Improved Continuous Approximation of PEPA Models through Epidemiological Examples

Soufiene Benkirane[a]   Jane Hillston[b]   Chris McCaig[a]
Rachel Norman[a]   Carron Shankland[a,1]

[a] *Department of Computing Science and Mathematics, University of Stirling, UK*

[b] *School of Informatics, University of Edinburgh, UK*

Abstract

We present two individual based models of disease systems using PEPA (Performance Evaluation Process Algebra). The models explore contrasting mechanisms of disease transmission: direct transmission (e.g. measles) and indirect transmission (e.g. malaria, via mosquitos). We extract ordinary differential equations (ODEs) as a continuous approximation to the PEPA models using the Hillston method and compare these with the traditionally used ODE disease models and with the results of stochastic simulation. Improvements to the Hillston method of ODE extraction for this context are proposed, and the new results compare favourably with stochastic simulation results and to ODEs derived for equivalent models in WSCCS (Weighted Synchronous Calculus of Communicating Systems).

Keywords: PEPA, process algebra, changing scale, individual based model, population based model

1   Introduction

Increasingly, theoretical computer science techniques such as process algebras are being used to tackle biological problems [3,15,14]. The belief is that formal specification will lead to more insight into the problem being described. Certainly, process algebras provide different forms of analysis not previously available to biologists. In the case of epidemiology, the study of disease spread, process algebra gives us a way to describe individual based models (drawn from observations of individual behaviour) and to then automatically derive population level models (allowing investigation of the properties of epidemics, and the result of control methods). Largely, theoretical biologists have worked at one scale (individual based) or the other (population based), but had no way to move between these scales. Process algebra

---

semantics allow a straightforward move from the specification of individual components (agents) to the derived behaviour of those components operating in parallel (the labelled transition system). Recent developments [8,13,4] have additionally allowed the population level behaviour of a process algebra model to be expressed as ordinary differential equations (ODEs) or mean field equations (MFEs). This opens up further possibilities for mathematical analysis and facilitates comparison with existing mathematical models for these systems. The goal of our work is to contribute new ways of working in epidemiology through process algebra.

This paper presents models of biological epidemics in the PEPA process algebra [7]. This is a novel use of PEPA. Our aim is to evaluate the suitability of language features of PEPA for epidemiology, and to compare the results with our previous modelling work in WSCCS [12]. The most important difference between the formalisms is the move from actions occurring in discrete time and weighted choice (WSCCS), to actions occurring in continuous time at a specified rate and choice driven by competition between rates (PEPA). In addition, PEPA allows synchronisation between multiple components.

A further reason for choosing PEPA over alternative formalisms is the ability to translate PEPA models into ODEs [8] (the Hillston method). The epidemiological models presented in this paper provide a way to evaluate this method of deriving ODEs. The method places some limitations on the form of the PEPA model:

- Cooperating components must share the same local rate (no passive components).
- Actions may not be hidden.
- Cooperation must include all common actions.
- Components of the same type do not cooperate.

Through the models presented here we discover a fifth limitation:

- There should not be implicit choice between independent but distinct components offering the same action for synchronisation.

We present a modified version of the method here (the Stirling amendment) which addresses this last limitation and that of passive rates. The contribution of the paper is therefore twofold: a novel use of PEPA in epidemiology, and an improved method of deriving ODEs for this context.

The structure of the paper is as follows. Section 2 provides an introduction to PEPA and ODE derivation as previously described in [8]. Section 3 presents the epidemiological models and compares the derived ODEs with simulation results. The insights gained through these models lead to the modified ODE derivation method presented in Section 4. Finally, we discuss the results in Section 5.

2 PEPA

PEPA has been used to study the performance of a wide variety of systems [9,1,2,16,10]. As in all process algebras, systems are represented in PEPA as the composition of *components* which undertake *actions*. In PEPA the actions are assumed to have a duration, or delay. Thus the expression $(\alpha, r).P$ denotes a component which can undertake an α action, at rate r (where rate is 1/delay) to

evolve into a component P. Here $\alpha \in \mathcal{A}$ where $\mathcal{A}$ is the set of action types and $P \in \mathcal{C}$ where $\mathcal{C}$ is the set of component types.

PEPA has a small set of combinators, allowing system descriptions to be built up as the concurrent execution and interaction of simple sequential components. We informally introduce the syntax below. More detail can be found in [7]. The structured operational semantics are shown in Figure 1.

