Higher moment analysis of a spatial stochastic process algebra

Marcel C. Guenther Jeremy T. Bradley

Imperial College London, 180 Queen's Gate,
London SW7 2AZ, United Kingdom,
Email: {mcg05,jb}@doc.ic.ac.uk

Abstract. We introduce a spatial stochastic process algebra called MASSPA, which provides a formal behavioural description of Markovian Agent Models, a spatial stochastic modelling framework. We provide a translation to a master equation which governs the underlying transition behaviour. This provides a means of simulation and thus comparison of numerical results with simulation that was previously not available. On the theoretical side, we develop a higher moment analysis to allow quantities such as variance to be produced for spatial stochastic models in performance analysis for the first time. We compare the simulation results against resulting ODEs for both mean and standard deviations of model component counts and finish by analysing a distributed wireless sensor network model.

Keywords: Higher Moment Analysis, Spatial Stochastic Process Algebra, Spatial Modelling, MAM, MASSPA

1 Introduction

Spatial modelling paradigms take into account localised behaviour of individuals or processes in the evaluation of a system. While some domains such as crowd dynamics modelling [20,3] are inherently linked to their environment, other areas which were traditionally analysed without the notion of space gained new insights from spatial dynamics, e.g. the spatial Lotka–Volterra model [8]. As further examples, spatial topology has a modelling impact in wide variety of application domains such as epidemiology [24], wireless sensor networks [13], fire propagation [7] and traffic modelling [6].

Capturing spatial information in the analysis of models comes at a higher computational cost. In discrete spatial modelling paradigms [24,20,13,10], which allow modellers to create lumped CTMCs with a finite number of locations, the state space explosion is even more severe than in non-spatial CTMCs as we need to keep track of the population size for each state in every location. In the past, spatial models could therefore only be analysed using stochastic simulation [12]. Today fluid approximation [17,14] can handle spatial models even if the population and the number of locations become large. Recent work on discrete space, continuous time models, has focused on the approximation of mean component counts, i.e. the mean population sizes of different states in different locations [24,20,13]. While approximations of the mean population sizes are important metrics, the evaluation of higher order moments is crucial to get a better understanding of the underlying stochastic process. Furthermore it has been shown that higher order moments can be used to get more accurate boundaries for passage time distributions [15]. In this paper we investigate the computation of higher (joint) moments for discrete spatial models. In particular we formalise the Markovian Agent Model (MAM) paradigm [13] by expressing it in a spatial stochastic process algebra and apply higher moment ODE analysis to MAMs for the first time. Moreover, we use stochastic simulation to verify our ODE approximation for mean and standard deviation of component counts in two examples.

The paper is organised as follows. In Sect. 2 we briefly introduce the MAM paradigm. Subsequently in Sect. 3 we summarise Engblom's approach [9] for the generation of ODEs for higher order stochastic moments of molecule counts in systems expressed by the chemical *Master Equation* [25]. This serves as the basis for the derivation of higher moment ODEs in MAMs (Sect. 4). In Sect. 5 we define a simple but expressive spatial stochastic process algebra for MAMs and show how to translate it to a set of mass action type reactions. In Sect. 6 we describe a wireless sensor network MAM and estimate its mean and standard deviation using ODEs.

2 Markovian Agent Models

In this section we briefly describe the key concepts of Markovian Agent Models (MAM)s. In [13] Gribaudo *et al.* describe a novel agent-based spatio-temporal model called Markovian Agent Model (MAM). The underlying CTMC of the lumped process is approximated using techniques explained in [1]. Space is assumed to be continuous, but for evaluation purposes it is discretised, for instance into a regular 2-dimensional grid. Agents are assumed to be distributed in space according to a spatial Poisson process. A Markovian Agent (MA) in a MAM is a simple sequential component that can have local transitions which occur at a specified exponential rate and possibly emit messages. Additionally each MA can have message induced transitions. When emitting a message, all neighbours of the emitting agent, i.e. agents that are able to receive the message, may execute an induced transition. The function which defines the notion of neighbourhood is the perception function $u(\cdot)$. In MAMs each agent is assumed to act autonomously, i.e. each agent can decide whether to process or ignore incoming messages. Therefore each induced transition in a MA has a probability of accepting an incoming message. Note that in contrast to other process algebras such as PEPA [16], MAMs have no strict form of synchronisation.

We assume that the density of agents of type $c \in C$ in state $i \in S^c$ at time t at location $v \in V$ is given by $\rho_i^c(t, v)$. Vector $\boldsymbol{\rho^c}(t, v)$ contains all $\rho_i^c(t, v),\ \forall i \in S^c$ and matrix $\boldsymbol{\rho}(t, v)$ contains all $\boldsymbol{\rho^c}(t, v),\ \forall c \in C$. If the constant total number of agents of type c in location v is $N^c(v)$ then there are

$$N_i^c(t, v) = \rho_i^c(t, v) N^c(v) \tag{1}$$

agents of type c in state i at location v at time t. Hence, we need to have $\sum \rho_i^c(t, v) = 1$. Moreover, each agent of type c in state j produces messages at a Poisson rate of

$$\beta_j^c = \lambda_j^c g_{jj}^c + \sum_{k \in S^c, k \neq j} q_{jk}^c g_{jk}^c \tag{2}$$

where g_{jk}^c is the number of messages that the agent produces when making a transition from state j to k. λ_j^c denotes the rate at which the agent produces messages while sojourning in state j and q_{jk} is the rate at which the agent moves from state j to k. Now let $u(j, c_j, v_j, i, c_i, v_i)$ denote the perception function that scales the intensity with which an agent of type c_i in state i at location v_i receives messages from another agent of type c_j in state j at location v_j. Generally $u(\cdot) \geq 0$. Two agents can communicate iff $u(\cdot) > 0$ for those agents. The rate at which an agent of type c receives messages in state i computed as follows

$$\gamma_{ii}^c(t, v) = \sum_{c' \in C} \sum_{j \in S^{c'}} \sum_{v_j \in V} u(j, c', v_j, i, c, v) \beta_j^{c'} \rho_j^{c'}(t, v_j) N^{c'}(v_j) \tag{3}$$

