A comparison of the ODE semantics of PEPA with timed continuous Petri nets

Vashti Galpin*

Abstract

The behaviour of PEPA models with large numbers of components can be approximated in a continuous fashion by a set of coupled ordinary differential equations (ODEs). Timed continuous Petri nets can be used to approximate behaviour via ODEs in a scenario of many servers. These two formalisms are compared, a translation from a PEPA model to a Petri net is given, and it is shown that the ODEs obtained in each case are the same. Different server semantics for Petri nets are also considered.

1 Introduction

The continuous approximation technique for PEPA [1] provides coupled ordinary differential equations (ODEs) which describe changes in behaviour over time [2]. This paper compares this approach with timed continuous Petri nets [3, 4] and the associated ODEs. In both cases, the continuous approximation deals with the problem of state space explosion. For PEPA, the state space is the labelled multi-transition system obtained from the structured operational semantics and in (discrete) bounded stochastic Petri nets, it is the reachability graph which describes the possible markings of a Petri net. For both PEPA and stochastic Petri nets, if the rates of activities/transitions are exponentially distributed, then the transition system/reachability graph is the basis for a continuous time Markov chain (CTMC) that represents the behaviour with respect to time of the system [5]. The continuous approximations avoid the calculation of the state spaces, and the resultant ODEs are amenable to analysis [2].

The comparison of these two continuous approximations will deepen our understanding of them and whether they are interchangeable or not. If they are similar then techniques for the one may be applied to the other.

Another aspect of timed-based Petri nets is whether infinite server semantics or finite server semantics (multiple server or single server) are used [3]. In infinite server semantics as many firings of a transition as are enabled can take place. In finite server semantics, there is an upper bound on how many simultaneous firings can occur. As part of the comparison, the type of server semantics used by PEPA will be identified. In what follows, infinite server semantics will be assumed as it is the more general case [3], and finite server semantics will be discussed later.

*Vashti Galpin, LFCS, University of Edinburgh `Email: vashti@gaon.net` (This research was carried out in part while the author was on sabbatical leave from the School of Computer Science, University of the Witwatersrand, South Africa.)

In the next section, continuous Petri nets are introduced, after which a section follows on the ODE semantics of PEPA. The fourth section of the paper provides the comparison, followed by a discussion of server semantics and the final section covers conclusions and further work.

2 Continuous Petri nets

The definitions in this section are based on those in [3, 5]. Let $\mathbb{R}^+ = \{x \mid x \geqslant 0\}$.

Definition 1 *A* (timed) continuous Petri net *is a pair* $\langle N, m_0 \rangle$ *where* $N = (P, T, Pre, Post)$ *with* P *the set of places,* T *the set of transitions with* $P \cap T = \emptyset$*, and* $Pre : P \times T \to \mathbb{R}^+$ *and* $Post : P \times T \to \mathbb{R}^+$ *are the pre and post incidence matrices respectively which give the arc weights between places and transitions. A net is* ordinary *if the maximum value over all arc weights is 1. The token flow matrix is* $C = Post - Pre$*. Additionally*

$$
\begin{aligned}
{}^\bullet t &= \{p \mid Pre(p,t) = 1\} & \qquad t^\bullet &= \{p \mid Post(p,t) = 1\} \\
{}^\bullet p &= \{t \mid Post(p,t) = 1\} & \qquad p^\bullet &= \{t \mid Pre(p,t) = 1\}.
\end{aligned}
$$

A marking associates values with places at a specific point in time, and is defined as a function $m : P \times \mathbb{R}^+ \to \mathbb{R}^+$*. The initial marking is* $m_0 = m(\cdot, 0)$*. The function* $\lambda : T \to \mathbb{R}^+$ *associates with each transition a firing rate.*

The main feature of continuous Petri nets is that markings are no longer restricted to integer values only. The focus here is on ordinary nets. Hence, whenever $p \in {}^\bullet t$, then $Pre(p,t) = 1$ and vice versa. This simplifies a number of definitions. However, for completeness, they are written in full and simplified if necessary. Infinite server semantics are assumed. Note that the definition of a timed continuous Petri net given above is very similar to that of a stochastic Petri net except for the real values for markings.

