From Individuals to Populations: a mean field semantics for process algebra

Chris McCaig, Rachel Norman, Carron Shankland

Department of Computing Science and Mathematics, University of Stirling, Stirling, FK9 4LA

Abstract

A new semantics in terms of Mean Field Equations is presented for WSCCS (Weighted Synchronous Calculus of Communicating Systems). The semantics captures the average behaviour of the system over time, but without computing the entire state space, therefore avoiding the state space explosion problem. This allows easy investigation of models with large numbers of components. The new semantics is shown to be equivalent to the standard Discrete Time Markov Chain semantics of WSCCS as the number of processes tends to infinity. The method of deriving the semantics is illustrated with examples drawn from biology and from computing.

Keywords: Process Algebra, Mean Field Equations, Semantics

1. Introduction

The collected evolution of a group of individuals through time is of significance in many fields. Of particular interest is the emergent behaviour of a whole population given a description of the low level interactions of individuals. For example, in epidemiology, the focus is on the number of infected individuals in the population and how a small number of initial infections can lead to a large epidemic. In biochemistry, fluctuating concentrations of molecules form intra- and inter-cellular signals. In computing science, networks are formed by clients and servers, and how information flows through that network is of importance. Process algebra [2] provides a convenient way to describe such individual-based models, and to obtain the overall population behaviour.

While simply modelling a system can lead to deeper understanding of it, usually some additional forms of analyses are desirable to capitalise on the description. Simulation of the model, for example, provides one view of overall system behaviour but a single simulation gives only one route through the state space. Given a large population, many different behaviours are possible, with

Email addresses: cmc@maths.stir.ac.uk (Chris McCaig), ran@maths.stir.ac.uk (Rachel Norman), ces@cs.stir.ac.uk (Carron Shankland)

significantly different outcomes. To capture the overall system behaviour we may consider the *average* behaviour of the system. Calculating this as the average of all simulations is clearly computationally expensive. An alternative is to calculate the *steady state* of a system. This ignores potentially interesting transient dynamics, and is again computationally expensive. To avoid state space computation, we may turn to Ordinary Differential Equations (ODEs). Mathematical biologists have for many years used ODEs to capture important transient dynamics of systems, e.g. [1, 15]. Historically, this approach also has drawbacks: assumptions must be made about how the continuous population-based dynamics of the system emerge from discrete individual interactions.

This paper presents a novel method of combining the benefits of an individual-based modelling approach with those of a population-based modelling approach. We give an alternative, but equivalent, semantics for the process algebra WSCCS [30] in terms of Mean Field Equations (MFEs). This approach gives a deterministic approximation of the discrete time, discrete space emergent behaviour of WSCCS models in terms of discrete time, continuous space MFEs. In addition to providing an alternative view of the system, a further benefit is the wide range of algebraic and numerical analyses available for MFEs.

Although originally inspired by epidemiological modelling, this work has wider applicability. Any system built in a bottom-up way from replicated individual components is amenable to the technique. In computing science terms, a new semantics has been developed for an existing language which facilitates exploration of systems with large numbers of components.

Related Work

Our initial studies [25] were inspired by Sumpter's derivation of mean field equations from WSCCS descriptions [27, 28] (using an informal heuristic). His main application area is social insect behaviour, but his thesis [27] includes a simple epidemic model and a simple population growth model. The current paper is based on the thesis of McCaig [19] and the related technical report [22].

Independently, other authors have developed ODE semantics for other process algebras: Calder et al. [8], Hillston [13] and Cardelli [9]. Our work tackles a different technical problem from those in two ways. Firstly, the process algebras used by Calder et al. [8], Hillston [13] and Cardelli [9] are continuous time. This means the rate of moving from one state to another is already available in the transition system, whereas here the transition rate must be derived from probabilistic transitions. In terms of expressiveness of modelling we find probabilistic choice (as in WSCCS) a more natural way to express individual behaviour than rates of activity (as in, e.g. PEPA [13]). Secondly, our focus has been on extracting the average behaviour of the existing Markov chain semantics of WSCCS. Hillston [13] also takes this approach, although some strong restrictions on the form of the model are imposed, making the method unsuitable for epidemiological models. Our recent joint work [5] removes those restrictions. Cardelli [9] and Calder et al. [8] make a mass action assumption about interaction. This yields a different semantics to the standard process algebra semantics, which is based largely on one to one communication. This difference seems to have

arisen from the application area: while we wish to mechanistically translate the behaviour of individuals in epidemics to system dynamics, they are concerned with biochemical applications in which the mass action assumption of interaction is appropriate. In Cardelli's work in particular, moving between discrete and continuous state requires translation via a volumetric factor γ. In our setting, this would be similar to dividing all rates by the total population size. While this is sometimes appropriate for describing interaction rates in epidemiology, there are some cases where it is incorrect (see Section 3.5 and our paper on transmission rates [20]).

Tangentially related is the work of Brodo et al. [7] who derive numeric rate information for π-calculus models. Their work is concerned with performance analysis, and allowing loose initial specification of a system, with greater refinement as more information becomes available. Relevant information includes network topology, throughput, latency, and protocol complexity. While it would be interesting to use their system to look at epidemiological models and to derive transition rates of components top-down from observation of the system as a whole, our goal is to work bottom-up: to use the observed individual behaviour to derive the behaviour of the system as a whole.

Outline of the Paper

The paper is structured as follows. Section 2 presents an overview of WSCCS and gives additional motivation for calculating an approximation of the average system dynamics. The formal definition of the algorithm to translate WSCCS models into MFEs is given in Section 3. The method is not suitable for all WSCCS models: restrictions to the applicability of the algorithm are presented in Section 3.1. It is not our aim to simply present an alternative semantics for WSCCS; the equivalence between the standard WSCCS semantics and the new MFE semantics is given in Section 3.6. Section 4 presents two worked examples of the approach, one drawn from biology and the other from computing. We conclude with some comments about useful applications of the approach.

2. Process Algebra

The particular process algebra used here is WSCCS *Weighted Synchronous Calculus of Communicating Systems* [30]. WSCCS is a discrete time process algebra, with synchronous activity. WSCCS has a distinguished record in use for biological applications (insect behaviour [27, 29], genetic evolution [10, 12], epidemiology [25, 27], immunology [24]). Its semantics are well suited to epidemiology, in which observations of individuals in the population are taken at discrete time points and probabilistic choice is a natural way to express alternative courses of action. The results presented here could be translated to any similar process algebra [2].

2.1. WSCCS Syntax and Semantics

WSCCS was developed by Tofts [30]. In WSCCS the basic components are *actions* ($a \in Act$) and the *processes* (or *agents*, $A \in \mathcal{A}$) that carry out those actions. The actions are chosen by the modeller to represent activities in the system. For example, *infect*, *send*, *receive*, *throw dice*, and so on. The formal syntax and semantics of WSCCS is presented in Tofts [30]. The main details are repeated here in Appendix A for the convenience of the reader. To illustrate the language, a simple epidemiological example is given in Fig. 1. The example comes from McCaig's thesis [19] and is based on the same assumptions as the classical ODE model of an epidemic by Kermack and McKendrick [15].

The population is divided into three groups: susceptibles (agents $S1$, $S2$ and $SI2$) have never had the disease, infecteds (agents $I1$, $I2$ and $Trans1$) currently have the disease and can pass it on to others, and recovereds (agents $R1$ and $R2$) have previously had the disease and are immune to future infection. WSCCS is a synchronous calculus, therefore all agents change state together. A convenient modelling discipline is to conceive the model in a number of *stages* and to give appropriate agent names indicating stage. For example, the model of Fig. 1 has two stages, and agent names include the label 1 and 2. Adherence to this convention improves readability, but failure to do so does not affect the ability to derive mean field equations.

$$
\begin{aligned}
S1 &\stackrel{\text{def}}{=} \omega.\mathit{infect} : SI2 + 1.\surd : S2 \\
I1 &\stackrel{\text{def}}{=} \omega.\mathit{infect} : I2 + 1.\surd : I2 \\
\mathit{Trans}1 &\stackrel{\text{def}}{=} \omega.\overline{\mathit{infect}} : 0 + 1.\surd : 0 \\
R1 &\stackrel{\text{def}}{=} \omega.\mathit{infect} : R2 + 1.\surd : R2 \\
\\
S2 &\stackrel{\text{def}}{=} 1.\surd : S1 \\
SI2 &\stackrel{\text{def}}{=} p_i.\surd : (I1 \times \mathit{Trans}1) + (1 - p_i).\surd : S1 \\
I2 &\stackrel{\text{def}}{=} p_r.\surd : R1 + (1 - p_r).\surd : (I1 \times \mathit{Trans}1) \\
R2 &\stackrel{\text{def}}{=} 1.\surd : R1 \\
\\
\mathit{Population} &\stackrel{\text{def}}{=} \Theta((S1\{s\} \times I1\{i\} \times \mathit{Trans}1\{i\} \times R1\{r\}) \lceil \{\surd\})
\end{aligned}
$$

Figure 1: Epidemiology model with contact followed by probabilistic choice

The initial state of the model of Fig. 1 comprises s $S1$, i $I1$, i $Trans1$ and r $R1$ individuals in parallel (where $\times$ indicates parallelism and $\{n\}$ indicates n copies of an agent). The first stage of the model (with agents labelled 1) is the contact stage. *Trans*1 agents represent an aspect of behaviour of infected agents. They may communicate with other agents (through the action *infect*) to try to pass on the disease. All other agents, including infected agents and recovered agents may

be communicated with in this way. This does mean that an individual $I1$ agent may interact with the *Trans*1 agent that represents the input behaviour of the same individual (i.e. individuals can attempt to infect themselves). Traditional ODE models (e.g. [3, 15], which this model seeks to emulate) feature terms to capture transmission based on the reasonable simplification that infected individuals can contact any member of the population (including themselves). The rationale behind this modelling choice is that for large populations (the only situation in which the mean field approximation offered by ODEs is useful) the difference between $I/(S + I + R)$ and $I/(S + I - 1 + R)$ is very small.

Processes synchronise pairwise on the action *infect* and its complement $\overline{infect}$. The *infect* action cannot happen without synchronising with $\overline{infect}$: this is enforced by the use of the restriction operator $\lceil\{\surd\}$. Similarly, $\overline{infect}$ cannot happen without synchronising with *infect*. We refer to actions such as *infect* and $\overline{infect}$ as *communicating actions* since they cannot proceed without synchronising with a complementary action (representing communication between agents).

The + operator indicates a weighted choice: transitions are selected with a probability corresponding to their weight divided by the weight of all possible transitions from that agent. In $S1$ the left hand option is weighted ω which is a special weight. In conjunction with the priority operator, Θ, ω weights become prioritised, meaning that if the *infect* action can happen, it must happen. Semantically, use of the Θ operator removes all non-priority choices. The : operator sequences an action and a process. For example, following an *infect* action, $S1$ becomes $SI2$. In stage two, a probabilistic decision is made in agents $SI2$ as to whether the infectious contact has resulted in infection or not. Agent $I2$ also makes a probabilistic decision to recover or not. Other agents simply mark time with the special action $\surd$. The agent 0 is the null agent, the process doing nothing.

The semantics of WSCCS is transition based, defining the actions that a process can perform and the weight with which a state can be reached. The operational rules of WSCCS are presented in Table A.7 of Appendix A. Importantly for this present work, the transition system can be interpreted as a Discrete Time Markov Chain (DTMC) under abstract bisimulation [30]. Abstract bisimulation essentially accumulates and normalises the weights associated with a choice, so that transitions are labelled by their action and probability of occurring.

2.2. Average Behaviour of WSCCS models

As mentioned in the Introduction, the transient dynamics of a system is of interest, i.e. the time series evolution of the model. While a single time series evolution of the model is revealing, the *average* behaviour may be more useful for further analysis.