Note that γ_{ii}^c is a rate of a convoluted Poisson process $X_1^{c'} + X_2^{c'} + \ldots$ where each $X_\cdot^{c'}$ represents a rate $\beta_j^{c'}$ modulated by $u(\cdot)$ and scaled by the population size of the sending agent. This convolution is only accurate if we have a spatially independent population distribution. Finally let

$$K^c(t, v) = Q^c + \varGamma^c(t, v)[A^c - I] \tag{4}$$

be the infinitesimal matrix for the CTMC of agents of type c at time t in location v where Q^c is the time invariant infinitesimal matrix describing the rates of internal state transitions for agents of type c, I the identity matrix and A^c the acceptance matrix where each element a_{ij}^c describes the probability of accepting the message and move from state i to j. We say $a_{ii}^c = 1 - \sum_{j \neq i} a_{ij}^c$ is the probability of ignoring the message and assume $\sum_{j \neq i} a_{ij}^c \leq 1$. Hence each row in A^c sums to 1. $\varGamma^c$ is the diagonal matrix with entries γ_{ii} and since all rows in $A^c - I$ sum to 0 so do the rows in $\varGamma^c(t, v)[A^c - I]$. Thus $K^c(t, v)$ is a valid infinitesimal matrix for the CTMC describing the evolution of $\boldsymbol{\rho^c}(t, v)$ at time t. Vector $\boldsymbol{\rho^c}(t, v)$, which represents the mean values of the underlying stochastic agent densities for states of agent class c at location v, can be approximated using the following equations

$$\begin{cases} \boldsymbol{\rho^c}(0, v) & t = 0 \\ \frac{\delta \boldsymbol{\rho^c}(t,v)}{\delta t} = \boldsymbol{\rho^c}(t, v) K^c(t, v) & t > 0 \end{cases} \tag{5}$$

Note that Eq. 5 uses the mean field assumption $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$ (see [1,18,22]). Further generalisations for MAMs which allow multiple types of messages and location dependent transition rates and dynamic MAMs exist but are omitted here. For more details see [4,6].

3 Higher moment ODEs for the Master Equation

In [9] Engblom derives a general expression for the derivation of ODEs for higher (joint) moments for models that can be expressed by the *Master Equation* [25]. Similar derivations can be found in [2,11]. In this section we summarise Engblom's work briefly. Later in Sect. 4 we use his technique to obtain ODEs for any higher (joint) moments in MAMs.

In a model of chemical reactions there are D different species and $|R|$ different reactions. $p(x,t)$ describes the probability distribution of the molecule count vector $x \in \mathbb{Z}^{D+} = \{0,1,2,\dots\}^D$ at time t. Note that $X(t)$ is a stochastic process. The propensity rate of a reaction $r \in R$ is described by $w_r : \mathbb{Z}^{D+} \to \mathbb{R}$. The change in x for reaction $r \in R$ is defined by $x_r = x + n_r \xrightarrow{w_r(x_r)} x$ where n_r are the negated stoichiometric coefficients. The resulting Master Equation is

$$\frac{\delta p(x,t)}{\delta t} = \sum_{\substack{r=1\\ x+n_r^- \geq 0}}^{R} \underbrace{w_r(x+n_r)p(x+n_r,t)}_{\text{incoming rate}} - \sum_{\substack{r=1\\ x-n_r^+ \geq 0}}^{R} \underbrace{w_r(x)p(x,t)}_{\text{outgoing rate}} \tag{6}$$

where $n_r = n_r^+ + n_r^-$ reflect the decrease and increase in the molecule count for reaction $r \in R$ respectively. In contrast to systems biology literature where reaction r usually changes x to $x + n_r^+$ and $x + n_r^-$, Engblom assumes that reaction r changes x to $x - n_r^+$ and $x - n_r^-$, i.e. $n_r^i = -1$ implies that the number of molecule i increases by 1 as a result of reaction r. Also note that integrating the Master Equation will yield the Chapman–Kolmogorov equation for the underlying CTMC. Engblom further shows that

$$\sum_{x \geq 0} T(x) \frac{\delta p(x,t)}{\delta t} = \sum_{r=1}^{R} \mathbb{E}[(T(X - n_r) - T(X))w_r(X)] \tag{7}$$

where $T : Z^{D+} \to R$ is a suitable test-function in form of a polynomial and $w_r(x) = 0 \ \forall x \ngeq n_r^+$. Taking $T(x) = x_i$ for example yields the differential equation for the mean of molecule count x_i

$$\dot{\mathbb{E}}[X_i] = \frac{\delta \mathbb{E}[X_i]}{\delta t} = \frac{\delta \mu_i}{\delta t} = -\sum_{r=1}^{R} n_r^i \mathbb{E}[w_r(X)] \tag{8}$$

Extending Eq. 7 to higher (joint) moments is straightforward.

4 Higher (joint) moments in MAMs

To date there is no derivation of ODEs for the computation of higher (joint) moments for component counts in MAMs. In [1,4] the authors only approximate the evolution of the mean of the component densities assuming $\mathbb{E}[XY] \approx \mathbb{E}[X]\mathbb{E}[Y]$. In this section we derive an expression for the second moment for MAM component counts using Engblom's technique outlined in Sect. 3.

