CONTINUOUS APPROXIMATION OF PEPA MODELS AND PETRI NETS

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

KEYWORDS
PEPA, Petri nets, continuous approximation, state space explosion, server semantics

ABSTRACT

Modelling very large systems that consist of many similar components can lead to state space explosion. Continuous approximation can avoid this. In the stochastic process algebra PEPA, models with large numbers of identical components can be approximated in a continuous fashion by a set of coupled ordinary differential equations (ODEs). Similarly, timed continuous Petri nets can be used to approximate behaviour via ODEs where there are many servers. These two approaches are compared and infinite and finite server semantics are considered.

INTRODUCTION

When modelling large systems such as one providing Web services to many clients, state space explosion imposes restrictions on the size of system that can be analysed. An approximation approach can mitigate the state space explosion problem by making it unnecessary to construct the state space. Ordinary differential equations (ODEs) extracted from the stochastic process algebra model of the Web services system were analysed in less than a tenth of a second of compute time even though there were 3^N states for N clients (Gilmore and Tribastone 2006).

The continuous approximation technique for the stochastic process algebra PEPA (Hillston 1996) provides coupled ODEs which model changes in behaviour over time (Hillston 2005). This paper compares this approach with timed continuous Petri nets (Mahulea et al. 2006, David and Alla 2005) 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 (Ajmone Marsan et al. 1995), 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 of the system over time (Hillston et al. 2001). The continuous approximations avoid the derivation of the state space, and the models become amenable to analysis (Hillston 2005).

In timed-based Petri nets, nets have either infinite server semantics or finite server semantics (multiple or single) (Mahulea et al. 2006). In infinite server semantics as many simultaneous firings of a transition as are enabled can take place. In finite server semantics, there is an finite bound on this number. These represent two different approaches to coordinating the interactions of components. The two semantics will be compared.

In this paper, a translation from PEPA models to continuous Petri nets and *vice versa* will be presented and it will be shown that the ODEs for each approach are the same. Additionally, it will be shown that the continuous semantics for PEPA are infinite server semantics. These results allow the use of techniques and theory for one approach to be applied to the other, and offer a choice between two formalisms to represent systems, either graphical or textual.

This research is novel. PEPA and stochastic Petri nets have been compared (Hillston et al. 2001, Ribaudo 1995) but the continuous approximations have not. Furthermore no investigation has been conducted as to whether the continuous approximation for PEPA has finite or infinite server semantics. These are important to consider as because they deepen our understanding of the two modelling formalisms.

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

CONTINUOUS PETRI NETS

This section presents existing definitions (Mahulea et al. 2006, Hillston et al. 2001). Let $\mathbb{R}^+ = \{x \mid x \geqslant 0\}$.

Definition 1 *A* (timed) continuous Petri net (CPN) *is a pair* $\langle N, m_0 \rangle$ *where* $N = (P, T, Pre, Post, \lambda)$ *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 all arc weights have value 0 or 1. The token flow matrix is* $C = Post - Pre$*. Additionally for* $t \in T$, ${}^\bullet t = \{p \mid Pre(p,t) > 0\}$, $t^\bullet = \{p \mid Post(p,t) > 0\}$*, and for* $p \in P$, ${}^\bullet p = \{t \mid Post(p,t) > 0\}$, $p^\bullet = \{t \mid Pre(p,t) > 0\}$*. The function* $\lambda : T \to \mathbb{R}^+$ *associates with each transition a firing rate. 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* $m_0 = m(\cdot, 0)$.

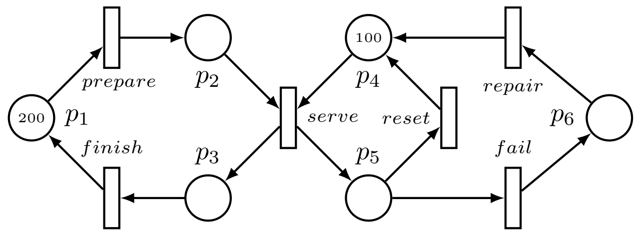

Figure 1: A CPN with Initial Marking

CPNs are drawn in the standard manner using circles for places and rectangles for transitions as illustrated in Figure 1 which describes a system with 200 clients and 100 unreliable servers that fail and then are repaired. The clients prepare work, obtain a service from a server and then finish the work, before starting on the next job. The initial marking is $(200, 0, 0, 100, 0, 0)$. The transition firing rates are $\lambda(prepare) = p$, $\lambda(serve) = s$, $\lambda(finish) = f$, $\lambda(reset) = r$, $\lambda(fail) = a$ and $\lambda(repair) = e$.

The distinguishing feature of CPNs is that markings are not restricted to integer values. In this paper ordinary nets are used. Hence, whenever $p \in {}^\bullet t$ then $Pre(p,t) = 1$ and whenever $p \in t^\bullet$ then $Post(p,t) = 1$. Results can be generalised since a net with arc weights greater than one can be converted to an ordinary Petri net (David and Alla 2005). Infinite server semantics are assumed as they are the more general case (Mahulea et al. 2006).

