PALOMA: A process algebra for located Markovian agents

PALOMA: A Process Algebra for Located Markovian Agents

Cheng Feng and Jane Hillston

LFCS, School of Informatics, University of Edinburgh
s1109873@sms.ed.ac.uk, jane.hillston@ed.ac.uk
http://www.quanticol.eu

Abstract. We present a novel stochastic process algebra that allows the expression of models representing systems comprised of populations of agents distributed over space, where the relative positions of agents influence their interaction. This language, PALOMA, is given both discrete and continuous semantics and it captures multi-class, multi-message Markovian agent models (M^2MAM). Here we present the definition of the language and both forms of semantics, and demonstrate the use of the language to model a flu epidemic under various quarantine regimes.

1 Introduction

Collective systems, comprised of many communicating entities and without centralised control, are becoming pervasive. Without any global knowledge, entities interact locally to create a system with discernible characteristics at the global level; a phenomenon sometimes termed *emergence*.

The notion of *locality* has spatial relationship implicit within it, and thus to faithfully capture these systems we have to be able to represent the spatial arrangement of entities and the constraints that this places on their communication. For example, interactions may only be allowed for entities which are co-located or within a certain physical distance of each other, or space may be segmented in such a way that even physically close entities are unable to communicate. Furthermore movement can be a crucial aspect of the behaviour of entities within the system. Therefore it becomes essential to develop modelling formalisms in which space is captured explicitly, and in which the same entity in different locations can be distinguished. Meanwhile, given the scale of collective systems, which often rely on large populations of entities in order to meet their objectives, we must also find efficient mechanisms both to express and to analyse the developed models.

Multi-class Multi-message Markovian Agents Models (M^2MAM) have recently been proposed by Cerrotti *et al.* as a suitable framework for modelling collective systems comprised of populations of agents which are spatially distributed [1]. Several case studies [2–4] demonstrated that this is a powerful and useful framework. However the model specification is in terms of matrices which capture the possible interactions and influences between agents. This form of specification is highly demanding on the modeller and prone to error. In this paper we

propose a process algebra to capture models within the M^{2}MAM framework and circumvent the rather cumbersome matrix specification. The process algebra, PALOMA, is equipped with both discrete event and differential semantics. This high-level specification is human-readable, less error-prone, amenable to automated checking and supports automated derivation of the executable models defined by the semantics. More specifically, the discrete semantics provides the theoretical foundation for discrete event simulation whilst the differential semantics allows us to automatically derive the matrices, and thus the underlying mean-field model of the M^{2}MAM in the form of initial value problems.

The paper is structured as follows. We briefly introduce the concepts of M^{2}MAMs in the next section. Section 3 presents the syntax and discrete semantics of PALOMA. This is followed by the differential semantics in Section 4. In Section 5, a case study in which we apply PALOMA to the modelling of the spread of flu in a multi-community society is presented. Finally, Sections 6 and 7 discuss related work, future research and draw final conclusions.

2 M^{2}MAM

In this section, we briefly introduce the key concepts of M^{2}MAMs, originally presented by Cerrotti *et al.* in [1]. M^{2}MAMs consist of a collection of Markov agents (MAs) distributed over space, which is represented by a finite set of locations. Each MA has a location attribute and can be denoted by a finite state machine in which two types of transitions can happen: *local* transitions and *induced* transitions. Local transitions occur whenever the MA changes its state spontaneously with a delay governed by an exponential distribution. Local transitions can also possibly emit messages that can cause the occurrence of induced transitions in MAs at the same or other locations. This enables location-based asynchronous interaction between MAs. The reception of a message is governed by the *perception function*, which depends on both the location and state of the sender and receiver MAs. When a MA receives a message, it can either ignore or accept it. In the latter case, the agent will change its state immediately by performing an induced transition.

Following [1], we use $\mathrm{MA}^c(\ell)$ to denote a MA of class c in location ℓ. A $\mathrm{MA}^c(\ell)$ can be defined as a tuple $\{Q^c(\ell), \Lambda^c(\ell), G^c(\ell, m), A^c(\ell, m), \pi_0^c(\ell)\}$, in which:

- $Q^c(\ell) = [q_{ij}^c(\ell)]$ is a $n_c \times n_c$ matrix, in which each element $q_{ij}^c(\ell)$ represents the rate of the local transition from state i to state j, with $q_{ii}^c(\ell) = -\sum_{j \neq i}^{n_c} q_{ij}^c(\ell)$ where n_c is the number of states of a MA of class c.
- $\Lambda^c(\ell) = [\lambda_i^c(\ell)]$ is a vector, in which each element $\lambda_i^c(\ell)$ denotes the rate of a self-jump transition which reenters the same state i, for a MA of class c.
- $G^c(\ell, m) = [g_{ij}^c(\ell, m)]$ is a $n_c \times n_c$ matrix in which each element $g_{ij}^c(\ell, m)$ describes the probability of $\mathrm{MA}^c(\ell)$ generating a message of type m during a local transition from state i to state j.
- $A^c(\ell, m) = [a_{ij}^c(\ell, m)]$ is a $n_c \times n_c$ matrix, in which each element $a_{ij}^c(\ell, m)$ $(i \neq j)$ describes the acceptance probability of message type m for the $\mathrm{MA}^c(\ell)$,

with induced transition from state i to state j whereas $a^c_{ii}(\ell, m)$ denotes the probability of dropping this message, and $a^c_{ii}(\ell, m) = 1 - \sum_{j \neq i} a^c_{ij}(\ell, m)$.
- $\pi^c_0(\ell)$ is the initial state probability distribution of an agent of class c in location ℓ.

2.1 Model Analysis

The density of agents of class c in state i, in location ℓ at time t is denoted $p^c_i(\ell, t)$. In M^{2}MAMs, the density of agents of each class in a location is assumed to remain constant, i.e. the value of $\sum_{i=1}^{n_c} p^c_i(\ell, t) = P^c(\ell)$ is invariant. We use a vector $p^c(\ell, t) = [p^c_i(\ell, t)]$ to denote the state density distribution of agents of class c in location ℓ and at time t. The analysis of interest is the transient evolution of $p^c(\ell, t)$. It can be computed by solving a set of coupled ODEs.