To map MAMs to the Master Equation we need to define an equivalent set of reactions along with the propensities and the negated stoichiometric coefficients. For simplicity we assume that a MAM has a single location and one type of agent. In [**?**] we show that this assumption can be generalised to derive ODEs for more complex MAM extensions. Since MAs are autonomous there is no synchronisation between any two agents. In the language of chemical reactions that implies that any reaction expressing an agent transition from state $i \in S$ to state $j \in S$ has the following form

$$\boldsymbol{N}(t) + n_r \xrightarrow{N_i(t)K_{ij}(\boldsymbol{N}(t))} \boldsymbol{N}(t) \tag{9}$$

with negated stoichiometric coefficient $n_r^i = 1$, $n_r^j = -1$ and $n_r^k = 0 \; \forall k \in S\backslash\{i,j\}$ and propensity rate $w_r = w_{ij}(\boldsymbol{N}(t)) = N_i(t)K_{ij}(\boldsymbol{N}(t))$. Note that K_{ij} is indeed the ij^{th} element of the K matrix. In contrast to the definition of K in Eq. 4 the parameters of the matrix have changed from (t,v) to $\boldsymbol{N}(t)$. Formally the parameters are equivalent, since all factors in Eq. 3 apart from the agent distributions are constant with respect to time and location when considering a single location MAM. We can now create such a reaction for every pair of states $(i,j) \in S \times S$, $i \neq j$ with negated stoichiometric coefficients $n^i = 1$ and $n^j = -1$. Note that all reactions of type (i,j) encapsulate both internal agent transitions and transitions induced by incoming messages as the K matrix combines both internal and induced transition rates. If no transition is possible then $K_{ij}(\boldsymbol{N}(t)) = 0$. Having translated our simplified MAM into a set of reactions we can use Eq. 7 to derive the ODEs for the mean value, the first joint moment and the second order moment of the component counts. We start with the mean. Using Eq. 8 we get

$$\begin{aligned}\dot{\mathbb{E}}[N_i(t)] &= -\sum_{r}^{R} n_r^i \mathbb{E}[w_r(\boldsymbol{N}(t))] \\ &= \sum_{j\in S, j\neq i} \mathbb{E}[N_j(t)K_{ji}(\boldsymbol{N}(t))] - \sum_{j\in S, j\neq i} \mathbb{E}[N_i(t)K_{ij}(\boldsymbol{N}(t))] \end{aligned} \tag{10}$$

since

$$\sum_{j\in S, j\neq i} K_{ij}(\boldsymbol{N}(t)) = -K_{ii}(\boldsymbol{N}(t)) \tag{11}$$

Eq. 10 becomes

$$\dot{\mathbb{E}}[N_i(t)] = \sum_{j\in S} \mathbb{E}[N_j(t)K_{ji}(\boldsymbol{N}(t))] \tag{12}$$

applying the mean field approximation to Eq. 12 yields the component count equivalent to Eq. 5 for a MAM with only one location and one agent class

$$\dot{\mathbb{E}}[N_i(t)] \approx \sum_{j \in S} \mathbb{E}[N_j(t)] K_{ji}(\mathbb{E}[\boldsymbol{N}(t)]) \tag{13}$$

Similarly using Eq. 7 with $T(\boldsymbol{N(t)}) = N_i(t)N_j(t)$ we get the following ODE for the first joint moment

$$\begin{aligned}\dot{\mathbb{E}}[N_i(t)N_j(t)] &= \sum_r^R \mathbb{E}[((N_i(t) - n_r^i)(N_j(t) - n_r^j) - N_i(t)N_j(t))w_r(\boldsymbol{N}(t))] \\ &= -\sum_r^R n_r^i \mathbb{E}[N_j(t) w_r(\boldsymbol{N}(t))] - \sum_r^R n_r^j \mathbb{E}[N_i(t) w_r(\boldsymbol{N}(t))] \\ &\quad + \sum_r^R n_r^i n_r^j \mathbb{E}[w_r(\boldsymbol{N}(t))] \end{aligned} \tag{14}$$

which becomes

$$\begin{aligned}\dot{\mathbb{E}}[N_i(t)N_j(t)] = \mathbb{E}[N_i(t) \sum_{k \in S} N_k(t) K_{kj}(\boldsymbol{N}(t))] + \mathbb{E}[N_j(t) \sum_{k \in S} N_k(t) K_{ki}(\boldsymbol{N}(t))] \\ + \sum_r^R n_r^i n_r^j \mathbb{E}[w_r(\boldsymbol{N}(t))] \end{aligned} \tag{15}$$

where $\sum_r^R n_r^i n_r^j \mathbb{E}[w_r(\boldsymbol{N}(t))]$

$$= \begin{cases} -\mathbb{E}[N_i(t)K_{ij}(\boldsymbol{N}(t))] - \mathbb{E}[N_j(t)K_{ji}(\boldsymbol{N}(t))] & i \neq j \\ -\mathbb{E}[N_i(t)K_{ii}(\boldsymbol{N}(t))] + \mathbb{E}[\sum_{j \in S, j \neq i} N_j(t)K_{ji}(\boldsymbol{N}(t))] & i = j \end{cases} \tag{16}$$

The expansion for $i \neq j$ follows from the observation that the only reactions which have non-zero negated stoichiometric coefficient products $n_r^i n_r^j$ are those resulting from transitions from i to j and from j to i. For $i = j$, however, we can consider all reactions involving i, since all of these will yield $(n_r^i)^2 = 1$. Substituting Eq. 16 into Eq. 15 gives the required definition of the first joint moment and the second order moment in MAMs with one location and one agent class. In [**?**] we show the corresponding ODEs for MAMs with multiple agents, messages and locations.

