Towards a spatial stochastic process algebra

Vashti Galpin*
LFCS, School of Informatics, University of Edinburgh
Email: Vashti.Galpin@ed.ac.uk

Abstract

This paper introduces spatial notions to the stochastic process algebra PEPA. This is motivated by both computer networks and epidemiology where location of actions or processes with respect to other parts of a system may affect the time taken by an event. A very general stochastic process algebra with locations is defined. Locations are introduced to both actions and processes, and are provided with a weighted graph structure.

1 Introduction

Stochastic process algebras have been developed as a tool for modelling computer systems in terms of performance [1, 2, 3]. Models described using stochastic process algebra can be analysed by the use of continuous time Markov chains (CTMCs). Spatial aspects and locations are relevant to the quantitative modelling of systems because systems can be physically distributed in space, and this distribution can affect the length of time required an activity to occur. To date, there has been some work on spatial aspects, notably PEPA nets [4], a Petri net influenced model where tokens are PEPA components that can move places, and StoKlaim [5] which locates processes at nodes in a network. Spatial models have also been developed within a biological framework [6, 7].

This paper considers spatial concepts within PEPA by introducing of named locations to process terms. The aim is to be as general as possible, unlike most previous approaches that are tailored for specific applications. This general approach facilitates the development of a spatial stochastic process algebra with a theory and with analyses which can then be made more concrete for a specific purpose. The general results will then carry through to the more concrete process algebra as will the general analyses. The general process algebra is a long term goal, and this paper is a starting point towards that goal.

Locations will be associated with processes and also with actions. This will give a rich variety of terms and behaviour in the most abstract case of the operational semantics, and the operational semantics can be instantiated with specific functions that will capture more specific roles of locations. Locations will be related to each other by a structure over locations which will provide

*The author is supported by the EPSRC SIGNAL Project, Grant EP/E031439/1.

additional information about what communication is possible between locations, and how likely it is to occur or how much it may be retarded or accelerated.

Using locations is useful from a modelling point of view because it can separate of concerns. In networks there is the actual physical infrastructure with certain parameters of operation together with the logical links that connect computers [8]. In epidemiology of plants, both the temporal and spatial behaviours are of interest [9]. There is the distance between plants as well as the probability of infection once the infectious agent reaches the plant to be considered.

Another reason for having a separate location structure is that it may be possible to only discover the state space once, and then to modify the CTMC by the information given in the location structure after which the steady state probabilities can be solved for. The process algebra will also be designed so that annotation of the process algebra terms is kept to a minimum. This will ensure that introducing locations does not lead to additional state space explosion.

2 Additions of locations to PEPA

Let $\mathcal{L}$ be a set of locations, and let $\mathcal{P}_{\mathcal{L}} = 2^{\mathcal{L}}$ be the powerset of $\mathcal{L}$. There is a weighted hypergraph structure on locations $G = (\mathcal{L}, E, w)$ with $E \subseteq \mathcal{P}_{\mathcal{L}}$[1] and $w : E \to \mathbb{R}$ a weighting function.

There is no particular interpretation imposed on locations or the functions defined over them. These locations may or may not have structure. For example, they could be nodes in a directed graph representing computers in a network or locations in n-dimensional space. The graph edges could represent simple network connectivity, or a tree structure representing nesting of locations. Additionally, the function w could be obtained from their structure. For example, if locations are points in two-dimensional space then in the case of an edge between a pair of locations, w could be obtained from a distance function that maps the two points to a scalar value.

The syntax for components is a two level syntax of sequential components and model components based on PEPA

$$S ::= (\alpha@L, r).S \mid S + S \mid C_s@L \qquad P ::= P \underset{M}{\bowtie} P \mid P/M \mid C.$$

with $L \in \mathcal{P}_{\mathcal{L}}$. The class of models with locations $\mathcal{L}$ is denoted $\mathcal{C}_{\mathcal{L}}$. Let α denote $\alpha@\emptyset$ and $\alpha@l$ denote $\alpha@\{l\}$, and similarly S denotes $S@\emptyset$ and $S@l$ denotes $S@\{l\}$. This syntax permits the association of a set of locations with an action and with a sequential process. This permits unconstrained processes such as $A@L_1 \overset{def}{=} (\alpha@L_2, r).A@L_3 + (\beta@L_3).B@L_4$ with $L_1, \ldots, L_4$ all different.