First of all, the total rate at which messages of type m are generated by a MA of class c in state j and location ℓ can be computed by:

$$\beta^c_j(\ell, m) = \lambda^c_j(\ell) g^c_{jj}(\ell, m) + \sum_{k \neq j} q^c_{jk}(\ell) g^c_{jk}(\ell, m) \tag{1}$$

where the first term on the right hand side of the above equation gives the rate at which messages of type m are generated by the MA in state j by a self-jump transition, whereas the second term denotes the rate of message generation by the MA during a local transition from state j to another state.

With $\beta^c_j(\ell, m)$, we can compute $\gamma^c_{ii}(\ell, m, t)$, the total reception rate of messages of type m by a MA of class c in state i and location ℓ, at time t. The rate $\gamma^c_{ii}(\ell, m, t)$ is contributed to by all the messages of type m generated by MAs of all classes in all states and all locations, as long as they can be perceived by the receiver MA. Thus, $\gamma^c_{ii}(\ell, m, t)$ is obtained by the following equation:

$$\gamma^c_{ii}(\ell, m, t) = \sum_{\ell' \in \mathcal{V}} \sum_{c'=1}^{C} \sum_{j=1}^{n_{c'}} u_m(\ell, c, i, \ell', c', j) \beta^{c'}_j(\ell', m) p^{c'}_j(\ell', t) \tag{2}$$

where $\mathcal{C} = \{1, \ldots, C\}$ is the set of agent classes in the model, $\mathcal{V}$ is the location set $u_m(\ell, c, i, \ell', c', j)$ is the perception function of message m, whose value represents the probability that an agent of class c, in state i, and in location ℓ perceives a message m sent by an agent of class c' in state j and in position ℓ'.

Finally, we use a diagonal matrix, $\Gamma^c(\ell, m, t) = diag(\gamma^c_{ii}(\ell, m, t))$ to collect the rates in Equation (2), and the infinitesimal generator matrix $K^c(\ell, t)$ for the population CTMC of agents of class c in location ℓ at time t can be obtained by:

$$K^c(\ell, t) = Q^c(\ell) + \sum_m \Gamma^c(\ell, m, t)[A^c(\ell, m) - I] \tag{3}$$

where I is the identity matrix, the first term on the right hand side of (3) is the infinitesimal generator matrix of the CTMC for local transitions, and the second term gives the infinitesimal generator matrix for induced transitions which uses

the averaged effect to approximate the effect of all other agents' interactions with an agent of class c in location ℓ at time t.

Shifting to a mean field view, the transient evolution of $p^c(\ell, t)$ is captured by standard Kolmogorov equations with initial conditions $p^c(\ell, t_0) = P^c(\ell)\pi_0^c(\ell)$ for all ℓ and c:

$$\frac{\mathrm{d}p^c(\ell, t)}{\mathrm{d}t} = p^c(\ell, t)K^c(\ell, t) \quad \forall(\ell, c) \tag{4}$$

3 PALOMA

PALOMA, the Process Algebra of Located Markovian Agents, is intended to provide a simple process algebra-based formalism which can be used to generate models in the M^{2}MAM framework. M^{2}MAM is used to generate a mean field model, but being based on Markovian agents it is also amenable to a discrete interpretation. As mentioned previously, PALOMA is equipped with both discrete and differential semantics. In this section, we first introduce the discrete interpretation, considering individual agents. We will then make the shift to population CTMC and ultimately a mean field model in the next section.

3.1 Syntax

In keeping with the M^{2}MAM framework, in PALOMA each agent is a finite state machine and the language is *conservative* in the sense that no agents are spawned or destroyed during the evolution of a model (although they can cease to change state). Thus the language has a two level grammar:

$$X(\ell) ::= !(\alpha, r).X(\ell) \mid ?(\alpha, p).X(\ell) \mid X(\ell) + X(\ell) \qquad P ::= X(\ell) \mid P \parallel P$$

Agents are parameterised by a *location*, here denoted by ℓ. Agents can undertake two types of actions, *spontaneous* actions, denoted $!(\alpha, r)$, and *induced* actions, denoted $?(\alpha, p)$. When an agent performs a spontaneous action, it does so with a given rate r, which is taken to be the parameter of an exponential distribution, where $1/r$ is the expected duration of the action. Spontaneous actions are broadcast to the entire system, and can induce change in any other agent which enables an induced action with the matching type α. An induced action has an associated probability p, which records the probability that the agent responds to a spontaneous action of the same type. In the style of the Calculus of Broadcasting Systems [5], this can be thought of as the probability that the agent *listens* as opposed to simply *hearing*. Alternative behaviours are represented by the standard choice operator, $+$. A choice between spontaneous actions is resolved via the race policy, based on their corresponding rates. We assume that there is never a choice between induced actions of the same type.

A model, P, consists of a number of agents composed in parallel. There is no direct communication between agents, for example in the style of shared actions in PEPA [6]. Instead, all interaction is via spontaneous/induced actions. When

an action is induced in an agent the extent of its impact is specified by a perception function, $u(\alpha, \ell, X, \ell', X')$[1]. This is a further probability which, given the locations of the two agents, their current states and action type involved, determines the likelihood that the induced action occurs. For example, the perception function might have value 1 when the two agents are within a communication radius r of each other, but a value of 0 whenever the distance between them is greater than r. Obviously this gives a rich set of possible styles of interaction, but note that each agent with an induced action chooses independently whether to respond or not.

3.2 Semantics

The semantics proceeds in sequences of alternating steps. This can be regarded as a semi-Markov process: the first step, corresponding to the spontaneous action, determines a delay, whilst the second step is probabilistic and determines what the next state will be, as each possible induced action makes the choice of whether to respond, based both on its inherent probability of "listening" and the perception function. Since each agent makes such a decision independently, the probabilities can be multiplied to obtain the overall probability of a given next state. Correspondingly we define two transition relations $\longrightarrow$ and $\longrightarrow_{\mathcal{P}}$. These are shown in Figures 1 and 2 respectively.

In order to keep track of which agents have "heard" the messages which are broadcast by spontaneous actions we associate an *ether* element with the system, which provides the environment for *all* agents. This has a distinguished empty state E_0. As shown in rule SpA, a spontaneous action can only be initiated if the ether is currently empty, and no probabilistic transitions are enabled ($\not\longrightarrow_{\mathcal{P}}$). The resulting local state records that the ether contains the message α which originated with rate r at location ℓ from the state $!(\alpha, r).X(\ell)$, and that the continuation is subject to a probabilistic resolution. Any state awaiting probabilistic resolution is denoted $S^{\mathcal{P}}$. Note that $S^{\mathcal{P}}$ states will not be in the CTMC.