5 Markovian Agent Spatial Stochastic Process Algebra

In this section we formally introduce a new spatial stochastic process algebra for Markovian Agent Models (MAM)s which we term MASSPA. First we describe a process algebra that allows us to define Markovian Agents. This process algebra is a blend between π-calculus [21] and a purely sequential version of PEPA [16], with passive actions and action names removed. To describe the spatial dynamics of MAMs we subsequently define the space, the distribution of agents in space and the perception function $u(\cdot)$. The notation for the spatial aspects of the model is similar to the one presented in [10,**?**]. Having defined MASSPA we give a simple example (Sect. 5.1) and show how MASSPA can be translated into chemical reactions (Sect. 5.2). The grammar of MASSPA is as follows:

$$\boxed{\begin{array}{l} S ::= \alpha.S \mid S + S \mid ?(m,p).S \mid \alpha!(m,g).S \mid C_S \mid \emptyset \\ P ::= P \underset{u(\cdot)}{\bowtie} P \end{array}}$$

where S denotes a sequential MA, C_S a sequential constant and $\emptyset$ the nil process. The basic MASSPA operators can be interpreted as follows:

Prefix: $\alpha.S'$ describes the possibility of a transition from the current process to process S'. This transition happens at rate α.
Choice: At a given time a process defined as $S + T$ can either behave as S or as T.
Constant: Assign names to patterns of behaviour associated with components, e.g. $N \overset{def}{=} S$. In this case N behaves the exact same way as S.
Message Sending: $\alpha!(m,g).S'$ describes a transition from the current process S to process S' where on average $g \in \mathbb{R}^{>0}$ messages of type m are sent as part of this transition. In fact g represents the parameter of a Poisson distribution, which defines the random number of messages generated as part of the transition.
Message Reception: $?(m,p).S'$ describes the possibility of a transition from the current process to process S' that can be induced by an incoming messages of type m. p is the probability of accepting such a message. For each type of message $m \in M$ that a process S listens to, we have to have $0 \leq \sum_{(m,p)\in S} p \leq 1$, i.e. each process can consume at most one type of message at a time.
Parallel: $P \underset{u(\cdot)}{\bowtie} Q$ means that agent populations (see definition below) P and Q, possibly located in different locations, operate in parallel. The perception function $u(\cdot)$ governs the message exchange between P and Q. This operator is only used to specify our operational semantics. In general we assume all agents in all locations act in parallel under $u(\cdot)$.

We now look at the definition of space, the agent populations and the perception function $u(\cdot)$. We begin with the space. Generally the space can take any discretised form as long as it is finite. Two basic examples are finite 2/3-dimensional regular rectangular and radial grids where each cell/location in the grid has a unique label $l \in L$, e.g. $L = \{A, B, C\}$ or $L = \{(0,0), (0,1), \ldots, (x,z)\}$. Within a certain cell we assume a spatial Poisson distribution of agents. In practice we may simply fix the number of agents in a certain state in a given location l and argue that we can find a corresponding rate for a spatial Poisson process that would on average generate that many agents in l. Assume that we have defined the sequential agents in MASSPA and that C is the set of distinct agent types, i.e. agents that do not share derivative states. Let S_i be the set of all derivative states for a sequential MASSPA agent of type $i \in C$ (cf. Sect. 2), such that $S_i \cap S_j = \emptyset$, $\forall i, j \in C$, $i \neq j$. Furthermore let $S = \bigcup_{i \in C} S_i$ be the set of all agent states. We define the set of agent populations for agents of type $i \in C$ as $P_i = \{s@l : s \in S_i,\ l \in L\}$ and also $P = \bigcup_{i \in C} P_i$. The initial agent population distribution is defined by a mapping $d : P \to \mathbb{N}^{\geq 0}$. Finally we define $u : Ch \to \mathbb{R}^{\geq 0}$ where $Ch \subseteq P \times P \times M$ is the set of channels and M the set of all message labels. We say that population $c_1@l_1$ can send messages of type m to $c_2@l_2$ iff $u(\cdot)$ is defined for $(c_1@l_1, c_2@l_2, m) \in Ch$.

The corresponding structured operational semantics are described in Fig. 1. It provides a translation to a labelled transition system which consists of transitions $a \in \mathcal{L}_\alpha \cup \mathcal{L}_M$ where $\mathcal{L}_\alpha$ is the set of exponentially delayed transitions and $\mathcal{L}_M$, consisting of (α, m, g_m)-sending events and (α, m, p_m)-reception events, is the set of message transitions. Assume $l_1, l_2 \in L$, possibly such that $l_1 = l_2$. $F \xrightarrow{(\alpha,m,g_m)} F$ means F sends g_m messages of type m at rate α and $E@l_1 \underset{u(\cdot)}{\bowtie} F@l_2 \xrightarrow{(x*\alpha*g_m,m,p_m)} E'@l_1 \underset{u(\cdot)}{\bowtie} F@l_2$ states that $E@l_1$ goes to $E'@l_1$ at rate $x * \alpha * g_m$ with probability p_m as the result of receiving message(s) of type m.

