Semi-Markov PEPA: a contradiction in terms?

Jeremy T. Bradley

Department of Computing, Imperial College London
180 Queen's Gate, London SW7 2BZ, United Kingdom
Email: jb@doc.ic.ac.uk

Abstract

In this report, we consider a semi-Markov version of PEPA. In particular, we discuss the seeming conflict between semi-Markov modelling, which appears to be sequential and lacking in concurrency, and process algebraic modelling, which whole *raison d'être* is to model concurrency.

1 Introduction

PEPA [1] is a very Markovian process algebra that is easy to use and has a wide user community [2]. It is perhaps unique among stochastic process algebras for this. It would nevertheless be useful for the community if the type of distribution that PEPA could use were extended beyond the confines of the exponential distribution.

While some SPAs have sought refuge in ever more intricate syntax, PEPA has maintained a very small set of base operators. Thus, any attempt to modify or extend PEPA should hold to the original design quality, that of syntactic simplicity. A further requirement is that there should be an easy recovery of the existing body of work—it would be all too easy to produce a PEPA-with-extra-distributions which could not take advantage of the many results that Markovian PEPA has given rise to.

With these in mind, we propose a semi-Markov variant of PEPA which has an SMP [3] as its underlying stochastic model but permits purely Markovian cliques of operation.

2 Semi-Markov Models: What are they good for?

In a way, semi-Markov processes are a low-level model searching for a high-level formalism. Markov models mapped particularly well onto concurrency-modelling paradigms, such as Petri nets and process algebras; largely due to the memoryless nature of the exponential distribution, which allows for the construction of an *expansion law* for concurrent Markov processes.

Semi-Markov models have no such obvious mapping. SMPs running concurrently are in fact GSMPs and they have no easy expansion law[1] that allows you to write down the residual distribution for an interrupted action.

In order to define large SMPs [5], you do have to abstract yourself away from the transition matrix level. To this end, we invented SM-SPN—or semi-Markov Stochastic Petri Nets [6], an instantiation of ESPN [7]. In an SMP, from a single state to one of its many possible successor states, a probabilistic choice is made and an arbitrary random delay is observed. In the same way, in a SM-SPN a probabilistic choice is made between concurrently enabled transitions (based on pre-assigned weights) and an arbitrary random firing delay is observed. There is no concurrent running of transition clocks—only a single transition clock runs at a time.

Here lies the heart of the issue: if only a single clock or distribution can be selected to run at a time in a semi-Markov model, how can it be any use for modelling concurrent processes? We suggest two uses:

- the first is: you have a concurrent design but a single-threaded architecture on which to run it, e.g. a Java environment on any single processor machine. The machine must timeslice between the different threads of the program: it must select a thread to run and then complete a certain amount of work on that thread before getting the chance to select a new thread to execute. This is ideally suited to the mutual exclusion style of semi-Markov processes; the selecting of a new thread being the probabilistic selection of a particular distribution.
- the second recognises the fact that the modeller will still want to have actual Markovian concurrency at their disposal. Markov processes are a strict subset of SMPs and thus it is possible to let the modeller have true concurrency while there are no general distributions enabled. As soon as a general distribution becomes enabled, the paradigm switches into the probabilistic selection of actions, as described above. This is particularly useful for describing failure/recovery modes in systems, where normal concurrent operation is suspended and a single recovery process is repairing the system.

3 Extending PEPA

We would propose modifying standard PEPA to give a syntax for a semi-Markov PEPA:

$$P \ ::= \ (a, D).P \;\big|\; P + P \;\big|\; P \underset{S}{\bowtie} P \;\big|\; P/L \;\big|\; A \tag{1}$$

[1]Process algebra semantics do exist for GSMPs in the form of Strulo's SPADES [4] and Bravetti's iGSMPA, but they are not interleaving.

where:

$$D \quad ::= \quad \lambda \;\big|\; \omega : L(s) \tag{2}$$

where λ can incorporate the passive $\top$ action.

Here D is an arbitrary delay parameter which can either take on the normal exponential rate λ or a weight, ω, and Laplace transform of a general distribution, $L(s)$.

We now prioritise the general distributions by saying that if, at any stage, a generally-distributed transition is enabled then all Markovian transitions are disabled by definition. If more than one generally-distributed transition is enabled then the weight ω is used to form a probabilistic choice between them.

Finally, if only Markovian transitions are enabled then standard PEPA semantic rules apply.

A first attempt at presenting a formal semantics for a semi-Markov PEPA is presented at the end of the report. Fig. 1 attempts to preserve the standard Markovian PEPA behaviour, while Fig. 2 adds those rules necessary to cope with the semi-Markov extensions to the formalism.

One notable design decision is that we have forbidden anything other than passive synchronisation on generally distributed transitions. It would just be too complicated to attempt to design an active synchronisation style for these transitions.

4 Conclusion

We have presented a semi-Markov version of PEPA, so that greater distribution expressiveness can be used in PEPA. We have identified two uses for a semi-Markov environment for PEPA, and attempted to keep the traditional Markovian PEPA at the core of the formalism. I would welcome any comments, improvements or help with formal semantics!

Acknowledgements

I would like to thank Jane Hillston for her comments and suggestions made on the concepts behind semi-Markov PEPA during my previous visit to Edinburgh.