Prefix

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

Choice

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

Constant

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

Hiding

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E/L \xrightarrow{(\alpha,r)} E'/L} \ (\alpha \notin L) \qquad\qquad \frac{E \xrightarrow{(\alpha,r)} E'}{E/L \xrightarrow{(\tau,r)} E'/L} \ (\alpha \in L)$$

Cooperation

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E \bowtie_L F \xrightarrow{(\alpha,r)} E' \bowtie_L F} \ (\alpha \notin L) \qquad\qquad \frac{F \xrightarrow{(\alpha,r)} F'}{E \bowtie_L F \xrightarrow{(\alpha,r)} E \bowtie_L F'} \ (\alpha \notin L)$$

$$\frac{E \xrightarrow{(\alpha,r_1)} E' \ F \xrightarrow{(\alpha,r_2)} F'}{E \bowtie_L F \xrightarrow{(\alpha,R)} E' \bowtie_L F'} \ (\alpha \in L), \qquad R = \frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)} \min(r_\alpha(E), r_\alpha(F))$$

Fig. 1. Operational Semantics of PEPA

Prefix: The basic mechanism for describing the behaviour of a system with a PEPA model is to give a component a designated first action using the prefix

combinator, denoted by a full stop, which was introduced above. As explained, $(\alpha, r).P$ carries out an α action with rate r, and it subsequently behaves as P.

Choice: The component $P + Q$ represents a system which may behave either as P or as Q. The activities of both P and Q are enabled. The first activity to complete distinguishes one of them: the other is discarded. The system will behave as the derivative resulting from the evolution of the chosen component.

Constant: It is convenient to be able to assign names to patterns of behaviour associated with components. Constants are components whose meaning is given by a defining equation. The notation for this is $X \stackrel{def}{=} E$. The name X is in scope in the expression on the right hand side meaning that, for example, $X \stackrel{def}{=} (\alpha, r).X$ performs α at rate r forever.

Hiding: The possibility to abstract away some aspects of a component's behaviour is provided by the hiding operator, denoted P/L. Here, the set L identifies those activities which are to be considered internal or private to the component and which will appear as the unknown type τ.

Cooperation: We write $P \bowtie_L Q$ to denote cooperation between P and Q over L. The set which is used as the subscript to the cooperation symbol, the *cooperation set* L, determines those activities on which the *cooperands* are forced to synchronise. For action types not in L, the components proceed independently and concurrently with their enabled activities. We write $P \parallel Q$ as an abbreviation for $P \bowtie_L Q$ when L is empty.

If a component enables an activity whose action type is in the cooperation set it will not be able to proceed with that activity until the other component also enables an activity of that type. The two components then proceed together to complete the *shared activity*. The rate of the shared activity may be altered to reflect the work carried out by both components to complete the activity. The total capacity of a component C to carry out activities of type α is termed the *apparent rate* of α in P, denoted $r_\alpha(P)$, see Section 2.1 for the definition. Unlike some other stochastic process algebras, PEPA assumes *bounded capacity*: a component cannot be made to perform an activity faster by cooperation, so the rate of a shared activity is the minimum of the rates of the activity in the cooperating components.

In some cases, when an activity is known to be carried out in cooperation with another component, a component may be *passive* with respect to that activity. This means that the rate of the activity is left unspecified (denoted $\top$) and is determined upon cooperation by the rate of the activity in the other component. All passive actions must be synchronised in the final model.

The syntax may be formally introduced by means of the following grammar:

$$S ::= (\alpha, r).S \mid S + S \mid C_S$$
$$P ::= P \bowtie_L P \mid P/L \mid C$$

where S denotes a *sequential component* and P denotes a *model component* which executes in parallel. C stands for a constant which denotes either a sequential component or a model component. C_S stands for constants which denote sequential components. The effect of this syntactic separation between these types of constants is to constrain legal PEPA components to be cooperations of sequential processes, a necessary condition for an ergodic underlying Markov process.

2.1 Apparent Rate

The *apparent rate* at which an action type occurs within a component is of importance when comparing components or when defining how they interact. We assume that the apparent rate of an action type represents the totally capacity of a component to carry out activities of that type when it is in its current state.

Definition 2.1 (Apparent Rate) *The* apparent rate *of action of type α in a component P, denoted $r_\alpha(P)$, is the sum of the rates of all activities of type α in $\mathcal{A}ct(P)$.*

(i) $r_\alpha((\beta, r).P) = \begin{cases} r & \text{if } \beta = \alpha \\ 0 & \text{if } \beta \neq \alpha \end{cases}$

(ii) $r_\alpha(P + Q) = r_\alpha(P) + r_\alpha(Q)$

(iii) $r_\alpha(P/L) = \begin{cases} r_\alpha(P) & \text{if } \alpha \notin L \\ 0 & \text{if } \alpha \in L \end{cases}$

(iv) $r_\alpha(P \underset{L}{\bowtie} Q) = \begin{cases} \min(r_\alpha(P), r_\alpha(Q)) & \text{if } \alpha \in L \\ r_\alpha(P) + r_\alpha(Q) & \text{if } \alpha \notin L \end{cases}$

The apparent rate will be *undefined* for component expressions containing *unguarded* variables, i.e. variables which are not prefixed by an activity. Consequently we do not allow a component to be defined by such an expression.