Prefix

$$\frac{}{E \xrightarrow{\alpha} E'} (E \overset{def}{=} \alpha.E') \qquad \frac{}{F \xrightarrow{\alpha} F'} (F \overset{def}{=} \alpha!(m, g_m).F') \qquad \frac{}{F \xrightarrow{(\alpha,m,g_m)} F} (F \overset{def}{=} \alpha!(m, g_m).F')$$

Competitive Choice

$$\frac{E \xrightarrow{a} E'}{E+F \xrightarrow{a} E'} \qquad \frac{F \xrightarrow{a} F'}{E+F \xrightarrow{a} F'}$$

Parallel

$$\frac{E \xrightarrow{a} E'}{E@l_1 \underset{u(\cdot)}{\bowtie} F@l_2 \xrightarrow{a} E'@l_1 \underset{u(\cdot)}{\bowtie} F@l_2} \qquad \frac{F \xrightarrow{a} F'}{E@l_1 \underset{u(\cdot)}{\bowtie} F@l_2 \xrightarrow{a} E@l_1 \underset{u(\cdot)}{\bowtie} F'@l_2}$$

Message Exchange

$$\frac{F \xrightarrow{(\alpha,m,g_m)} F}{E@l_1 \underset{u(\cdot)}{\bowtie} F@l_2 \xrightarrow{(x*\alpha*g_m,m,p_m)} E'@l_1 \underset{u(\cdot)}{\bowtie} F@l_2} (E \overset{def}{=} ?(m, p_m).E' + \ldots, u(F@l_2, E@l_1, m) = x)$$

Constant

$$\frac{E \xrightarrow{\alpha} E'}{A \xrightarrow{\alpha} E'} \ (A \overset{def}{=} E)$$

Fig. 1. Operational semantics of MASSPA.

5.1 A simple MASSPA example

In the following we give a simple MASSPA example. In Sect. 6 we look at more complex model. The *simple MAM* (see Fig. 2) has locations $L = \{A, B, C\}$ containing populations of identical agents so that set of all states $S = \{on, \mathit{off}\}$. The agent populations are $P = \{on@A, \mathit{off}@A, on@B, \mathit{off}@B, on@C, \mathit{off}@C\}$.

$$\mathit{off} \overset{def}{=} ?(m,1).on$$
$$on \overset{def}{=} \mu!(m,1).\mathit{off}$$

$$u(on@A, \mathit{off}@B, m) = 1 \quad u(on@B, \mathit{off}@A, m) = 1$$
$$u(on@B, \mathit{off}@C, m) = 1 \quad u(on@C, \mathit{off}@B, m) = 1$$

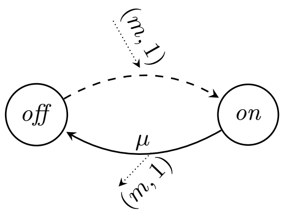

Fig. 2. A simple MAM with 3 locations.

5.2 Translating MASSPA into mass action type reactions

In this section we discuss how any MASSPA model can be translated into a system of chemical reactions (cf. Sect. 3). It is easy to see that the number of different agent populations $|P|$ is equivalent to the number of "molecule" species D. Hence, P is the set of available species. In the simple example above there is only one agent type, i.e. $|C| = 1$. As agents do not synchronise, each reaction has the following form $r : a@l \xrightarrow{w_r} b@l,\ a@l, b@l \in P_i,\ i \in C$. This is a different way of stating the assumption that all reactions for MAMs describe the evolution for a specific type of agent in one location (cf. [**?**]). Clearly the negated stoichiometric coefficients are $1, -1$ for $a@l, b@l$ respectively. To derive the reactions we need to define the w_r terms. By Eq. 9 we have $w_r = w_{a@l\ b@l} = N_{a@l}(t)K^i_{a@l\ b@l}(\boldsymbol{N}(t)) = N_a(t,l)K^i_{ab}(t,l),\ a, b \in S_i,\ i \in C,\ l \in L$. $N_{a@l}(t)$ is the number of agents of type i in state a at location l. To derive $K^i_{a@l\ b@l}(\boldsymbol{N}(t))$ we need to sum the local transition rate from $a@l$ to $b@l$ and the corresponding transition rate induced by incoming messages. The local rate of transitions from $a@l$ to $b@l$ can be obtained directly from the MASSPA agent definition, as this rate is location independent. Let states a, b be derived from process definitions A, B where $A \overset{def}{=} \cdots + \alpha.B + \cdots + \beta!(m, g_m).B$. To obtain the local transition rate we sum up all α and β rates for all prefix operations $\alpha.B$ and sending operations $\beta!(\cdot,\cdot).B$ in the definition of process A. The resulting rate is the ab^{th} element of matrix Q^i (cf. Eq. 4). Now we describe the rate of message induced transitions from $a@l$ to $b@l$. For any term $?(m, p_m).B$ in the definition of A we first need to find all channels $(x@l_1, a@l, m) \in Ch$ for which $u(\cdot)$ is defined. Note that $x \in S_j$ and potentially $a, b \notin S_j$. Let M_a be the set of all messages that can trigger process A to become process B and Ch_a the channels of type $(x@l_1, a@l, m)$ for which $u(\cdot)$ is defined. The total rate of message induced transitions in $K^i_{a@l\ b@l}(\boldsymbol{N}(t))$ from population $a@l$ to $b@l$ is

$$\sum_{m \in M_a} p_m(a,b) \sum_{(x@l_1, a@l, m) \in Ch_a} u(x@l_1, a@l, m)\ \beta(x,m)\ N_{x@l_1}(t) \tag{17}$$

where $p_m(a,b)$ is probability of an agent in state a to accept a message of type m and transit to state b and $\beta(x,m)$ is the total sending rate of messages m of an agent in state x. The former is the sum of all probabilities p_m for message $m \in M$ that are used in the $?(m, p_m).B$ terms in A. The latter is the sum of all rates $\beta * g_m$ from the $\beta!(m, g_m).X'$ terms in the definition of process X that state x refers to. Hence Eq. 17 is the same as the ab^{th} element of $\mathit{\Gamma}^i(t,l)$ in Eq. 4. Therefore MASSPA models do indeed capture the dynamics of static MAMs described in Sect. 2. To give a practical example we now show the reactions for the simple MAM described in Sect. 5.1. First we derive the matrices $K(t,v)$ for each of the three locations.