Definition 2 *A transition* t *is enabled at time* τ *if for all* $p \in {}^\bullet t$*,* $m(p, \tau) > 0$ *and* t *has enabling degree*

$$enab(t, \tau) = \min_{p \in {}^\bullet t} \left\{ \frac{m(p,\tau)}{Pre(p,t)} \right\}$$

An enabled transition can fire an amount of α *with* $0 < \alpha \leq enab(t, \tau)$*. After this firing the new marking will be* $m(\cdot, \tau') = m(\cdot, \tau) + \alpha C(\cdot, t)$*.*

Mahulea *et al* [3] state that the continuous approximation under infinite server semantics is appropriate when there are many clients and many servers. Then it is possible to take a deterministic continuous approximation of the discrete case by assuming that the firing delays can be approximated by their mean values [6].

The fundamental equation which defines how markings change over time is defined as follows for continuous Petri nets [3].

$$m(\cdot, \tau + \delta\tau) = m(\cdot, \tau) + C(\cdot, t) \cdot \sigma(\tau)$$

Differentiating this equation with respect to time results in

$$\frac{dm(\cdot,\tau)}{d\tau} = C \cdot f(\cdot,\tau)$$

where the flow $f(\cdot,\tau) = \sigma'(\tau)$ is the derivative of the firing sequence and is defined for transition t as

$$f(t,\tau) = \lambda(t) \cdot \min_{p \in {}^\bullet t}\left\{\frac{m(p,\tau)}{Pre(p,t)}\right\} = \lambda(t) \cdot \min_{p \in {}^\bullet t}\left\{m(p,\tau)\right\}$$

where $\lambda(t)$ is the firing rate [3]. Since the net is ordinary the equation can be simplified. The change in the marking of a single place p can be expressed as follows. Let n be the number of transitions. Then

$$\begin{aligned}
\frac{dm(p,\tau)}{d\tau} &= \sum_{j=1}^{n} C(p,t_j).f(t_j,\tau) \\
&= \sum_{j=1}^{n} Post(p,t_j).f(t_j,\tau) - \sum_{j=1}^{n} Pre(p,t_j).f(t_j,\tau)) \\
&= \sum_{t \in {}^\bullet p} f(t,\tau) - \sum_{t \in p^\bullet} f(t,\tau) \\
&= \sum_{t \in {}^\bullet p} \lambda(t) \cdot \min_{p' \in {}^\bullet t}\{m(p',\tau)\} - \sum_{t \in p^\bullet} \lambda(t) \cdot \min_{p' \in {}^\bullet t}\{m(p',\tau)\}.
\end{aligned}$$

3 PEPA and ODE semantics

The definitions in this section are based on those in [2]. Consider the standard syntax for PEPA

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

where C names a component, C_s names a sequential component, α is an action, r is a rate, and together they form an activity, and L is a set of actions. Also consider the standard operational semantics of PEPA [1]. This syntax limits PEPA components so that the resulting CTMC is ergodic [2]. In this paper, the following subset of PEPA will be considered.

- Identical components cannot synchronise on actions, ie. L must be empty when identical components are put in parallel.
- When an activity is shared, the rates must be identical, and no passive rates are allowed.
- All shared actions of non-identical components must synchronise.
- Hiding is not used.

The numerical vector form can be used to obtain the vector state space which is smaller than the labelled multi-transition system when there are many identical components [2].

Definition 3 *For an arbitrary PEPA model M with n component types C_i for $i = 1, \dots n$ each with N_i distinct derivatives, the numerical vector form of M, $V(M)$ is a vector with $N = \sum_{i=1}^{n} N_i$ entries. Each entry v_{i_j} records how many instances of the jth local derivative of component type C_i are present in the current state.*

The vector state space gives an aggregated model and can be used in the usual way to obtain a CTMC. The numerical vector form can also be used as the basis of a continuous approximation in the case of large numbers of identical components using the following differential equation for $N(C_{i_j}, \tau) = v_{i_j}$ the number of derivatives of type C_{i_j} at time τ [2].

$$\frac{dN(C_{i_j},\tau)}{d\tau} = \sum_{(\alpha,r)\in En(C_{i_j})} r \times \min_{C\in Ex(\alpha,r)} \{N(C,\tau)\} - \sum_{(\alpha,r)\in Ex(C_{i_j})} r \times \min_{C\in Ex(\alpha,r)} \{N(C,\tau)\}$$

where

$$Ex(D) = \{(\alpha, r) \mid D \xrightarrow{(\alpha,r)}\} \qquad Ex(\alpha, r) = \{D \mid D \xrightarrow{(\alpha,r)}\}$$

$$En(D) = \{(\alpha, r) \mid \exists D', D' \xrightarrow{(\alpha,r)} D\}.$$

Hence $Ex(D)$ captures those activities that decrease the number of D's (exit activities), $En(D)$ captures those activities that increase the number of D's (entry activities) and $Ex(\alpha, r)$ describes those derivatives that can perform (α, r) activities. Therefore the change in the number of a derivative D is expressed in terms of the number of decreases and increases at the given rate where the changes are bounded by the minimum number of derivatives available to perform the activity.