To illustrate the idea, consider the transition diagram presented in Fig. 2. Here, the first two steps of evolution of the model is shown, with weights and actions combined under abstract bisimulation. Monte Carlo simulation can be

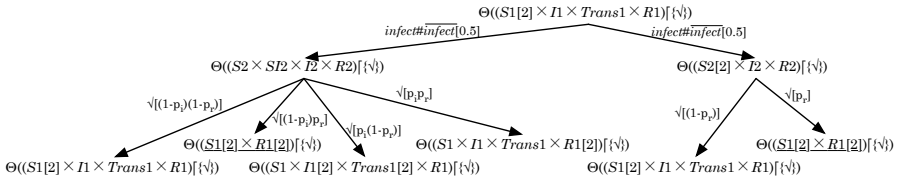

Figure 2: Evolution of the Simple SIR model. Deadlocked states are underlined.

used to generate a single path through the state space; however, more simulations are required to explore the whole state space and to calculate the average behaviour. For example, given values $p_r = 0.1$ and $p_i = 0.5$, multiple simulations would give the average state after two steps of $\Theta((S1\{1.75\} \times I1\{1.15\} \times Trans1\{1.15\} \times R1\{1.1\})\lceil\{\surd\}) $. Note that since this is a mean value, we have moved from the discrete state space of the WSCCS model into continuous state space.

While advances in dealing with the state space of larger processes have been made, we are still limited by the constraints of computer memory and time. Probabilistic workbench [31], the tool for WSCCS, can handle systems up to 500 components [32] but imposes restrictions on how these models can be interpreted. More generally, systems with state spaces of around 10^{12} is the current limitation (see [26] for a recent review of state space techniques for model checking). While this seems large, state space is exponential in the number of components, and in biological systems components can number in the millions.

Again considering the diagram of Fig. 2, another way to calculate the average state after two steps would be to multiply the probability of getting to each state by the component numbers in that state. For example, given the values $p_r = 0.1$ and $p_i = 0.5$, the probability of getting to $\Theta((S1 \times I1\{2\} \times Trans1\{2\} \times R1)\lceil\{\surd\})$ is 0.225. The contribution of this state to the average number of $I1$ agents is therefore 2*0.225. Calculating the state space piecewise in order to work out the average state is also expensive, but a similar result can be achieved by examining the syntax of the model. Fig. 1 holds all the information required about state evolution, given the semantics of WSCCS. We propose algebraic calculation of the average time series behaviour via Mean Field Equations (MFEs). The terms of the MFE capture the calculation above: the probability of getting to each state multiplied by the number of components in that state.

2.3. State Evolution and State Vector

For any model, the transition system may be viewed as the evolution of the initial state $(A1\{n_1\}, \ldots, Am\{n_m\})$ through time, where A_i are WSCCS agents. We denote the number of agents Ai at time t by Ai_t. For example, consider the state space shown in Fig. 2. The initial state is

$$(S1\{2\}, I1\{1\}, Trans1\{1\}, R1\{1\}, S2\{0\}, SI2\{0\}, I2\{0\}, R2\{0\}) .$$

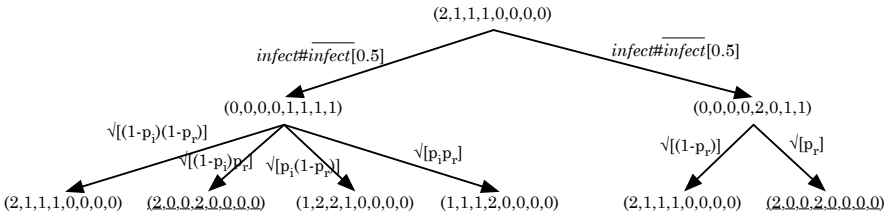

Figure 3: Evolution of the Simple SIR model: state vector

It is convenient, and more compact, to represent this as a numerical vector rather than as process algebra syntax

$$(2, 1, 1, 1, 0, 0, 0, 0) \ .$$

The state vector includes values for all possible agents. This representation identifies members of the same equivalence class under direct bisimulation [30]. (Direct bisimulation is a more discriminating relation than abstract bisimulation, mentioned earlier, because the numeric weights given to transitions are preserved and significant: a process $A = 1.a : A1 + 1.a : A2$ is not equivalent to $B = 2.a : B1 + 2.a : B2$ under direct bisimulation but A and B are equivalent under abstract bisimulation.) Fig. 3 shows the transition system of Fig. 2 in terms of state vectors.

Definition 1 (State Vector). *For an arbitrary WSCCS model $\mathcal{M}$ with m agent types $Ai, i = 1, \ldots, m$, the* numerical vector form *of $\mathcal{M}$, $\mathcal{V}(\mathcal{M}, t)$, is a vector with m entries. The entry $v_i(t)$ records how many instances of agent type Ai are exhibited at time t.*

Evolution of the system through time simply means changing the values in the state vector. Formally, for a source state $(A1_t, \ldots, Am_t)$ any one of a number of transitions $(A1_t, \ldots, Am_t) \xrightarrow{a[p]} (A1_{t+1}, \ldots, Am_{t+1})$ may be taken, with a range of values for a and p.

A transition has three possible effects on any particular Ai_t in the source state:

exit activity Following the transition, the process evolves to some other agent Aj therefore the number of Ai agents is decreased over time ($Ai_{t+1} < Ai_t$).

entry activity In symmetry with an exit activity for Ai above must be an entry activity for Aj. The number of Aj agents increases over time ($Aj_{t+1} > Aj_t$).

none The process becomes Ai and there is no change in number of Ai agents ($Ai_{t+1} = Ai_t$).

Since WSCCS is a synchronous calculus, in each time step, for every agent in the system, one of the above activities will occur. Note that sometimes there is a one to one relationship between Ai_t and Aj_{t+1} while in other cases a single exit activity for Ai_t results in entry activities for a set $\{Aj_{t+1}\}, j \in 1 \ldots m$. For example, in Fig. 1 when agent $S2$ evolves to agent $S1$ this is a one to one relationship, but when $SI2$ evolves (left hand choice), the number of $Trans1$ and $I1$ both increase.

Our goal is not directly to calculate state spaces in terms of the numerical state vector as in Fig. 3. Instead, we give the mean change from one state to another in terms of Mean Field Equations, i.e. the value of $v_i(t)$ is given as an expression in $A_1(t), \ldots, A_m(t)$. Effectively, this is done by constructing a symbolic expression capturing the product of the change of state and the probability of making that change, from any given state. The syntax of the model indicates the choices available at each point, but does not directly give the value of p, the probability of change, in the DTMC. To calculate the average state change in each step, we construct a transition table noting these exit and entry activities for all combinations of agents and actions.

3. Deriving Mean Field Equations

This section presents the method to translate WSCCS models of the form

$$Population \stackrel{\text{def}}{=} (A1\{n_1\} \times A2\{n_2\} \times \ldots \times Am\{n_m\}) \lceil \{L\}$$

(possibly with the priority operator Θ) into Mean Field Equations. As before, the Ai are all WSCCS agents and may include further parallelism. There are some restrictions on the form of the model; not all WSCCS models are suitable to be translated into MFEs. Restrictions are presented in Section 3.1.

The computational expense of generating the state space and/or simulation is avoided because the method is based entirely on manipulation of the syntax of a WSCCS model. The method described in this Section is $O(g^2c)$ where g is the total number of types of agents and c is the total number of distinct actions in the WSCCS description. There are a number of additional auxiliary functions involved in the computation, but these can all be calculated in a single pass of the WSCCS description.

3.1. Restrictions

The method presented here cannot be used to obtain accurate mean field equations for every possible well-formed WSCCS model. Firstly, the focus of the model should be to investigate the numbers of agents of each type present after some fixed period. Secondly, the system considered must have sufficiently large numbers of each agent. It is a well known result that deterministic models do not accurately capture the behaviour of small systems where stochastic effects can have a great influence. Most importantly, in disease systems it is known that the initial proportion of infected individuals greatly affects the convergence of deterministic equations to a discrete stochastic system [34].

In addition to these more obvious restrictions we also place some restrictions on the way that models must be written to be amenable to the method. In all of the examples given in this section the action a, and its complement $\overline{a}$, are communicating actions and all others are not communicating actions.

1. All weights associated with communication must be 1 or ω. For single actions, there should be only one alternative (non-communicating) action to the communication action, which is weighted 1.
2. Probabilistic choice steps must be separate from communication steps. These first two restrictions are mainly about convenience; the terms of Section 3.5 are more easily calculated. It might also be argued that using weights other than 1 or ω makes communication more difficult to understand: users are best to avoid it. An example of an agent that would not be allowed under these first two restrictions is

$$A = 1.a : A1 + 0.5.b : A2 + 0.5.c : A3 \ .$$

3. There should be no combination of distinct communicating actions $a\#b$. (Multiple distinct actions are allowed as choices, as are multiple communicating actions $\overline{a}^n$.) It is possible to reformulate any such system so that the two different communicating actions take place in successive stages. Given that concurrency in WSCCS is interleaving in any case this is not a bar to expressivity.
4. Collaborating agents performing a single instance of the output action may evolve to different states, depending on whether they communicate or not; collaborating agents that perform multiple instances of the action must evolve to the same state, regardless of whether they communicate or not (and irrespective of how many instances of the action they perform). For instance the agent

$$C = \omega.\overline{a} : C1 + 1.b : C2$$

   is allowed, as is

$$C = \omega^2.\overline{a}^2 : C1 + \omega.\overline{a} : C1 + 1.b : C1 \ ,$$

   but the agent

$$C = \omega^2.\overline{a}^2 : C1 + \omega.\overline{a} : C2 + 1.b : C3$$

   is not (assuming $C1$, $C2$ and $C3$ are distinct agents). Biologically there seems to be little need to allow evolution to different states depending on the number of instances of an action performed.
5. Processes should not include nested permission sets, i.e. all communication takes place between all processes (potentially), and not between subgroups defined by restriction. The reason for this is that the restriction operator cannot be distributed over parallelism [30]. From a modelling perspective, this appears to be a reasonable restriction, being equivalent to assuming random mixing since all agents can (potentially) communicate with all

others. If communication between subgroups is required, it can be accommodated by renaming. For instance, agents $A1$ and $C1$ communicate on *actionA* while agents $A2$ and $C2$ communicate on *actionB*.

6. Processes must always be guarded, e.g. $I \stackrel{\text{def}}{=} T \times Trans$ is not allowed. This simplifies construction of the transition table.

These restrictions make the definition of the general terms in Section 3.5 simpler and impose modelling disciplines to aid clarity. Removing the restrictions, while possible, would therefore make derivation of MFEs more complex, and make models more difficult to understand. Having constructed a range of models [19, 20, 21, 23], we can say that the restrictions have not proved limiting in terms of the systems that we have been able to model, and that there is therefore no advantage in removing these restrictions.

3.2. Preliminary Definitions

Some auxiliary definitions are required. Formal definitions are given in Appendix B; here we give informal motivation. Processes can be classified by syntactic features as: *communicating* (having an action enabled that is involved in a communication), *probabilistic* (having only actions enabled that are not involved in communication), and *priority* (communicating and using ω weights). Given any process A, the function *transitions(A)* returns all outgoing transitions from the source state A. The function *get_comm_trans(A)* returns the transitions involving communicating actions from A. Conversely, the function get_non_comm_trans(A) returns the (single) transition not involving a communicating action from A. Given a process A comprising agents $A1, \ldots, An$ in parallel (with no guards), the function *components(A)* returns agents $A1, \ldots, An$. For a process communicating on action a, two groups of processes involved in the synchronisation are defined: *collaborators* are those processes with the matching action $\overline{a}$, and *competitiors* are those processes with the same action a.

3.3. The Transition Table and Mean Field Equations

In Section 2.2 the notion of exit and entry activities was described. The transition table TT notes these exit and entry activities and forms the core of the translation from WSCCS to MFEs. Pseudocode to construct the table is given in Fig. 4, Section 3.4.