Note that in cases of cooperation, the apparent rate of the shared activity will be the minimum of the apparent rates of the components involved, where

$$m\top < n\top : \text{for } m < n \text{ and } m, n \in \mathbb{Q}$$
$$r < n\top : \text{for all } r \in \mathbb{R}, n \in \mathbb{Q}$$
$$m\top + n\top = (m + n)\top : m, n \in \mathbb{Q}$$
$$\frac{m\top}{n\top} = \frac{m}{n} : m, n \in \mathbb{Q}$$

In the case of a passive action it is assumed that the corresponding component does not contribute at all to the work required to complete the shared activity.

2.2 Derivation of ODEs

The semantics of a PEPA model is given by repeatedly applying the rules of Figure 1 to form a *labelled transition system* (usually termed the *derivation graph* in PEPA). When a Markovian interpretation is put on the PEPA model the duration of each activity is assumed to be a random variable governed by a negative exponential distribution. In this case the derivation graph can be considered to be the state transition diagram of a continuous time Markov chain (CTMC). Thus one Markovian state is associated with each syntactic term. Performance analysis is then carried out in terms of the steady state probability distribution, or the transient probability distribution, which is extremely costly when the state space is large.

Hillston [8] introduced a novel interpretation of PEPA models, as Ordinary Differential Equations. We do not propose to repeat the method here, but to give an overview of the main points. Essentially the Hillston method produces an alterna-

tive semantics for PEPA which can be viewed as a continuous approximation of the model behaviour. The approximation is of the state space, and a crucial motivator is that this is done *without* generating the underlying CTMC. This circumvents the state space explosion problem, and makes investigation of large populations with state space $\gg 10^{10}$ possible. The alternative semantics provides a good approximation of the CTMC semantics (as shown in [8] by comparing the ODEs to the steady state likelihoods of the components).

The method is focussed on systems having a large number of repeated components (since this is where much improvement can be gained in state space calculation). Instead of viewing each state in the derivation graph as a syntactic PEPA form, it is viewed as a numerical vector in which each component is represented by the number of instances of that component. This aggregation exploits strong equivalence in PEPA. The numerical vector form for an arbitrary PEPA model is defined as follows.

Definition 2.2 [Numerical Vector Form] For an arbitrary PEPA model $\mathcal{M}$ with n component types $\mathcal{C}_i, i = 1, \ldots, n$, each with N_i distinct derivatives, the *numerical vector form* of $\mathcal{M}$, $\mathcal{V}(\mathcal{M})$, is a vector with $N = \sum_{i=1}^{n} N_i$ entries. The entry v_{i_j} records how many instances of the jth local derivative of component type $\mathcal{C}_i$ are exhibited in the current state.

For example, the model of Figure 2 begins with the numerical state vector $((990, 0, 10, 0), (10, 990))$ where the first subvector corresponds to numbers of the local derivatives of component $S(SC, I$ and $R)$, and the second subvector corresponds to numbers of the local derivatives of component $Transmitter(Dormant)$.

The evolution of this state may be regarded as a continuous process represented by coupled ODEs. For example, for component $\mathcal{C}_{i_j}$ (the *j*th local derivative of component C_i), the change in number of $\mathcal{C}_{i_j}$ (denoted $v_{i_j}(t)$) is given by:

$$
\frac{dv_{i_j}(t)}{dt} = - \sum_{(\alpha,r) \in Ex(\mathcal{C}_{i_j})} r \times \min_{\mathcal{C}_{k_l} \in Ex(\alpha,r)} (v_{k_l}(t))
$$
$$
+ \sum_{(\alpha,r) \in En(\mathcal{C}_{i_j})} r \times \min_{\mathcal{C}_{k_l} \in Ex(\alpha,r)} (v_{k_l}(t))
$$

where $Ex(\mathcal{C}_{i_j})$ is the set of activities which are *exit* activities from a component, i.e. (α, r) such that there is a transition $\mathcal{C}_{i_j} \xrightarrow{(\alpha,r)} \mathcal{C}_{i_k}$, for some k. $Ex(\alpha, r)$ denotes the corresponding set of local derivatives $\mathcal{C}_{i_k}$. Exit activities result in a *decrease* in the number of component $\mathcal{C}_{i_j}$. $En(\mathcal{C}_{i_k})$ is the set of actions for which β is an *entry* activity, i.e. (β, r) such that there is a transition $\mathcal{C}_{i_j} \xrightarrow{(\beta,r)} \mathcal{C}_{i_k}$, for some j. Entry activities result in an *increase* in the number of instances of component $\mathcal{C}_{i_k}$.

3 SIR models

Kermack and McKendrick [11] introduced the classic SIR model of disease spread which has been widely used since then. The model is described by three coupled

ordinary differential equations. The population is split into three groups:

Susceptibles have never had the disease, and may contract it after exposure.

Infecteds have the disease and can pass it on to susceptibles.

Recovereds have previously had the disease and are assumed to be immune to future infection.

This basic pattern can be easily modified, for example, to allow recovered individuals to lose immunity and be recycled to the susceptible pool (SIRS model). The models presented here are both SIRS models.