Extracting the specific ODEs for a given model can be automated based on an activity graph and an activity matrix [2].

Definition 4 *An activity graph is a bipartite graph (N,A). The nodes are partitioned into N_t, the activities, and N_p, the derivatives. $A \subseteq (N_t \times N_p) \cup (N_p \times N_t)$, where $a = (n_t, n_p) \in A$ if n_t is an entry activity of derivative n_p, and $a = (n_p, n_t) \in A$ if n_t is an exit activity of n_p.*

Definition 5 *For a model with N_t activities and N_p derivatives, the activity matrix M_a is an $N_p \times N_t$ matrix and entries are defined as follows.*

$$M_a(p_i, t_j) = \begin{cases} +1 & \textit{if } t_j \in En(p_i) \\ -1 & \textit{if } t_j \in Ex(p_i) \\ 0 & \textit{otherwise.} \end{cases}$$

4 Expressing a PEPA model as a continuous Petri net

Working with the restricted subset of PEPA given above, it is possible to convert this to a timed continuous Petri net. First, construct the activity graph. This is can be viewed as a net of the form $(N_p, N_t, M_{pre}, M_{post})$ where

$$M_{pre}(p_i, t_j) = \begin{cases} +1 & \text{if } t_j \in Ex(p_i) \\ 0 & \text{otherwise} \end{cases} \qquad M_{post}(p_i, t_j) = \begin{cases} +1 & \text{if } t_j \in En(p_i) \\ 0 & \text{otherwise.} \end{cases}$$

The activity matrix M_a is then $M_{post} - M_{pre}$ which is just the token flow matrix. The initial marking is determined by the numbers of each component;

$m(p, 0) = N(p, 0)$. Additionally, $\lambda(t) = r$ where $t = (\alpha, r) \in N_t$. Therefore we have constructed a continuous Petri net (which we could also view as a bounded stochastic Petri net) from a PEPA model, and the activity graph and the Petri net are isomorphic.

Moreover, for each distinct type of derivative in the model, we have a place in the Petri net and the tokens in that place represent the number of copies of the derivative in the model. The vector state space that can be obtained from the PEPA model is isomorphic to the reachability graph of the continuous Petri net if we were to view it as a discrete Petri net, since the behaviour of both the PEPA model and the net is the same. If an activity reduces by one the count of a particular derivative and increases by one the count of another derivative in the PEPA model, then in the Petri net, the transition that is that activity will fire and remove one token from the place that is that particular derivative and add a token to the place that is the other derivative. Similarly the token changes associated with a firing of a transition can be expressed as changes in the number of derivatives in the PEPA model. Hence, $m(p, \tau) = N(p, \tau)$.

However, the focus here is continuous approximations. In Section 2, we gave the ODEs based on markings and in Section 3, the ODEs for the continuous approximation for a PEPA model. We are interested in whether these are the same considering the construction of a continuous Petri net given above. Consider the ODE from Section 3 for the place p.

$$\begin{aligned}
\frac{dN(p,\tau)}{dt} &= \sum_{t \in En(p)} r \times \min_{p' \in Ex(t)} \{N(p',\tau)\} - \sum_{t \in Ex(p)} r \times \min_{p' \in Ex(t)} \{N(p',\tau)\} \\
&= \sum_{t \in {}^\bullet p} \lambda(t) \times \min_{p' \in {}^\bullet t} \{m(p',\tau)\} - \sum_{t \in p^\bullet} \lambda(t) \times \min_{p' \in {}^\bullet t} \{m(p',\tau)\} \\
&= \frac{dm(p,\tau)}{d\tau}.
\end{aligned}$$

The equation holds since $t \in En(p)$ is equivalent to $t \in {}^\bullet p$, $t \in Ex(p)$ is equivalent to $t \in p^\bullet$ and $p' \in Ex(t)$ is equivalent to $p' \in {}^\bullet t$.

