

Reconciling Population and Agent Models for Crowd Dynamics

Andrea Bracciali

CNR-ISTI,

Pisa, Italy.

`bracciali@isti.cnr.it`, Jane Hillston

School of Informatics,

University of Edinburgh.

`Jane.Hillston@ed.ac.uk`, Diego Latella

CNR-ISTI,

Pisa, Italy.

`latella@isti.cnr.it` and

Mieke Massink

CNR-ISTI,

Pisa, Italy.

`massink@isti.cnr.it`

Abstract

We propose an approach to the quantitative modelling of crowd dynamics, viz. the behaviour of systems of large numbers of mobile agents. The approach relies on a stochastic process algebra as specification language (Bio-PEPA), and combines stochastic simulation techniques and continuous fluid flow approximation. The approach encompasses the *agent* modelling viewpoint, as system behaviour emerges from the specified agent interaction, and the *population* modelling viewpoint, when continuous analysis is used. The result is expressive, as we will show by discussing a few examples, and efficient, by the adoption of the fluid flow analysis techniques, which approximate system dynamics as continuous variations of population.

1 Introduction

The widespread diffusion of mobile and communicating computational devices is enabling a rich new set of services and software applications, which rely on open, dynamic and large interconnection networks and on a high level of interaction between the components of such a distributed system. These systems are pervasive and comprise a large number of components, agents or individuals. Hence, they represent a large source of data useful for understanding system dynamics and, at various degrees, the parts played by the components.

The dynamics of a crowd of individuals equipped with mobile devices is one of the possible forms of this kind of system. Mobile phones and satellite navigators can be used to *i*) provide localised and context sensitive information to individuals, e.g. maps, attractions and updated traffic information, and *ii*) gather information about the individuals and the emergent behaviour of the overall system, e.g. mining the system behaviour to understand recurrent traffic patterns or to validate the functioning of the system against desired criteria, such as the quality of

service requirements. Clearly, the aspects in *i*) and *ii*) are mutually dependent, such as when the traffic is automatically monitored and the occurrence of an accident can be communicated to individuals together with alternative “optimal” routes, so that the overall traffic patterns can in turn be affected. Other examples can be found in airports and air traffic, cities, open air public events and the like. Interesting issues to be addressed in the design of such systems include aspects like the identification of congestion and interference when different groups of individuals move through the same common physical spaces; arrival times under different assumptions about the presence of other individuals sharing the spaces and routing information provided to them; the effect of common individual behaviour deviating from standard behaviour through error or because of other circumstances; and the effect of dynamic changes in the physical space on the flow of people.

The traditionally difficult issue of appropriate modelling and analysis of these systems is made worse by aspects like the large scale of the system, the difficulty in precisely characterising the interactive behaviour of each one of the components, which are often loosely coordinated, and an intrinsic indeterminacy of the system. How can the choices of a number of drivers in a traffic jam during a peak hour be suitably described? Formal modelling and analysis of human/agents collective behaviour within spatial environments is still a challenging and partially unexplored problem.

Here we focus on *crowd dynamics*, understood as the emergent behaviour of a multitude of mobile agents within a common, possibly open, environment. We recall two general modelling approaches.

Following a *population viewpoint* approach, dynamics is observed in terms of classes of agents homogeneously following the same patterns of behaviour. For instance, the class of commuters from one city to another could be observed in terms of the occupancy of a connecting highway. These descriptions are generally based on continuous mathematical models that describe the deterministic average time course of relevant quantities and are typically specified in terms of sets of differential equations (ODEs).

In an *agent viewpoint* approach, the behaviour of each single agent is described and the overall behaviour of the system emerges from the interaction of agents over time. For instance, a set of rules could be used to code the behaviour of an agent in an highway and its routing plans to move from one city to another. These descriptions typically rely on discrete models, which often encompass some probability to describe the inherent indeterminacy or lack of complete knowledge about the system. These models can be used to produce simulations of the system behaviour and reasoning about them, for instance in the tradition of the use of modal logic predicating over (probabilistic) state-transition models.

The former modelling approach can often be computationally efficient, whenever equations can be efficiently solved numerically. This approach provides an informative representation of the system, at least for its average behaviour, and has been extensively used so far for modelling a wide range of physical phenomena. However, it may fall short in those cases where the system behaviour is inherently stochastic, can be influenced by the occurrence of single events, is sensitive to the action of a few individuals, or whenever the average behaviour is not of much interest or generates a system of ODEs which are difficult to solve. For instance, in the case of commuters, one might be interested not only in the overall resulting flux between two cities, but also in how much this flux can vary and how it may be affected by road works happening with an estimated frequency and duration.

In this paper, we approach the modelling of crowd dynamics by adopting standard modelling techniques rooted in stochastic process algebras enhanced with an efficient analysis technique, called *Fluid Flow Approximation* [11]. From process algebras we gain an expressive linguistic

description of agent behaviour, which can be thought of modularly, can be easily revised, is grounded on a neat formal semantics also encompassing the stochastic aspects, and is supported by a suite of computational tools. Fluid flow aims at tackling the state space combinatorial explosion that arises as a consequence of interleaving the behaviour of the many independent individuals and services in the system. Fluid flow relies on an abstract quantitative description of the system, i.e. variables represent amounts of agents with “homogeneous” behaviour, and on the approximating hypothesis that such quantities vary continuously over time. Intuitively speaking, the continuity hypothesis can be understood as an acceptable approximation in the presence of large numbers of agents, as for the cases of our interest, and considering that these quantities vary over time by small discrete steps, one or few units, negligible with respect to the large numbers involved. The high level process algebraic model can be automatically turned into a system of ODEs, which efficiently approximate the averaged behaviour of the system. This approach is similar to symbolic models, like counting abstractions, where the identities of single processes are abstracted away when they are gathered together in classes that are homogeneous for the verification purposes. It is worth remarking that in other formalisms both continuous and discrete values have been considered for modelling purposes. For instance, hybrid automata [9] model dynamical systems that may consist of a discrete, state-based control, but have interfaces with a continuously varying world. Here instead, the continuous variables are approximations of discrete quantities such as the number of individuals in a particular class.