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. (Mahulea et al. 2006) note that the continuous approximation under infinite server semantics is appropriate for many clients and many servers. A deterministic continuous approximation of the discrete case can be done by assuming that the firing delays can be approximated by their mean values (Silva and Recalde 2005). This applies to mono-T-semiflow reducible nets which includes the class of equal conflict nets (Mahulea et al. 2006).

The fundamental equation which defines how markings change over time is defined as $m(\cdot,\tau) = m_0 + C\sigma(\tau)$ where $\sigma(\tau)$ is the firing vector (Mahulea et al. 2006). Differentiating this equation with respect to time gives

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

with

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

where the flow $f(\cdot,\tau) = \sigma'(\tau)$ is the derivative (in the mathematical sense) of the firing sequence (Mahulea et al. 2006). The change in the marking of a single place p can be expressed as follows. Let n be the number of transitions. Then

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

PEPA AND ODE SEMANTICS

Consider the standard syntax for PEPA (Hillston 2005) with Hiding omitted: $P ::= P \underset{L}{\bowtie} P \mid C$ and $S ::= (\alpha, r).S \mid S + S \mid C_s$ where C names a model or sequential 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 given in Figure 2 (Hillston 1996) with the following operators.

Prefix $(\alpha, r).P$ can be understood as the process that can perform the action α with a delay from the exponential distribution determined by r and which then behaves as P.

Choice $P_1 + P_2$ represents the choice between behaving as either P_1 or P_2. The process that completes first will proceed and the other will be discarded.

Cooperation $P_1 \underset{L}{\bowtie} P_2$ can act as P_1 independently of P_2 (and vice versa) for any actions not in L. For actions in L, P_1 and P_2 can only proceed when they both can perform the action and the rate is determined by the slower of the two. $P_1 || P_2$ is used for $P_1 \underset{\emptyset}{\bowtie} P_2$, and $P[n]$ for n copies of P in parallel without cooperation.

Constant $C \stackrel{def}{=} P$ defines the constant C which has the same behaviour as P.

The following subset of PEPA will be considered, as this is the subset for which the ODE semantics are defined (Hillston 2005). It only allows communication between components that are not identical.

1. All shared actions of non-identical components must synchronise.
2. Identical components cannot synchronise on actions.
3. Rates must be identical for a shared activity, and no passive rates are allowed.

Definition 3 (Hillston 2005) *For an arbitrary PEPA model M with n component types C_i for $i = 1, \ldots n$ each with N_i distinct derivatives (successor states of components), the* numerical vector form *of M, $V(M)$ is a vector with $N = \sum_{i=1}^{n} N_i$ entries. Each v_{i_j} records how many instances of*

Prefix

$$\frac{}{(\alpha, r).E \stackrel{(\alpha,r)}{\rightarrow} E}$$

Constant

$$\frac{E \stackrel{(\alpha,r)}{\rightarrow} E'}{A \stackrel{(\alpha,r)}{\rightarrow} E'} (A \stackrel{def}{=} E)$$

Choice

$$\frac{E \stackrel{(\alpha,r)}{\rightarrow} E'}{E + F \stackrel{(\alpha,r)}{\rightarrow} E'} \qquad \frac{F \stackrel{(\alpha,r)}{\rightarrow} F'}{E + F \stackrel{(\alpha,r)}{\rightarrow} F'}$$

Cooperation

$$\frac{E \stackrel{(\alpha,r)}{\rightarrow} E'}{E \underset{L}{\bowtie} F \stackrel{(\alpha,r)}{\rightarrow} E' \underset{L}{\bowtie} F} (\alpha \notin L)$$

$$\frac{F \stackrel{(\alpha,r)}{\rightarrow} F'}{E \underset{L}{\bowtie} F \stackrel{(\alpha,r)}{\rightarrow} E \underset{L}{\bowtie} F'} (\alpha \notin L)$$

$$\frac{E \stackrel{(\alpha,r_1)}{\rightarrow} E' \quad F \stackrel{(\alpha,r_2)}{\rightarrow} F'}{E \underset{L}{\bowtie} F \stackrel{(\alpha,R)}{\rightarrow} E' \underset{L}{\bowtie} F'} (\alpha \in L)$$

where $R = \dfrac{r_1}{r_\alpha(E)} \dfrac{r_2}{r_\alpha(F)} \min(r_\alpha(E), r_\alpha(F))$

Figure 2: PEPA Structured Operational Semantics

the jth local derivative of component type C_i are present in the current state.

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 (Hillston 2005). The vector state space gives an aggregated model and can generate a CTMC. The numerical vector form is used for the 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 (Hillston 2005).