If the ether contains a message α then an agent enabling an induced α action may progress to a probabilistic state in which, with probability p, the continuation $X(\ell)$ is chosen, and with probability $1-p$, the continuation $?(\alpha, p).X(\ell)$ is chosen (rule InA). For other agents, their spontaneous actions are blocked until the current one has been fully broadcast and probabilistically resolved (rule NoSp). If the ether contains a message of type α then an agent enabling a spontaneous action of any type (including α) witnesses the ongoing action, enters a probabilistic state and awaits resolution. This ensures that only one spontaneous action can be in progress at a time. Note that there is no possibility of an agent "sharing" the α action as would be possible in a language such as CSP or PEPA. Similarly, if the ether contains a message of type α then an agent that enables an induced message of any other type simply witnesses the ongoing action, enters a probabilistic state and awaits resolution (rule NoIn). In some cases a spontaneous

[1] Here we do not need to explicitly specify the class of sender and receiver agents as it can be deduced by the state and the location attributes.

$$\mathsf{SpA} \quad E_0, !(\alpha, r).X(\ell) \xrightarrow{(\alpha,r)} [\alpha, r, \ell, X], X(\ell)^{\mathcal{P}} \quad (\not\rightarrow_{\mathcal{P}})$$

$$\mathsf{InA} \quad [\alpha, r, \ell', X'], ?(\alpha, p).X(\ell) \xrightarrow{(\alpha,r)} [\alpha, r, \ell', X'], (?(\alpha, p).X(\ell) +^{p} X(\ell))^{\mathcal{P}}$$

$$\mathsf{NoSp} \quad [\alpha, r, \ell', X'], !(\beta, s).X(\ell) \xrightarrow{(\alpha,r)} [\alpha, r, \ell', X'], !(\beta, s).X(\ell)^{\mathcal{P}}$$

$$\mathsf{NoIn} \quad [\alpha, r, \ell', X'], ?(\beta, p).X(\ell) \xrightarrow{(\alpha,r)} [\alpha, r, \ell', X'], ?(\beta, p).X(\ell)^{\mathcal{P}} \quad (\beta \neq \alpha)$$

$$\mathsf{Ch1} \ \frac{E, X_1(\ell) \xrightarrow{(\alpha,r)} E', X_1'(\ell')^{\mathcal{P}}}{E, X_1(\ell)+X_2(\ell) \xrightarrow{(\alpha,r)} E', X_1'(\ell')^{\mathcal{P}}} \qquad \mathsf{Ch2} \ \frac{E, X_2(\ell) \xrightarrow{(\alpha,r)} E', X_2'(\ell')^{\mathcal{P}}}{E, X_1(\ell)+X_2(\ell) \xrightarrow{(\alpha,r)} E', X_2'(\ell')^{\mathcal{P}}}$$

$$\mathsf{Par} \ \frac{E_1, X_1(\ell_1) \xrightarrow{(\alpha,r)} E', X_1'(\ell_1')^{\mathcal{P}} \qquad E_2, X_2(\ell_2) \xrightarrow{(\alpha,r)} E', X_2'(\ell_2')^{\mathcal{P}}}{(E_1, X_1(\ell_1)) \parallel (E_2, X_2(\ell_2)) \xrightarrow{(a,r)} E', (X_1'(\ell_1') \parallel X_2'(\ell_2'))^{\mathcal{P}}}$$

Fig. 1. The delay transition relation for PALOMA

action may not induce any actions in other agents. If this is the case the message will propagate, without impacting any other agents, except to put them into the trivial probabilistic state. Choice behaves as we would anticipate. We assume that within a choice both elements are in the same location as they correspond to a single agent. Parallel agents must agree on the single spontaneous action to take place, and consequently update the ether in the same way. A spontaneous action is deemed to be complete when all agents have moved to a probabilistic state. In this case a probabilistic resolution must be made to determine the next state. This is defined by the probabilistic transition relation, which will clear the ether and create the opportunity for the next spontaneous message.

Probabilistic resolutions are determined by a second transition relation $\longrightarrow_{\mathcal{P}}$, shown in Figure 2. The only probabilistic states which genuinely have different possible outcomes are those which resulted from an induced action. In this case there are two different resolutions according to whether the induced action is "listened to" or simply "heard". In either case the ether is emptied when the probabilistic resolution is made (rule PR1). First, a choice is made whether to hear the message or not, but secondly, if the message is heard, its impact is adjusted according to the perception function. This is consistent with the M^{2}MAM formalism. For other states the probabilistic resolution will simply clear the ether and return the agent to an active state again (rule PR2). Parallel agents undergo probabilistic resolution independently and their probabilities are multiplied (rule ParP).

4 Differential Semantics of PALOMA models

Obtaining performance metrics via discrete event simulation can become very expensive or even infeasible for PALOMA models when there is a large number of

$$\mathsf{PR1}\ [\alpha, r, \ell', X'], (?(\alpha, p).X(\ell) +^{p} X(\ell))^{\mathcal{P}} \begin{cases} \xrightarrow{(\alpha, p \times u(\alpha,\ell,X,\ell',X'))}_{\mathcal{P}} E_0, X(\ell) \\ \xrightarrow{(\alpha, 1-p \times u(\alpha,\ell,X,\ell',X'))}_{\mathcal{P}} E_0, ?(\alpha, p).X(\ell) \end{cases}$$

$$\mathsf{PR2}\ [\alpha, r, \ell', X'], X(\ell)^{\mathcal{P}} \xrightarrow{(\alpha,1)}_{\mathcal{P}} E_0, X(\ell)$$

$$\mathsf{ParP}\ \frac{E, X_1(\ell_1)^{\mathcal{P}} \xrightarrow{(\alpha,p)}_{\mathcal{P}} E_0, X_1'(\ell_1') \qquad E, X_2(\ell_2)^{\mathcal{P}} \xrightarrow{(\alpha,q)}_{\mathcal{P}} E_0, X_2'(\ell_2')}{E, (X_1(\ell_1) \parallel X_2(\ell_2))^{\mathcal{P}} \xrightarrow{(\alpha, p \times q)}_{\mathcal{P}} (E_0, X_1'(\ell_1')) \parallel (E_0, X_2'(\ell_2'))}$$

Fig. 2. The probabilistic transition relation for PALOMA

agents in the model. Thus, it is advantageous to define the differential semantics for PALOMA which can automatically derive the mean-field model in the form of initial value problems (a set of coupled ODEs with initial values) as done for M^{2}MAM. As solving the mean-field model is independent of the number of agents in the system, this enables scalable analysis of PALOMA models. In this section, we introduce the differential semantics of PALOMA models, first developing a population-based structured operational semantics which lifts the individual-based PALOMA model to a population-level view. This serves as an intermediate tool for the generation of the mean-field model.