Our work lies in the tradition of the quantitative, often discrete, analysis of multi-agent systems, but is specifically centered on the combination of an expressive and abstract linguistic description of the system behaviour, by the use of a process calculi, and of a mathematically sound, but potentially efficient, kind of analysis. The agent view and stochastic modelling are integrated within the framework, allowing the analysis of non-averaged behaviour to be carried out with standard techniques. For instance one may wish to understand and even tune the average behaviour of the system and then deepen some particular cases of interest, before the actual fielding of the system. Indeed, the computational costs of stochastic simulation analysis may be justified when a final system design is analysed, but are often prohibitive when used to explore many options during early phases of design.

We illustrate this approach by recalling the main definitions of the framework and then by developing an analysis of a realistic emergency egress scenario. This consists of the evacuation of a building. Agents have a plan to reach the most convenient exit and are constrained by the structure of the building and by their reciprocal behaviour. We can assume that the evacuation directions are communicated to agents instant by instant by a central coordinator, although we only model the delay effect of these communications. Following their plan, agents move towards the assigned exit. The egress case study is taken from [12], where a modelling framework based on flux optimisation analysis is presented. A first approach exploring the feasibility of fluid flow analysis applied to guidance systems has been presented in [7, 16, 15], where the Performance Evaluation Process Algebra PEPA [10] has been used.

Here, we illustrate the expressiveness of the model by adding the possibility for agents to change the provided plan based on local information they might sense from the environment. In order to dynamically use contextual information, we have adopted the Bio-PEPA process algebra [5], originally devised for modelling bio-molecular interaction. Bio-PEPA supports location aware and context dependent information, an important feature that is not available in PEPA. That is, Bio-PEPA embeds a notion of spatial location, intended to model biological compartments, which allows us to describe the building structure and capacity and locate agents within it. Also, the agent behaviour can be expressed as a function of the current state of the system, like the level of congestion in given rooms, an abstraction of the act of sensing the

RA 211		18w 18e		SE 13
LW 25	HA 133	LE 16		
SW 22	RB 92	16w	RC 98	18e

Figure 1: Three floors of example building from EVACNET4 manual (NC/IC)

environment, common to standard agent models.

Thanks to these features, we can embed simple planning behaviour in agent models: whenever an agent gets to know that one of the two alternative paths to an exit is congested, it may change its mind and attempt to reach the other exit. This individual, probabilistic, behaviour is shown to lead to the emergence of faster evacuation patterns, as the occupancy of the two exit paths is kept balanced.

Although we discuss mainly the modelling aspects and fluid flow analysis, the framework we present can be easily integrated with other analysis techniques. Stochastic model checking appears to be a natural candidate, which has already been implemented in several related analysis tools. Although state explosion undermines the applicability of “exact” model checking, approximated versions of it could be adopted. Particularly, checking samples of model simulations provides a versatile compromise between precision and efficiency. These issues are scope for future work. Furthermore, on a longer term, we are interested in devising a suitable modelling language and associated analysis framework for crowd dynamics.

In Section 2 the egress case study is presented. Bio-PEPA and the fluid flow technique are introduced in Section 3. The egress modelling is detailed in Section 4. Final remarks are in Section 5.

2 Agent emergency egress

We discuss the modelling of the emergency egress of a building in Bio-PEPA as a paradigmatic example of crowd dynamics modelling. We consider and extend a study described in [12], where a detailed model of a three story building is used to determine an *optimal* (fastest) evacuation plan by means of an advanced capacitated network flow transshipment algorithm, a specialized algorithm used in solving linear programming problems with network structure. The constraints imposed by the building morphology to the evacuee flux and typical human behaviour, such as the average speed of an evacuee and the average door crossing rate per time unit are taken into consideration.

The plan of the building is depicted in Figure 1: there are three rooms (RA, RB, RC), one hall (HA) and two staircases and landings, on the east (LE, SE) and west side (LW, SW) of the building, respectively, all connected by a number of doors (gaps in the lines) and each with its capacity. Evacuees escape from the building either from SE or SW. When the alarm goes off there are 36 agents in RA, half of which will be directed towards east and half to west, 16 in RB heading to west and 18 heading to east. The average communication time to be instructed about directions in every room is estimated to be about 2 seconds. Agents move at an average

speed of 1 m/s, and average traversal time for each room can be calculated according to the room dimensions. See [12] for further details.

3 Bio-PEPA and Fluid Flow Analysis

In this section we give a short description of Bio-PEPA [4, 5, 3], a language that has recently been developed for the modelling and analysis of biochemical systems. The main components of a Bio-PEPA system are the “*species*” *components*, describing the behaviour of individual entities, and the *model component*, describing the interactions between the various species. The initial amounts of each type of entity or species are given in the model component.

The syntax of the Bio-PEPA components is defined as:

$$S ::= (\alpha, \kappa) \text{ op } S \mid S + S \mid C \quad \text{with op} = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot \quad P ::= P \underset{\mathcal{L}}{\bowtie} P \mid S(x)$$

where S is the *species component* and P is the *model component*. In the prefix term $(\alpha, \kappa) \text{ op } S$, κ is the *stoichiometry coefficient* of species S in action α . This arises from the original formulation of the process algebra for modelling biochemical reactions, where the stoichiometric coefficient captures how many molecules of a species are required for a reaction. However it may be interpreted more generally as the multiples of an entity involved in a successful action. The *prefix combinator* “op” represents the role of S in the action, or conversely the impact that the action has on the species. Specifically, \downarrow indicates a *reactant* which will be consumed in the action, \uparrow a *product* which is produced as a result of the action, \oplus an *activator*, \ominus an *inhibitor* and \odot a generic *modifier*, all of which play a role in an action without being produced or consumed. The operator “+” expresses the choice between possible actions, and the constant C is defined by an equation $C = S$. The process $P \underset{\mathcal{L}}{\bowtie} Q$ denotes synchronisation between components P and Q , the set \mathcal{L} determines those activities on which the operands are forced to synchronise, with \bowtie denoting a synchronisation on all common action types. In the model component $S(x)$, the parameter $x \in \mathbb{R}$ represents the initial amount of the species.