The most commonly used formulation of the SIRS model is:

$$
\begin{aligned}
\frac{dS}{dt} &= -\beta SI/N + \alpha R \ , \\
\frac{dI}{dt} &= -\gamma I + \beta SI/N \ , \\
\frac{dR}{dt} &= \gamma I - \alpha R \ ,
\end{aligned}
\tag{1}
$$

where β is the rate of infection having been contacted, γ is the rate of recovery from infection and becoming immune, α is the rate of losing immunity and becoming susceptible again, and $N = S + I + R$.

Frequency dependent transmission of the disease is used here, since Norman and Shankland [14] showed that this arises most naturally from process algebra models. Frequency dependent transmission indicates that the number of contacts made by a single individual remains fixed, regardless of population size.

In previous work, McCaig [12] has used WSCCS (Weighted Synchronous Calculus of Communicating Systems) [17] to describe a variety of SIR models and to analyse the Mean Field Equations (MFEs) obtained. As a first step in comparing WSCCS and PEPA for epidemiology, we look at a fundamental aspect of transmission mechanism. Two particular variants are of interest: direct transmission and indirect transmission. McCaig proposed in his thesis [12] that, from a process algebra point of view, all forms of transmission can be reduced to one of these two types. By considering examples of both types of transmission we have some confidence that PEPA is generally applicable in epidemiology.

Our models take the form *Individuals* $\bowtie_L$ *Infrastructure*, where the *Individuals* are independently operating agents representing the individuals of the system (e.g. the susceptible, the infected and recovered individuals) and the *Infrastructure* models the communication mechanisms between individuals, whether direct or indirect.

3.1 Direct Transmission

Direct transmission requires physical contact between susceptible and infected individuals. The PEPA model for direct transmission can be seen in Figure 2. The basic agents are S, I and R, representing *Susceptible*, *Infected* and *Recovered* individuals respectively. SC represents an aspect of a susceptible individual: one who has been contacted by an infected individual. Such contact may not automatically lead to infection (e.g. the immune system fights off the disease with probability p_i at rate ir). These agents all act independently, with no communication between

$$S \stackrel{def}{=} (contact, \top).SC$$

$$SC \stackrel{def}{=} (infected, p_i * ir).I + (no\_infection, (1 - p_i) * ir).S$$

$$I \stackrel{def}{=} (contact, \top).I + (recover, rr).R$$

$$R \stackrel{def}{=} (contact, \top).R + (lose\_immunity, li).S$$

$$Transmitter \stackrel{def}{=} (contact, cr).Transmitter + (recover, \top).Dormant$$

$$Dormant \stackrel{def}{=} (infected, \top).Transmitter$$

$$S[990] \parallel I[10] \underset{\{infected, contact, recover\}}{\bowtie} Transmitter[10] \parallel Dormant[990]$$

Fig. 2. Direct transmission

them (indicated by use of the $\parallel$ operator).

So how does infection happen? The *Transmitter* agent represents an aspect of the *I* agent: the portion which is able to pass on the infection. Once activated, the *Transmitter* repeatedly carries out the *contact* action at rate *cr*. The number of *Transmitter* agents is equal to the number of *I* agents. When an *I* agent recovers (at rate *rr*), a *Transmitter* becomes *Dormant*.

For biological realism all of the *S*, *I* and *R* are available to contact, but only the *S* agents may be changed by this contact. If an *I* agent contacts an *R* agent (via *Transmitter*), then no new infection results.

This model is designed to be as near equivalent as possible to the basic SIR model presented in McCaig's thesis [12, p47, Figure 3.1], except for the feature of losing immunity at rate *li* introduced here for more biological flexibility. Setting $li = 0$ makes the models equivalent. We also set $p_i = 1$ so the *SC* state acts as a delay. Note that this does not guarantee infection since all the *I* may recover before all the *S* get the disease.

As it stands, the Hillston method cannot be used to derive ODEs for this model due to the use of passive rates. This problem is easily overcome by setting all passive rates to the matching local rate (e.g. all actions $(contact, \top)$ become $(contact, cr)$); however, this is unsatisfactory since passive rates are a natural way to express our model. Biologically, individuals are contacted by infected individuals whether they like it or not. Following simplification to remove *min* terms and replace *Transmitter* with the equivalent *I*, the derived ODEs for this revised model are:

$$dS/dt = -cr.S + (1 - p_i).ir.SC + li.R$$
$$dSC/dt = cr.S - SC$$
$$dI/dt = p_i.ir.SC - rr.I$$
$$dR/dt = rr.I - li.R \tag{2}$$