4.1 Population-based Structured Operational Semantics

In PALOMA, as agents in the same state and location are identical, it is advantageous to use a population-based state vector to represent the state of the model in which symmetric states are aggregated to mitigate the well-known state space explosion problem. For example, consider the following simple two-location SIS model in PALOMA, which we refer to as Example 1:

$$\begin{aligned} S(\ell) &= ?(contact, p).I(\ell) + !(move, q).S(\ell') \\ I(\ell) &= !(contact, \beta).I(\ell) + !(recover, \gamma).S(\ell) + !(move, q).I(\ell') \\ S(\ell') &= ?(contact, p).I(\ell') + !(move', q').S(\ell) \\ I(\ell') &= !(contact, \beta).I(\ell') + !(recover, \gamma).S(\ell') + !(move', q').I(\ell) \end{aligned}$$

$$u(contact, \ell, X, \ell', X') = \begin{cases} \dfrac{1}{S_\ell + I_\ell} & \text{if } (\ell = \ell' \wedge X = S) \\ 0 & \text{otherwise} \end{cases}$$

$$S(\ell)[N_{S(\ell)}] \parallel I(\ell)[N_{I(\ell)}] \parallel S(\ell')[N_{S(\ell')}] \parallel I(\ell')[N_{I(\ell')}]$$

The model captures a disease spread scenario, in which an infective agent (I) makes a *contact* action spontaneously at the rate β. A susceptible agent (S) gets infected by accepting a *contact* message with the probability p. The perception function of the *contact* message can be explained as follows. If the message is received by a susceptible agent in the *same location* as the message sender, it can be perceived with probability $\frac{1}{S_\ell + I_\ell}$, where X_ℓ denotes the number of agents in state X in location ℓ. Otherwise, the message cannot be perceived. Intuitively,

the perception function of message *contact* means that an infective agent contact one arbitrary agent in its current location. Thus, a susceptible agent in the same location as the infective agent can perceive the *contact* message with probability $1/N_\ell$, where $N_\ell = S_\ell + I_\ell$ is the total number of agents in the location.

Agents in location ℓ move to location ℓ' by performing a spontaneous action *move* at the rate q, and move back by a spontaneous action $move'$ at the rate q'. The spontaneous actions *move* and $move'$ do not have any corresponding induced actions and so can be thought of as not emitting a message. An infective agent can also do a *recover* action spontaneously without message emission at rate γ.

Lastly, the final equation gives the initial populations of agents, where for example, $S(\ell)[N_{S(\ell)}]$ denotes $N_{S(\ell)}$ agents in the state $S(\ell)$ in parallel. This is syntactic sugar to ease the definition of large population models.

Now suppose we use a counting abstraction, constructing a state vector $\mathbf{X} = (x_1, x_2, x_3, x_4)$ to represent the current state of the model, in which x_1 denotes the number of agents in state $S(\ell)$, x_2, the number of agents in state $I(\ell)$, x_3, the number of agents in state $S(\ell')$ and x_4, the number of agents in state $I(\ell')$. Then the size of the state space is reduced from $O(4^N)$ to $O(N^4)$, where $N = \sum x_i$.

Using this notation, the x_2 enabled transitions caused by the spontaneous actions *contact* made by the x_2 agents in state $I(\ell)$ can be aggregated to a population-level transition as follows:

$$E_0, \mathbf{X} \xrightarrow{(contact, \beta \times x_2)} [contact, \beta \times x_2, \ell, I], \mathbf{X}^{\mathcal{P}} \quad \not\rightarrow_{\mathcal{P}} \tag{5}$$

The probabilistic resolutions following from this transition can be aggregated:

$$[contact, \beta x_2, \ell, I], \mathbf{X}^{\mathcal{P}} \left\{ \begin{array}{c} \cdots \\ \xrightarrow{(contact, \binom{x_1}{i} \times (pu)^i \times (1-pu)^{x_1 - i})}_{\mathcal{P}} E_0, \mathbf{X} + (-i, i, 0, 0) \\ \cdots \end{array} \right. \tag{6}$$

where $i = (0, 1, ..., x_1)$, u is the value of the perception function. Note that $\binom{x_1}{i} \times (pu)^i \times (1 - pu)^{x_1 - i}$ is the probability that there are i out of x_1 induced transitions of the form $S(\ell) =?(contact, p).I(\ell)$ actually fired, where $(-i, i, 0, 0)$ is the associated net change on the state vector caused by these transitions.

Furthermore, as the probabilistic resolutions occur immediately and finish instantaneously after a spontaneous transition fired, we can use a new transition relation $\longrightarrow_*$, which we call the *population-based* transition relation to form the following transitions to represent the transitions in equations (5) and (6):

$$\mathbf{X} \xrightarrow{(\tau, \beta \times x_2)}_* \mathbf{X} + (0, 0, 0, 0) \tag{7}$$

$$\mathbf{X} \left\{ \begin{array}{c} \cdots \\ \xrightarrow{(\tau_i, \beta \times x_2 \times \binom{x_1}{i} \times (pu)^i \times (1-pu)^{x_1 - i})}_* \mathbf{X} + (-i, i, 0, 0) \\ \cdots \end{array} \right. \tag{8}$$

where τ, τ_i are the transition names, $\beta \times x_2$, $\beta \times x_2 \times \binom{x_1}{i} \times (pu)^i \times (1-pu)^{x_1 - i}$ are the rates of the transitions, $(0, 0, 0, 0)$ and $(-i, i, 0, 0)$ are the net change of the elements in the state vector caused by the transitions. Note that in *population-based* transitions, induced transitions have rates derived from the spontaneous