In a Bio-PEPA model different manifestations of an entity are represented by distinct species components. For example in the emergency egress model the behaviour of a person in a room is captured by the different species component from that which describes the behaviour of a person in the hall even though in reality one person might expect to experience each of these manifestations.

A Bio-PEPA *system* with locations consists of a set of species components, also called sequential processes, a model component, and a context (locations, kinetics rates, parameters, etc.). The prefix term $(\alpha, \kappa) \text{ op } S@l$ is used to specify that the action is performed by S in location l .

Definition A Bio-PEPA system \mathcal{P} is a 6-tuple $\langle \mathcal{L}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, \text{Comp}, P \rangle$, where: \mathcal{L} is the set of locations, \mathcal{N} is the set of auxiliary information, \mathcal{K} is the set of parameters, \mathcal{F}_R is the set of functional rates, Comp is the set of species components, P is the model component.

Bio-PEPA is given an operational semantics [5]. In this context species amounts are abstracted by discrete levels, representing intervals of values. There are two relations over the processes: the *capability relation*, which supports the derivation of qualitative information, and the *stochastic relation*, defined in terms of the capability relation and equipped with rates for the associated action types. There is a rate function r_α associated with the action type α and its value is calculated in each state according to the current state of the system and its context. In modelling biochemistry this rate function is often the so-called *mass action* law which

stipulates that the rate of the reaction is the product of the amounts of its reactants and a rate constant. At each time instant each action is assumed to be governed by an exponentially distributed random variable which determines its duration: the rate function gives the parameter of this distribution. This gives rise to an underlying continuous time Markov chain (CTMC). The dynamic behaviour of processes is determined by a *race condition*: all actions which are enabled attempt to proceed but only the fastest succeeds.

The Bio-PEPA language is supported by a suite of software tools which automatically process Bio-PEPA models and generate internal representations suitable for different types of analysis [5, 2, 1]. These tools capture mappings from Bio-PEPA to differential equations, stochastic simulation models [6], continuous time Markov chains (CTMC) with levels [4] and PRISM models [14].

3.1 Fluid Flow

As mentioned above, the Bio-PEPA semantics allows for the application of different analysis and evaluation techniques including fluid flow analysis originally defined in terms of the related stochastic process algebra PEPA. We give a very brief summary of the approach here; for details see [11].

A Bio-PEPA model consists of a number of sequential components each of which represents a number of entities in a distinct state. The result of an action is to increase the number of some entities and decrease the number of others, these adjustments reflecting the stoichiometry with respect to the action. Thus we can represent the total state of the system at any time as a vector with entries capturing the counts of each species component. This gives rise to a discrete state system which undergoes discrete events. The idea of fluid flow analysis is to approximate these discrete jumps by continuous flows between the states of the system. This approximation becomes good when entities are present in such high numbers as to make the frequency of actions high and the relative change from single events small. In this case, when the continuous flow represents the average change in the number of species components well, we can derive a set of ordinary differential equations (ODEs) which approximate the average behaviour of the CTMC.

4 Emergency Egress: a Case Study

We illustrate how the combination of the Bio-PEPA stochastic process algebras and fluid flow analysis can support the modelling of crowd dynamics for the specific case of emergency egress introduced in Section 2. First the model is described and then some simulation results are discussed.

4.1 The Bio-PEPA model

The basic ingredients of our Bio-PEPA model are *i*) the definition of the spatial structure; *ii*) the definition of the agent behaviour, which basically corresponds to sequential processes in the process algebra terminology; and *iii*) the interaction structure amongst agents, i.e. how an agent can interact within the system, which corresponds to structured processes.

Excerpts from the egress model are reported in Figure 2. The spatial structure of the building consists of a set of locations without any nesting. These represent rooms, e.g. `ra`, and doors, e.g. `d1_ra_ha`, i.e. door 1 between room `ra` and hall `ha`. It is possible to define dimensions within the structure definition, here `normal_room` and `normal_door`. These dimensions could

be used, for instance, when defining the rates relative to actions involving movement. As explained, rates determine exponential negative distributions, which represent action duration or, equivalently, occurrence time. It is worth recalling that the expected value of an exponential distribution of parameter r , i.e. the average occurrence time, is $1/r$. Time is measured in minutes. For instance, we have determined the rate `to_ra_d1 = 6`, accounting for $1/6$ of a minute, i.e. ten seconds, since this time is the sum of the time for getting directions (around 3 seconds) and reaching door 1 in room `ra` (about 7 seconds to walk 7 meters at about $1m/s$). Analogously, `form_d1` accounts for the door exiting rate.

Other variables and parameters have been defined. For instance, `occupancy_d1` represents the occupancy of door 1 as the sum of agents that are currently occupying the door (see below for a description of `D1_ra_e`, `D1_ra_w`, ... agents); `full_d1` is a boolean test about the door being fully occupied by agents in transit, where `capacity_d1` stands for the maximum number of agents that can occupy a door, 2 in our model, `H(.)` is a non-negative-test function, and `0/1` arithmetic implements the boolean algebra; `switch_d1` is a boolean flag accounting for a door being traversable, i.e. open and not full; `safeD1_ra3_e` is again a boolean value accounting for the existence of at least a `D1_ra_e` agent ready to interact; and `allowance_ha3` represents the number of available free places in room `ha3`.

These variables represent useful contextual information that can be used to finely control the model. For instance, a door traversal rate can be influenced by the instantaneous door occupancy, or doors can be opened and closed, i.e. suitable rates made null or positive, by simply switching parameters, like `open_d1`. Furthermore, variables are used to summarise relevant information about the dynamics of the system for the analysis of simulations and for defining context dependent behaviour (see below about the measurement of path congestion).

