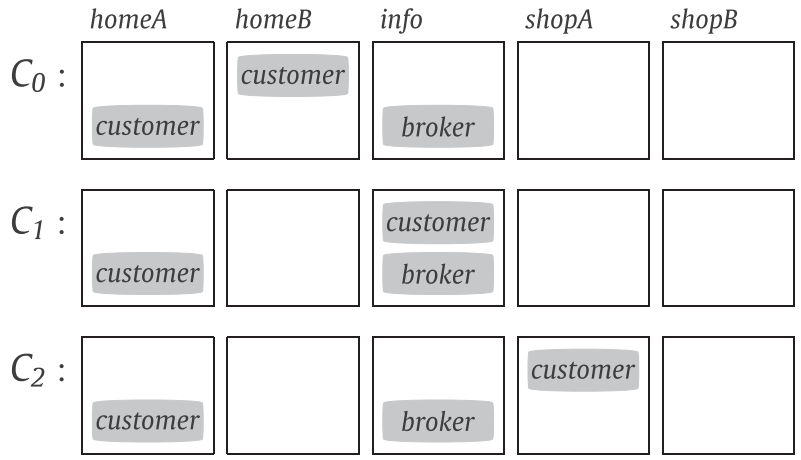

Fig. 1. Three configurations in an evolution of the running example (time progress is not represented).

The paper is structured in the following way. We first describe the syntax and semantics of TIMO. After that we introduce the net algebra used in the translation from TIMO expressions to Petri nets, and then describe the translation itself. We also explain the nature of behavioural equivalence of the resulting Petri net model and the original expression.

1.1. Running example

To introduce the basic concepts of TIMO, we use a *simple e-shops (SES)* running example illustrated in Fig. 1. In this scenario, we have two customer processes initially residing in their respective home locations *homeA* and *homeB*, and looking for (the address of) an e-shop where the same desirable e-item can be purchased. To find this out, each customer moves to the location *info* in order to acquire the relevant address (this move takes up to *5* time units). After waiting for *2* time units at location *info* without getting the desired address, the e-item loses its importance and the customer is no longer interested in acquiring it. The location *info* contains a broker who knows all about the e-shops stocking the desired e-item. For up to *5* time units the right e-shop is that at the location *shopA*, and after that for up to *7* time units at location *shopB* (these changes of availability are cyclical and happen also if a location is communicated to a customer). It is important to point out that any interaction between processes can only happen within the same location, and so it is necessary for a customer to move to the broker location in order to get the desirable address. The timers can define a coordination in time of the customers, and take care of the relative time of interaction of the processes residing at the same location.

Fig. 1 portrays three possible configurations in an evolution of the running example in which there are two customers. The active customer is initially residing in location *homeB* (configuration C_0), and then moving to location *info* to acquire the address of an e-shop (configuration C_1). After receiving such an address from the broker, the customer moves to the corresponding location *shopA* (configuration C_2).

2. TIMO: a process algebra for Timed Mobility

We start by giving the syntax and semantics of TIMO which uses timing constraints allowing, for example, to specify what is the longest time it takes a mobile process to move to another location. In TIMO, waiting for a communication on a channel or a movement to a new location can be constrained. If an action does not happen before a predefined deadline, the waiting process switches its operation to an alternate mode. This approach leads to a method of sharing the channels over time. A timer (such as $\Delta 7$) of an output action $a^{\Delta 7}$ **!** makes it available for communication only for the period of *7* time units. We use timers for both input and output actions. The reason for having the latter stems from the fact that in a distributed system there are both multiple clients and multiple servers, and so clients may decide to switch from one server to another depending on the waiting time.

2.1. Syntax

We assume suitable data types together with associated operations, including a set *Loc* of *locations* and a set *Chan* of *communication channels*. We also use a set *Id* of process identifiers, and each $id \in Id$ has arity m_{id}. In what follows, we use $\mathbf{x}$ to denote a finite tuple of elements $(x_1, \ldots, x_k)$ whenever it does not lead to a confusion.