TT is a $(\mathcal{A} \times Act) \times \mathcal{A}$ matrix, with symbolic expressions over $\mathcal{A}$ as entries. From any agent Ai at a given time there may be several transitions a_j. The term in the MFE for Ai is built from subterms, each one derived using the *construct* function and corresponding to each a_j. The definition of *construct(A,w,a)*, and therefore the derivation of these subterms, is fully determined by the context of the action carried out (e.g. part of a probabilistic choice, or part of a communication) and the composition of the population (i.e. how many of each different agent there are). The classification is as follows:

NonCommunicating Agents and Simple Probabilistic Choice: a is a non communicating action, and the context is probabilistic choice.

Communicating Agents: a is a communicating action, where

Single communicating action, Prioritised: the communication involves only single actions (a and $\overline{a}$), and is prioritised.

Single communicating action, Not prioritised: the communication involves only single actions (a and $\overline{a}$), and is not prioritised.

Multiple parallel communicating actions, Prioritised: the communication involves multiple parallel actions (e.g. $\overline{a}\#\overline{a}\#\overline{a}$), and is prioritised.

Multiple parallel communicating actions, Not prioritised: the communication involves multiple parallel actions (e.g. $\overline{a}\#\overline{a}\#\overline{a}$), and is not prioritised.

Multiple distinct communicating actions, Prioritised: the communication involves choice between multiple distinct actions (e.g. a_1 and a_2), and is prioritised.

Details of this derivation are given in Section 3.5. For convenience the terms associated with each form above are summarised in Table 1. The pseudocode to compute the entries in the transition table is given in Fig. 5, Section 3.4.

Returning to TT, a single row of TT relates to the source state agent Ai at time t and one of its enabled actions a_j. Each entry (derived by *construct*) in that row is an expression in Ai_t denoting the proportion of Ai_t agents performing a_j to become some new destination state agent Ak_{t+1}. The columns of the matrix are labelled by the destination states Ak.

Where Ai_t evolves to the same agent Ak_{t+1} irrespective of which action it performs, a single row is used for that agent which is labelled $Ai_t *$. An example of such an agent is the $I1$ agent in Fig. 1. Note that if the new agent Ak_{t+1} is a parallel agent then entries are made in the columns corresponding to every component of Ak_{t+1}.

	prioritised	not prioritised
simple probabilistic choice	disallowed	$p_j A_t$
single communicating action	$\min\left(A_t, \frac{A_t C_t}{N_t}\right)$	$\frac{A_t C_t}{N_t + C_t}$
multiple parallel communicating actions	$\min\left(A_t, \frac{A_t \sum_i c_i C i_t}{N_t}\right)$	see Appendix C
multiple distinct communicating actions	$\min\left(\frac{A_t * C m_t}{N_t}, \frac{A_t * C m_t}{\sum_i^n C_{(A,a_i),t}}\right)$	omitted

Table 1: Summary of general terms for evolution of A_t agents following action a (a_m for the last case)

		0	$S2$	$SI2$	$I2$	$R2$
$S1$	*infect*			$\frac{S1_t * Trans1_t}{S1_t + I1_t + R1_t}$		
$S1$	$\surd$		$S1_t - \frac{S1_t * Trans1_t}{S1_t + I1_t + R1_t}$			
$I1$	*				$I1_t$	
$Trans1$	*	$Trans1_t$				
$R1$	*					$R1_t$

Table 2: State transition table for $S1, I1$, $Trans1$ and $R1$ agents in Fig. 1

		$S1$	$I1$	$R1$	$Trans1$
$S2$	$\surd$	$S2_t$			
$SI2$	$\surd$	$(1-p_i)SI2_t$	$p_i SI2_t$		$p_i SI2_t$
$I2$	$\surd$		$(1-p_r)I2_t$	$p_r I2_t$	$(1-p_r)I2_t$
$R2$	$\surd$			$R2_t$	

Table 3: State transition table for $S2, SI2, I2$ and $R2$ agents in Fig. 1

Given the constructed table, the mean field equation for Ak_{t+1} is obtained by summing the terms in the column Ak. The Mean Field Equations are generated by the pseudocode of the second half of Fig. 4. The algorithm produces a MFE for every agent, but further simplification is usually desirable. This is illustrated by the example of the next section.

3.3.1. Transition Table Example

The transition table corresponding to the model of Fig. 1 is given in Tables 2 and 3 to illustrate the form of the table. The table is sparse, so only the populated sections are given.

To complete the example based on Fig. 1 the MFEs are constructed. As specified in Fig. 4, summing each column gives the MFE for that state. The method yields the following equations

$$\begin{aligned}
S1_{t+1} &= S2_t + (1-p_i)SI2_t \\
I1_{t+1} &= p_i SI2_t + (1-p_r)I2_t \\
R1_{t+1} &= p_r I2_t + R2_t \\
Trans1_{t+1} &= p_i SI2_t + (1-p_r)I2_t \\
S2_{t+1} &= S1_t - \frac{S1_t * Trans1_t}{S1_t + I1_t + R1_t} \\
SI2_{t+1} &= \frac{S1_t * Trans1_t}{S1_t + I1_t + R1_t} \\
I2_{t+1} &= I1_t \\
R2_{t+1} &= R1_t \ .
\end{aligned}$$

When the model was constructed, the idea was to build it in stages. So there is the idea that one step of time should encompass both stages in the model.

That is, the point of interest is how the $S1$, $I1$ and $R1$ agents change over time, treating the $S2, SI2, I2$ and $R2$ agents as intermediaries. The equations above can be combined to remove mention of these intermediate states by substituting equations for $S2_{t+1}, SI2_{t+1}, I2_{t+1}$ and $R2_{t+1}$ into equations for $S1_{t+2}, I1_{t+2}$ and $R1_{t+2}$. This involves some adjustment of the time subscripts. The equation for *Trans*1 is removed since we are not interested in these agents, but note that $\mathit{Trans}1_t = I1_t$. The new system of MFEs is

$$\begin{aligned} S1_{t+2} &= S1_t - \frac{S1_t * I1_t}{S1_t + I1_t + R1_t} + (1 - p_i)\frac{S1_t * I1_t}{S1_t + I1_t + R1_t} \\ I1_{t+2} &= p_i \frac{S1_t * I1_t}{S1_t + I1_t + R1_t} + (1 - p_r) I1_t \\ R1_{t+2} &= p_r I1_t + R1_t \ . \end{aligned}$$

Finally, algebraic simplification is carried out, adjusting the time subscript to reflect a single time step. The final form of the MFEs is

$$\begin{aligned} S1_{t+1} &= S1_t - p_i \frac{S1_t * I1_t}{S1_t + I1_t + R1_t} \\ I1_{t+1} &= I1_t + p_i \frac{S1_t * I1_t}{S1_t + I1_t + R1_t} - p_r I1_t \\ R1_{t+1} &= R1_t + p_r I1_t \ . \end{aligned}$$

Further manipulation of the MFEs may be carried out as required. The advantages of having this alternative view of the model are that different properties (e.g. on the relationship of various parameters) may become more obvious, and that algebraic analysis can be carried out.

3.4. Pseudocode for MFE generation

Pseudocode to construct the transition table described in Section 3.3 is given in Figs. 4 and 5. The presentation of this pseudocode is simplified by assuming that the entries are given as symbolic expressions over agents A_i (*SymbolicTerm*). We assume that a *SymbolicTerm* may be assigned, that they may be added together using +, and that there is an empty *SymbolicTerm* which may be used for initialisation.

Five auxiliary functions are introduced corresponding to concepts detailed in the next section; their definitions are given in Appendix B. *competitors(A,a)* are those processes with the same action a and *collaborators(A,a)* are those processes with the matching action $\overline{a}$. If the second argument is omitted (as in the case of multiple distinct actions) then *competitors(A)* denotes all competitors on all actions. Similarly for *collaborators(A,a)*. The cases of single communicating action, multiple parallel communicating action, and multiple distinct communicating actions must be distinguished. This is done using the functions *single_act*, *parallel_act* and *multiple_act* respectively, based on the syntax of the agent or of its collaborators.

```
1   /* All agents in the specification have been identified and enumerated.
    Process each agent in turn. */
2   for each agent Ai {
3       /* Two cases of interest: communication or probability (restriction 2) */
4       if communicating(Ai) then {
5           /* This is a communicating agent therefore there will be at least one
            transition with a communicating action. Construct a term for
            each communicating transition.
            The sum of these terms (called total_term) is used in constructing
            the part of the MFE for the non-communicating action (lines 16-20).
            term and total_term are symbolic expressions in agents A. */
6           total_term = empty;
7           for each (w_j, a_j, Ak) ∈ get_comm_trans(Ai) {
8               /* Construct the term for exactly one outgoing transition a_j. */
9               term = construct(Ai, w_j, a_j);
10              /* The destination state for the transition may be a single agent A,
                or a parallel agent. TT must be updated for each destination state. */
11              for each Am ∈ components(Ak)
12                  TT[(Ai, a_j), Am] = TT[(Ai, a_j), Am] + term;
13              /* Add this term to the total_term constructed so far. */
14              total_term = total_term + term;
15          }
16          /* Get the single non-communicating alternative (restriction 1). */
17          (w, b, Ak) = get_non_comm_trans(Ai);
18          /* Update TT for each destination state (similar to lines 11-12),
            using A_i - total_term (the remainder of A_i after communication). */
19          for each Am ∈ components(Ak)
20              TT[(Ai, b), Am] = TT[(Ai, b), Am] + Ai_t - total_term;
21      }
22      else /* The second case of interest: probabilistic action. */
23          /* Several possible transitions from Ai, with associated weights, */
24          for each (w_j, a_j, Ak) ∈ transitions(Ai)
25              /* evolving to single or parallel agents (as in lines 11-12). */
26              for each Am ∈ components(Ak)
27                  /* Construct the probabilistic term for each transition,
                    and add it to the appropriate table entries. */
28                  TT[(Ai, a_j), Am] = TT[(Ai, a_j), Am] + construct(Ai, w_j, a_j)
29  }
30  /* Construct the MFE for each agent by summing relevant TT entries. */
31  for each agent Ak
32      for each action a_j
33          for each Ai
34              MFE[Ak] := MFE[Ak] + TT[(Ai, a_j), Ak]
```

Figure 4: Pseudocode to generate MFEs

```
1   function construct (A, w, a): SymbolicTerm {
2   /* construct a symbolic term relating to the change in A
    given the transition a. Cases correspond to Tab. 1 */
3   case A in {
4   /* Simple probabilistic choice, Tab. 1: row 1, column 2 */
5       probabilistic(A): return w/sum_weights(A) * A_t;
6   /* Single communicating action, prioritised, Tab. 1: row 2, column 1 */
7       communicating(A) and priority(A) and single_act(collaborators(A,a)):
8           term = min(A_t, A_t * collaborators(A,a)/(A_t + competitors(A,a)));
9   /* Single comm. action, not prioritised, Tab. 1: row 2, column 2 */
10      communicating(A) and not priority(A) and single_act(collaborators(A,a)):
11          term = A_t * collaborators(A,a)
                        /(A_t + collaborators(A,a) + competitors(A,a));
12  /* Multiple parallel comm. actions, prioritised, Tab. 1: row 3, column 1 */
13      communicating(A) and priority(A) and parallel_act(collaborators(A,a)):
14          term = min(A_t, A_t * sum_all_acts(collaborators(A,a))
                        /(A_t + competitors(A,a)));
15  /* Multiple distinct comm. actions, prioritised, Tab. 1: row 4, column 1 */
16      communicating(A) and priority(A) and multiple_act(A):
17          term = min(A_t * collaborators(A,a)/(A_t + competitors(A)),
                        A_t * collaborators(A,a)/collaborators(A));
18  /* no other cases are dealt with */
19      otherwise: return error;
20  }}
```

Figure 5: Pseudocode to construct a term describing the evolution of A agents at time $t+1$

The pseudocode for *construct* assumes two further auxiliary functions: *sum_weights* which takes a probabilistic agent and returns the sum of the weights of all outgoing transitions, and *sum_all_acts* used in the multiple parallel actions case which takes a set of agents (collaborators) and returns a *SymbolicTerm* capturing the total number of $\overline{a}$ actions performed by all collaborators.