We distinguish agents heading towards east from those heading towards west by means of distinct classes of sequential processes. Also, we model the behaviour of an agent in every location by means of a distinct sequential process. Hence, `RA_e` is the process relative to an agent in room `ra`, heading towards east. Such an agent can engage in several different kinds of interaction, amongst which it can perform the action `(ra_e_in_d1_ra_ha, 1)`. The definition of the sequential process `D1_ra_e` represents the class of agents who have entered door 1 from room `ra`. They can perform the action `(ra_e_in_d1_ra_ha, 1)`, too. Whenever agents of the two classes synchronise on the execution of this action, which has 1 as stoichiometric parameter, one instance of `RA_e` disappears in `ra` and a new instance of `D1_ra_e` occupies one of the free spaces in door 1 in between `ra` and `ha`.¹ The effects of these interactions are clearly located in a specific compartment of the model by the `@` construct, so that `<< RA_e@ra` represents the loss of an `RA_e` agent in room `ra`. It is important to remark that not necessarily an agent must already be occupying the door but it may be created as an effect of this interaction. The pair of disappearing and appearing agents model movement from a compartment to another. In the following, when we will say that two processes synchronise we will also comprise such a case where one of the two “interacting” processes is created as an effect of the interaction.

As prescribed by the construct `kineticLawOf`, this action is ruled by the mass action law: `fMA(to_ra_d1 * switch_d1 * ra_e_in_safe)`, where `to_ra_d1` is the rate and `switch_d1` the parameter discussed above, and `ra_e_in_safe` is a boolean flag guaranteeing that at least one `RA_e` is ready to synchronise on the action. Other actions are ruled by the same kinetic law, such as `ha_e_out_d1_ra_ha` that regards the exiting from a door and it is constrained by the

¹The syntax accepted by the Bio-PEPA modelling tool is slightly different from the one in the formal definition of Bio-PEPA reported in Section 3: sequential processes $(\alpha, \kappa) \text{ op } S$, with $\text{op} = \downarrow \mid \uparrow \mid \odot$ are written as $(\text{op}, n) * S'$, with $*$ = `<<`, `>>`, and `(.)`, respectively, and S' can carry some localisation information, e.g. `RA_e@ra`; the synchronisation operator \bowtie is written as `<*>` (`< >`, when no synchronisation is required).

```

// BUILDING LAYOUT (COMPARTMENTS)

location ra :      size = normal_room, type= compartment;
location d1_ra_ha : size = normal_door, type= compartment;
...
// PARAMETERS SET UP

to_ra_d1      = 6;                // 3 + (60/7);
from_d1       = door_exit_rate;
...
occupancy_d1 = D1_ra_e@d1_ra_ha + D1_ra_w@d1_ra_ha + ... :
full_d1      = H(capacity_d1 - occupancy_d1);
switch_d1    = open_d1 * full_d1;
safeD1_ra3_e = H(D1_ra3_e@d1_ra3_ha3);
allowance_ha3 = 133 - occupancy_ha3;
...

// AGENT DYNAMICS (FUNCTIONAL RATES)

// From ra to ha through d1
kineticLawOf ra_e_in__d1_ra_ha : fMA(to_ra_d1 * switch_d1 * ra_e_in_safe);
kineticLawOf ha_e_out__d1_ra_ha : fMA(from_d1 * open_d1 * safeD1_ra_e * allowance_ha);
kineticLawOf ra_w_in__d1_ra_ha : ...
kineticLawOf ha_w_out__d1_ra_ha : ...
// ... and back
kineticLawOf ha_e_in__d1_ra_ha : ...
...

// AGENT DEFINITIONS (SEQUENTIAL PROCESSES)

RA_e = (ra_e_in__d1_ra_ha, 1) << RA_e@ra + ...
      (ra_e_out__d1_ra_ha, 1) >> RA_e@ra + ...
RA_w = ...
HA_e = ...
...
D1_ra_e = (ra_e_in__d1_ra_ha, 1) >> D1_ra_e@d1_ra_ha +
          (ha_e_out__d1_ra_ha, 1) << D1_ra_e@d1_ra_ha;
D1_ra_w = ...
D1_ha_e = ...
D1_ha_w = ...
...
// SYSTEM DEFINITION (MODEL COMPONENT)

building ::= (RA_e@ra[18] <> RA_w@ra[18] <> HA_e@ha[0] <> ...)
           <*>
           (D1_ra_e@d1_ra_ha[0] <> D1_ra_w@d1_ra_ha[0] <> ...)

```

Figure 2: Excerpts from the Emergency Egress BioPEPA model.

current occupancy of the arrival room, i.e. `allowance_ha`.

The mass action law might not appear a very natural candidate for modelling the dynamics of emergency egress, since, being dependent on the numbers of the participating agents, it would imply that the more agents are in a room the faster they traverse a door. However, we have chosen to adopt it because it guarantees Markovian properties and has the intuitive reading of linking the expected values of distributions to the inverse of their rates. This choice is pragmatically justified by the fact that the capacity control on doors, i.e. max two agents can traverse a door together, makes the flux of evacuees through doors almost linear in time

and independent of the number of agents in a room. This leads to an emergent overall egress behaviour very coherent with other results in literature, as discussed below. Not surprisingly, the door allowance turns out to be a very sensitive parameter within egress scenarios, as also shown by models based on different approaches, such as in [13] where Langevin equations have been used.

It is worth noting how information has been encoded in the model specification, such as labeling each agent with its final destination. On one hand, this follows the choice of having a detailed model, so that, for instance, it could be possible to close a door only for agents heading towards one of the two possible exits. On the other hand, this is imposed by the nature of the Bio-PEPA language that has been designed striving for simplicity, and has been equipped with a parsimonious set of operators (and also has been specifically devised for modelling basic molecular interaction). However, the compactness of the language makes the translation from higher level, domain oriented, specification languages into Bio-PEPA quite easy and facilitates the adoption of a more abstract specification language, as done for instance in [16] for PEPA. It is also worth noting that the chosen specification language has allowed us to precisely describe the individual agent behaviour, which cannot be addressed in the settings of [12].