$$\textsf{PbSp}\quad \frac{E_0, !(\alpha, r).X(\ell) \xrightarrow{(\alpha,r)} [\alpha, r, \ell, X], X(\ell)^{\mathcal{P}} \not\rightarrow_{\mathcal{P}}}{\mathbf{X} \xrightarrow{(\tau, \mathbf{X}[i_1]\times r)}_* \mathbf{X}'\{\mathbf{X}'[i_1] = \mathbf{X}[i_1] - 1, \mathbf{X}'[i_2] = \mathbf{X}[i_2] + 1\}} \quad \mathbf{X}[i_1] > 0$$

$$\textsf{PbIn}\quad \frac{[\alpha, r, \ell, X], (?(\alpha, p).X'(\ell') +^p X'(\ell'))^{\mathcal{P}} \begin{cases} \xrightarrow{(\alpha, p\times u)}_{\mathcal{P}} & E_0, X'(\ell') \\ \xrightarrow{(\alpha, 1-p\times u)}_{\mathcal{P}} & E_0, ?(\alpha, p).X'(\ell') \end{cases}}{\mathbf{X} \begin{cases} \cdots \\ \xrightarrow{(\tau_k, r_{\tau_k})}_* \mathbf{X}'\{\mathbf{X}'[i_2] = \mathbf{X}[i_2] - k, \mathbf{X}'[i_3] = \mathbf{X}[i_3] + k\} \\ \cdots \end{cases}}$$

with $\mathbf{X}[i_1] > 0$, $k = (0, ..., \mathbf{X}[i_2])$, $r_{\tau_k} = \mathbf{X}[i_1] \times r \times \binom{\mathbf{X}[i_2]}{k} \times (pu)^k \times (1 - pu)^{\mathbf{X}[i_2]-k}$

Fig. 3. The Population-based Structured Operational Semantics

transitions. By doing this, we can analyse the influence of the spontaneous transitions and induced transitions on the population level dynamics of the model separately. This simplifies the analysis of PALOMA models at the population level because there are now no probabilistic transitions at this level.

We formally define the population-based structured operational semantics with rules for population-based transitions for PALOMA in Figure 3. The premises of these two rules describe the behaviour of single agents whereas the conclusions gives the collective dynamics of the populations of agents. More specifically, the rule PbSp infers a population-based transition from a spontaneous transition of a single agent, in which $\mathbf{X}$ and $\mathbf{X}'$ are the state vectors representing the states of the model before and after the transition. i_1, i_2 are the indexes of count variables in the state vector for agents in states $!(\alpha, r).X(\ell)$ and $X(\ell)$ respectively. We do not need to explicitly specify the count of agents in other states because they remain invariant after the transition in the premise of rule PbSp. The rule PbIn infers a set of population-based transitions from a transition of a single agent induced by a sptontaneous action α fired at the rate r which is performed by an agent in state X and in location ℓ. i_1, i_2, i_3 are the indexes of count variables in the state vector for agents in states $X(\ell)$, $?(\alpha, p).X'(\ell')$ and $X'(\ell')$ respectively.

4.2 Population-based CTMC Model for PALOMA

With the above state aggregation and population-based structured operational semantics, we can define the population-based CTMC model for PALOMA. Formally, the population-based CTMC model for PALOMA is defined as a tuple $\mathcal{P} = (\mathbf{X}, \mathcal{D}, \mathcal{T}, \mathbf{x_0})$, where:

- $\mathbf{X} = (x_1, ..., x_n)$ is a state vector format, where each vector element is the count variable of agents in a specific state and location.
- Each x_i takes a value in a finite domain $\mathcal{D}_i \subset \mathbb{Z}^+$. Thus, $\mathcal{D} = \mathcal{D}_1 \times ... \times \mathcal{D}_n$ is the state space of the model.

- $\mathcal{T}(\mathbf{X}) = \{\tau_1, ..., \tau_m\}$ is the set of population-based transitions enabled in state $\mathbf{X}$, of the form $\tau_i = (r(\mathbf{X}), \mathbf{d})$, where:
  1. $r : \mathcal{D} \to \mathbb{R}_{\geq 0}$ is the rate function which depends on the current state of the system.
  2. $\mathbf{d} \in \mathbb{Z}^n$, is the update vector which gives the net change on each element of $\mathbf{X}$ caused by the transition.
- $\mathbf{x_0}$ is the initial state of the model.

4.3 The Mean-field Model

The population-based CTMC model for PALOMA is used to extract ODEs for the mean-field model similarly to [7]. We will explain how this can be done in this subsection. Firstly, if we are interested in the mean behaviour of the system dynamics, the set of population-based transitions in the conclusion of the rule PbIn can be aggregated by a single transition as:

$$\mathbf{X} \xrightarrow{(\tau, \mathbf{X}[i_1] \times r)}_{*} \mathbf{X} + \mathbf{d}_\tau \tag{9}$$

where $\mathbf{d}_\tau[i_2] = -pu \times \mathbf{X}[i_2]$, $\mathbf{d}_\tau[i_3] = pu \times \mathbf{X}[i_2]$ and for $j \notin \{i_2, i_3\}$, $\mathbf{d}_\tau[j] = 0$. Specifically, $\mathbf{X}[i_1] \times r$ is the rate at which the spontaneous transition α in the premise of PbIn occurs, and it is also treated as the rate of the above aggregated transition. $\binom{\mathbf{X}[i_2]}{k} \times (pu)^k \times (1-pu)^{\mathbf{X}[i_2]-k}$ is the probability that there are k out of $\mathbf{X}[i_2]$ enabled induced transitions actually fired in the probabilistic resolutions of the spontaneous transition, and in this case the net change in the count variables for agents in state $?(\alpha, p).X'(\ell')$ and $X'(\ell')$ is $(-k, k)$. Thus, by summing up the product of all the possible net changes and their associated probabilities, we can get the expected net change in the population level of agents in these two states caused by the transitions induced by the α action performed by the agents in state $X(\ell)$ as follows:

$$\mathrm{E}_{\mathbf{d}[i_2,i_3]} = \sum_{k=0}^{\mathbf{X}[i_2]} \binom{\mathbf{X}[i_2]}{k} (pu)^k \times (1-pu)^{\mathbf{X}[i_2]-k} \times (-k, k) = (-pu \times \mathbf{X}[i_2], pu \times \mathbf{X}[i_2])$$