3.5. Entries of the Transition Table

This section gives a more detailed derivation for each of the terms in the Table 1:

- Non-communicating agents and simple probabilistic choice (Section 3.5.1)
- Single communicating action: Prioritised (Section 3.5.2)
- Single communicating action: Not prioritised (Section 3.5.3)
- Multiple parallel communicating actions: Prioritised (Section 3.5.4)
- Multiple parallel communicating actions: Not prioritised (Section 3.5.5)
- Multiple distinct communicating actions: Prioritised (Section 3.5.6)

Agent evolution through communication depends not just on the action itself (single or multiple, prioritised or not), but also on the mix of agents in the population available to communicate on the chosen action. The mean outcome is based on the weighted multinomial choice of all possible outcomes.

Consider a general system with agents A, Ci, Xi and Pi.

A is the agent of interest, i.e. the A in *construct*(A, w, a), and a is the action of interest. We wish to calculate how many of A will communicate on action a and evolve to the next state. As noted above, the number of A agents at time t is denoted A_t.

$Ci_{(A,a)}$ are the agents collaborating with A on a, i.e. the agents who have the corresponding action $\overline{a}$. The total number of collaborators, denoted $C_{(A,a,t)}$, is defined as $\sum_i Ci_{(A,a),t}$. The number of *types* of collaborator is denoted m_c.

$Xi_{(A,a)}$ are the other agents interacting with Ci on $\overline{a}$, i.e. agents other than A doing the action a. These may be regarded as being in competition with agent A instances since they may absorb instances of the action $\overline{a}$ leaving fewer for collaboration with A. Competitors are assumed to always have the same syntactic form with regard to single, multiple, prioritised, and not prioritised actions as A. The total number of competitors, denoted $X_{(A,a,t)}$, is defined as $\sum_i Xi_{(A,a),t}$. The number of *types* of competitor is denoted m_x.

The total number of agents doing the a action $A_t + X_{(A,a),t}$ is denoted N_t.

Pi are those agents not involved in the a action, either as competitors or collaborators.

To aid clarity, the subscripts (A, a) and (A, a, t) are omitted below since these can be understood from the context. In the following, the binomial coefficient $\binom{n}{m}$ is used, representing the number of unordered ways to choose m objects from a group of n distinct objects.

3.5.1. Non-communicating agents and simple probabilistic choice

Computation of *construct*(A, w, a) is straightforward for steps involving only non-communicating actions. The agent in the source state takes the form

$$A \stackrel{\text{def}}{=} w_1.a_1 : A1 + w_2.a_2 : A2 + \ldots + w_m.a_m : Am \ ,$$

where $\exists! j : 1 \ldots m$ such that $w = w_j$ and $a = a_j$. Agent A evolves independently without communicating with any other agent. This evolution is governed by the multinomial distribution. The semantics of WSCCS [30] states that in a large number of repeated experiments of this process, we expect to see A_j chosen with relative frequency $w_j / \sum_{k=1}^{m} w_k$. Therefore, from standard theory, the probability that A will become one of its destination processes Aj is

$$p_j = \frac{w_j}{\sum_{k=1}^{m} w_k} \ .$$

The mean number of A agents that become one of the agents Aj in the next time step is

$$\textit{construct}(A, w, a) = p_j A_t \ .$$

3.5.2. Single communicating action: Prioritised

The agent in the source state takes the form

$$A = \omega.a : A1 + 1.b : A2 \ ,$$

where a is a communicating action and b is not a communicating action. We wish to calculate how many A_t evolve to $A1_{t+1}$. We assume collaborators Ci of the form

$$Ci = \omega.\overline{a} : Ci' + 1.c : Ci'' \ ,$$

where $\overline{a}$ is a communicating action and c is not a communicating action (c may be the same action as b, or a different action). Given a particular A agent, there are two choices. The A agent communicates with a Ci agent to evolve to an $A1$ agent, or it does not communicate with a Ci agent and evolves to an $A2$ agent. Since communication is prioritised, failure to communicate with a Ci agent arises either because there are not sufficient numbers of Ci agents with which to communicate, or because there are Xi agents communicating with the Ci agents. In the general case, the number of resulting $A1$ agents ranges from 0 to A_t, depending on both the number of Ci and the number of Xi. There is a further complication: some possibilities in this range are more likely than others. For example, there are A_t ways for just one A to communicate with a single Ci agent, but there is exactly one way for all A agents to communicate with A_t Ci agents.

For a single action with prioritised communication, given that the weights of alternative actions are 1 (restriction 1, Section 3.1), the mean change in agent A is expressed as

$$\frac{\sum_{k=1}^{A_t} k\binom{A_t}{k}\binom{N_t-A_t}{C_t-k}}{\sum_{k=1}^{A_t}\binom{A_t}{k}\binom{N_t-A_t}{C_t-k}} \,. \tag{1}$$

For this example the limits of the sum have been shown explicitly. These are omitted hereafter since they can be understood from the context, and one of the binomial coefficients will be zero if the limits are exceeded. On the numerator we have the weighted sum of all possible evolutions of A agents to $A1$ agents. That is, if the evolution is to a state with 42 $A1$ agents, then we multiply by 42. Similarly, if the evolution is to a state with a single $A1$ agent, then we multiply by 1. This is k in the expression above. The second component of the numerator indicates the number of A communicating with Ci agents, and the third component indicates the number of Xi agents communicating with Ci agents. The denominator is the same sum, unweighted, representing all possible evolutions of A.

This term can be simplified via Vandermonde's Convolution [11] and standard theory regarding the binomial coefficient to

$$\frac{A_t C_t}{N_t} \,.$$

If $N_t < C_t$ more agents are available to do $\overline{a}$ than a and the term is limited to A_t, therefore

$$construct(A, w, a) = \min\left(A_t, \frac{A_t C_t}{N_t}\right) \,.$$

In practice, it is possible to eliminate the min in this term by formulating the model so that $N_t \geq C_t$ is always true. For example, pairing all Ci agents with a parallel Xi agent as in the example of Fig. 1.

3.5.3. Single communicating action: Not prioritised

The agent in the source state takes the form

$$A = 1.a : A1 + 1.b : A2 \,,$$

where a is a communicating action and b is not a communicating action (restriction 1, Section 3.1). Following the single unprioritised a action, the mean number of A agents evolving to $A1$ at time $t+1$ is given by

$$\frac{\sum_i \binom{C_t}{i}\sum_k k\binom{A_t}{k}\binom{N_t-A_t}{i-k}}{\sum_i \binom{C_t}{i}\sum_k \binom{A_t}{k}\binom{N_t-A_t}{i-k}} \,.$$

The added term here expresses that the alternative action b may be chosen more frequently than in the prioritised communication case. Again, the term simplifies, yielding

$$construct(A, w, a) = \frac{A_t C_t}{N_t + C_t} \,.$$

In this case $N_t + C_t \geq C_t$ for all values of N_t and C_t and there is no need for a min term.

3.5.4. Multiple parallel communicating actions: Prioritised

The agent in the source state takes the form

$$A = \omega.a : A1 + 1.b : A2 \ ,$$

as in the *Single communicating action, Prioritised* case. The difference here is the number of actions performed by the collaborating Ci agents. The $\overline{a}$ actions may be multiple, i.e. $\overline{a}\#\overline{a}\#\overline{a}\#\ldots\#\overline{a}$, written $\overline{a}^n$ for n instances of $\overline{a}$. We assume that there are m different Ci agents, each performing up to c_i instances of $\overline{a}$. In particular, we assume Ci takes the form

$$Ci = \omega^{c_i}.\overline{a}^{c_i} : Ci' + \omega^{c_i-1}.\overline{a}^{c_i-1} : Ci' + ... + \omega.\overline{a} : Ci' + 1.b : Ci' \ .$$

This means that Ci will do as many $\overline{a}$ actions as it can (because of priority).

The mean number of A which evolve to $A1$ is given by

$$\frac{\sum_k k\binom{A_t}{k}\binom{N_t-A_t}{\sum_i c_i Ci_t}}{\sum_k \binom{A_t}{k}\binom{N_t-A_t}{\sum_i c_i Ci_t}} \ .$$

The term $\sum_i c_i Ci_t$ gives the total number of $\overline{a}$ actions available.

This can be simplified (see the right hand term of the min expression below). As before, the total number of actions is limited by the capabilities of A. In particular, if $N_t < \sum_i c_i Ci_t$, then fewer actions are performed in total (leading to the min term).

$$\mathit{construct}(A, w, a) = \min\left(A_t, \frac{A_t \sum_i c_i Ci_t}{N_t}\right) \ .$$

Unlike the case for single action, prioritised communication, it is not possible to eliminate this min term through manipulation of the model while maintaining competition between A and its competitors.

3.5.5. Multiple parallel communicating actions: Not prioritised

In general, models without priority featuring agents that perform multiple instances of the output action lead to MFE that are intractable. A full derivation of the general term is included in Appendix C for completeness but here we present the case where an agent can perform at most two instances of the action. Agents performing multiple instances of the output action must always evolve to the same state (restriction 4, Section 3.1) so the collaborating agents take the general form

$$Ci = 1.\overline{a}^2 : Ci' + 1.\overline{a} : Ci' + 1.b : Ci' \ .$$

The agent in the source state takes the form

$$A = 1.a : A1 + 1.b : A2 \ ,$$

with competitors taking the same form

$$Xi = 1.a : X1 + 1.b : X2 \; .$$

In the following, the multinomial coefficient $\binom{m}{p,q,r}$ is used. This represents the number of unordered ways to choose a group of p objects, a group of q objects and a group of r objects from a group of m distinct objects, with $m = p+q+r$.

The mean number of A that become $A1$ is given by

$$\frac{A \sum_{n_1} \sum_{n_2} \binom{C_t}{n_1, n_2, C_t - n_1 - n_2} \binom{A_t + X_t - 1}{2 \times n_2 + n_1 - 1}}{\sum_{n_1} \sum_{n_2} \binom{C_t}{n_1, n_2, C_t - n_1 - n_2} \binom{A_t + X_t}{2 \times n_2 + n_1}} \; .$$

For even a simple case such as this the term is quite unwieldy and does not lend itself to the sort of analyses we typically wish to use to study our models. In the general case this problem is amplified. We have not found it biologically necessary to use this form of communication in our case studies so far.

3.5.6. Multiple distinct communicating actions: Prioritised

We consider first the case of two distinct actions since this arises more commonly (e.g. the supershedders model of McCaig [19]). The agent in the source state takes the form

$$A = \omega.a_1 : A1 + \omega.a_2 : A2 + 1.b : A3 \; .$$

Two sets of collaborating agents $C1$ and $C2$ perform the actions $\overline{a_1}$ and $\overline{a_2}$ respectively. Communication is prioritised. The agent A can perform either a_1 or a_2, evolving differently in each case, but cannot perform both actions together (restriction 3, Section 3.1). The action b is a non-communicating action and because of priority will only be executed if neither a_1 or a_2 can synchronise with another process.