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

for $Ex(D) = \{(\alpha, r) \mid D \stackrel{(\alpha,r)}{\rightarrow}\}$, $Ex(\alpha, r) = \{D \mid D \stackrel{(\alpha,r)}{\rightarrow}\}$ and $En(D) = \{(\alpha, r) \mid \exists D', D' \stackrel{(\alpha,r)}{\rightarrow} D\}$. 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 or an activity matrix (Hillston 2005).

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. The* activity matrix *M_a is an $N_p \times N_t$ matrix and entries are defined as follows.*

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

The example in Figure 3 consists of two different types of servers, possibly offering the same service, but at different rates, plus clients who do not mind which server they use. The servers provide the service then reset, and a client interacts with a server and then does something before interacting with a server again. The activity matrix is given in Figure 5 together with the ODEs for this model. The activity graph for this system is given in Figure 4.

FROM A PEPA MODEL TO A CPN

Working with the subset of PEPA specified previously, it is possible to convert this to a timed CPN. First, construct the activity graph. This is a net of the form $(N_p, N_t, M_{pre}, M_{post})$ where

$$M_{pre}(p_i, t_j) = \begin{cases} +1 \text{ if } t_j \in Ex(p_i) \\ 0 \quad \text{otherwise} \end{cases}$$

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

The activity matrix M_a is then $M_{post} - M_{pre}$ which is 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 $t = (\alpha, r) \in N_t$. Hence a CPN can be constructed from a PEPA model, and the activity graph and the CPN are isomorphic.

Moreover, for each distinct type of derivative in the model, there is 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 obtained from the PEPA model is isomorphic to the reachability graph of the CPN if it were viewed 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 represents that particular derivative and add a token to the place that represents the other derivative. Similarly the token changes associated

$$C \stackrel{def}{=} (serv_1, s_1).C' + (serv_2, s_2).C' \quad C' \stackrel{def}{=} (do, d).C \quad S_i' \stackrel{def}{=} (reset_i, r_i).S_i \quad S_i \stackrel{def}{=} (serv_i, s_i).S_i'$$

$$Sys \stackrel{def}{=} C[100] \underset{\{serv_1, serv_2\}}{\bowtie} \left(S_1[50] || S_2[50]\right)$$

Figure 3: Servers Example

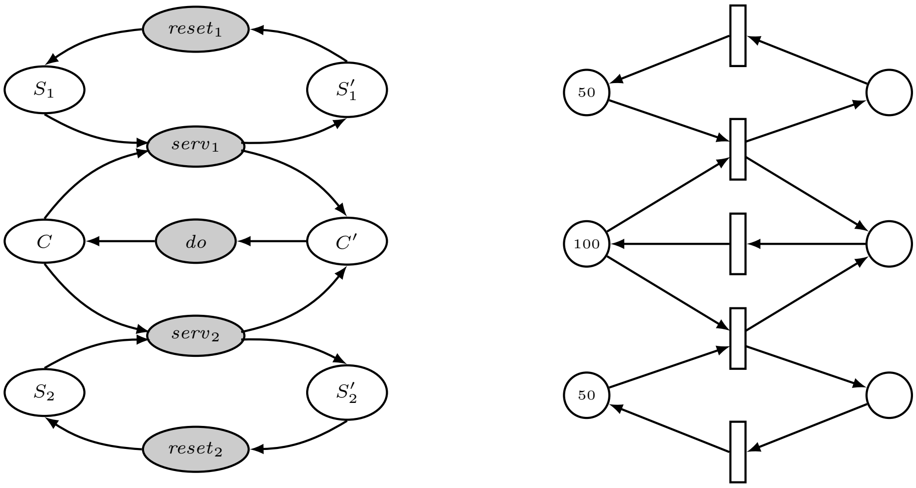

Figure 4: The Activity Graph and its Translation to a CPN with Initial Marking for the Servers Example

with a firing of a transition can be expressed as changes in the number of derivatives in the PEPA model. Hence, $m(p, \tau) = N(p, \tau)$. The resulting net is bounded because the total number of derivatives for each component is fixed in the PEPA model.

Considering the continuous case, the ODEs defined from PEPA for the place p are as follows.

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

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 CPN under infinite server semantics.

Figure 4 shows the CPN obtained from the activity graph for the servers example, together with its initial marking. Pre and $Post$ can easily be determined, and it is clear, for example, that $Ex(C)$ is the same as the post set of the place with initial marking 100.

FROM A CPN TO A PEPA MODEL

It is more complex to do the reverse translation. First, it is necessary to add implicit places to the net and to show this does not affect the ODEs that are obtained. Once implicit places have been added, the net can be transformed to a PEPA model using an existing algorithm (Hillston et al. 2001). An implicit place is a place in a net that can be removed without changing the behaviour of the net (Silva et al. 1998).