Finally, after the definition of each agent, their composition into the overall system is formalised. According to this model there are two sets of agents in the building: those representing evacuees in rooms and those representing evacuees that are traversing a door. The agents of each one of these sets do not synchronise with each other on any action $\langle \rangle$. Differently, agents in the two different sets synchronise on every action they share $\langle * \rangle$: whenever an agent moves from or into a room, it has to do it together with the agent representing an evacuee that is crossing the relevant door (this matching is guaranteed by a suitable matching of action names, as discussed for `ra_e_in_d1_ra_ha` above). The overall system definition also determines the initial distribution of agents within the building at the time of an alarm, e.g. there are 18 `RA_e` and 18 `RA_w` in the building in room `ra`.

4.2 Some Analysis Results

Once the model and initial conditions have been specified, several kinds of analysis can be carried out automatically. Stochastic simulations are based on the Gillespie Stochastic Simulation Algorithm (SSA), a Monte-Carlo sampling algorithm [6]. Intuitively, according to the current state, the quantities of the present agents and their definitions, the next most likely to occur interaction and its occurrence time are chosen. The state is updated according to the effects of the chosen interaction: the amount of some agents may vary and the current time is updated. Fluid flow analysis interprets the system in terms of a set of ODEs, which are numerically solved (see Section 3.1).

All analysis results are reported in figures over a suitable time interval depicting the variations, either continuous (ODE), discrete (G) or averaged (Nx G), of the numbers of relevant agents.

Figure 3 reports the occupancy of each room during evacuation. In Figure 3(a) a single stochastic simulation is depicted. It can be observed how the rooms where agents initially are (`ra`, `rb`, `rc`) become quickly empty within the first half minute. Hall `ha` becomes then crowded, while other rooms, like the staircases (`se`, `sw`) have a quite low occupancy during the entire time interval. This can be explained by the staircases directly and quickly leading to an exit. The number of exited agents grows linearly until the end of evacuation, within the first two and a half minutes. Several of these details, i.e. those relative to small numbers like staircase

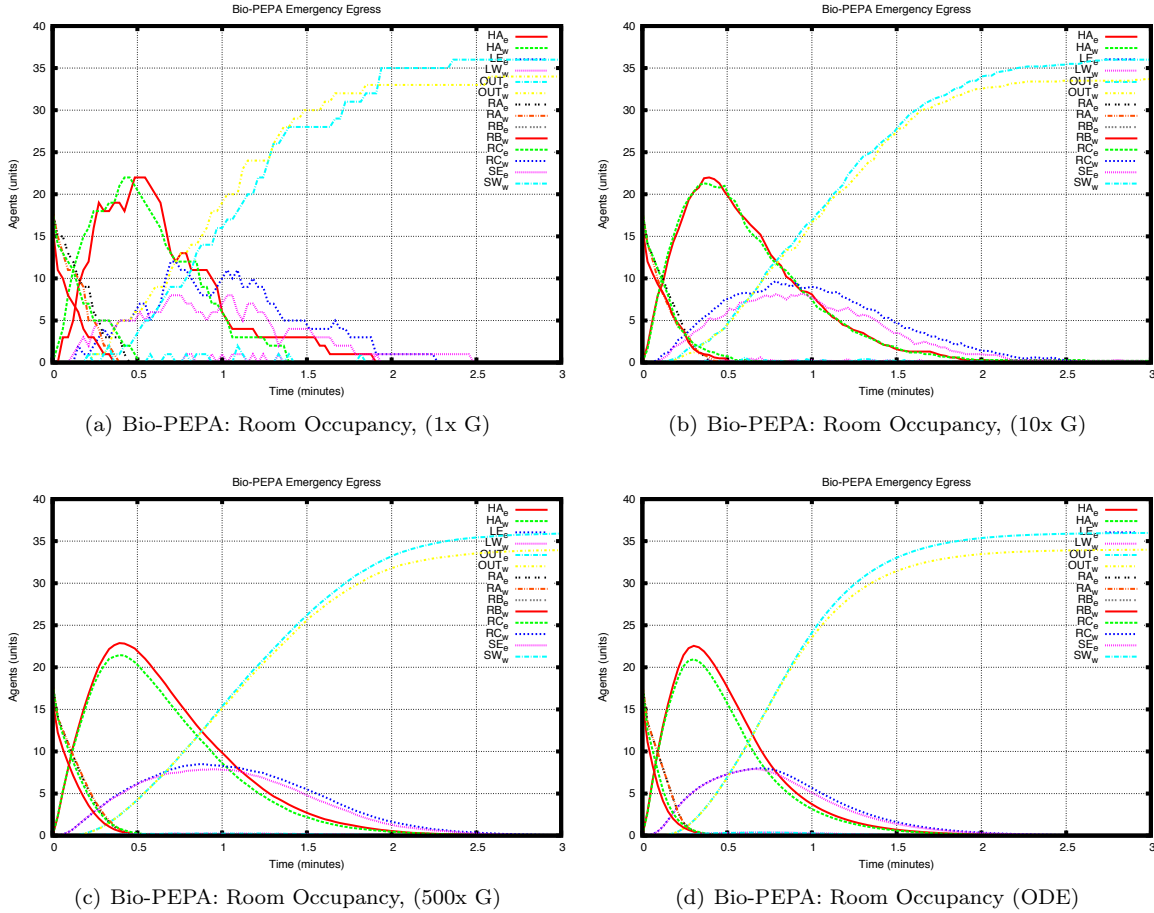


Figure 3: Bio-PEPA Emergency Egress: Results.

occupancy, tend to be blurred in Figure 3(b), where the average of 10 stochastic simulations is reported. Averaging over a larger number of stochastic simulations, 500 in Figure 3(c), tends quite precisely to the results of the approximation obtained by means of fluid flow analysis and reported in Figure 3(d). This correspondence is expected for systems consisting of a large number of agents, but happens, as in this case, also for not so large systems, proving that fluid flow analysis can provide informative results for a wide range of cases. For the sake of brevity, we refrain from presenting here a comparison of the efficiency of the analysis techniques, but refer to the study in [17]. There, the ODE analysis of a large system of 20.000 agents costed about 16 mins, while a *single* stochastic trace for the same time interval ran for 450 mins.