The number of A_t which communicate with $C1_t$ is

$$\frac{\sum_k \sum_j k \binom{A_t}{k, j, A_t - k - j} \binom{N_t - A_t}{C1_t - k, C2_t - j, N_t - A_t - C1_t - C2_t + k + j}}{\sum_k \sum_j \binom{A_t}{k, j, A_t - k - j} \binom{N_t - A_t}{C1_t - k, C2_t - j, N_t - A_t - C1_t - C2_t + k + j}}$$

This can be simplified, so that

$$\frac{A_t C1_t}{N_t} \; .$$

This term is valid only when $N_t \geq C1_t + C2_t$. If $N_t < C1_t + C2_t$ then we have

$$\frac{A_t C1_t}{C1_t + C2_t} \; .$$

Therefore, the general term for the number of A_t agents which communicate with $C1_t$ is

$$\mathit{construct}(A, w, a) = \min \left(\frac{A_t C1_t}{N_t}, \frac{A_t C1_t}{C1_t + C2_t} \right) \; .$$

The corresponding term for the number of A_t agents which communicate with $C2_t$ is found similarly

$$construct(A, w, b) = \min\left(\frac{A_t C2_t}{N_t}, \frac{A_t C2_t}{C1_t + C2_t}\right) .$$

As for the case of single action prioritised communication, this min term can be avoided by using parallel agents.

n actions, prioritised communication. The results for two actions can be generalised to cover cases where there are n different actions $a_1, a_2, ..., a_n$, giving

$$construct(A, w, a_m) = \min\left(\frac{A_t * Cm_t}{N_t}, \frac{A_t * Cm_t}{\sum_i^n C_{(A,a_i),t}}\right) .$$

where m ranges over $1 \ldots n$ and Cm denotes the set of agents collaborating on action a_m. As above, this min term can be avoided by using parallel agents.

3.5.7. General Properties of the MFE Semantics

Given a model of the form

$$Population \stackrel{\text{def}}{=} A1\{n_1\} \times A2\{n_2\} \times \ldots \times Am\{n_m\}\lceil\{L\}$$

(possibly with the priority operator Θ) the MFEs derived will always be first order difference equations, i.e. of the general form

$$\mathbf{A}_{t+1} = f(\mathbf{A}_t) , \tag{2}$$

where $\mathbf{A}$ is the vector of the different agent types for which equations are being derived,

$$\mathbf{A} = (A_1 \ldots A_m) .$$

This is a consequence of the 'memoryless' Markovian nature of WSCCS, i.e. the future state of the system depends only on the current state of the system and not on previous states.

Higher order equations are obtained by substitution. For example, noting that (2) implies

$$\mathbf{A}_{t+2} = f(\mathbf{A}_{t+1}) ,$$

and substituting for $\mathbf{A}_{t+1}$ to find

$$\begin{aligned} \mathbf{A}_{t+2} &= f(\mathbf{A}_{t+1}) \\ &= f(f(\mathbf{A}_t)) , \end{aligned}$$

as was done for the example of Section 3.3.1. If such simplification is carried out, the modeller should be sure that the intermediate state $\mathbf{A}_{t+1}$ is not of interest. The second order equations are likely to be algebraically more complicated.

For models featuring only probabilistic choice the derived equations will be first order linear difference equations (i.e. each term in f is linear in one of the components of $\mathbf{A}$) and for models featuring communication they will be non-linear.

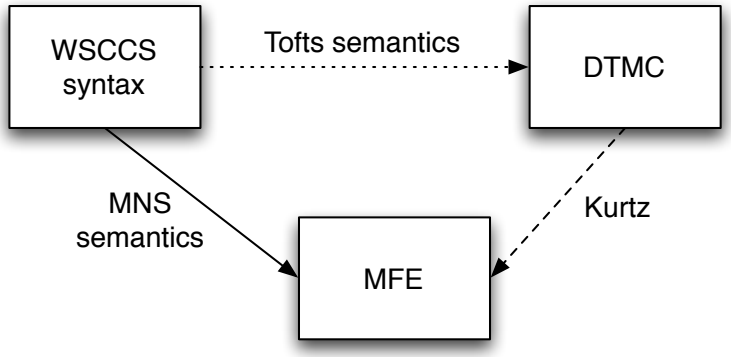

Figure 6: Relationship between MFE and Markov chain semantics of WSCCS

3.6. Correctness

In this Section the relationship between the derived MFEs and the standard Markov semantics for WSCCS, as represented in Fig. 6 is established. The method presented here offers an alternative semantics for WSCCS in terms of Mean Field Equations (shown as the solid line in Fig. 6). While this may be useful and interesting in itself, in terms of describing and analysing models our particular goal is to capture the existing behaviour [30] (represented by the dotted arrow in Fig. 6). This will allow conclusions to be drawn about how individual behaviour results in emergent properties in the population dynamics. The key to this relationship is a result of Kurtz [17], who presented limit theorems relating the mean of Continuous Time Markov chain (CTMC) and Discrete Time Markov Chain (DTMC) to ordinary differential equations. This is shown as a dashed line in Fig. 6. Kurtz showed that it was possible to derive ODEs as an approximation of the average behaviour of a DTMC. At the limit, where the DTMC consists of infinitely many agents, the mean of the Markov chain is *equivalent* to the derived ODEs. An intermediate step of Kurtz's proof produces terms equivalent to those given in Section 3.5. We use this to show the correctness of our derivation of MFEs from WSCCS.

Kurtz laid out conditions under which his limit theorem holds. We present these here and then repeat them below, together with an explanation of how they relate to our WSCCS models.

Consider $X_n(k)$, a sequence of discrete time Markov processes, with measurable state spaces $(E_n, \mathscr{B}_n)$, $E_n \in \mathscr{B}^k$, the Borel sets [16] in $\mathbb{R}^k$. When considering processes over $\{0, 1, ..., n\}$ Kurtz rescales to $[0, 1]$ by dividing through by n and letting $n \to \infty$.

The one step transition function of $X_n(k)$ is denoted by

$$\mu_n(x, \Gamma) = P\{X_n(k+1) \in \Gamma \mid X_n(k) = x\}$$

i.e. $\mu_n(x, \Gamma)$ is the probability of moving from x to a point in the set Γ in one timestep. Suppose there exist sequences of positive numbers α_n and ε_n such

that

$$\lim_{n\to\infty} \alpha_n = \infty \qquad \text{and} \qquad \lim_{n\to\infty} \varepsilon_n = 0 \ ,$$

$$\sup_n \sup_{x\in E_n} \alpha_n \int_{E_n} |z-x| \mu_n(x, dz) < \infty$$

and

$$\lim_{n\to\infty} \sup_{x\in E_n} \alpha_n \int_{|z-x|>\varepsilon_n} |z-x| \mu_n(x, dz) = 0 \ .$$

Kurtz shows that the difference between state changes in the Markov chain is equivalent to that expressed in the relevant ODE. For every $\delta > 0,\ t > 0$

$$\lim_{n\to\infty} \sup_{x\in E_n} P\Big\{ \sup_{k\leq \alpha_n t} |X_n(k) - X_n(0) - \sum_{l=0}^{k} \frac{1}{\alpha_n} F_n(X_n(l))| > \delta \\ \text{where } X_n(0) = x \Big\} = 0 \ ,$$

where $F_n(x) = \alpha_n \int_{E_n} (z-x) \mu_n(x, dz)$. It is to this intermediate stage in the proof of the correctness of ODEs that we compare our method of deriving MFEs.

We now relate these conditions to our method for deriving MFE:

1. **[Kurtz]** $X_n(k)$ is a sequence of discrete time Markov processes, with measurable state spaces $(E_n, \mathscr{B}_n)$, $E_n \in \mathscr{B}^k$, the Borel sets [16] in $\mathbb{R}^k$.
   [WSCCS] The states of WSCCS models here are in $\mathbb{N}^k$, where k is the number of types of agents in the model. This is a consequence of the vector representation of state space chosen in Section 2.3. All subsets of $\mathbb{R}^k$ are Borel sets and $\mathbb{N}^k \subset \mathbb{R}^k$.
2. **[Kurtz]** When considering processes over $\{0, 1, ..., n\}$ Kurtz rescales to $[0, 1]$ by dividing through by n and letting $n \to \infty$.
   [WSCCS] Processes range over $\{0, 1, ..., n\}$, where n is the initial number of agents in the system. The same rescaling is carried out to match Kurtz's conditions.
3. **[Kurtz]** The one step transition function is denoted by

   $$\mu_n(x, \Gamma) = P\{X_n(k+1) \in \Gamma \mid X_n(k) = x\}$$

   i.e. $\mu_n(x, \Gamma)$ is the probability of moving from x to a point in the set Γ in one timestep.
   [WSCCS] Consider the LTS of a WSCCS model under abstract bisimulation. The one step transition function can be extracted from the LTS by the following.

   $$\mu_n(A, \{A' \mid A \xrightarrow{a[p]} A'\}) = p$$

   where p is the probability that this action occurs, derived from all the preceding weighted choices as given in Appendix A.2.1.

4. [**Kurtz**] Suppose there exist sequences of positive numbers α_n and ε_n such that

$$\lim_{n\to\infty} \alpha_n = \infty \qquad \text{and} \qquad \lim_{n\to\infty} \varepsilon_n = 0 \,,$$

$$\sup_n \sup_{x\in E_n} \alpha_n \int_{E_n} |z-x|\mu_n(x,dz) < \infty \tag{3}$$

and

$$\lim_{n\to\infty} \sup_{x\in E_n} \alpha_n \int_{|z-x|>\varepsilon_n} |z-x|\mu_n(x,dz) = 0 \,. \tag{4}$$

[**WSCCS**] In WSCCS terms, we think of z and x as being state vectors with a component representing each type of agent in the system. The term $|z-x|$, which appears in both (3) and (4), is the magnitude of the difference between the start state, x, and the destination state, z, i.e. how much change there has been in the components. This means that $|z-x|$ is the norm of the vector travelled in one timestep.
As $n \to \infty$ the number of states that can be reached in one step becomes very large (since WSCCS is synchronous, many components change simultaneously). As demonstrated earlier, there is a higher probability of moving to a state with a small change from the previous state when there are lots of components since there are lots of ways to make that change. Conversely, the states for which the change is high are less likely to occur, since this requires the coincidence of many choices. Formally, the distribution of weights on the LTS is either Gaussian (for probabilistic choices) or hyper-geometric (for communicating actions).
The vectors are scaled by dividing by n (point 2 above), therefore $0 \leq |z-x| \leq 1$. By definition $\mu(x,z)$ is a probability, therefore $0 \leq \mu(x,z) \leq 1$. The states z for which $\mu(x,z)$ is greatest will be close to x (such that $|z-x|$ is close to 0). For z where $|z-x|$ is larger, the probability of reaching z will be close to 0. This means that $\int_{E_n} |z-x|\mu_n(x,dz)$ is infinitesimal and at the limit (where $n=\infty$) $\alpha_n = \infty$ and $\alpha_n \int_{E_n} |z-x|\mu_n(x,dz) < \infty$ is true and (3) is satisfied.
In contrast, the core of (4) represents the states which are unreachable, i.e. for which $\mu(x,z) = 0$. As $\varepsilon_n \to 0$, (4) captures that the reachable states are nearer to x, i.e. the change $|z-x|$ is small.

Kurtz shows that the difference between state changes in the Markov chain is equivalent to that expressed in the ODE. For every $\delta > 0,\ t > 0$

$$\lim_{n\to\infty} \sup_{x\in E_n} P\Bigg\{ \sup_{k\leq \alpha_n t} |X_n(k) - X_n(0) - \sum_{l=0}^{k} \frac{1}{\alpha_n} F_n(X_n(l))| > \delta \\ \text{where } X_n(0) = x \Bigg\} = 0 \,, \tag{5}$$

where $F_n(x) = \alpha_n \int_{E_n} (z-x)\mu_n(x,dz)$.

Applied to process behaviour over only one timestep (as here), (5) becomes

$$\lim_{n\to\infty} \sup_{x\in E_n} P\Bigg\{ |X_n(1) - X_n(0) - \int_{E_n} (z-x)\mu_n(x,dz)| > \delta \\ \text{where } X_n(0) = x \Bigg\} = 0 \ . \tag{6}$$

Introducing a function

$$G(x) = \int_{E_n} (z-x)\mu_n(x,dz) \ , \tag{7}$$

(6) means that at the limit $n \to \infty$, the difference

$$X_n(1) - \{X_n(0) + G(X_n(0))\} \ ,$$

is infinitesimal; therefore, it can be assumed that

$$X_n(1) = X_n(0) + G(X_n(0)) \ .$$

Markov processes have no memory of previous states, allowing a further generalisation:

$$X_n(k+1) = X_n(k) + G(X_n(k)) \ . \tag{8}$$

The form of $G(x) = \int_{E_n} (z-x)\mu_n(x,dz)$ is equivalent to the way in which MFEs are constructed in Section 3. We interpret the integral here as a summation. The integral, across the entire state space, of the product of the change of state and the probability of making that change, gives the mean change of state. By adding this to the previous state of the models, (8), we obtain the MFEs derived by our method.