Definition 5 *An implicit place $p \in P$ is a place such that whenever $t \in p^\bullet$ is enabled at time τ, $m(p, \tau) \geq$ $\min_{p' \in {}^\bullet t \setminus \{p\}} \{m(p', \tau)\}$*

The addition of an implicit or complementary place for each place in a net is called complementation. For each place p, a new place $\overline{p}$ is added such that ${}^\bullet\overline{p} = p^\bullet$ and $\overline{p}^\bullet = {}^\bullet p$ using the following construction (Hillston et al. 2001). The construction requires that the net be bounded, and for each place p there is an upper bound $b(p)$ on the value that the place p can have in any marking reachable from m_0.

Definition 6 *Given a net $S = \langle N, m_0 \rangle$ with $N = (P, T, Pre, Post, \lambda)$, its* complementation *is a net $S' = \langle N', m_0' \rangle$ with $N' = (P', T, Pre', Post', \lambda)$ where $P' = P \cup \overline{P}$ with $\overline{P} = \{\overline{p} \mid p \in P\}$.*

$$Pre' = \begin{bmatrix} Pre \\ Post \end{bmatrix} \; Post' = \begin{bmatrix} Post \\ Pre \end{bmatrix} \; m_0' = \begin{bmatrix} m_0 \\ b - m_0 \end{bmatrix}$$

A new place $\overline{p}$ can be shown to be implicit by proof by contradiction on the bound $b(p)$. Also $m_{S'}(p, \tau) = m_S(p, \tau)$ for all $p \in P$, where m_R refers to the marking in the context

	$serv_1$	$serv_2$	$reset_1$	$reset_2$	do
C	-1	-1	0	0	$+1$
C'	$+1$	$+1$	0	0	-1
S_1	-1	0	$+1$	0	0
S_1'	$+1$	0	-1	0	0
S_2	0	-1	0	$+1$	0
S_2'	0	$+1$	0	-1	0

$$\frac{dN(C,\tau)}{d\tau} = d.N(C',\tau)-s_1.\min(N(C,\tau),N(S_1,\tau))-s_2.\min(N(C,\tau),N(S_2,\tau))$$

$$\frac{dN(C',\tau)}{d\tau} = s_1.\min(N(C,\tau),N(S_1,\tau))+s_2.\min(N(C,\tau),N(S_2,\tau))-d.N(C',\tau)$$

$$\frac{dN(S_i,\tau)}{d\tau} = r_i.N(S_i')-s_i.\min(N(C,\tau),N(S_i,\tau)), \quad i=1,2$$

$$\frac{dN(S_i',\tau)}{d\tau} = s_i.\min(N(C,\tau),N(S_i,\tau))-r_i.N(S_i'), \quad i=1,2$$

Figure 5: The Activity Matrix and the ODEs for the Servers Example

of the net R. Furthermore $enab_{S'}(t,\tau) = enab_S(t,\tau)$ for all $t \in T$ where $enab_R$ refers to the enabling degree in the context of the net R. The following ODEs can be obtained from the CPN S' created by complementation of S.

$$\begin{aligned}\frac{dm_{S'}(p,\tau)}{d\tau} &= \sum_{t\in {}^\bullet p}\lambda(t).\min_{p'\in {}^\bullet t}\{m_{S'}(p',\tau)\} \\ &- \sum_{t\in p^\bullet}\lambda(t).\min_{p'\in {}^\bullet t}\{m_{S'}(p',\tau)\}.\end{aligned}$$

The focus here is on the original places in P. Note that

$$\begin{aligned}\min_{p'\in {}^\bullet t}\{m_{S'}(p',\tau)\} &= \min\big(\min_{p\in {}^\bullet t\cap P}\{m_S(p,\tau)\},\min_{\overline{p}\in {}^\bullet t\cap \overline{P}}\{m_{S'}(\overline{p},\tau)\}\big) \\ &= \min_{p\in {}^\bullet t\cap P}\{m_S(p,\tau)\}\end{aligned}$$

since for each $\overline{p}$, $m(\overline{p},\tau) \geq \min_{p'\in {}^\bullet t\cap P}\{m(p',\tau)\}$, and so the value of the marking at that implicit place can be ignored when determining the minimum. Considering a place p in P, we obtain the same equation as that obtained before complementation.

$$\begin{aligned}\frac{dm_{S'}(p,\tau)}{d\tau} &= \sum_{t\in {}^\bullet p}\lambda(t).\min_{p'\in {}^\bullet t\cap P}\{m_S(p',\tau)\} \\ &- \sum_{t\in p^\bullet}\lambda(t).\min_{p'\in {}^\bullet t\cap P}\{m_S(p',\tau)\}.\end{aligned}$$

The net formed by complementation can be translated into a PEPA model using the algorithm presented by Hillston et al. (Hillston et al. 2001). The algorithm has one minor change; the rates are not passive in the sequential components. This is necessary to meet the restriction on the form a PEPA model can take and does not affect the correctness of the algorithm. It is possible to derive ODEs from this PEPA model as done above. However, it is simpler to use the high/low approach for modelling biological systems in a reagent-centric fashion (Calder et al. 2005). In this approach, each reagent can be at a high or low concentration, and this is modelled using a single component with two states. The ODEs are then obtained for the reagent rather than for each of its states. In the cited reference, mass action is used in the ODEs since it is appropriate for the biological model. In the context of this paper, min is the appropriate function to use as it captures possible synchronisations.