$$K(t,A) = K(t,C) = \begin{pmatrix} -\mu & \mu \\ N_{on@B}(t)\mu & -N_{on@B}(t)\mu \end{pmatrix} \tag{18}$$

$$K(t,B) = \begin{pmatrix} -\mu & \mu \\ (N_{on@A}(t) + N_{on@C}(t))\mu & -(N_{on@A}(t) + N_{on@C}(t))\mu \end{pmatrix} \tag{19}$$

The top left element in the rate matrix $K(t,X)$ is $K_{on@X\ on@X}$, the top right element $K_{on@X\ off@X}$, the bottom left element $K_{off@X\ on@X}$ and the bottom right element $K_{off@X\ off@X}$. The reactions for the model are

$$\begin{array}{lll} on@A & \xrightarrow{N_{on@A}(t)\mu} & off@A \\ off@A & \xrightarrow{N_{off@A}(t)N_{on@B}(t)\mu} & on@A \\ on@B & \xrightarrow{N_{on@B}(t)\mu} & off@B \\ off@B & \xrightarrow{N_{off@B}(t)(N_{on@A}(t)+N_{on@C}(t))\mu} & on@B \\ on@C & \xrightarrow{N_{on@C}(t)\mu} & off@C \\ off@C & \xrightarrow{N_{off@C}(t)N_{on@B}(t)\mu} & on@C \end{array} \tag{20}$$

It is important to note that sending agent populations (e.g. $on@B$ in the second reaction) act as catalysts but do not change state as they send a message as there is no synchronisation in MAMs. Hence messages are only ever sent while the sender sojourns in its current state.

Figure 3 shows numerical results for the simple MAM. We found that for small populations the mean approximation deteriorates as time goes on. In this model this is partially due to the fact that simulations for small agent populations tend to a state where no more communication is possible, i.e. $on@A = on@B =$

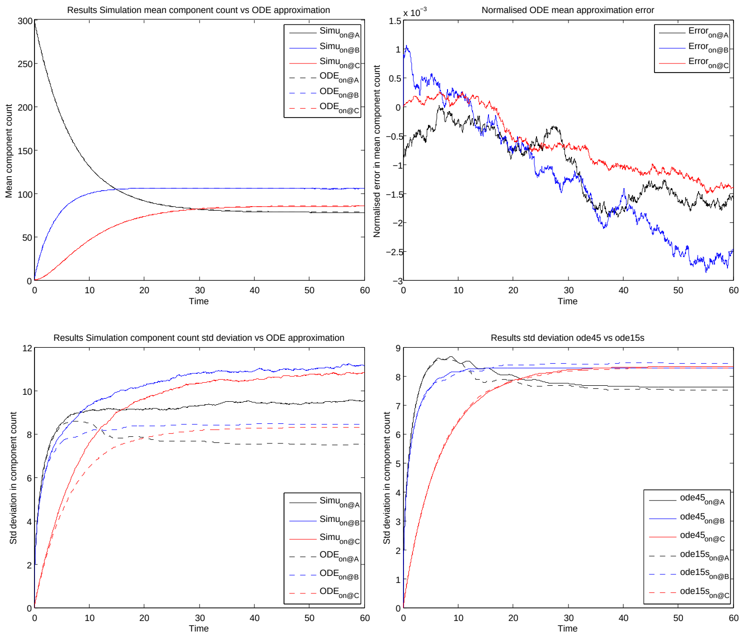

Fig. 3. Numerical results $on@A$, $on@B$, $on@C$ for the simple MAM with initial values $300, 0, 0, 300, 0, 450$ for $on@A$, $off@A$, $on@B$, $off@B$, $on@C$, $off@C$ and $\mu = 0.1$. Rates of reactions derived from message induced transitions were divided by the population size of the receiving agent type (see [?]). The error was computed by subtracting the ODE approximation from 10,000 averaged simulation traces and dividing by the population size of the corresponding agent. The first three graphs contain the ODE approximation solved using ode15s.

$on@C = 0$. It is easy to show that this another fixed point for the ODEs, too. However, the normalised error in the mean became smaller as we increased the population size, which agrees with theoretical mean field results [1,18]. The standard deviation approximation for the simple MAM is quantitatively inaccurate, but qualitatively good as it preserves the relative difference between the standard deviation of $on@A$, $on@B$ and $on@C$ around $t = 60$. More interestingly the comparison between the explicit *ode45* solver and the implicit *ode15s* solver in Matlab shows that the ODEs which determine the second order moments also have multiple fixed points. In this example *ode45* gives quantitatively and qualitatively worse results than *ode15s*.

6 Worked example: A simplified spatial WSN

Our worked example is a simplified version of the Wireless Sensor Network presented in [13]. In this MAM there are wireless sensor nodes (WSN)s which sample some quantity of their environment, e.g. temperature or humidity, and forward the samples to a sink location via a number of intermediate WSNs. The wireless sensor node is defined in Figs. 4. Each agent samples its environment at rate λ and propagates the measurement to the next agent that is closer to the sink. If it receives a sample from a WSN that is further away from the sink, it first buffers it and then sends it on to the next link at rate μ. Furthermore nodes may go to sleep when they have no buffered messages in order to save energy. WSNs may also fail without chance of recovery.