Consider for instance an alternative derivation of the term for single communicating action with prioritised contact, Eqn. (1) of Section 3.5.2. In this case

$$\frac{\binom{A_t}{j}\binom{N_t-A_t}{C_t-j}}{\sum_{k=1}^{A_t}\binom{A_t}{k}\binom{N_t-A_t}{C_t-k}}$$

represents the probability of j of the agents of interest communicating with the collaborators C_t. If we then multiply by j (equivalent to $(z-x)$ in (7)) and sum across all values of j (equivalent to the integral across E_n in (7)) we have

$$\frac{\sum_{j=1}^{A_t} j\binom{A_t}{j}\binom{N_t-A_t}{C_t-j}}{\sum_{k=1}^{A_t}\binom{A_t}{k}\binom{N_t-A_t}{C_t-k}} = \frac{A_t C_t}{N_t} \ ,$$

which is (1).

4. Examples

We present two further examples to illustrate the method. The first, on population growth is from McCaig et al [23]. The second, on resource allocation, is novel.

$$
\begin{aligned}
N1 &\stackrel{\text{def}}{=} 1.get : N2 + 1.\surd : 0 \\
Res1 &\stackrel{\text{def}}{=} 1.\overline{get} : Res2 + 1.\surd : Res2 \\
N2 &\stackrel{\text{def}}{=} p_b.\surd : (N1 \times N1) + p_d.\surd : 0 + (1 - p_b - p_d).\surd : N1 \\
Res2 &\stackrel{\text{def}}{=} 1.\surd : Res1 \\
Population &\stackrel{\text{def}}{=} (N1\{n\} \times Res1\{f\})\lceil\{\surd\}
\end{aligned}
$$

Figure 7: Density dependence on deaths with non-prioritised communication

4.1. Population Dynamics

An important feature of many biological models is a dynamic population, i.e. a population with the ability to grow and shrink. While it is possible to simply add fixed probabilities of birth and death, it is often more desirable to allow the probability of birth and death to vary depending on the size of the population at each instant in time (density dependence). For example, as the population grows, resources such as food and shelter become scarce, therefore individuals become weaker and are more likely to die. Alternatively this weakness may manifest itself as a reduced fecundity and a reduction in the birth rates.

The model of Fig. 7 demonstrates an individual-based approach to modelling density dependent population growth. Agents represent population members ($N1$, $N2$) and "resource" ($R1$, $R2$), e.g. food, shelter, or space. The resource is required by individuals to survive. It is finite and individuals must compete for it. It does not last forever, therefore must be reacquired at regular intervals. Access to this resource can be used to determine the likelihood of either birth or death; the model of Fig. 7 has density dependent death.

Acquiring a resource is modelled with non-prioritised communication. This means that individuals might not obtain the resource, even when it is available, and is therefore more biologically plausible. In Fig. 7 the $N1$ agents can get the resource, becoming the agent $N2$, but if they do not get the resource they die, becoming the null agent 0 (this is density dependent death). The $N2$ agents give birth probabilistically and die probabilistically (e.g. due to old age).

The transition table for this model is shown in Table 4 and the resulting MFE (substituting f for $Res1$ since resource is constant) is

$$N_{t+1} = (1 + p_b - p_d)\frac{fN_t}{f + N_t} \ , \tag{9}$$

where the term $fN_t/(f + N_t)$ represents the proportion of the population who survive the competition for resource, with the factor $(1 + p_b - p_d)$ representing the increase in the population due to births and the decrease due to probabilistic death. Equation (9) can be rearranged to give

$$N_{t+1} = \frac{aN_t}{1 + bN_t} \ , \tag{10}$$

	0	$N1_{t+1}$	$N2_{t+1}$	$Res1_{t+1}$	$Res2_{t+1}$
$(N1_t, get)$			$\frac{N1_t * Res1_t}{N1_t + Res1_t}$		
$(N1_t, \surd)$	$N1_t - \frac{N1_t * Res1_t}{N1_t + Res1_t}$				
$(Res1_t, *)$					$Res1_t$
$(N2_t, \surd)$	$p_d N2_t$	$(1 - p_b - p_d)N2_t$ $+2p_b N2_t$			
$(Res2_t, *)$				$Res2_t$	

Table 4: State transition table for $N1, N2, Res1$ and $Res2$ agents in Fig. 7

where $a = (1 + p_b - p_d)$ and $b = 1/f$. Equation (10) is the Beverton-Holt model [6], originally proposed as a model of salmon populations displaying density dependent birth. Even though our model is based on density dependent death the interpretations of a and b here are similar to the original Beverton and Holt definitions. Parameter a corresponds to the proliferation rate per generation and parameter b corresponds to $1/M$ where M is a measure of the maximal population size. Our derivation endorses the plausibility of the Beverton-Holt model, which is commonly used in models of plant populations but not so widely used for animal populations.

Setting $N_{t+1} = N_t = N^*$ in (9) and solving for N^* yields the steady state

$$N^* = (p_b - p_d)f \ .$$

To ensure the steady state is positive we require $p_b > p_d$. Such analysis is not obvious from the original model.

Finding equation (9) for a population featuring density dependent death highlights the advantage of our approach. We know that the population model described by (9) is a direct consequence of the individual level assumptions that have been made. The traditional method of deriving population level equations relies on assumptions about how individual level behaviour influences the dynamics of the system as a whole, when this is not always well understood.

We also previously developed a model featuring competition for food with density dependent birth [23] with the resulting MFE,

$$N_{t+1} = (1 - p_d)N_t + \frac{p_b f N_t}{f + N_t} \ , \tag{11}$$

featuring a birth term similar to the single term found for the density dependent death model, (9), but with death captured by a separate term. Here the density dependent birth term could be rewritten as the single term in the Beverton-Holt model, by choosing $a = p_b$ and $b = 1/f$, but we still have a separate term to capture the fixed probability of death. Once again this MFE is a direct result of the individual behaviour described in the underlying WSCCS model.

$$
\begin{aligned}
\mathit{Booth\_A} &\stackrel{\text{def}}{=} \omega.\mathit{occupy} : \mathit{Booth\_U} + 1.\surd : \mathit{Booth\_A} \\
\mathit{Booth\_U} &\stackrel{\text{def}}{=} \omega.\mathit{vacate} : \mathit{Booth\_A} + 1.\surd : \mathit{Booth\_U} \\
\mathit{Canvasser} &\stackrel{\text{def}}{=} q.\mathit{success} : (\mathit{Waiting} \times \mathit{Canvasser}) + (1-q).\mathit{fail} : \mathit{Canvasser} \\
\mathit{Waiting} &\stackrel{\text{def}}{=} \omega.\overline{\mathit{occupy}} : \mathit{Busy} + 1.\surd : \mathit{Waiting} \\
\mathit{Busy} &\stackrel{\text{def}}{=} p.\mathit{work} : \mathit{Done} + (1-p).\surd : \mathit{Busy} \\
\mathit{Done} &\stackrel{\text{def}}{=} \omega.\overline{\mathit{vacate}} : 0 \\
\mathit{Cafe} &\stackrel{\text{def}}{=} \Theta((\mathit{Canvasser}\{c\} \times \mathit{Booth\_A}\{b\})\lceil\{\mathit{fail}, \mathit{success}, \mathit{work}, \surd\})
\end{aligned}
$$

Figure 8: The Internet Cafe

	0	C_{t+1}	W_{t+1}	B_{t+1}	D_{t+1}
$(C_t, \mathit{success})$		qC_t	qC_t		
(C_t, fail)		$(1-q)C_t$			
(W_t, occupy)				$min(W_t, BA_t)$	
$(W_t, \surd)$			W_t $-min(W_t, BA_t)$		
(B_t, work)					pB_t
$(B_t, \surd)$				$(1-p)B_t$	
(D_t, vacate)	D_t				

Table 5: Transition table for Canvasser agents in Fig. 8

4.2. The Internet Cafe

Resource allocation is a common problem in distributed computing. This example was inspired by the storage allocation problem of [14], phrasing the problem in terms of an Internet cafe.

The Internet cafe has a fixed number of booths, b. Customers arrive to make use of the booths, but there may be no booths available, so a queue of waiting customers forms. Waiting customers use booths as they are vacated (the queuing system here is not orderly). The model is shown in Fig. 8. Arrivals at the cafe are modelled by a Bernoulli process (*Canvasser*), which represents employees outside the cafe attracting customers inside. Note the separation of the probabilistic choice to continue working or not (in *Busy*) and the communication to signal that the booth is free (in *Done*).

Parts of the transition table for the model are given in Tables 5 and 6 (using abbreviations for agent names). The resulting MFEs are given below. The term

	BA_{t+1}	BU_{t+1}
$(BA_t, occupy)$		$min(W_t, BA_t)$
$(BA_t, \surd)$	$BA_t - min(W_t, BA_t)$	
$(BU_t, vacate)$		$BU_t - min(D_t, BU_t)$
$(BU_t, \surd)$	$min(D_t, BU_t)$	

Table 6: Transition table for Booth agents in Fig. 8

$min(D_t, BU_t)$ from Table 6 simplifies to D_t since $BU_t < D_t$ is impossible.

$$\begin{aligned} BA_{t+1} &= BA_t - min(BA_t, W_t) + D_t \ , \\ BU_{t+1} &= BU_t + min(BA_t, W_t) - D_t \ , \\ C_{t+1} &= C_t \ , \\ W_{t+1} &= W_t - min(BA_t, W_t) + qC_t \ , \\ B_{t+1} &= (1-p)B_t + min(BA_t, W_t) \ , \\ D_{t+1} &= pB_t \ . \end{aligned} \tag{12}$$

From these equations we can derive new information. For example, consider the steady state of the system, i.e. where $X_{t+1} = X_t$ for any X. The number of booths is constant over time, $BA_{t+1} + BU_{t+1} = BA_t + BU_t = b$, as is the number of canvassers $C_{t+1} = C_t = c$. We consider two cases in turn: $BA < W$ and $BA \geq W$ (because of the min term in (12)). If $BA < W$ a contradiction is derived: there is no steady state in this case. In fact, the length of the queue grows unboundedly. If $BA \geq W$ then the equations can be solved, yielding the condition $b - qc(1 + 1/p) \geq qc$ for the steady state to exist. This equation, for example, allows the cafe owner to calculate the number of booths required $b \geq qc(2 + 1/p)$ based on known customer arrival and departure rates.

In the source paper [14] the conjecture was that utilisation of file space becomes maximal as the length of time files stayed in store increased. Translated to the above scenario, this is $BA/BU \to 0$ as $p \to 0$. In the steady state

$$\frac{BA}{BU} = \frac{b - qc(1 + 1/p)}{qc(1 + 1/p)} \ ,$$

and the result follows.

As above, these observations only become clear when the MFEs are derived from the model; however, it is important to have made the individual-based model in the first place since this is the source of our observations.

5. Conclusion

An alternative, yet equivalent, semantics has been presented for WSCCS. The advantages of the new semantics are that they allow time series information

to be calculated quickly without calculating the whole state space, and that they give a means of translating between the individual-based world of processes and the population-based dynamics of mean field equations. This last point is particularly crucial in the biological setting. For many years biologists have sought a rigorous way of moving between scales [4, 18].