The PEPA model obtained by complementation can be viewed similarly, so component C_p is the high concentration and its derivative $C_{\overline{p}}$ is the low concentration. An activity graph and an activity matrix can be constructed (Calder et al. 2005).

Definition 7 *An* HL activity graph *is a bipartite graph (N,A). The nodes are partitioned into N_t the activities and $N_p = \{C_p \mid p \in P\}$. $A \subseteq (N_t \times N_p) \cup (N_p \times N_t)$, where $a = (n_t, n_p) \in A$ if n_t if n_t is an entry activity of C_p (and an exit activity of $C_{\overline{p}}$) and $a = (n_p, n_t) \in A$ if n_t an exit activity of C_p (and an entry activity of $C_{\overline{p}}$). The* HL activity matrix *M_a is an $N_p \times N_t$ matrix and entries are defined as follows.*

$$M_a(C_p,t) = \begin{cases} +1 & \textit{if } t \textit{ is an entry activity of } C_p \\ -1 & \textit{if } t \textit{ is an exit activity of } C_p \\ 0 & \textit{otherwise.}\end{cases}$$

Note that here the activities are just the transitions T from the net. However, N_p is the same size as the set of original places, and hence the activity graph is not isomorphic to the complemented net but rather to the original CPN. The ODEs that can be extracted from this model are as follows.

$$\begin{aligned}\frac{dN(C_p,\tau)}{d\tau} &= \sum_{(t,\lambda(t))\in En(C_p)}\lambda(t) \times \min_{C_{p'}\in Ex(t,\lambda(t))}\{N(C_{p'},\tau)\} \\ &- \sum_{(t,\lambda(t))\in Ex(C_p)}\lambda(t) \times \min_{C_{p'}\in Ex(t,\lambda(t))}\{N(C_{p'},\tau)\}\end{aligned}$$

Note that $N(C_p,\tau) = m(p,\tau)$, $(t,\lambda(t)) \in En(C_p)$ is equivalent to $t \in {}^\bullet p$ and $(t,\lambda(t)) \in Ex(C_p)$ is equivalent to $t \in p^\bullet$. Moreover, $C_{p'} \in Ex(t,\lambda(t))$ is equivalent to $p' \in {}^\bullet t \cap P$ since only the components created from $p \in P$ and not the complementary places, are under consideration. Hence, the ODEs are the same.

Hillston et al. (Hillston et al. 2001) also present an algorithm that reduces the number of implicit places required. For the algorithm presented above, this improvement is not necessary since using the high/low reagent approach means that a place and its complement are treated as a single object resulting in ODEs that are the same as those found for the original CPN.

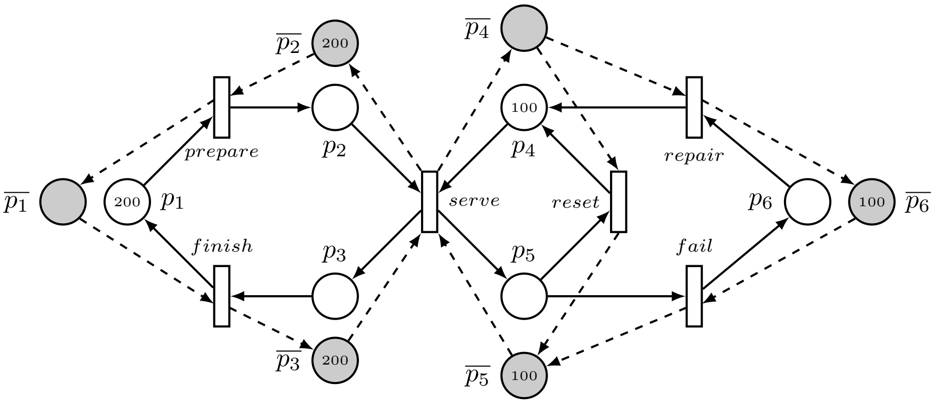

Figure 6: A CPN with Complementation

	prepare	*serve*	*finish*	*reset*	*fail*	*repair*
C_{p_1}	-1	0	$+1$	0	0	0
C_{p_2}	$+1$	-1	0	0	0	0
C_{p_3}	0	$+1$	-1	0	0	0
C_{p_4}	0	-1	0	$+1$	0	$+1$
C_{p_5}	0	$+1$	0	-1	-1	0
C_{p_6}	0	0	0	0	$+1$	-1