Now, consider a population-based CTMC model for PALOMA (in which all the population-based transitions in the conclusion of the rule PbIn are aggregated in the style of Equation (9)) which is currently in state $\mathbf{X}$ with an enabled transition τ. This means that in every $1/r_\tau$ time units on average, a change in the population level of some agents denoted by $\mathbf{X}' = \mathbf{X} + \mathbf{d}_\tau$ occurs. If we approximate such a discrete change in a continuous fashion, then the change in the population level of the agents over a finite time interval Δt is:

$$\mathbf{X}(t + \Delta t) = \mathbf{X}(t) + r_\tau \times \mathbf{d}_\tau \times \Delta t$$

Rearranging the above equation and taking the limit $\Delta t \to 0$, we obtain the ODE which describes the (approximated) transient evolution of the population level of the agents in the system caused by transition τ as: $\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = r_\tau \times \mathbf{d}_\tau$

Taking all enabled transitions $\mathcal{T}(\mathbf{X}) = \{\tau_1, ..., \tau_m\}$ into account, the ODE which describes the (approximated) transient evolution of the complete population-level system dynamics, has initial condition $\mathbf{X}(0) = \mathbf{x_0}$ and is defined as:

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = \sum_{i=1}^{m} r_{\tau_i} \times \mathbf{d}_{\tau_i}.$$

The Motivating Example We use Example 1 to illustrate our approach. From the induced transitions in each location $\hat{\ell} \in \{\ell, \ell'\}$:

$$S(\hat{\ell}) = ?(contact, p).I(\hat{\ell}) \quad \text{induced by} \quad I(\hat{\ell}) = !(contact, \beta).I(\hat{\ell})$$

we obtain $\tau_1 = (\beta x_2, (-pux_1, pux_1, 0, 0))$ and $\tau_2 = (\beta x_4, (0, 0, -pu'x_3, pu'x_3))$ respectively, where $u = \dfrac{1}{x_1 + x_2}$ and $u' = \dfrac{1}{x_3 + x_4}$.

From the following spontaneous transitions:

$$\begin{array}{lll} S(\ell) = !(move, q).S(\ell') & S(\ell') = !(move', q').S(\ell) & I(\ell) = !(move, q).I(\ell') \\ I(\ell') = !(move', q').I(\ell) & I(\ell) = !(recover, \gamma).S(\ell) & I(\ell') = !(recover, \gamma).S(\ell') \\ I(\ell) = !(contact, \beta).I(\ell) & I(\ell') = !(contact, \beta).I(\ell') & \end{array}$$

We get the following corresponding population-based transitions:

$$\begin{array}{lll} \tau_3 = (qx_1, (-1, 0, 1, 0)) & \tau_4 = (q'x_3, (1, 0, -1, 0)) & \tau_5 = (qx_2, (0, -1, 0, 1)) \\ \tau_6 = (q'x_4, (0, 1, 0, -1)) & \tau_7 = (\gamma x_2, (1, -1, 0, 0)) & \tau_8 = (\gamma x_4, (0, 0, 1, -1)) \\ \tau_9 = (\beta x_2, (0, 0, 0, 0)) & \tau_{10} = (\beta x_4, (0, 0, 0, 0)) & \end{array}$$

Therefore, the mean-field model for Example 1 is $\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = \sum_{i=1}^{i=10} r_{\tau_i} \times \mathbf{d}_{\tau_i}$. The associated ODE model for each state count variable is:

$$\begin{aligned} \frac{\mathrm{d}x_1(t)}{\mathrm{d}t} &= -\beta \times x_2 \times p \times u \times x_1 - q \times x_1 + q' \times x_3 + \gamma \times x_2 \\ \frac{\mathrm{d}x_2(t)}{\mathrm{d}t} &= \beta \times x_2 \times p \times u \times x_1 - q \times x_2 + q' \times x_4 - \gamma \times x_2 \\ \frac{\mathrm{d}x_3(t)}{\mathrm{d}t} &= -\beta \times x_4 \times p \times u' \times x_3 + q \times x_1 - q' \times x_3 + \gamma \times x_4 \\ \frac{\mathrm{d}x_4(t)}{\mathrm{d}t} &= \beta \times x_4 \times p \times u' \times x_3 + q \times x_2 - q' \times x_4 - \gamma \times x_4 \end{aligned}$$

The mean-field model matches our intuition from the M^{2}MAM definition.

5 Case Study: Modelling the Spread of Flu

In this section, we extend the PALOMA model in Example 1 to model the spread of flu in a multi-community society. The model captures a simplified scenario

of the 1918-1919 flu epidemic in central Canada, which was originally described in [8]. Consider an isolated society which consists of m communities. Q is the rate matrix in which each element q_{ij} is the rate at which a resident travels from community i to community j. We use β_i to denote the number of contacts per person per day in community i. For various reasons, the number of contacts per person per day is not the same in all communities. For example, suppose community i is the business centre of the society, then the number of contacts per person per day in community i should be higher than other communities.

In the epidemic model, a resident has three states: State S for susceptible, I for infected and R for recovered. When a susceptible resident makes contact with an infected resident, he will be infected by the flu with the probability p. On average, it takes about $1/\gamma$ days for an infected resident to recover from the flu. Once a resident recovers, he will not be infected again. We are interested in how many residents are infected by the flu from the beginning to the end of the outbreak. This can be captured by the following PALOMA model:

$$S(\ell_i) = ?(contact, p).I(\ell_i) + \sum_{j \neq i}^{m} !(move_{ij}, q_{ij}).S(\ell_j)$$

$$I(\ell_i) = !(contact, \beta_i).I(\ell_i) + !(recover, \gamma).R(\ell_i) + \sum_{j \neq i}^{m} !(move_{ij}, q_{ij}).I(\ell_j)$$

$$R(\ell_i) = \sum_{j \neq i}^{m} !(move_{ij}, q_{ij}).R(\ell_j)$$

$$u(contact, \ell, X, \ell', X') = \begin{cases} \dfrac{1}{S_\ell + I_\ell + R_\ell} & \text{if } (\ell = \ell' \wedge X = S) \\ 0 & \text{otherwise} \end{cases}$$

$$S(\ell_1)[N_{S(\ell_1)}] \parallel I(\ell_1)[I_{I(\ell_1)}] \parallel \ldots \parallel S(\ell_m)[N_{S(\ell_m)}] \parallel I(\ell_m)[I_{I(\ell_m)}]$$

where the perception function of message *contact* also means that an infected resident can make contact with an arbitrary resident in their current community. Thus, a susceptible resident in the same location as the infective agent can perceive the *contact* message with probability $1/N_\ell$, where $N_\ell = S_\ell + I_\ell + R_\ell$ is the total number of residents in the community ℓ.