Having established the theoretical basis for the translation between WSCCS and MFEs future work will focus on application to specific problems. Several epidemiological systems have already been investigated with some success:

- Using this technique it is possible to explore how different types of local interaction can be translated into transmission at the global level [20]. In particular, interaction at the local level may be frequency dependent (a fixed number of contacts regardless of population size). Mathematical biologists typically reflect this assumption at the population level by frequency dependent transmission $\beta SI/N$. Alternatively, interaction at the local level may be density dependent (a variable number of contacts depending on population size). Mathematical biologists typically reflect this assumption at the population level by density dependent transmission βSI. Turner et al. [33] suggest that frequency dependent transmission results at the global level, regardless of local interaction. Our results showed that density dependent interaction at the local level was matched by the traditional density dependent transmission term at the global level. Similarly a fixed probability of contact at the local level leads to a frequency dependent transmission term at the global level.

- Population dynamics, with density dependent births and deaths are described in [23], together with an extended disease model which has been successfully matched to HIV-AIDS data.

- The method allows easy comparison of different mechanisms for disease spread. A recent study [21] considers *superspreaders*. Models encode whether superspreaders are more infectious than other infected individuals, or whether they are more gregarious than other infected individuals. Our surprising result was that mechanism is not important for average behaviour in this case.

The approach suits any problem in which a system is composed of a number of replicated individuals, where there is interaction between individuals, and where we are interested in how this interaction contributes to emergent system properties. The approach will most obviously be useful for biologists of various kinds, and for exploration of distributed computer systems.

Acknowledgement

This work was supported by EPSRC through a Doctoral Training Grant (CM, from 2004–2007) and through *System Dynamics from Individual Interactions: A process algebra approach to epidemiology* (EP/E006280/1, all authors, 2007–2010). We thank the anonymous referees for their helpful comments.

References

[1] R.M. Anderson and R.M. May. The population-dynamics of micro-parasites and their invertebrate hosts. *Philosophical transactions of the Royal Society of London Series B*, 291:451–524, 1981.

[2] J.C.M. Baeten. A brief history of process algebra. *Theoretical Computer Science*, 335(2/3):131–146, 2005.

[3] M. Begon, M. Bennet, R.G. Bowers, N.P. French, S.M. Hazel, and J. Turner. A clarification of transmission terms in host-microparasite models: numbers, densities and areas. *Epidemiology and infection*, 129:147–153, 2002.

[4] M. Begon, C.R. Townsend, and J.L. Harper. *Ecology: From Individuals to Ecosystems*. Wiley-Blackwell, 2005. 4th edition.

[5] S. Benkirane, J Hillston, C. McCaig, R. Norman, and C. Shankland. Improved continuous approximation of PEPA models through epidemiological examples. In *From Biology to Concurrency and back, FBTC 2008*, volume 229 of *ENTCS*, pages 59–74. Elsevier, 2008.

[6] R.J.H. Beverton and S.J. Holt. *On the dynamics of exploited fish populations*, volume 19 of *Fisheries Investigations, Series 2*. H.M.S.O., 1957.

[7] L. Brodo, P. Degano, and C. Priami. A tool for quantitative analysis of calculus processes. In *ICALP Satellite Workshops*, pages 535–550, 2000.

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

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

[10] A.M. Dunn, M.J. Hatcher, R.S. Terry, and C. Tofts. Evolutionary ecology of vertically transmitted paprasites: transovarial transmission of a microsporidian sex ratio distorer in *Gammarus duebeni*. *Parasitology*, 111:S91–S109, 1995.

[11] R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete Mathematics: A foundation for computer science*. Addison-Wesley, 1989.

[12] M. J. Hatcher and C. Tofts. The evolution of polygenic sex determination with potential for environmental manipulation. Technical Report UMCS-95-4-2, Department of Computer Science, University of Manchester, 1995.

[13] 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.

[14] E.G. Coffman Jr., T.T. Kadota, and L.A. Shepp. On the asymptotic optimality of first-fit storage allocation. *IEEE Trans. Soft. Eng.*, 11:235–239, 1985.

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

[16] K. Kuratowski. *Topology: Volume 1.* Polish Scientific Publishers, 1966.

[17] T.G. Kurtz. Solutions of ordinary differential equations as limits of pure jump markov processes. *Journal of Applied Probability*, 7:49–58, 1970.

[18] S.A. Levin, B. Grenfell, A.Hastings, and A.S. Perelson. Mathematical and computational challenges in population biology and ecosystems science. *Science*, 275:334 – 343, 1997.

[19] C. McCaig. *From individuals to populations: changing scale in process algebra models of biological systems.* PhD thesis, University of Stirling, 2007. http://hdl.handle.net/1893/398.

[20] C. McCaig, M. Begon, R. Norman, and C. Shankland. A rigorous approach to investigating common assumptions about disease transmission: Process algebra as an emerging modelling methodology for epidemiology. *Theory in Biosciences*, 2010. Special issue on emerging modelling methodologies. In press.

[21] C. McCaig, M. Begon, R. Norman, and C. Shankland. A symbolic investigation of superspreaders. *Bulletin of Mathematical Biology*, 2010. Special issue on Algebraic Biology. In press.

[22] C. McCaig, R. Norman, and C. Shankland. Deriving mean field equations from large process algebra models. Technical Report CSM-175, Department of Computing Science and Mathematics, University of Stirling, March 2008. http://hdl.handle.net/1893/1584.

[23] C. McCaig, R. Norman, and C. Shankland. From individuals to populations: A symbolic process algebra approach to epidemiology. *Mathematics in Computer Science*, 2(3):139–155, 2009.

[24] R Monroy. A process algebra model of the immune system. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 3214 of *Lecture Notes in Computer Science*, pages 526–533. Springer-Verlag, 2004.

[25] 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.

[26] R. Pelánek. Fighting state space explosion: Review and evaluation. In *Proc. of Formal Methods for Industrial Critical Systems (FMICS'08)*, volume 5596, pages 37–52, 2009.

[27] D. Sumpter. *From Bee to Society: an agent based investigation of honeybee colonies*. PhD thesis, UMIST, 2000.

[28] D.J.T. Sumpter and D.S. Broomhead. Relating individual behaviour to population dynamics. *Proceedings of the Royal Society of London Series B*, 268:925–932, 2001.

[29] C. Tofts. Describing social insect behaviour using process algebra. *Transactions of the Society for Computer Simulation*, 9:227–283, 1992.

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

[31] C. Tofts. Exact, analytic, and locally approximate solutions to discrete event-simulation problems. *Simulation Practice and Theory*, 6:721–759, 1998.

[32] C. Tofts. Exploiting strong attractors to slaughter monsters - taming 10^{1500} states and beyond. Technical Report HPL-2006-121, HP Laboratories, Bristol, 2006.

[33] J. Turner, M. Begon, and R.G. Bowers. Modelling pathogen transmission: the interrelationship between local and global approaches. *Proceedings of the Royal Society of London Series B*, 270:105–112, 2002.

[34] R.W. West and J.R. Thompson. Models for the simple epidemic. *Mathematical Biosciences*, 141:29–39, 1997.

Appendix A. WSCCS

In this Appendix we summarise the formal syntax and semantics of WSCCS [30].

Appendix A.1. Syntax

Appendix A.1.1. Actions

Action names, $a \in Act$, are chosen from an arbitrary set and should be suggestive of the system being described. The inverse of the action a (typically input) is $\overline{a}$ (typically output) and the identity action is denoted by $\surd$. When actions must occur in parallel we denote the multiplication by $\#$ such that $a\#\overline{a} = \surd$. Actions form an abelian group with identity $\surd$ and the inverse of action a being $\overline{a}$. Actions occur instantaneously and have no duration.

Appendix A.1.2. Relative frequency expressions

Relative Frequency Expressions, e, are defined as follows with x ranging over a set of variable names and c ranging over a fixed field (e.g. $\mathbb{N}$ or $\mathbb{R}$):

$$e ::= x \mid c \mid e + e \mid e \times e \ .$$

In these expressions we have commutative and associative multiplication and addition, with multiplication distributing over addition.

Appendix A.1.3. Weights

The set of WSCCS weights $\mathscr{W}$, denoted by w_i, are of the form $e\omega^k$, $e = e\omega^0$. In such weights e is the relative frequency with which this choice should be taken and k is the priority of this choice with ω an infinite object, $\omega > e \ \forall \ e$. The following multiplication and addition rules apply with $k \geq k'$:

$$\begin{aligned} e\omega^k + f\omega^{k'} &= e\omega^k = f\omega^{k'} + e\omega^k \ , \\ e\omega^k + f\omega^k &= (e+f)\omega^k = f\omega^k + e\omega^k \ , \\ e\omega^k * f\omega^{k'} &= (ef)\omega^{k+k'} = f\omega^{k'} * e\omega^k \ . \end{aligned}$$

Appendix A.1.4. Grammar

The possible WSCCS expressions are given by the following BNF grammar:

$$A ::= X \mid a:A \mid \Sigma\{w_i.A_i \mid i \in I\} \mid A \times A \mid A\lceil L \mid \Theta(A) \mid A[S] \mid X \stackrel{\text{def}}{=} A \ .$$

Here $X \in \mathit{Var}$, a set of process variables; $a \in \mathit{Act}$, an action group; $w_i \in \mathscr{W}$, a set of weights; S is a set of renaming functions, $S : \mathit{Act} \to \mathit{Act}$ such that $S(\surd) = \surd$ and $\overline{S(a)} = S(\overline{a})$; action subsets $L \subseteq \mathit{Act}$ with $\surd \in L$; and arbitrary indexing sets I . The informal interpretation of the operators is as follows:

- 0 is defined as $\Sigma\{w_i.A_i \mid i \in \emptyset\}$ and is a process which cannot proceed, representing deadlock ;
- X the process bound to the variable X ;
- $a{:}A$ a process which can perform the action a becoming the process A ;
- $\Sigma\{w_i.A_i \mid i \in I\}$ the weighted choice between processes A_i , the weight of A_i being w_i . Considering a large number of repeated experiments of this process, we expect to see A_i chosen with relative frequency $w_i/\Sigma_{i\in I}w_i$. The binary plus operator can be used in place of the indexed sum i.e. writing $\Sigma\{1_1.a{:}0, 2_2.b{:}0\}$ as $1.a{:}0 + 2.b{:}0$;
- $A \times B$ the synchronous parallel composition of A and B . At each stage each process must perform an action with the composed process performing the composition (denoted #) of the individual actions, e.g. $a{:}A \times b{:}B$ yields $a\#b{:}(A \times B)$. This is a powerful operator: models are constructed by describing simple individuals and composing a number of those in parallel. McCaig [19] introduces an extended notation $A\{n\}$ which is syntactic sugar for n instances of process A in parallel, where $n \in \mathbb{N}$;

- $A\lceil L$ a process which can only perform actions in the group L . These are referred to as the *free* actions. This operator is used to enforce communication on actions $b \notin L$. Two processes in parallel may communicate when one carries out an action and the other carries out the matching co-action, e.g. *infect* and $\overline{infect}$. Communication can be used to model passing of information from one process to another, or to coordinate activity. Such communication is strictly two-way ;
- $\Theta(A)$ represents taking the prioritised parts of the process A only. A WSCCS model is not considered well-formed if ω weights appear and the $\Theta(A)$ operator is not used. The $\Theta(A)$ operator must be applied every time ω weights appear in the model, since its use makes clear that prioritised actions are executed in preference to other actions ;
- $A[S]$ represents A relabelled by the function S ;
- $X \stackrel{\text{def}}{=} A$ represents binding the process variable X to the expression A .

Appendix A.2. Semantics

The semantics of WSCCS is transition based, defining the actions that a process can perform and the weight with which a state can be reached. The operational rules of WSCCS, presented in Table A.7, follow the informal description of the operators given above. In particular note the two different arrows that feature in the table: $\stackrel{a}{\longrightarrow}$ represents a transition, associated with the action a ; and $\stackrel{w}{\longmapsto}$ represents a transition associated with a weight w . The auxiliary predicate $does_A(E)$, which denotes the ability of E to perform A after zero or more probabilistic actions, is well defined since only finitely branching choice expressions are allowed.