$$\begin{aligned}
WSN_{(0)} &\overset{def}{=} \lambda!(m,1).\,WSN_{(0)} + ?(m,1).\,WSN_{(1)} + \\
&\quad\ \phi_0.\,WSN_{(fail)} + \sigma_0.\,WSN_{(sleep)} \\
WSN_{(i)} &\overset{def}{=} \lambda!(m,1).\,WSN_{(i)} + ?(m,1).\,WSN_{(i+1)} + \\
&\quad\ \mu!(m,1).\,WSN_{(i-1)} + \phi_i.\,WSN_{(fail)} \\
WSN_{(b)} &\overset{def}{=} \lambda!(m,1).\,WSN_{(b)} + \mu!(m,1).\,WSN_{(b-1)} + \\
&\quad\ \phi_b.\,WSN_{(fail)} \\
WSN_{(sleep)} &\overset{def}{=} \sigma_s.\,WSN_{(0)} + \phi_s.\,WSN_{(fail)} \\
WSN_{(fail)} &\overset{def}{=} \emptyset
\end{aligned}$$

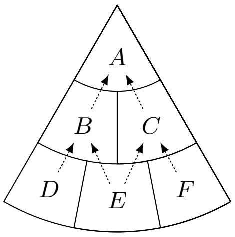

Fig. 4. A simple WSN in MASSPA. b denotes the buffer size. Arrows in the spatial layout on the right define $u(\cdot) = 1$ for locations L={A,B,C,D,E,F}. The set of states is S={0,1,. . . ,b,sleep,fail}.

.

We now compare the ODE approximations for the mean and standard deviation for the WSN MAM with the exact solution. The ODEs were solved in Matlab using *ode45* and *ode15s*. We experimented with various moment closures for the third order terms in the second order moment ODEs (see [**?**]). For the ODE traces below we used the $\mathbb{E}[XYZ] \approx \mathbb{E}[XY]\mathbb{E}[Z]$. For stochastic simulation we used Matlab's *simbiology* tool.

As can be seen in Fig. 5 the mean is approximated well by the ODEs. This holds for all quantities not only for the mean of the component counts in location B shown in Fig. 5. Furthermore we did not observe multiple fixed points for the first order ODEs, which could be due to the fact that the communication will never stop completely as WSNs always sample at rate λ. The more interesting observation made in this model is the behaviour of the ODEs for the second order moments. While the ODEs for WSN components counts in locations A, C, D, E and F were remarkably accurate irrespective of the choice of ODE solver, we

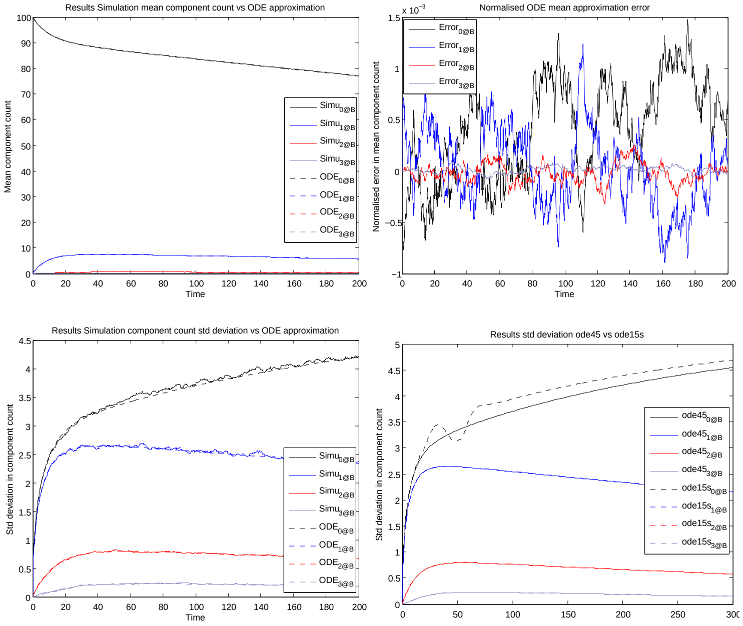

Fig. 5. Numerical results $0@B, 1@B, 2@B, 3@B$ for the WSN MAM with initial values $20, 100, 50, 20, 10, 90$ for $0@A, 0@B, 0@C, 0@D, 0@E, 0@F$ and $b = 3, \mu = 0.1, \lambda = 0.03, o_s = 0.0001, o_0 = 0.001$ and zero probability of node failure. Message induced reactions rates were divided by the population size of the receiving agent type (see [**?**]). The error was computed by subtracting the ODE approximation from 3,500 averaged simulation traces and dividing by the population size of the corresponding agent. The first three graphs contain the ODE approximations solved using ode45.

found that in location B there was a significant difference in the solution when applying *ode15s* as opposed to *ode45* in Matlab. In Fig. 5d *ode45* is more stable than *ode15s*. The two traces eventually converge around time $t = 2000$ so there is no indication that there are multiple fixed points in the second order ODEs.

7 Conclusion

In this paper we have derived a simple mapping from MAMs to the language of chemical reactions. This mapping is then used to derive second order ODEs for MAMs for the first time. To formalise the description of MAMs further we defined MASSPA, a process algebra for MAMs and showed how MASSPA can be translated into mass action type reactions. Moreover, we have given numerical examples for two models that we defined in MASSPA and shown that their first order ODE approximations are generally good for large populations. As for the second order ODE approximation we got very good results in the WSN model and less accurate results in the simple MAM. We also observed that comparing stochastic simulation traces to the ODE traces for small population sizes gives a good indication as to whether the second order ODEs become more accurate as we increase the overall population size. The $\mathbb{E}[XYZ] \approx \mathbb{E}[XY]\mathbb{E}[Z]$ moment closure assumption proved to be a simple but effective choice. The only closure which gave slightly better results for the simple MAM model was $\mathbb{E}[XYZ] \approx \mathbb{E}[X]\mathbb{E}[YZ]$, but for the WSN model this closure gave rather inaccurate second order approximations. Although the closure behaviour could be entirely model dependent it is not unlikely that $\mathbb{E}[XYZ] \approx \mathbb{E}[XY]\mathbb{E}[Z]$ works well for MAMs as the Z term always represents contributions from the K matrix. Further research is needed to investigate moment closures for MAMs and to find indicators for well behaved spatially motivated moment closures similar to those discussed in [22]. Should good moment closures for MAMs be hard to determine, we might need to look for further evaluation techniques such as those discussed in [23]. Having computed higher moments for ODEs it would be interesting to look at resulting passage time bounds that can be deduced using techniques described in [15].