5.1 Investigate the Effect of Quarantine on the Spread of the Flu

In this subsection we present the results of some experiments run on the model to investigate the effect of different quarantine policies on the spread of the flu. For example, quarantine may be applied to a whole community which is believed to be the source of the outbreak, reducing the likelihood of travel to other communities. Alternatively, individuals who develop flu in any community may be individually isolated and prevented from travelling.

Community-level Quarantine: Here we assume that the flu originates from community i. We model the effect of *community-level quarantine* by adding a quarantine factor $0 < \sigma < 1$ to the rate at which residents travel into and out of community i. More specifically, the value of $Q(i, j)$ and $Q(j, i)$ becomes $q_{ij} \times \sigma$ and $q_{ji} \times \sigma$ repectively.

Individual-level Quarantine: Alternatively the focus may be on the individuals with the disease. Suppose that on average, an infected person exhibits symptoms $1/\eta$ days after infection. Once an infected person is discovered, they will be isolated immediately until recovery. To model this, we introduce a new state $ISO(\ell_i)$, which represents an isolated infected resident currently in community i. Note that not all infected individuals will exhibit symptoms. The modifications to the PALOMA model with individual-level quarantine are given as follows:

$$I(\ell_i) = !(contact, \beta_i).I(\ell_i) + !(recover, \gamma).R(\ell_i) + !(discovered, \eta).ISO(\ell_i) + \sum_{j \neq i}^{m} !(move_{ij}, q_{ij}).I(\ell_j)$$

$$ISO(\ell_i) = !(recover, \gamma).R(\ell_i)$$

$$u(contact, \ell, X, \ell', X') = \begin{cases} \dfrac{1}{S_\ell + I_\ell + R_\ell} & \text{if } (\ell = \ell' \wedge X = S) \\ 0 & \text{otherwise} \end{cases}$$

Note that an isolated resident cannot travel out of their current community or contact other residents. As a result, an agent in state $ISO(\ell_i)$ can only do a spontaneous action *recover* and go to the state $R(\ell_i)$. Moreover, ISO_ℓ is not included in the perception function of *contact* which also reflects that isolated residents do not have chances to contact other residents.

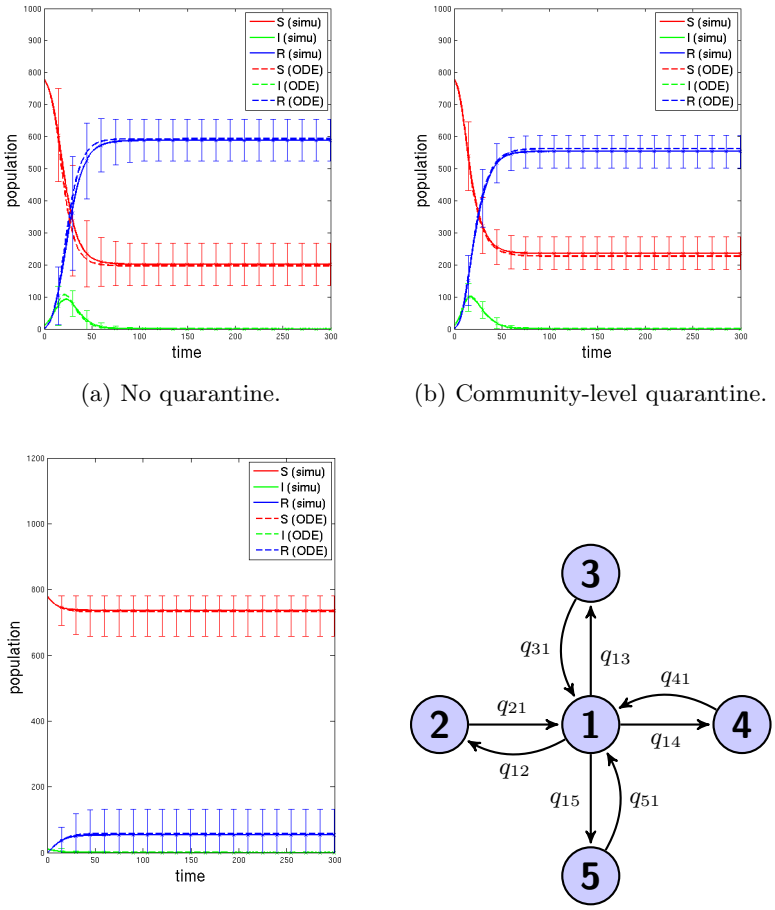
Model Analysis We consider five communities located in a star topology as illustrated in Figure 4(d): we assume community 1 is the business centre of the society and the flu also originates in Community 1. Thus, the community-level quarantine is imposed on Community 1. We assume that there are 300 residents of Community 1, 10 of whom are infected at the start of the study; 150 residents of Community 2, 140 residents of Community 3, 100 residents of Community 4 and 100 residents of Community 5, all of whom are susceptible at the start of the study. The values of parameters used in our simulation are given in Table 1.

Our simulation tool can parse PALOMA models and run discrete event simulations. The corresponding mean-field model is automatically generated in the form of Matlab scripts when the model is parsed, and can be run directly in Matlab. Figure 4(a), 4(b), 4(c) show our simulation results with 95% confidence interval in three different senarios. The results generated by the discrete event simulation (taking the average of 100 simulation runs) match well with the results of the mean-field model. The run time of a discrete event simulation for this model is about 70 seconds on a dual CORE i5 machine whereas the mean-field model can generate results instantly.

The results also give us some interesting information. It can be seen that community-level quarantine only reduces the number of infected residents to a

$p = 0.5$	$\beta_1 = 1$	$\beta_i = 0.5\ (i \neq 1)$	$\gamma = 0.2$	$\eta = 0.25$	$\sigma = 0.1$	$q_{12} = 0.1$
$q_{13} = 0.12$	$q_{14} = 0.13$	$q_{15} = 0.11$	$q_{21} = 0.4$	$q_{31} = 0.4$	$q_{41} = 0.3$	$q_{51} = 0.35$

Table 1. Parameters used in the simulation



(a) No quarantine.