From our experience with mathematical models of disease these equations are clearly seen to be rather different from expected (e.g. Eqn. (1)). Moreover, the ODEs do not provide a close approximation for the transient dynamics of the system. In epidemiology, transient dynamics are of more interest than steady state, since we are concerned with whether an epidemic happens, how fast it happens, where it peaks, and so on. Figure 3 compares the ODE for *I* only to the results
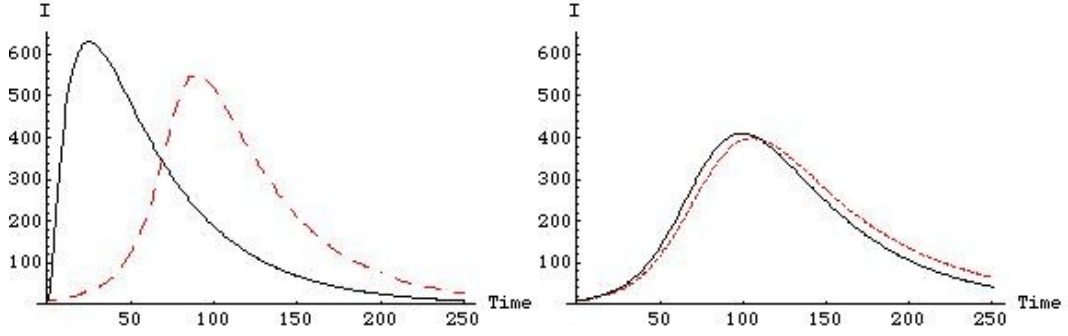
Fig. 3. Direct Transmission. On the left: PEPA ODE (solid line), SSA with local rates (dashed line). On the right: SSA with passive rates (dotted line), SIRS ODE (solid line). Parameter values: $cr = 0.08, ir = 1, li = 0, p_i = 1, rr = 0.02$. RMS = 269 (left) 25 (right).

of stochastic simulation (PEPA Eclipse plugin [6], mean of 500 simulations using Gillespie's algorithm [5]). The graph on the left compares the ODE with simulations at the local rate. The match is clearly not good, shown by the Root Mean Square (RMS) value. The graph on the right shows our goal in the form of Eqn. (1) plotted with simulations at the passive rate. These are, as expected, rather different from the ODE for the local rate. Note that changing the passive rate to a local rate speeds up the infection (the simulation peak on the left is higher and earlier than that on the right).

We move on to indirect transmission.

3.2 Indirect Transmission

Indirect transmission requires no physical contact between susceptible and infected individuals; the disease is passed via an infected vector (e.g. mosquitos, ticks, rats) or via an infected touch-surface (e.g. door handles, shared latrines). The PEPA model for indirect transmission can be seen in Figure 4. This model has been developed to be equivalent to ODEs derived from the WSSCS model of McCaig [12, p142, Figure 6.1]. This is a model in which the amount of environment to be infected is finite and infection in the environment decays probabilistically.

Indirect transmission is stylistically a more natural expression of transmission for PEPA: the agents on the right hand side of the $\bowtie$ operator represent infected

$$S \stackrel{def}{=} (contact, \top).SC$$

$$SC \stackrel{def}{=} (infected, p_i * ir).I + (no\_infection, (1 - p_i) * ir).S$$

$$I \stackrel{def}{=} (infect\_env, ier).I + (contact, \top).I + (recover, rr).R$$

$$R \stackrel{def}{=} (contact, \top).R + (lose\_immunity, lir).S$$

$$InfEnv \stackrel{def}{=} (contact, cr).InfEnv + (decay, dr).Env$$

$$Env \stackrel{def}{=} (infect\_env, \top).InfEnv$$

$$S[990] \parallel I[10] \underset{\{contact, infect\_env\}}{\bowtie} Env[10000]$$
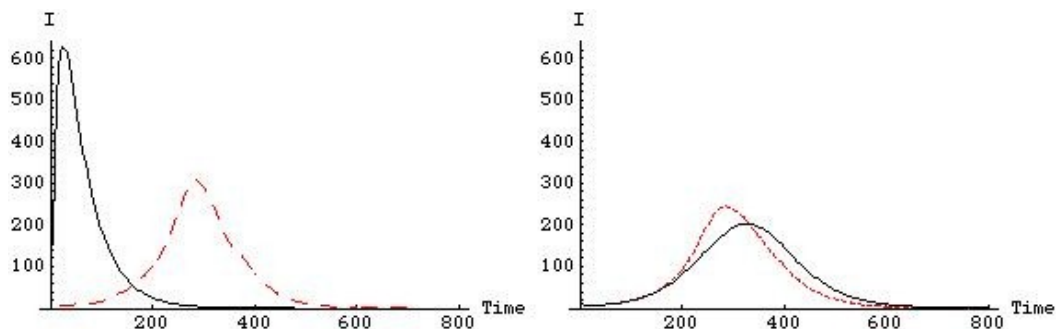
Fig. 4. Indirect transmission

Fig. 5. Indirect Transmission. On the left: PEPA ODE (solid line), SSA with local rates (dashed line). On the right: SSA with passive rates (dotted line), WSCCS ODE (solid line). Parameter values: $cr = 0.08, dr = 0.005, ier = 0.01, ir = 1, li = 0, p_i = 1, rr = 0.02$. RMS = 165 (left) 23 (right). Note longer time scale than direct transmission.

and non-infected *environment* (*InfEnv* and *Env*) rather than different aspects of the I agents. Agents S, I and R come into contact with the environment, potentially resulting in infection.