Acknowledgements

We would like to thank Davide Cerotti and Marco Gribaudo for giving us valuable insights into the MAM paradigm. Moreover, we would like to thank Anton Stefanek and Richard Hayden for their input on moment closures techniques. Finally we would also like to thank the referees for their time and efforts in reviewing the original submission and their helpful comments which have certainly improved the paper.

References

1. Andrea Bobbio, Marco Gribaudo, and Miklos Telek. Mean Field Methods in Performance Analysis. In *Fifth International Conference on Quantitative Evaluation of SysTems*, pages 215–224. QEST, 2008.
2. Luca Bortolussi. On the Approximation of Stochastic Concurrent Constraint Programming by Master Equation. *Electronic Notes in Theoretical Computer Science*, 220(3):163–180, 2008.
3. Andrea Bracciali, Jane Hillston, Diego Latella, and Mieke Massink. Reconciling Population and Agent Models for Crowd Dynamics. In *3rd International Workshop on Logics, Agents, and Mobility*. LAM'10, 2010.
4. Davide Cerotti. *Interacting Markovian Agents.* PhD thesis, University of Torino, 2010.
5. Davide Cerotti, Marco Gribaudo, and Andrea Bobbio. Disaster Propagation in Heterogeneous Media via Markovian Agents. In Roberto Setola and Stefan Geretshuber, editors, *3rd International Workshop on Critical Information Infrastructures Security CRITIS*, volume 5508 of *Lecture Notes in Computer Science*, pages 328–335. Springer Berlin Heidelberg, 2009.

6. Davide Cerotti, Marco Gribaudo, and Andrea Bobbio. Presenting Dynamic Markovian Agents with a road tunnel application. In *IEEE International Symposium on Modeling Analysis Simulation of Computer and Telecommunication Systems MASCOTS*, pages 1–4. IEEE, 2009.
7. Davide Cerotti, Marco Gribaudo, Andrea Bobbio, Carlos Calafate, and Pietro Manzoni. A markovian agent model for fire propagation in outdoor environments. In *7th European Performance Engineering Workshop EPEW*, pages 131–146, 2010.
8. Ulf Dieckmann and Richard Law. *Relaxation projections and the method of moments*, chapter 21, pages 412–455. Number December in Cambridge studies in adaptive dynamics. Cambridge University Press, 2000.
9. Stefan Engblom. Computing the moments of high dimensional solutions of the master equation. *Applied Mathematics and Computation*, 180(2):498–515, 2006.
10. Vashti Galpin. Towards a spatial stochastic process algebra. In *Proceedings of the 7th Workshop on Process Algebra and Stochastically Timed Activities (PASTA)*, Edinburgh, 2008.
11. C S Gillespie. Moment-closure approximations for mass-action models. *IET systems biology*, 3(1):52–58, 2009.
12. Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361, 1977.
13. Marco Gribaudo, Davide Cerotti, and Andrea Bobbio. Analysis of on-off policies in sensor networks using interacting markovian agents. In *6th IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 300–305, 2008.
14. Richard A Hayden and Jeremy T Bradley. A fluid analysis framework for a Markovian process algebra. *Theoretical Computer Science*, 411(22-24):2260–2297, 2010.
15. Richard A Hayden, Anton Stefanek, and Jeremy T Bradley. Fluid computation of passage time distributions in large Markov models. *submitted to Theoretical Computer Science*, 2010.
16. Jane Hillston. A Compositional Approach to Performance Modelling. *Cambridge University Press*, page 158, 1996.
17. Jane Hillston. Fluid flow approximation of PEPA models. *Second International Conference on the Quantitative Evaluation of Systems QEST05*, pages 33–42, 2005.
18. Jean-Yves Le Boudec, David McDonald, and Jochen Mundinger. A Generic Mean Field Convergence Result for Systems of Interacting Objects. *Fourth International Conference on the Quantitative Evaluation of Systems QEST 2007*, pages 3–18, 2007.
19. Mieke Massink, D Latella, A Bracciali, and M D Harrison. A Scalable Fluid Flow Process Algebraic Approach to Emergency Egress Analysis. In *8th IEEE International Conference on Software Engineering and Formal Methods*, pages 169–180. IEEE, 2010.
20. Mieke Massink, Diego Latella, Andrea Bracciali, and Jane Hillston. Modelling Crowd Dynamics in Bio-PEPA. In *Proceedings of the 9th Workshop PASTA/Bio-PASTA*, pages 1–11, 2010.
21. Robin Milner. *Communicating and Mobile Systems: the Pi-Calculus.* Cambridge University Press, 1999.
22. David J Murrell, Ulf Dieckmann, and Richard Law. On moment closures for population dynamics in continuous space. *Journal of Theoretical Biology*, 229(3):421–432, 2004.
23. Otso Ovaskainen and Stephen J Cornell. Space and stochasticity in population dynamics. *Proceedings of the National Academy of Sciences of the United States of America*, 103(34):12781–12786, 2006.

24. Anton Stefanek, Maria Vigliotti, and Jeremy T Bradley. Spatial extension of stochastic pi calculus. In *8th Workshop on Process Algebra and Stochastically Timed Activities*, pages 109–117, 2009.
25. N G Van Kampen. *Stochastic Processes in Physics and Chemistry*, volume 11 of *North-Holland personal library*. North-Holland, 1992.