$$
\begin{aligned}
\frac{dm(p_1,\tau)}{d\tau} &= f.m(p_3,\tau) - p.m(p_1,\tau) \\
\frac{dm(p_2,\tau)}{d\tau} &= p.m(p_1,\tau) - s.min\{m(p_2,\tau), m(p_4,\tau)\} \\
\frac{dm(p_3,\tau)}{d\tau} &= s.min\{m(p_2,\tau), m(p_4,\tau)\} - f.m(p_3,\tau) \\
\frac{dm(p_4,\tau)}{d\tau} &= r.m(p_5,\tau) + e.m(p_6,\tau) - s.min\{m(p_2,\tau), m(p_4,\tau)\} \\
\frac{dm(p_5,\tau)}{d\tau} &= s.min\{m(p_2,\tau), m(p_4,\tau)\} - r.m(p_5,\tau) - a.m(p_5,\tau) \\
\frac{dm(p_6,\tau)}{d\tau} &= a.m(p_5,\tau) - e.m(p_6,\tau)
\end{aligned}
$$

Figure 7: The Activity Matrix Obtained from the Unreliable Servers PEPA Model and the ODEs Obtained from the Unreliable Servers CPN in Figure 1

Figure 6 shows the unreliable servers example from Figure 1 under complementation with appropriate bounds. The ODEs from the uncomplemented net are given in Figure 7. From the complemented net the PEPA model in Figure 8 is obtained. The HL activity matrix is given in Figure 7, and the same ODEs as in Figure 7 can be extracted from this matrix.

SERVER SEMANTICS

As mentioned above, there are two types of server semantics. In infinite server semantics, a transition can fire simultaneously as much as it is enabled. This is also called marking dependent or variable speed (David and Alla 2005). For the finite server or constant speed case (David and Alla 2005), the number of simultaneous firings of transition is bounded. The transformations above have shown that the ODE semantics of PEPA are infinite server semantics. Infinite server semantics are the more general case. In the discrete case, finite server semantics can be obtained in a net with infinite server semantics by 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 (Mahulea et al. 2006).

A similar approach can be taken in PEPA. When a particular activity is to be limited, a new component can be added of the form $B_\alpha \stackrel{\scriptscriptstyle def}{=} (\alpha, r).B_\alpha$ with k_α copies where (α, r) is the activity to be constrained. Letting $N(C)$ refer only to the original components and their derivatives, then the resulting ODEs can be expressed as

$$
\begin{aligned}
\frac{dN(D,\tau)}{d\tau} = &\sum_{(\alpha,r)\in En(D)} r \times \min\{k_\alpha, \min_{C\in Ex(\alpha,r)} \{N(C,\tau)\}\} \\
&- \sum_{(\alpha,r)\in Ex(D)} r \times \min\{k_\alpha, \min_{C\in Ex(\alpha,r)} \{N(C,\tau)\}\}. \quad (*)
\end{aligned}
$$

Note that the ODE for each B_α has the form $\frac{dN(B_\alpha)}{d\tau} = 0$ since the number of copies is always k_α. An example is given in Figures 10 and 9. This represents senders that collect data and pass it on to receivers that deliver the data. The more abstract version (on the left in Figure 10 and $SendRec$ in Figure 9) has infinite server semantics and represents the system without constraints. The second version (on the right and $SendRecK$) has additional places/components that constrain the transitions/activities. This represents a more concrete version of the system where there is a medium between senders and receivers that only permits a certain number of exchanges at one time, and both the sender and receiver must each interact with a buffer of limited capacity in collecting or delivering the data. The behaviour over time of this system

$$
\begin{array}{llllll}
C_{p_1} \stackrel{def}{=} (prepare, p).C_{\overline{p}_1} & & C_{p_2} \stackrel{def}{=} (serve, s).C_{\overline{p}_2} & & C_{p_3} \stackrel{def}{=} (finish, f).C_{\overline{p}_3} \\
C_{\overline{p}_1} \stackrel{def}{=} (finish, f).C_{p_1} & & C_{\overline{p}_2} \stackrel{def}{=} (prepare, p).C_{p_2} & & C_{\overline{p}_3} \stackrel{def}{=} (serve, s).C_{p_3} \\
C_{p_4} \stackrel{def}{=} (serve, s).C_{\overline{p}_4} & & C_{p_5} \stackrel{def}{=} (reset, r).C_{\overline{p}_5} + (fail, a).C_{\overline{p}_5} & & C_{p_6} \stackrel{def}{=} (repair, e).C_{\overline{p}_6} \\
C_{\overline{p}_4} \stackrel{def}{=} (reset, r).C_{p_4} + (repair, e).C_{p_4} & & C_{\overline{p}_5} \stackrel{def}{=} (serve, s).C_{p_5} & & C_{\overline{p}_6} \stackrel{def}{=} (fail, a).C_{p_5}
\end{array}
$$

$$
\mathcal{M} \stackrel{def}{=} C_{p_1}[200] \underset{\{prepare,finish\}}{\bowtie} C_{\overline{p}_2}[200] \underset{\{serve\}}{\bowtie} C_{\overline{p}_3}[200] \underset{\{serve\}}{\bowtie} C_{p_4}[100] \underset{\{serve,reset,repair\}}{\bowtie} C_{\overline{p}_5}[100] \underset{\{fail\}}{\bowtie} C_{\overline{p}_6}[100]
$$