The derived ODEs (making the same modification to passive rates and simplifying as before) are:

$$
\begin{aligned}
dS/dt &= -cr.S + (1 - p_i).ir.SC + li.R \\
dSC/dt &= cr.S - SC \\
dI/dt &= p_i.ir.SC - rr.I \\
dR/dt &= rr.I - li.R \\
dInfEnv/dt &= ier.Env - dr.InfEnv \\
dEnv/dt &= -ier.Env + dr.InfEnv
\end{aligned}
\tag{3}
$$

It is clear something is wrong, because the first four equations above are exactly the same as Eqn. (2): the amount of infected environment is not used in calculating the number of new infections. Simulation results are shown in Figure 5. On the left a significant difference is seen between the local rate simulation results and the ODE for I from Eqn. (3). On the right, the passive rate simulation results are compared to ODEs derived from our WSCCS models.

Clearly the models presented here do not fit the pattern expected by the Hillston method (which works well for models of computer systems). In the following Section we propose a modification to the method, customised for epidemiological models.

4 Modified Derivation of ODEs: the Stirling amendment

The Hillston method does not give good results for our epidemiological models. Revisiting the direct transmission example above one reason might be that self-loops are ignored. In the context of a single agent this makes sense: if I is contacted, it evolves to agent I so there is no overall change in number of I agents. In the context of the whole system this does not make sense: I and R absorb contact actions, meaning there are fewer actions for S (and therefore less infection than indicated by the ODEs of Eqn. (2)). This latter point illustrates another difference:

64

there is an implicit choice between independent but distinct components offering the same action for synchronisation, i.e. a *Transmitter* agent may *contact* any one of a number of S, I and R agents. Similarly, in indirect transmission the self-looping of *InfEnv* determines the number of *contact* actions and is therefore important to the system dynamics. The impact of activities on components must be related to the overall apparent rate of an action in the system, and to the proportion of that component in the whole population. For example, in direct transmission the overall rate of *contact* is determined by the slowest component since S, I and R are passive. This yields *cr.Transmitter*. This activity must be "shared out" amongst the S, I and R, otherwise we are making the biologically implausible assumption that somehow all contacts get channelled to the susceptible individuals only.

This Section presents two modifications to the Hillston method [8], yielding a closer match with stochastic simulation results for passive rates. The first modification is to add more fine-grained classification of actions in the activity matrix by adding information about self loops. As a consequence, information about "side" is required because to calculate the rate correctly we need to know which agents are collaborating and which are competing. The second modification is to use the standard apparent rate calculation thereby removing the restriction on passive rates. We assume that competing components share the same rate (here they are all passive).

Definition 4.1 The Stirling amendment deals with systems of the form $P_L \bowtie_L P_R$.

Hillston introduced the *activity matrix*, representing exit and entry activities as -1 and +1. In the activity matrix each row corresponds to a single local derivative. In the representation of the model as a system of ODEs there is one equation for each state variable, i.e. for the current number of each local derivative exhibited. This equation details the impact of the rest of the system on the value of that state variable. This can be derived automatically from the activity matrix when we associate a state variable n_i with each row of the matrix and a rate constant r_j with each column of the matrix. The number of terms in the ODEs will be equal to the number of non-zero entries in the corresponding row, each term being based on the rate of the activity associated with that column. The Stirling amendment adds the notions of *side* and *self loops* to the activity matrix.

Definition 4.2 [Activity Matrix] For a model with N_A activities and N_D distinct local derivatives, the *activity matrix* M_a is an $N_D \times N_A$ matrix, and the entries are defined as follows.

$$(d_i, a_j) = \begin{cases} +1_L & \text{if } a_j \text{ is an entry activity of } d_i, d_i \in P_L \\ -1_L & \text{if } a_j \text{ is an exit activity of } d_i, d_i \in P_L \\ +1_R & \text{if } a_j \text{ is an entry activity of } d_i, d_i \in P_R \\ -1_R & \text{if } a_j \text{ is an exit activity of } d_i, d_i \in P_R \\ SL_L & \text{if } a_j \text{ is a self loop of } d_i, d_i \in P_L \\ SL_R & \text{if } a_j \text{ is a self loop of } d_i, d_i \in P_R \\ 0 & \text{otherwise.} \end{cases}$$

```
//Form one ODE for each local derivative/state variable
For  i = 1...N_D
   //Find the activities involving this derivative
     For  j = 1....N_A
      If  M_a(i, j) ≠ 0
       //Form exit sets and self loop sets for activity j
         Ex_L(j) = ∅;  Ex_R(j) = ∅
         Loop_L(j) = ∅;  Loop_R(j) = ∅
          For  k = 1...N_D
            If  M_a(k, j) = −1_S
              Ex_S(j) = Ex_S(j) ∪ {k}
            If  M_a(k, j) = −SL_S
              Loop_S(j) = Loop_S(j) ∪ {k}
       //Record the impact of each such activity
      If  M_a(i, j) = +1_S
        Add
```
$$+r_j(Sys) \times n_z(t)/ \sum_{k \in Ex_S(j) \cup Loop_S(j)} n_k(t)$$
```
        to the equation where n_z ──(j,r)──→ n_i.
      If  M_a(i, j) = −1_S
        Add
```
$$-r_j(Sys) \times n_i(t)/ \sum_{k \in Ex_S(j) \cup Loop_S(j)} n_k(t)$$
```
      to the equation
```