Figure 4 reports the case in which no evacuation plan is provided to agents, over a $[0, 10]$ minutes time interval. Technically, this can be easily obtained by means of the switching variables associated to each door. As explained, preferences degrees or boolean conditions can be associated to the traversal of each door in each room by each agent, accordingly to its final destination. This way of specifying may appear a bit ad-hoc, but it can be generalised whenever one uses a more abstract language, which will then be translated into low-level mechanisms,

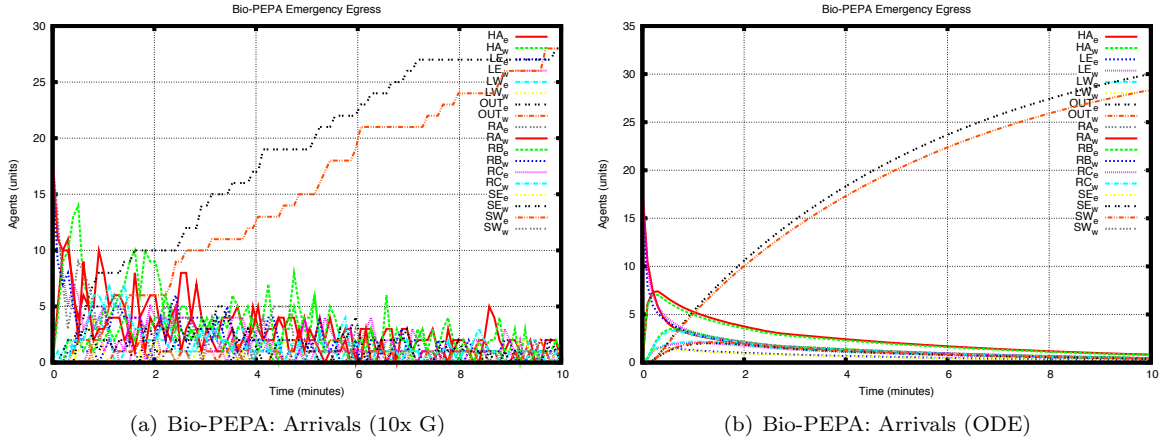


Figure 4: Bio-PEPA Emergency Egress: No Routing.

as the ones provided by Bio-PEPA. The lack of any such information corresponds to agents having no directions. Clearly, when agents move about randomly evacuation times increase. By observing the ODE approximation, one could understand that all rooms become almost empty around minute 6. Observing the stochastic simulation, instead, it is easy to realise that several agents are still bouncing from one room to another searching for an exit. This illustrates the approximation of fluid flow analysis in the presence of a low number of agents. Enriching the continuous analysis with variance and higher-order moments would result in a more informative analysis, but also a more computationally expensive one, see [8] for results on this topics.

Figure 5 reports a sanity check for our model. Our results are confronted with those obtained by the PEPA model in [17]. It is worth recalling that that model has been shown to be largely coherent with the results obtained from the original work in [12], up to the expected differences, since the results therein are given in terms of optimal fluxes. Figures 5(a)-(b) report the agents evacuation time course, which are quite coincident, although the Bio-PEPA model is slightly faster initially (it must be remembered that the two models are anyway based on different languages with different modelling abstractions, in particular different synchronisation-rate policies). Analogously, Figures 5(c)-(d) report room occupancy and the Bio-PEPA model results are again slightly faster in the room dynamics, but within an overall correspondence.

Agents can be equipped with a simple planning capability: when they get to know that one of the two possible exit paths is congested they may change their final destination. This is done by measuring the instantaneous congestion level of the `le` and `lw` landings, i.e. the difference between their capacity and occupancy. When the difference between the two congestion levels becomes greater than a threshold, a number of agents in the central hall `ha` change their destination (technically, they turn into agents with a different destination) with a speed ruled by an exponential distribution with a suitable rate (under the hypothesis that there are enough agents that can change destination). This is modelled by adding two actions to the `HA_e` and `HA_w` agents, which now must synchronise on these swapping actions:

$$HA_e@ha[n] \langle swap_to_ha_W, swap_to_ha_E \rangle HA_w@ha[n]$$

The rate of a swapping action (here the one for `HA_w`) is:

$$swap_on * H(allowance_le - allowance_lw - swap_threshold_WE_ha) *$$

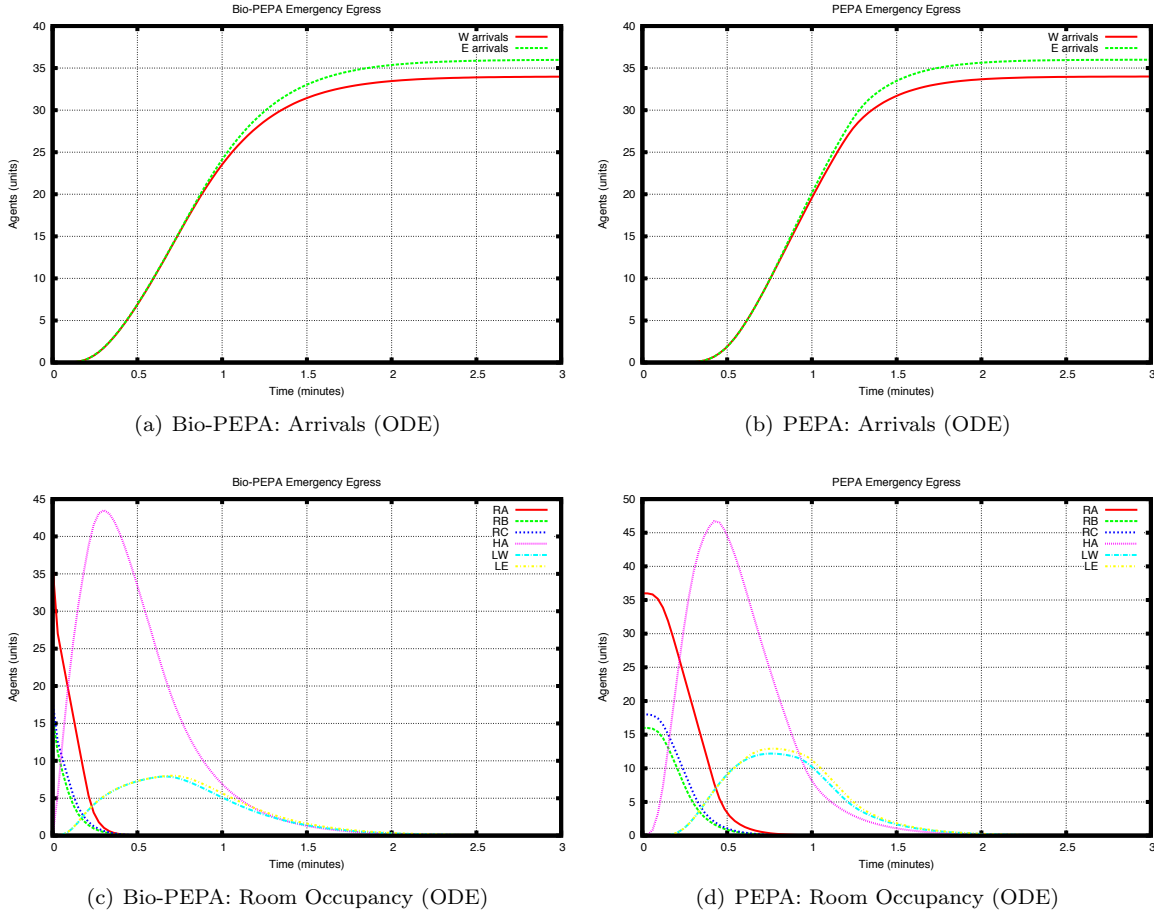


Figure 5: Bio-PEPA and PEPA Emergency Egress: Result Comparison (ODE).

$$H(\text{enough_HA_w_to_HA_e}) * HA_w@ha * \text{swap_speed_WE_ha};$$

where the first three terms are switches controlling the activation of planning, the threshold and that there are enough agents, while the last two terms represent a law of mass action of rate `swap_speed_WE_ha`.

The threshold, speed and numbers of swapping agents are three parameters whose effects can be studied for determining the most effective rerouting strategies.

Figure 6 illustrates analysis results for a case with and without rerouting capabilities. In order to better appreciate the effects of rerouting we have used a more populated building with 400 agents. We have further restricted capacity of the exit path towards east and we have augmented door capacity to speed-up system dynamics. By comparing a single stochastic simulation with and without rerouting capabilities in Figures 6(a)-(b), it emerges that the rerouting capability makes egress much faster: it is completed, according to these two simulations, within 3.5 mins instead of about 5. This is due to the fact that fewer agents try the east path with less capacity, see for instance the difference for `HA_w` in the two figures, and to the fact that the occupancy of the two exit paths is kept balanced, a best practice for load distribution. Bal-

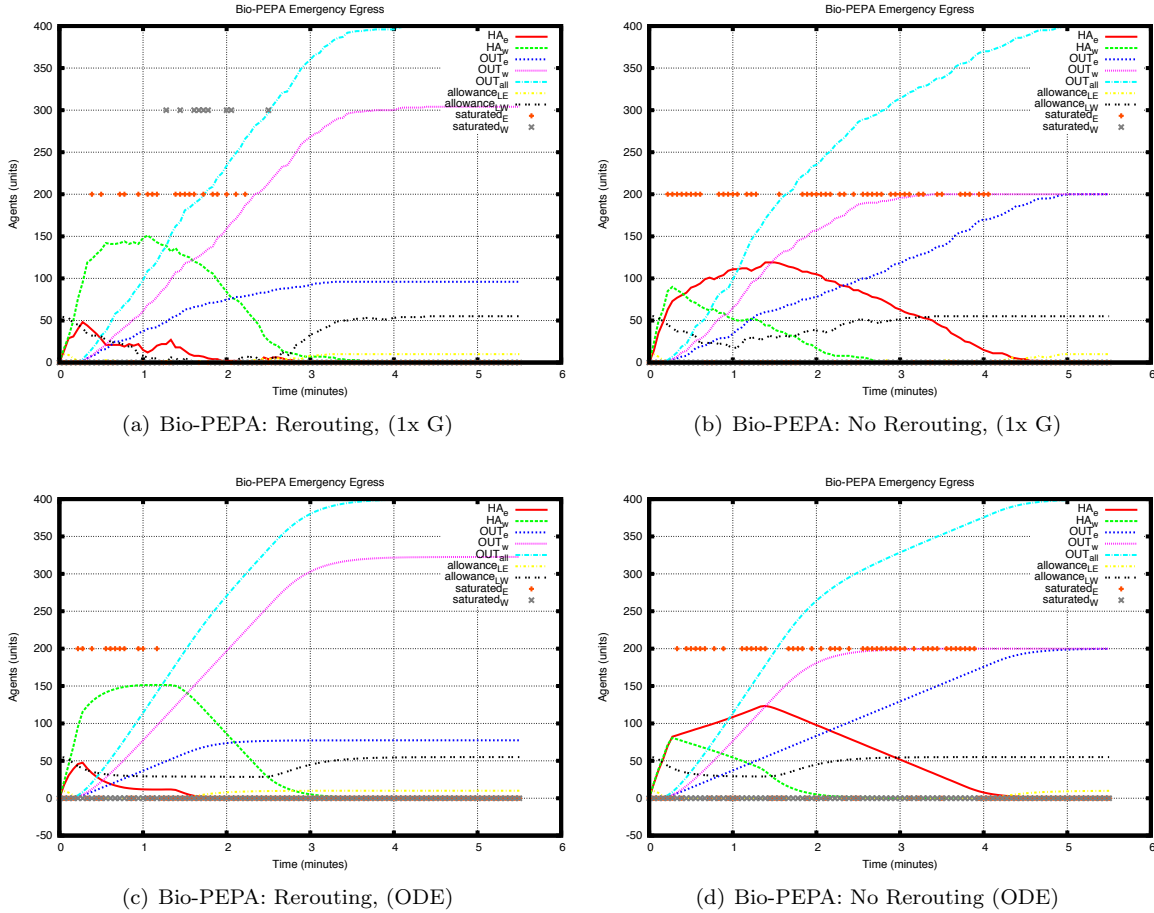


Figure 6: Bio-PEPA Emergency Egress: Rerouting.