The operational semantics are given in Table 1. The labels of the transition system come from $\mathcal{A} \times \mathcal{P}_{\mathcal{L}} \times \mathbb{R}^+$. These are parameterised by three functions. The first, *apref* takes a sequential component which possibly contains location information, and returns a location set. Here the aim is to resolve the location set associated with the action by considering the location set in the prefix term as well as other information that may be included in the component term. The second, *async* maps $\mathcal{P}_{\mathcal{L}} \times \mathcal{P}_{\mathcal{L}}$ to $\mathcal{P}_{\mathcal{L}}$, and determines the location set of a synchronisation. The last, *rsync* takes two rates, two model components, two

[1]In some cases, especially in the case of sets of size 2 which may represent edges in a directed graph, ordering will be important and will be assumed when necessary.

Prefix

$$\frac{}{(\alpha@L, r).S \xrightarrow{(\alpha@L', r)} S} \quad L' = \mathit{apref}((\alpha@L, r).S)$$

Constant

$$\frac{P \xrightarrow{(\alpha@L, r)} P'}{A \xrightarrow{(\alpha@L, r)} P'} \quad A \overset{\mathit{def}}{=} P$$

Choice

$$\frac{P_1 \xrightarrow{(\alpha@L, r)} P_1'}{P_1 + P_2 \xrightarrow{(\alpha@L, r)} P_1'} \qquad \frac{P_2 \xrightarrow{(\alpha@L, r)} P_2'}{P_1 + P_2 \xrightarrow{(\alpha@L, r)} P_2'}$$

Hiding

$$\frac{P \xrightarrow{(\alpha@L, r)} P'}{P/M \xrightarrow{(\alpha@L, r)} P'/M} \quad \alpha \notin M \qquad \frac{P \xrightarrow{(\alpha@L, r)} P'}{P/M \xrightarrow{(\tau@L, r)} P'/M} \quad \alpha \in M$$

Cooperation

$$\frac{P_1 \xrightarrow{(\alpha@L, r)} P_1'}{P_1 \underset{M}{\bowtie} P_2 \xrightarrow{(\alpha@L, r)} P_1' \underset{M}{\bowtie} P_2} \quad \alpha \notin M \qquad \frac{P_2 \xrightarrow{(\alpha@L, r)} P_2'}{P_1 \underset{M}{\bowtie} P_2 \xrightarrow{(\alpha@L, r)} P_1 \underset{M}{\bowtie} P_2'} \quad \alpha \notin M$$

$$\frac{P_1 \xrightarrow{(\alpha@L_1, r_1)} P_1' \quad P_2 \xrightarrow{(\alpha@L_2, r_2)} P_2'}{P_1 \underset{M}{\bowtie} P_2 \xrightarrow{(\alpha@L, R)} P_1' \underset{M}{\bowtie} P_2'} \quad \alpha \in M \quad \begin{array}{l} L = \mathit{async}(L_1, L_2) \\ R = \mathit{rsync}(r_1, r_2, P_1, P_2, L_1, L_2) \end{array}$$

Table 1: Operational semantics of PEPA with locations

location sets and returns a rate for the synchronisation of two components. All three functions can return $\perp$ which means that a transition is not possible.

Note that in the original PEPA definition the rate of a synchronisation is determined by the rates in the premises and the components involved (since the apparent rate of the action must be calculated with respect to each component) hence all that is being added to this is the location sets. The standard PEPA rate can written as

$$\mathit{RPEPA}(r_1, r_2, P_1, P_2, L_1, L_2) = \frac{r_1}{r_\alpha(P_1)} \frac{r_2}{r_\alpha(P_2)} \min(r_\alpha(P_1), r_\alpha(P_2)).$$

Since it may be necessary to work with the locations that are found in components, location identifying functions are defined. The set of locations of a component P is $\mathit{loc}(P) = \mathit{aloc}(P) \cup \mathit{ploc}(P)$ where $\mathit{ploc}(P)$, the process location function, extracts a location set from terms of the form $C_s@L$ in the obvious manner and $\mathit{aloc}(P)$, the action location function, extracts a location set from terms of the form $(\alpha@L, r).S$ in the obvious manner.

3 A more concrete process algebra

The process algebra in Section 2 is too abstract since there are many different functions that can be used to instantiate it. In this section, a more concrete process algebra that is suitable for modelling networks is developed.