Appendix A.2.1. Abstract Bisimulation

Let Pr denote the set of closed expressions of WSCCS, according to the grammar of Appendix A.1.4.

Definition 2. *We define the probability of a transition:* $P \stackrel{p}{\longmapsto} P'$ *if* $P \stackrel{w}{\longmapsto} P'$ *and the total weight of transitions from* P *is* w'*, with* $p = w/w'$*.*

Definition 3. *We define an abstract notion of evolution as follows:*

$$P \stackrel{a[p]}{\longrightarrow} P' \text{ iff } P \stackrel{p_1}{\longmapsto} \ldots \stackrel{p_n}{\longmapsto} \stackrel{a}{\longrightarrow} P' \text{ with } p = \prod p_i \ .$$

Definition 4. *Let* S *be a set of processes, then*

$$P \stackrel{a[p]}{\longrightarrow} S \text{ iff } p = \sum \{p_i \mid P \stackrel{a[p_i]}{\longrightarrow} Q \text{ for some } Q \in S\} \ .$$

Definition 5. *We say an equivalence relation* $R \subseteq \mathcal{A} \times \mathcal{A}$ *is an* abstract bisimulation *if* $(P, Q) \in R$ *implies that*

$$\text{for all } S \in Pr/R \text{ and for all } p \in [0,1],\ P \stackrel{a[p]}{\longrightarrow} S \text{ iff } Q \stackrel{a[p]}{\longrightarrow} S \ .$$

As Tofts [30] remarks, this relation is not a congruence, but is a useful notion of equivalence.

$\overline{a{:}A \xrightarrow{a} A}$	$\overline{\sum\{w_i.A_i \mid i \in I\} \overset{w_i}{\longmapsto} A_i}$
$\dfrac{A \xrightarrow{a} A' \quad B \xrightarrow{b} B'}{A \times B \xrightarrow{a \# b} A' \times B'}$	$\dfrac{A \overset{w}{\longmapsto} A' \quad B \overset{v}{\longmapsto} B'}{A \times B \overset{wv}{\longmapsto} A' \times B'}$
$\dfrac{A \xrightarrow{a} A' \quad B \overset{w}{\longmapsto} B'}{A \times B \overset{w}{\longmapsto} A \times B'}$	$\dfrac{A \overset{w}{\longmapsto} A' \quad B \xrightarrow{a} B'}{A \times B \overset{w}{\longmapsto} A' \times B}$
$\dfrac{A \xrightarrow{a} A' \quad a \in L}{does_L(A)}$	$\dfrac{A \overset{w}{\longmapsto} A' \quad does_L(A')}{does_L(A)}$
$\dfrac{A \xrightarrow{a} A' \quad a \in L}{A \lceil L \xrightarrow{a} A' \lceil L}$	$\dfrac{A \overset{w}{\longmapsto} A' \quad does_L(A')}{A \lceil L \overset{w}{\longmapsto} A' \lceil L}$
$\dfrac{A \xrightarrow{a} A'}{A[S] \xrightarrow{S(a)} A'[S]}$	$\dfrac{A \overset{w}{\longmapsto} A'}{A[S] \overset{w}{\longmapsto} A'[S]}$
$\dfrac{A \xrightarrow{a} A' \quad X \overset{\text{def}}{=} A}{X \xrightarrow{a} A'}$	$\dfrac{A \overset{w}{\longmapsto} A' \quad X \overset{\text{def}}{=} A}{X \overset{w}{\longmapsto} A'}$
$\dfrac{A \xrightarrow{a} A'}{\Theta(A) \xrightarrow{a} \Theta(A')}$	$\dfrac{A \overset{n\omega^i}{\longmapsto} A' \nexists (j>i).A \overset{m\omega^j}{\longmapsto} A''}{\Theta(A) \overset{n}{\longmapsto} \Theta(A')}$

Table A.7: Operational rules for WSCCS

Appendix B. Auxiliary Definitions

Here we present definitions for the auxiliary functions used in the pseudocode of Figs. 4 and 5. Firstly, define the general form of a model $\mathcal{M}$ as

$$\mathcal{M} = \Theta((A_1 \times \ldots \times A_n) \lceil \{L\}) \; .$$

Given a *serial* process $A = w_1.a_1 : A1 + w_2.a_2 : A2 + ... + w_m.a_m : Am$ define transitions$(A) = \{w_1.a_1{:}A1, w_2.a_2{:}A2, ..., w_m.a_m{:}Am\}$. Given a *parallel* process $A = A1 \times A2 \times ... \times Am$ define components$(A) = \{A1, A2, ...Am\}$.

For any model M, define the set of communicating actions inductively over the grammar of M as

$$\begin{aligned}
\text{comm_acts}(a : A) &= \{a\} \\
\text{comm_acts}(\Sigma\{w_i.A_i \mid i \in I\}) &= \bigcup_{i \in I} \text{comm_acts}(A_i) \\
\text{comm_acts}(A \times B) &= \text{comm_acts}(A) \cup \text{comm_acts}(B) \\
\text{comm_acts}(A \lceil L) &= \text{comm_acts}(A) \setminus L \\
\text{comm_acts}(\Theta(A)) &= \text{comm_acts}(A) \\
\text{comm_acts}(A[S]) &= \text{comm_acts}(A)[S] \\
&\quad \text{where S is a renaming function as above} \\
\text{comm_acts}(X) &= \text{comm_acts}(A) \text{ where } X \overset{\text{def}}{=} A
\end{aligned}$$

The function *get_comm_trans(A,M)* is defined in terms of the global set of communicating actions and the function *transitions*:

$$\begin{aligned} \text{get\_comm\_trans}(A, M) &= \{(w_i, a_i, Ai) \in \text{transitions}(A) \mid a_i \in \text{comm\_acts}(M)\} \\ \text{get\_non\_comm\_trans}(A, M) &= \{(w_i, a_i, Ai) \in \text{transitions}(A) \mid a_i \notin \text{comm\_acts}(M)\} \end{aligned}$$

For convenience, we drop the context M from the use of these functions since it is always for the whole model. Processes can then be classified as

$$\begin{aligned} \text{communicating}(A) &= (\text{get\_comm\_trans}(A) \neq \emptyset) \\ \text{probabilistic}(A) &= (\text{get\_comm\_trans}(A) = \emptyset) \\ \text{priority}(A) &= (\exists w = k\omega \mid k > 0 \text{ and } (w, a, A') \in \text{transitions}(A)) \ . \end{aligned}$$

Three predicates over agents were used in the pseudocode of Fig. 5. The first two are are only called with the argument collaborators(A,a) and the third is called with the agent A itself. All are defined using the syntax of agents.

$$\begin{aligned} \text{single\_act}(\text{collaborators}(A, a)) &= \forall B_i \in \text{collaborators}(A, a), \\ &\quad B_i \text{ has the form } \overline{a} : C + D \text{ where} \\ &\quad D \text{ does not include } \overline{a}. \\ \text{parallel\_act}(\text{collaborators}(A, a)) &= \forall B_i \in \text{collaborators}(A, a), \\ &\quad B_i \text{ has the form } \overline{a}^n : C + D \text{ where} \\ &\quad n > 1 \text{ and } D \text{ may include } \overline{a}. \\ \text{multiple\_act}(A) &= A \text{ has the form } a_1 : A1 + a_2 : A2 + A3 \text{ where} \\ &\quad A3 \text{ does not include } a_1 \text{ or } a_2, \text{ and} \\ &\quad a_1, a_2 \in \text{comm\_acts}(M). \end{aligned}$$

Finally, two functions were used to provide summation

$$\begin{aligned} \text{sum\_weights}(A) &= \Sigma\{w \mid (w, a, A) \in \text{transitions}(A)\} \\ \text{sum\_all\_acts}(\text{collaborators}(A, a)) &= \Sigma_i \text{replications}(c_i) * C_i \\ &\quad \text{where } (w_i, c_i, C_i) \in \mathit{transitions}(C) \text{ for} \\ &\quad C \in \mathit{collaborators}(A, a) \text{ and } c_i = \overline{a}^n \text{ where} \\ &\quad n > 1 \text{ and replications}(\overline{a}) = 1, \\ &\quad \text{replications}(\overline{a} \# \mathit{more}) = 1 + \text{replications}(\mathit{more}) \end{aligned}$$

Appendix C. Multiple parallel communicating actions, Not prioritised

Here we present the full general term for the Multiple non-prioritised contacts case discussed in Section 3.5.5.

The agent in the source state takes the form

$$A = 1.a : A1 + 1.b : A2 \ ,$$

as in the *Single, Non-prioritised* case. The Ci can perform multiple actions, as in the *Multiple Communicating Actions, Prioritised* case, but replacing all weights by 1

$$Ci = 1^{c_i}.\overline{a}^{c_i} : Ci' + 1^{c_i-1}.\overline{a}^{c_i-1} : Ci' + ... + 1.\overline{a} : Ci' + 1.b : Ci' \ .$$

We make use of the Multinomial coefficient

$$\frac{Ci_t!}{(\prod_{j=1}^{c_i} n_{i,j}!)(Ci_t - \sum_{k=1}^{c_i} n_{i,k})!} \ ,$$

for each of the Ci agents that perform the inverse action, where $n_{i,k}$ is the number of Ci agents performing k instances of $\overline{a}$ at a particular time. The binomial coefficients in (C.1) below

$$\binom{A_t + (\sum_{p=1}^{m_x} Xp_t) - 1}{(\sum_{q=1}^{m_c} \sum_{r=1}^{c_q} r * n_{q,r}) - 1} \quad \text{and} \quad \binom{A_t + (\sum_{p=1}^{m_x} Xp_t)}{\sum_{q=1}^{m_c} \sum_{r=1}^{c_q} r * n_{q,r}}$$

come from the simplification of the product of the individual binomial coefficients for the numbers of A and Xj that communicate with a Ci. The overall general term for this case is

$$construct(A, w, a) = A_t \frac{f\Big(\Big(\prod_{i=1}^{m_c} \frac{Ci_t!}{(\prod_{j=1}^{c_i} n_{i,j}!)(Ci_t - \sum_{k=1}^{c_i} n_{i,k})!}\Big)\binom{A_t + (\sum_{p=1}^{m_x} Xp_t) - 1}{(\sum_{q=1}^{m_c} \sum_{r=1}^{c_q} r * n_{q,r}) - 1}\Big)}{f\Big(\Big(\prod_{i=1}^{m_c} \frac{Ci_t!}{(\prod_{j=1}^{c_i} n_{i,j}!)(Ci_t - \sum_{k=1}^{c_i} n_{i,k})!}\Big)\binom{A_t + \sum_{p=1}^{m_x} Xp_t}{\sum_{q=1}^{m_c} \sum_{r=1}^{c_q} r * n_{q,r}}\Big)} \quad \text{(C.1)}$$

where

$$f(x) = \sum_{n_{m_c,c_{m_c}}=0}^{Cm_c} \sum_{n_{m_c,c_{m_c}-1}=0}^{Cm_c - n_{m_c,c_{m_c}}} \cdots \sum_{n_{m_c,1}=0}^{Cm_c - \sum_{s=1}^{c_{m_c}} n_{m_c,s}} \sum_{n_{m_c-1,c_{m_c-1}}=0}^{C(m_c-1)} \cdots \sum_{n_{1,1}=0}^{C1 - \sum_{u=1}^{c_1} n_{1,u}} x \ ,$$

omitting the time subscript t to avoid confusion. m_c and m_x are as defined previously and ci is the maximum number of instances of $\overline{a}$ that Ci can perform. Since the agents performing the action are able to make more than two choices multinomial coefficients rather than binomial coefficients are used. These cannot be simplified in the same way as previously, leaving (C.1) as the general term for this form of communication. This is intractable in the MFE, therefore such communication is generally omitted from our models.