

Chapter 30

Process Algebras for Locality

Rolando Blanco

University of Waterloo, Canada

Paulo Alencar

University of Waterloo, Canada

ABSTRACT

Several process algebras have been proposed in the literature to support the notion of locality. In this chapter the authors compare some of these process algebras, namely Distributed π , Ambient Calculus, Join Calculus, M-calculus, and Kell Calculus. Specifically, they look at the locality abstractions supported by the algebras and how these abstractions interact with the communication mechanisms in the algebras, higher-order constructs, and process passivation. The authors also look at the model checking support in the algebras that allow the verification of locality-based behaviour.

INTRODUCTION

Process algebras are formalisms for the specification and description of distributed, and in general, concurrent systems. Process algebras provide means to specify parallel composition (two processes operating at the same time), alternative composition (a choice between two processes) and sequential composition (a process operating after another process terminates) (Baeten et al. 2007).

Traditionally in process algebras, processes communicate with each other via *channels*. In a communication on a particular channel one of the

processes writes to the channel and another process reads from the channel. Channels are a kind of a more general construct, called *names*. A name is an identifier, and it is only useful for comparing for identity with other names (Needham, 1989). In the area of process algebras, channel reads are referred to as *abstractions*, while channel writes are referred to as *concretions*.

Process algebras were conceived at the end of the 1970s (Milner, 1980, Antony and Hoare, 1978, Bergstra and Klop, 1985). Based on one of these early algebras, the Calculus of Communicating Systems (Milner, 1980), Robin Milner,

Joachim Parrow, and David Walker developed the π -calculus (Milner et al., 1989). An extensively studied formalism, one of the main contributions of the π -calculus is the ability to establish and terminate communication channels dynamically. Process expressions in the π -calculus have the following syntax:

$$P ::= \mathbf{0} \mid \bar{a}(\tilde{e}).P \mid a(\tilde{e}).P \mid P+Q \mid P|Q \mid \mathbf{new} a P \mid P!$$

$\mathbf{0}$ represents the null process. $\bar{a}(\tilde{e}).P$ represents a process writing names \tilde{e} on channel a . In general, \tilde{e} is used to indicate zero, or one or more names. After the names are read, the process continues as P . $a(\tilde{e}).P$ represents a process waiting for input on channel a . When the input is received, the process continues as P . $P+Q$, called *summation* of processes, represents a process that behaves either like P or Q . $P|Q$ represents the parallel execution of processes P and Q . $\mathbf{new} a P$, called *restriction*, is used to specify a private name a used in process P . $P!$, called *replication*, represents an infinite number of copies of P , running in parallel. Some variations of the π -calculus support recursive processes in-lieu of replication.

Because only names can be transmitted in channel communications, π -calculus is a first-order process algebra. Process algebras where names and processes can be transmitted as part of a channel communication are said to be *higher-order algebras*.

The asynchronous π -calculus was one of the first variations proposed of the π -calculus (Honda and Tokoro, 1991, Boudol, 1992). In the asynchronous π -calculus, outputs on channels cannot be followed by a process, and the summation of processes is eliminated from the algebra.

Some process algebras provide abstractions for modelling localities. A locality provides the ability to group, name, and transmit processes. Some process algebras allow locations to be transmitted as part of channel communications. The ability to associate a process with a location is of interest

when formalizing and verifying locality-based processing. For example, one may be interested in formalizing that processing must occur at a certain location, or that a process may migrate from one location to another under certain conditions.

In this chapter we present algebras that provide some explicit notion of locality, that can be used to represent physical locations in a distributed system. The algebras considered are Distributed π , Ambient Calculus, Join Calculus, M-calculus, and Kell Calculus, which includes kell-m. These algebras trace their lineage back to the π -calculus (cf. Figure 1).

DISTRIBUTED π

The Distributed π calculus, $D\pi$, is a first-order calculus. There are both, synchronous and asynchronous versions of $D\pi$ (Hennessy and Riely, 1998, Hennessy, 2007). $D\pi$ was the first variation of the π -calculus to introduce an explicit notion of locality. Associated to each location is one or more processes (called π -calculus) running in parallel. An expression $K[P] \mid K[R]$, specifies that processes R and P are executing both in location K . Locations in the $D\pi$ are not themselves processes, therefore locations cannot be located within other locations.

When two processes at different locations wish to communicate, one of the processes must migrate to the location of the other process. For this purpose, $D\pi$ provides a *move* operator $T::P$ that allows the migration of process P to location T . For example, in $K[T::P] \mid T[Q]$ process P migrates to location T resulting in $T[P] \mid T[Q]$. When a process migrates to another location it loses the communication channels of the previous location.

Locations in $D\pi$ cannot be passivated. *Passivation* is the term used in process algebras to indicate the ability to suspend the process executing within a location. In some algebras, a passivation may be followed by the elimination

11 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/process-algebras-locality/66486

Related Content

Geographically Distributed Cloud-Based Collaborative Application

Bogdan Solomon, Dan Ionescu, Cristian Gadeaand Marin Litoiu (2013). *Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments* (pp. 248-274).

www.irma-international.org/chapter/geographically-distributed-cloud-based-collaborative/72220

Open-Source Software Issues

Sofiane Sahraoui (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 33-38).

www.irma-international.org/chapter/open-source-software-issues/29376

Single Image 3D Beard Face Reconstruction Approaches

Hafiz Muhammad Umair Munirand Waqar Shahid Qureshi (2022). *International Journal of Cyber-Physical Systems* (pp. 1-17).

www.irma-international.org/article/single-image-3d-beard-face-reconstruction-approaches/314572

Recommendation System for Hairstyle Based on Face Recognition Using AI and Machine Learning

Yogesh M. Kambleand Raj B. Kulkarni (2024). *International Journal of Software Innovation* (pp. 1-10).

www.irma-international.org/article/recommendation-system-for-hairstyle-based-on-face-recognition-using-ai-and-machine-learning/309960

Information Theoretic XSS Attack Detection in Web Applications

Hossain Shahriar, Sarah North, Wei-Chuen Chenand Edward Mawangi (2018). *Application Development and Design: Concepts, Methodologies, Tools, and Applications* (pp. 972-987).

www.irma-international.org/chapter/information-theoretic-xss-attack-detection-in-web-applications/188243