PEPA model	P	P_G	
$\downarrow$			
LMTS	$\mathcal{M}$	$\mathcal{M}_G$	$\mathcal{M} \circledcirc G = \mathcal{M}_G$
$\downarrow$			
CTMC	$\mathbf{Q}$	$\mathbf{Q}_G$	$\mathbf{Q} \boxdot G = \mathbf{Q}_G$
$\downarrow$			
steady state	$\mathbf{\Pi}$	$\mathbf{\Pi}_G$	$\mathbf{\Pi} \boxed{\boxdot} G = \mathbf{\Pi}_G$

Table 2: Structure of results

For the level of abstraction at which networks will be modelled here, communication between processes in the network is viewed as directional and only involves two processes at a time. Communication involves passing a packet from one process to another. It is assumed that processes are static (not mobile) and hence they have one fixed location (a singleton set) that does not change over time. More than one process can have the same location. Since processes have a single fixed location, the restriction to pairwise synchronisation implies that communication can take place between at most two locations. This means that the hypergraph over locations is just a directed graph and hence pairs are assumed instead of sets and location sets are drawn from $\mathcal{L} \cup (\mathcal{L} \times \mathcal{L})$. The required functions are as follows.

$$\mathit{apref}(S) = \begin{cases} l & \text{if } \mathit{ploc}(S) = \{l\} \\ \bot & \text{otherwise} \end{cases}$$

$$\mathit{async}(L_1, L_2) = \begin{cases} (l_1, l_2) & \text{if } L_1 = \{l_1\}, L_2 = \{l_2\} \\ \bot & \text{otherwise} \end{cases}$$

$$\begin{aligned} &\mathit{rsync}(r_1, r_2, P_1, P_2, L_1, L_2) \\ &\qquad = \begin{cases} \dfrac{r_1}{r_\alpha(P_1)} \dfrac{r_2}{r_\alpha(P_2)} \min(r_\alpha(P_1), r_\alpha(P_2)) \odot w((l_1, l_2)) & \\ \qquad\qquad\qquad \text{if } L_1 = \{l_1\}, L_2 = \{l_2\}, (l_1, l_2) \in E & \\ \bot \qquad\qquad\quad\; \text{otherwise} & \end{cases} \end{aligned}$$

Note that the rate obtained by *rsync* for synchronisation involves the operator $\odot$ and the weight of the edge involved. This operator can be instantiated with an appropriate function such as multiplication when modelling a specific scenario. The edge (l, l) can be viewed as always present, and either not affect the rate between two colocated processes or alternatively the weighting on this edge could represent faster communication between colocated processes.

The operational semantics with these functions give rise to a labelled multi-transition system with activities containing locations as labels and components as states. As in PEPA this is translated to a CTMC. If a steady state is required for this CTMC, it is necessary to ensure that the weightings do not remove edges that would prevent this.

4 Results and analysis

The type of theoretical results that will be beneficial to facilitating analysis are given in Table 2. The first column represents the stages in the analysis of a PEPA model using CTMCs. The second column contains the results without the use of locations at each stage of analysis and the third column contains the results with locations. The goal is to achieve theorems, as given the last column, that allow us to express the results with locations as a combination of the location graph together with the results for the model without location. Hence it is necessary to discover the appropriate function for the operators $\circledcirc$, $\boxdot$ and $\boxcircle$ if they exist. The benefits of this are that the state space needs only to be discovered once and then it will be possible to experiment with different location graphs in an efficient manner. This is ongoing research.

References

[1] J. Hillston, *A compositional approach to performance modelling.* Cambridge University Press, 1996.

[2] M. Bernardo and R. Gorrieri, “A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time,” *Theoretical Computer Science*, vol. 202, pp. 1–54, 1998.

[3] U. Herzog and V. Mertsiotakis, “Stochastic process algebras applied to failure modelling,” in *Proceedings of the Second Workshop on Process Algebras and Performance Modelling*, (Regensberg), July 1994.

[4] S. Gilmore, J. Hillston, L. Kloul, and M. Ribaudo, “PEPA nets: a structured performance modelling formalism,” *Performance Evaluation*, vol. 54, pp. 79–104, 2003.

[5] R. de Nicola, J.-P. Katoen, D. Latella, M. Loreti, and M. Massink, “KLAIM and its stochastic semantics,” tech. rep., Dipartimento di Sistemi e Informatica, Universitá di Firenze, 2006.

[6] A. Regev, E. Panina, W. Silverman, L. Cardelli, and E. Shapiro, “BioAmbients: an abstraction for biological compartments,” *Theoretical Computer Science*, vol. 325, pp. 141–167, 2004.

[7] C. Priami and P. Quaglia, “Beta binders for biological interactions,” in *CMSB 2004,* Paris, France, LNCS 3082 (V. Danos and V. Schächter, eds.), pp. 20–33, Springer, 2004.

[8] S. Low, F. Paganini, and J. Doyle, “Internet congestion control,” *IEEE Control Systems Magazine*, vol. 22, pp. 28–43, Feb 2002.

[9] A. van Maanen and X. X.-M., “Modelling plant disease epidemics,” *European Journal of Plant Pathology*, vol. 109, pp. 669–682, 2003.