Hence the ODEs generated by a PEPA model are the same as those for the associated continuous Petri net under infinite server semantics.

5 Server semantics

As mentioned previously, there are two types of server semantics, finite and infinite. In the above, we have shown that the ODEs are the same for infinite server semantics. Infinite server semantics is the more general case and, in the discrete case, finite server semantics can be obtained by assuming infinite server semantics for all transitions and then modifying the net. This requires the addition of a place with k tokens with arcs to and from the transition which is to be bounded [3]. A similar approach could be taken in PEPA. When a particular activity is to be limited, a new component can be added of the form $B \stackrel{\mathit{def}}{=} (\alpha, r).B$ with k copies where (α, r) is the activity to be limited.

However, Silva and Recalde [6] argue that in the continuous case, infinite and finite server semantics are about two different relaxations of the model, many servers and many clients for infinite, and few servers and many clients for finite server semantics and are therefore different. In finite server semantics, a transition has a maximal firing speed at which it can perform representing k times the speed of a single server [3] and a different expression is required for the flow of a transition. This is given by Mahulea *et al* [3] as

$$f(t,\tau) = \begin{cases} \lambda(t) \text{ if } \forall p \in {}^{\bullet}t, m(p) > 0 \\ \min\Big\{\lambda(t), \min\limits_{p \in {}^{\bullet}t \wedge m(p)=0} \Big\{ \sum\limits_{t' \in {}^{\bullet}p} \frac{f(t',\tau) Post(p,t')}{Pre(p,t)} \Big\} \Big\} \end{cases}$$

and somewhat differently by Alla and David for their constant speed continuous Petri nets [4]. It is not immediate as to whether the expressions are the same. This is an area for further work.

6 Conclusions and further work

In this paper, we have shown how to construct a ordinary timed continuous Petri net from a PEPA model (using a subset of the language). Moreover we have shown that when approximated continuously, the behaviour of both can be characterised by the same ODEs. Furthermore we have shown that this continuous approximation using PEPA has infinite server semantics, which means that as many activities can happen simultaneously as are able to happen.

Previous research has provided PEPA with stochastic Petri net semantics [7] working at the syntactic level of a PEPA model. This approach differs from the approach taken here where the activity graph is considered directly.

A question raised by this work is whether it is possible to take a ordinary timed continuous Petri net and construct a PEPA model from it, and whether the ODEs would be the same. Previous work by Hillston *et al* [5] demonstrates how a bounded stochastic Petri net can be transformed to a PEPA model. This involves adding complementary places, and each place and its complement can be viewed as a state machine and hence can be modelled as PEPA sequential components. A similar approach could be applied here, but it seems likely that the the PEPA model will be larger than the Petri net, and the similarities in terms of ODEs will not be immediate.

Both the PEPA approximation and the continuous Petri net approach are suitable for when there are large numbers of identical components. An interesting question is to how robust this approach is when component numbers decrease, and this is an area of ongoing research.

References

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

[2] J. Hillston, “Fluid flow approximation of PEPA models,” in *Second International Conference on the Quantitative Evaluation of Systems (QEST 2005),* 19-22 September 2005, Torino, Italy, pp. 33–43, IEEE Computer Society, 2005.

[3] C. Mahulea, L. Recalde, and M. Silva, “On performance monotonicity and basic servers semantics of continuous Petri nets,” in *WODES ’06: Eighth*

International Workshop on Discrete Event Systems, Ann Arbor, USA, July 2006.

[4] H. Alla and R. David, "Continuous and hybrid Petri nets," *Journal of Circuits, Systems, and Computers*, vol. 8, no. 1, pp. 159–188, 1998.

[5] J. Hillston, L. Recalde, M. Ribaudo, and M. Silva, "A comparison of the expressiveness of SPA and bounded SPN models," in *PNPM '01: Proceedings of the 9th international Workshop on Petri Nets and Performance Models (PNPM'01),* Washington, DC, USA, p. 197, IEEE Computer Society, 2001.

[6] M. Silva and L. Recalde, "Continuization of timed Petri nets: From performance evaluation to observation and control," in *ICATPN 2005* (G. Ciardo and P. Darondeau, eds.), vol. 3536 of *Lecture Notes in Computer Science*, pp. 26–47, Springer, 2005.

[7] M. Ribaudo, "Stochastic Petri net semantics for stochastic process algebras," in *Proceedings of the 6th International Workshop on Petri Nets and Performance Models,* Durham, North Carolina, 1995.