(b) Community-level quarantine.

(c) Individual-level quarantine.

(d) The topology of the 5 communities in the simulation.

Fig. 4. The simulation result and the community topology of the flu spread model

limited extent whereas individual-level quarantine has a more profound effect on controlling the spread of the flu.

6 Related Work

There have been many previous process algebras which encompass some spatial modelling, ranging from very abstract space and mobility in the π-calculus [9] to Cardelli and Gardner's elegant process algebra based around affine geometry [10]. Some also incorporate stochastic behaviour, such as stochastic π-calculus [11], Bio-PEPA [12], stochastic Bio-Ambients [13] and stochastic bigraphs [14]. But

in each of these space is abstractly represented and most focus on a containment relationship between locations.

Closer to our work are the process algebras PALPS [15] and MASSPA [16]. PALPS, the Process Algebra with Location for Population Systems, is designed for building ecological models, and offers language primitives targeted at this application domain. Moreover, only an individual-based semantics is developed, severely restricting the scalability of the models which can be developed. Like PALOMA, MASSPA, the Markovian Agent Spatial Stochastic Process Algebra, takes the M^2MAM framework as its starting point. MASSPA emulates message broadcast by allowing each spontaneous action to emit a number of messages, loosely based on the likelihood that the single spontaneous action will trigger that number of induced actions. This multiplication of actions affects the dynamics of the system and no individual-based semantics is established. Instead MASSPA models are translated into systems of chemical reactions and population-level discrete event simulation based on mass action dynamics is developed. In contrast, PALOMA is equipped with both individual- and population-based semantics, supports dynamic perception functions (i.e. based on the current system state), which enables PALOMA to model adaptive behaviour. Furthermore, we slightly extend M^2MAM to allow agents to move between locations. The original M^2MAM framework requires the number of agents in each location to remain constant because the derived ODEs describe the evolution of the state density distribution of agents in each location over time, and then use that to derive the number of agents in different states. However, in the differential semantics of PALOMA, we use the ODEs to directly describe the evolution of number of agents in different states in different locations. Thus we are able to allow agents to move in PALOMA models, as demonstrated in the case study.

In [17] the authors apply mean-field models with locations (or classes more generally), which are close to the mean-field models developed in this paper, to the performance evaluation of network systems. PALOMA high-level language to define models of this kind.

7 Conclusion

PALOMA is a novel stochastic process algebra which treats location as a primary feature of each agent, and allows the interaction of agents to be adapted according to their locations. Location is just one possible interpretation of this parameter, and more generally, PALOMA can be seen as supporting *attribute-based communication*. This style of communication has previously been investigated in languages such as SCEL [18], but in that context has not been amenable to scalable analysis. Here we demonstrate how PALOMA may be equipped with scalable analysis through both discrete event and mean field interpretations. Future work will consider suitable notions of equivalence and logic, e.g. it could be useful to consider agents which exert influence of the same type over an analogous region of space, to be equivalent even though their exact locations differ. We will also investigate abstractions of space within PALOMA models.

Acknowledgement

This work is partially supported by the EU project QUANTICOL, 600708.

References

1. Cerotti, D., Gribaudo, M., Bobbio, A., Calafate, C.T., Manzoni, P.: A Markovian agent model for fire propagation in outdoor environments. In: Computer Performance Engineering. Springer (2010) 131–146
2. Gribaudo, M., Cerotti, D., Bobbie, A.: Analysis of on-off policies in sensor networks using interacting Markovian agents. In: 6th Annual IEEE International Conference on Pervasive Computing and Communications, IEEE (2008) 300–305
3. Bruneo, D., Scarpa, M., Bobbio, A., Cerotti, D., Gribaudo, M.: Markovian agent modeling swarm intelligence algorithms in wireless sensor networks. Performance Evaluation **69**(3) (2012) 135–149
4. Cerotti, D., Gribaudo, M., Bobbio, A.: Disaster propagation in inhomogeneous media via Markovian agents. Critical Information Infrastructure Security (2008) 328–335
5. Prasad, K.: A calculus of broadcasting systems. Science of Computer Programming **25**(2) (1995) 285–327
6. Hillston, J.: A Compositional Approach to Performance Modelling. CUP (2005)
7. Tribastone, M., Gilmore, S., Hillston, J.: Scalable differential analysis of process algebra models. IEEE Transactions on Software Engineering **38**(1) (2012) 205–219
8. Sattenspiel, L., Herring, D.A.: Simulating the effect of quarantine on the spread of the 1918–19 flu in central Canada. Bull of Mathematical Biology **65**(1) (2003) 1–26
9. Sangiorgi, D., Walker, D.: The pi-calculus: a Theory of Mobile Processes. CUP (2003)
10. Cardelli, L., Gardner, P.: Processes in space. In: Programs, Proofs, Processes. Springer (2010) 78–87
11. Priami, C.: Stochastic π-calculus. The Computer Journal **38**(7) (1995) 578–589
12. Ciocchetta, F., Hillston, J.: Bio-PEPA: A framework for the modelling and analysis of biological systems. Theoretical Computer Science **410**(33) (2009) 3065–3084
13. Brodo, L., Degano, P., Priami, C.: A stochastic semantics for bioambients. In: Parallel Computing Technologies (PaCT 2007),. LNCS (2007) 22–34
14. Krivine, J., Milner, R., Troina, A.: Stochastic bigraphs. Electronic Notes in Theoretical Computer Science **218** (2008) 73–96
15. Efthymiou, X., Philippou, A.: A process calculus for spatially-explicit ecological models. Application of Membrane Computing, Concurrency and Agent-based Modelling in Population Biology (AMCA-POP 2010) (2010) 84–78
16. Guenther, M.C., Bradley, J.T.: Higher moment analysis of a spatial stochastic process algebra. In: Computer Performance Engineering. Springer (2011) 87–101
17. Bakhshi, R., Endrullis, J., Endrullis, S., Fokkink, W., Haverkort, B.: Automating the mean-field method for large dynamic gossip networks. In: 7th International Conference on the Quantitative Evaluation of Systems, IEEE (2010) 241–250
18. De Nicola, R., Ferrari, G., Loreti, M., Pugliese, R.: A language-based approach to autonomic computing. In: Formal Methods for Components and Objects, Springer (2013) 25–48