Fig. 6. Pseudo-code for generating the set of ODEs

In addition, the rate of activity is calculated differently. Hillston uses the local rate (and these were all assumed to be the same). The new rate here is the overall apparent rate $r_\alpha(Sys)$ multiplied by *component contribution*. For row d_i, if the matrix entry subscript is L (say) then the contribution of d_i to the activity is $n_i(t)/\sum_{k \in Ex_L(j) \cup Loop_L(j)} n_k(t)$. That is, the sum of all components tagged -1_L or SL_L in that column. This change is specified in the amended algorithm of Figure 6. As before there will be one ODE in the system for each row of the matrix. SL entries do not add terms to the ODEs.

Revisiting the model of Sections 3.1, the modified equations for direct transmission are:

$$dS/dt = -\frac{cr.S.I}{S+I+R} + (1-p_i).ir.SC + li.R$$

$$dSC/dt = \frac{cr.S.I}{S+I+R} - SC$$

$$dI/dt = p_i.ir.SC - rr.I$$
$$dR/dt = rr.I - li.R \tag{4}$$

Note that only the transmission term has changed. The apparent rate calculation for *contact* results in $min(cr.Transmitter, \top(S+I+R))$ which is equal to $cr.I$ since $N(Transmitter, t) = N(I, t)$.

For indirect transmission (Section 3.2) the Stirling amendment produces the following equations:

$$dS/dt = -\frac{cr.S.InfEnv}{S+I+R} + (1-p_i).ir.SC + li.R$$

$$dSC/dt = \frac{cr.S.InfEnv}{S+I+R} - SC$$

$$dI/dt = p_i.ir.SC - rr.I$$

$$dR/dt = rr.I - li.R$$

$$dInfEnv/dt = ier.I - dr.InfEnv$$

$$dEnv/dt = -ier.I + dr.InfEnv \tag{5}$$

The apparent rate calculation for *contact* results in $min(ier.I, \top.Env)$. We assume this results in $ier.I$ since $N(Env, 0)$ is chosen so that $\forall t.N(Env, t) > 0$.

The graphs of Figure 7 and Figure 8 show the Stirling amendment ODEs plotted against passive rate simulation results. Figure 7 additionally shows the standard ODE (1) for I while Figure 8 shows the ODE derived from our WSCCS models. It can be seen that there is now an excellent match with the other ODE (identical for indirect transmission), and a better match to the stochastic simulation results as shown by the RMS value. The match is rather poorer for indirect transmission
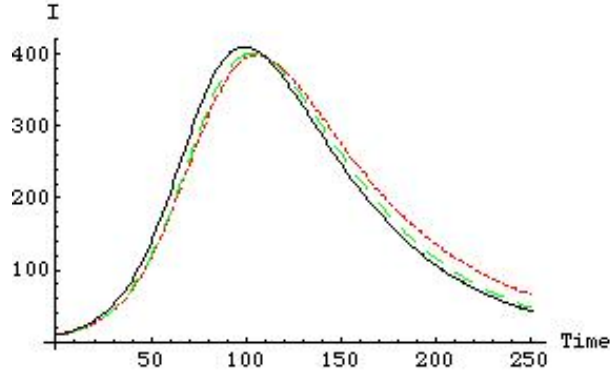

Fig. 7. Direct Transmission: SSA with passive rates (dotted line), SIRS ODE (solid line), new PEPA ODE (dashed line). Parameters as before. RMS = 13.
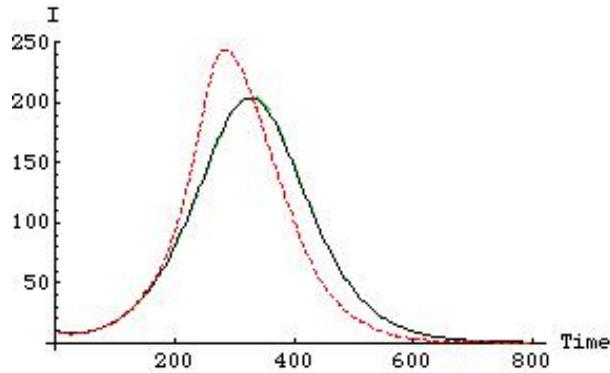

Fig. 8. Indirect Transmission: SSA with passive rates (dotted line), WSCCS ODE (solid line), new PEPA ODE (dashed line). Parameters as before. RMS = 23.

67

than for direct transmission.

For indirect transmission, an important feature of the WSCCS models was whether a finite or infinite environment is used. In the PEPA model we have a finite environment; however, we have set the number of environment components so high as to be effectively infinite (for this number of S, I and R). The equations obtained provide a good match with both the finite and infinite environment equations of McCaig [12, p141 and p152], and syntactically are closest to those for infinite environment.