Figure 8: Unreliable Servers Example

$$
P_1 \stackrel{def}{=} (t_2, \lambda_2).P_2 \qquad P_2 \stackrel{def}{=} (t_1, \lambda_1).P_1 \qquad P_3 \stackrel{def}{=} (t_2, \lambda_2).P_4 \qquad P_4 \stackrel{def}{=} (t_3, \lambda_3).P_3 \qquad B_{t_i} \stackrel{def}{=} (t_i, k_{t_i}).B_{t_i}
$$

$$
SendRec \stackrel{def}{=} P_1[100] \underset{t_2}{\bowtie} P_3[100] \qquad SendRecK \stackrel{def}{=} (P_1[100] \underset{t_1}{\bowtie} B_{t_1}[k_{t_1}]) \underset{t_2}{\bowtie} (B_{t_2}[k_{t_2}] \underset{t_2}{\bowtie} (P_3[100] \underset{t_3}{\bowtie} B_{t_3}[k_{t_3}]))
$$

Figure 9: PEPA Models of Sender and Receiver Example

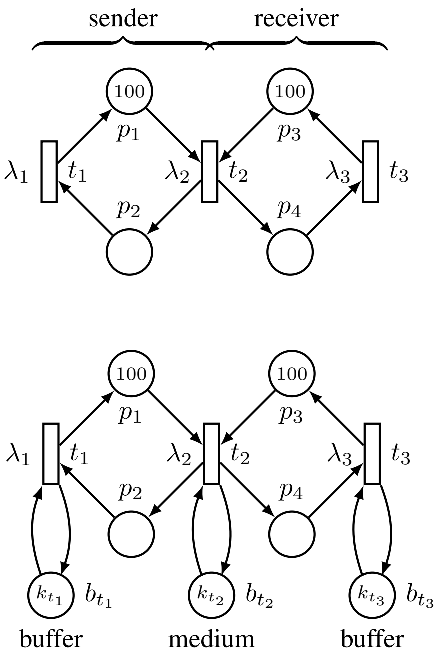

Figure 10: CPNs for Sender and Receiver Example

can then be expressed using the approximation given by the ODEs in $(*)$.

However, Silva and Recalde (Silva and Recalde 2005) argue that in the continuous case, infinite and finite server semantics are about two different relaxations of the model for approximation; many servers and many clients for infinite, and few servers and many clients for finite server semantics and are therefore different. In their view of finite server semantics, a transition has a maximal firing speed $\lambda'(t)$ at which it can perform representing k times the speed of a single server (Mahulea et al. 2006) and a different expression is required for the flow of a transition.

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

In the case that $enab(t,\tau) > 0$ then the flow is just the firing speed of the transition. Otherwise, the flow through the transition is determined by minimum of the speeds of the transitions supplying the empty places that supply the transition. Determining the speeds requires the solution of a linear programming problem (David and Alla 2005). Hence for ordinary nets, the ODEs have the form

$$
\frac{dm(p,\tau)}{d\tau} = \sum_{t \in {}^\bullet p} f(t,\tau) - \sum_{t \in p^\bullet} f(t,\tau)
$$

with $f(t,\tau)$ defined as above, and with the Pre and $Post$ terms equal to one. These equations result in linear piecewise ODEs (Mahulea et al. 2006). The net behaves as described by a set of equations until a place empties, and then the speeds change since the enabling of the net has changed (a place is now empty). This switching between equations continues until a steady state is reached where no more changes in the marking can happen. This steady state is guaranteed if there is no conflict in the net (David and Alla 2005).

Infinite server semantics also result in linear piecewise ODEs. Changes between equations occur because of the minimum function. Hence the ODEs for both types of server semantics can be viewed as a hybrid system. ie. a system with both discrete and continuous behaviours. The continuous behaviour in this case are the sets of ODEs and the discrete behaviour is the switching from one set of ODEs to another due to a specific event, either the change in the minimum or the emptying of a place. Hybrid systems are an area of ongoing research.

A question of interest is how these the finite server ODEs compare to the explicitly constrained infinite server ODES. Transforming the ODEs in ($*$) to the equivalent Petri net description gives

$$\begin{aligned}\frac{dm(p,\tau)}{d\tau} &= \sum_{t\in{}^\bullet p}\lambda(t).\min\{k(t),\min_{p'\in{}^\bullet t}\{m(p',\tau)\}\}\\ &- \sum_{t\in p^\bullet}\lambda(t).\min\{k(t),\min_{p'\in{}^\bullet t}\{m(p',\tau)\}\}\end{aligned}$$

where $k(t)$ is the equivalent of k_α. Since $\lambda(t)$ represents the rate at which transition t can fire, and $\lambda'(t)$ represents the speed at which $k(t)$ servers can be served, it can be assumed that $\lambda(t).k(t) = \lambda'(t)$. Hence the question becomes a comparison of the terms