References

[1] J. Hillston, *A Compositional Approach to Performance Modelling*, vol. 12 of *Distinguished Dissertations in Computer Science*. Cambridge University Press, 1996. ISBN 0 521 57189 8.

[2] "PEPA resource site." `http://www.dcs.ed.ac.uk/stg/pepa/`.

[3] R. Pyke, "Markov renewal processes: Definitions and preliminary properties," *Annals of Mathematical Statistics*, vol. 32, pp. 1231–1242, December 1961.

[4] P. G. Harrison and B. Strulo, "Process algebra for discrete event simulation," *Qualitative Methods in Parallel Systems*, 1995.

[5] J. T. Bradley, N. J. Dingle, W. J. Knottenbelt, and H. J. Wilson, "Hypergraph-based parallel computation of passage time densities in large semi-markov models," in *NSMC'03, Proceedings of the 4th International Workshop on Numerical Solutions of Markov Chains*, 2003. (to appear).

[6] J. T. Bradley, N. J. Dingle, W. J. Knottenbelt, and P. G. Harrison, "Performance queries on semi-Markov stochastic Petri nets with an extended Continuous Stochastic Logic," in *PNPM'03, Proceedings of Petri Nets and Performance Models*, 2003. (to appear).

[7] J. B. Dugan, K. S. Trivedi, R. M. Geist, and V. F. Nicola, "Extended stochastic Petri nets: Applications and analysis," in *Performance'84, Proceedings of 10th Intl. Symp. on Models of Computer System Performance* (E. Gelenbe, ed.), pp. 507–519, Elsevier, Paris, December 1984.

Prefix

$$\frac{}{(a,\lambda).P \xrightarrow{(a,D)} P}$$

Competitive Choice

$$\frac{P \xrightarrow{(a,\lambda)} P'}{P + Q \xrightarrow{(a,\lambda)} P'} \quad \text{if } Q \not\xrightarrow{(b,\omega:L(s))}$$

$$\frac{Q \xrightarrow{(a,\mu)} Q'}{P + Q \xrightarrow{(a,\mu)} Q'} \quad \text{if } P \not\xrightarrow{(b,\omega:L(s))}$$

Cooperation

$$\frac{P \xrightarrow{(a,\lambda)} P'}{P \underset{S}{\bowtie} Q \xrightarrow{(a,\lambda)} P' \underset{S}{\bowtie} Q} \quad \text{if } a \notin S \text{ and } Q \not\xrightarrow{(b,\omega:L(s))}$$

$$\frac{Q \xrightarrow{(a,\mu)} Q'}{P \underset{S}{\bowtie} Q \xrightarrow{(a,\mu)} P \underset{S}{\bowtie} Q'} \quad \text{if } a \notin S \text{ and } P \not\xrightarrow{(b,\omega:L(s))}$$

$$\frac{P \xrightarrow{(a,\lambda)} P' \quad Q \xrightarrow{(a,\mu)} Q'}{P \underset{S}{\bowtie} Q \xrightarrow{(a,R)} P' \underset{S}{\bowtie} Q'} \quad \text{if } a \in S$$

$$\text{where } R = \frac{\lambda}{r_a(P)} \frac{\mu}{r_a(Q)} \min(r_a(P), r_a(Q))$$

Hiding

$$\frac{P \xrightarrow{(a,D)} P'}{P\backslash L \xrightarrow{(a,D)} P'\backslash L} \quad \text{if } a \notin L \qquad \frac{P \xrightarrow{(a,D)} P'}{P\backslash L \xrightarrow{(\tau,D)} P'\backslash L} \quad \text{if } a \in L$$

Constant

$$\frac{P \xrightarrow{(a,D)} P'}{A \xrightarrow{(a,D)} P'} \quad A \stackrel{\mathrm{def}}{=} P$$

Fig. 1. Operational semantics for SM-PEPA: normal Markovian semantics are preserved if only Markovian actions are enabled.

Competitive Choice

$$\frac{P \xrightarrow{(a,\omega:L(s))} P'}{P + Q \xrightarrow{(a,\omega:L(s))} P'} \qquad \frac{Q \xrightarrow{(a,\omega:L(s))} Q'}{P + Q \xrightarrow{(a,\omega:L(s))} Q'}$$

Cooperation

$$\frac{P \xrightarrow{(a,\omega:L(s))} P'}{P \underset{S}{\bowtie} Q \xrightarrow{(a,\omega:L(s))} P' \underset{S}{\bowtie} Q} \quad \text{if } a \notin S$$

$$\frac{Q \xrightarrow{(a,\omega:L(s))} Q'}{P \underset{S}{\bowtie} Q \xrightarrow{(a,\omega:L(s))} P \underset{S}{\bowtie} Q'} \quad \text{if } a \notin S$$

$$\frac{P \xrightarrow{(a,\omega:L(s))} P' \quad Q \xrightarrow{(a,\top)} Q'}{P \underset{S}{\bowtie} Q \xrightarrow{(a,\omega:L(s))} P' \underset{S}{\bowtie} Q'} \quad \text{if } a \in S$$

$$\frac{P \xrightarrow{(a,\top)} P' \quad Q \xrightarrow{(a,\omega:L(s))} Q'}{P \underset{S}{\bowtie} Q \xrightarrow{(a,\omega:L(s))} P' \underset{S}{\bowtie} Q'} \quad \text{if } a \in S$$

Fig. 2. Operational semantics for SM-PEPA: semi-Markov extensions