5   Discussion

We have presented two models of disease spread in PEPA. These follow the main classification of transmission into direct and indirect transmission. The first observation is that it is possible to model these different mechanisms at all in PEPA. The second observation is that indirect transmission is slightly more naturally expressed in PEPA. Use of PEPA provides considerable advantages over use of WSSCS in terms of tool support, links to other formalisms, and active language developments for biology.

ODE derivation from these models highlighted limitations of the Hillston method, but it should be noted that the Hillston method was developed for models of computer systems and not for models of disease spread. In the context of epidemiology we presented a modification of this method, based on our experience with WSCCS. The resulting ODEs provide a better fit to the results of stochastic simulation. Current work is underway to fine tune the method, and to establish the precise relationship between the Hillston method and the Stirling amendment. Even if shown to be more general than the Hillston method, the method presented here does not apply to all PEPA models. For example, hidden actions are not considered.

Cardelli [4] has also developed a method of deriving ODEs from process algebra models. The motivation there is to provide a process algebra semantics matching the mass action behaviour of chemical systems. This therefore presents a new semantics which is not related to the usual synchronisation semantics of process algebras. In our work it is important to be able to move between the scales of individual and population in such a way that the transient dynamics of the system are preserved (i.e. that the ODE approximation matches the CTMC semantics).

Use of process algebra for epidemiology provides new insight into disease spread because it is possible to rigorously justify the population level ODEs in terms of observed individual behaviour. More work in this area remains to be carried out. This is planned as part of the System Dynamics project mentioned below. It is also planned to apply the knowledge gained in modelling biological diseases back to computer systems. An obvious candidate for such modelling is the spread of computer viruses and other malware.

Acknowledgement

References

[1] H. Bowman, J. Bryans, and J. Derrick. Analysis of a multimedia stream using stochastic process algebra. In C. Priami, editor, *Sixth International Workshop on Process Algebras and Performance Modelling*, pages 51–69, Nice, September 1998.

[2] Jeremy T Bradley, Nicholas J Dingle, Stephen T Gilmore, and William J Knottenbelt. Derivation of passage-time densities in PEPA models using ipc: the Imperial PEPA Compiler. In Gabriele Kotsis, editor, *Proc. of the 11th IEEE/ACM Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, pages 344–351, University of Central Florida, October 2003. IEEE Computer Society Press.

[3] M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. In *Proceedings of BioCONCUR 2004*, Electronic Notes in Theoretical Computer Science. Elsevier, 2004.

[4] L. Cardelli. On process rate semantics. *Theoretical Computer Science*, 391:190–215, 2008.

[5] Daniel T. Gillespie. Exact Stochastic Simulation of Coupled Chemical Reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[6] S. Gilmore, M. Tribastone, A. Duguid, and A. Clark. Pepa plug-in for eclipse. homepages.inf.ed.ac.uk/mtribast/plugin/, 2008.

[7] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[8] J. Hillston. Fluid Flow Approximation of PEPA models. In *QEST'05, Proceedings of the 2nd International Conference on Quantitative Evaluation of Systems*, pages 33–42. IEEE Computer Society Press, Torino, September 2005.

[9] J. Hillston. Tuning systems: From composition to performance. *The Computer Journal*, 48(4):385–400, May 2005. The Needham Lecture paper.

[10] Jane Hillston, Leïla Kloul, and Amdjed Mokhtari. Towards a feasible active networking scenario. *Telecommunication Systems*, 27(2–4):413–438, October 2004.

[11] W.O. Kermack and A.G. McKendrick. Contributions to the mathematical theory of epidemics. *Proceedings of the Royal Society of London A*, 115:700–721, 1927.

[12] C. McCaig. *From individuals to populations: changing scale in process algebra models of biological systems*. PhD thesis, University of Stirling, 2008. Available from www.cs.stir.ac.uk/~cmc/thesis.ps.

[13] C. McCaig, R. Norman, and C. Shankland. An algorithm for deriving mean field equations from large process algebra models. Technical Report 175, University of Stirling, 2008.

[14] R. Norman and C. Shankland. Developing the use of process algebra in the derivation and analysis of mathematical models of infectious disease. In *Computer Aided Systems Theory - EUROCAST 2003*, volume 2809 of *Lecture Notes in Computer Science*, pages 404–414. Springer-Verlag, 2003.

[15] A. Regev, E.M. Panina, W. Silverman, L. Cardelli, and E. Shapiro. Bioambients: an abstraction for biological compartments. *Theoretical Computer Science*, 325:141–167, 2004.

[16] Nigel Thomas, Jeremy T Bradley, and William J Knottenbelt. Stochastic analysis of scheduling strategies in a grid-based resource model. *IEE Software Engineering*, 151(5):232–239, September 2004.

[17] C. Tofts. Processes with probabilities, priority and time. *Formal Aspects of Computing*, 6:536–564, 1994.