$$\phi(t,\tau) = \lambda(t).\min\{k(t),\min_{p'\in{}^\bullet t}\{m(p',\tau)\}\} \quad \text{and}$$

$$f(t,\tau) = \begin{cases}\lambda(t).k(t), & \text{if } \forall p\in{}^\bullet t, m(p,\tau)>0\\ \min\Big\{\lambda(t).k(t), \min\limits_{\substack{p\in{}^\bullet t\wedge\\ m(p,\tau)=0}}\Big\{\sum\limits_{t'\in{}^\bullet p} f(t',\tau)\Big\}\Big\}, & \text{otherwise.}\end{cases}$$

Clearly the closer $k(t)$ is to zero, the closer the values of ϕ and f will be since the term involving $k(t)$ is likely to be the minimum and hence both equation are likely to have the value $\lambda(t).k(t)$. Understanding in more detail the relationship between these three approximations is future research.

CONCLUSION

This paper has shown how to construct an ordinary timed continuous Petri net from a PEPA model (using a subset of the language) and vice versa. Moreover it has been shown that when approximated continuously, the behaviour of both can be characterised by the same set of coupled ODEs. Furthermore the continuous approximation using PEPA has infinite server semantics. This paper has established links between two continuous approaches to modelling the performance of systems, one graphically based and the other textual, and hence techniques and results for one approach can be applied to the other. Both continuous approximation approaches are suitable for systems with many identical components.

Questions for ongoing research include how robust these modelling approaches are when component numbers decrease, and how the appropriate ODEs can be obtained when modelling large numbers of clients and a much smaller number of servers.

ACKNOWLEDGEMENTS

The author thanks Jane Hillston for her comments. The author is supported by the EPSRC SIGNAL Project, Grant EP/E031439/1. Part of this research was conducted while the author was on sabbatical leave from the University of the Witwatersrand.

REFERENCES

Ajmone Marsan M.; G. Balbo; G. Conte; S. Donatelli; and G. Franceschinis. 1995. *Modelling with generalized stochastic Petri nets*. Wiley.

Calder M.; S. Gilmore; and J. Hillston. 2005. "Automatically deriving ODEs from process algebra models of signalling pathways". In *Proceedings of Computational Methods in Systems Biology (CMSB 2005)*. 204–215.

David R. and H. Alla. 2005. *Discrete, continuous and hybrid Petri nets*. Springer.

Gilmore S.T. and M. Tribastone. 2006. "Evaluating the Scalability of a Web Service-Based Distributed e-Learning and Course Management System". In *Web Services and Formal Methods, Third International Workshop (WS-FM 2006)*. Springer, LNCS 4184, 214–226.

Hillston J. 1996. *A compositional approach to performance modelling*. Cambridge University Press.

Hillston J. 2005. "Fluid Flow Approximation of PEPA models". In *Second International Conference on the Quantitative Evaluation of Systems (QEST 2005)*. IEEE Computer Society, 33–43.

Hillston J.; L. Recalde; M. Ribaudo; and M. Silva. 2001. "A Comparison of the Expressiveness of SPA and Bounded SPN Models". In *Proceedings of the 9th International Workshop on Petri Nets and Performance Models (PNPM'01)*. IEEE Computer Society, 197–206.

Mahulea C.; L. Recalde; and M. Silva. 2006. "On performance monotonicity and basic servers semantics of continuous Petri nets". In *Eighth International Workshop on Discrete Event Systems (WODES '06)*.

Ribaudo M. 1995. "Stochastic Petri net semantics for stochastic process algebras". In *Sixth International Workshop on Petri Nets and Performance Models (PNPM'95)*. IEEE Computer Society, 148–157.

Silva M. and L. Recalde. 2005. "Continuization of Timed Petri Nets: From Performance Evaluation to Observation and Control". In *Applications and Theory of Petri Nets 2005, 26th International Conference (ICATPN 2005)*. Springer, LNCS 3536, 26–47.

Silva M.; E. Teruel; and J.M. Colom. 1998. "Linear Algebraic and Linear Programming Techniques for the Analysis of Place or Transition Net Systems". In *Lectures on Petri Nets I: Basic Models*. Springer, LNCS 1491, 309–373.

BIOGRAPHY

VASHTI GALPIN completed her tertiary education up to MSc level at the University of the Witwatersrand, Johannesburg, South Africa. She was awarded her PhD in Computer Science by the University of Edinburgh in 1998. Her thesis considered the metatheory of process algebras. After completing a one-year postdoctoral fellowship at the University of the Witwatersrand, she worked there as a lecturer for seven years. During this period her research focus was women in computing and computer science education research. She is currently a research associate in the Laboratory for Foundations of Computer Science in the School of Informatics at the University of Edinburgh and conducts research into stochastic process algebras as well as their application to modelling in systems biology.