anced occupancy can be appreciated in Figure 6(a), where the dots representing the saturation of the two paths are evenly distributed, while the ones for west path saturation are absent in Figure 6(b). Analogous results can be appreciated by means of fluid flow analysis and are reported in Figures 6(c)-(d). This quite good correspondence, validates the results based on a single stochastic simulation with stronger statistical significance. On the other hand, note that the ODE analysis hardly captures the singular points relative to west path saturation, while the more frequent ones relative to the east path are shown. This experiment shows well the complementarity of the two analysis techniques.

4.3 The Bio-PEPA plug-in

The modelling in Bio-PEPA is supported by a tool suite developed as a plug-in (<http://homepages.inf.ed.ac.uk/s9552712/bio-pepa/plugin.html>) for the Eclipse (<http://www.eclipse.org>) open cross-platform development environment, see Figure 7. The plug-in, currently still under development, comprises an editor to develop the model specification and a tool bar reporting information about the model, errors and warning, the file

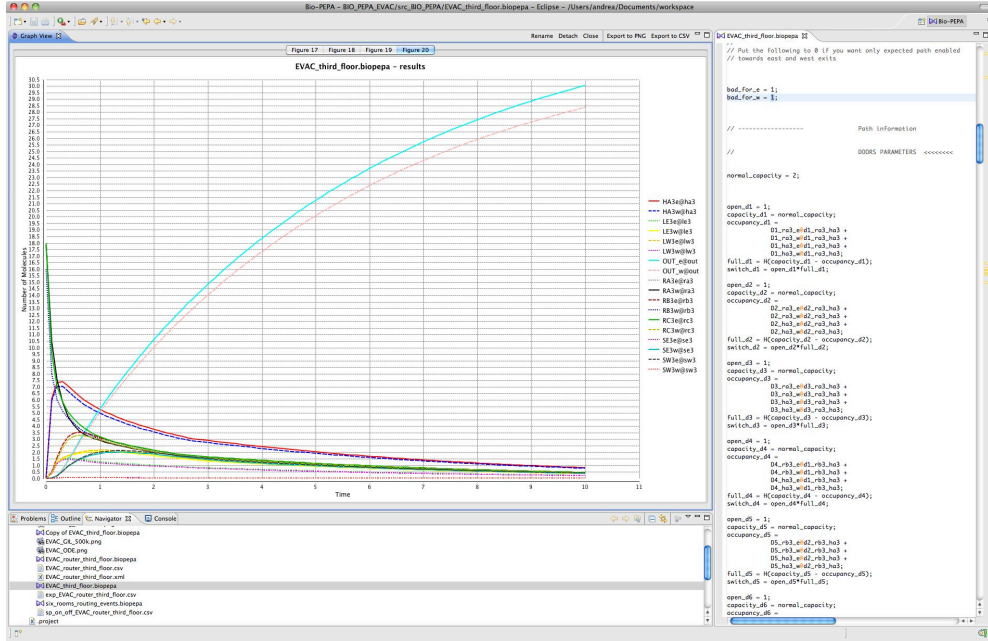


Figure 7: The Bio-PEPA Eclipse plug-in.

systems and the like. It embeds both stochastic simulation engines based on the Gillespie algorithm and several ODE solvers. Support for experimenting with parameter variations and reaction knock-down is provided. Simulation results can be saved as CSV files or visualised by means of an embedded graph view tool. The plug-in is open to be integrated with other tools. It is planned to allow the model specification to be exported in SBML and simulations to be represented in XML for Traviando, a simulation trace analysis tool (<http://www.cs.wm.edu/~kemper/traviando.html>).

5 Conclusions and future work

The application of modelling techniques to the analysis of crowd dynamics and systems containing a vast number of human and computational mobile agents appears to be an open research topic. We have explored the application of continuous and discrete quantitative techniques to this problem. Stochastic simulation is well suited to address indeterminacy and emergent behaviour that can be triggered by a low number of agents, but it is generally computationally expensive. Fluid flow analysis can approximate well the system behaviour, especially when agents are present in such high numbers as to make the frequency of actions high and the relative system change from one event small. Continuous analysis has given good approximations also for the case study we discussed. Continuous techniques can be much more computationally viable than discrete ones, making them much more suitable for use in the system design phase, when many design options and parameter tuning are an issue.

We have shown how the proposed approach combines both analysis techniques making reference to the same process algebraic specification. It also models both individual agent

behaviour, following simple rules, as well as dynamic context dependent aspects. In all cases, the process algebraic approach is provided with a well-defined formal semantics, which makes it possible to directly relate the results obtained via the various analysis techniques. Several tools support the automation of the analysis in an ongoing development and integration effort.

Our research interest is in further developing the approach along a few main directions. We are interested in developing suitable linguistic abstractions to more precisely describe the dynamics of systems of a large number of mobile agents displaced in a physical environment. This may require the use of more high-level specification languages than the (Bio-)PEPA family, possibly including specific constructs for capturing specific aspects of crowd dynamics. Then, we plan to study the possibility of embedding in a process algebraic context other mathematical theories, which could be useful to better capture emergent behaviour. A starting point is represented by Langevin stochastic differential equations, used for crowd dynamics in [13] and compared against stochastic simulations of PEPA models in [18]. Also, we intend to integrate existing mature verification techniques and develop new ones for this specific domain. A promising candidate is quantitative model checking whose application to this field would benefit from further advances in expressiveness, by the development of ad-hoc logics, and in computational efficiency, possibly considering approximated verification techniques, such as checking of a sample of the space of the possible traces of a model, as for instance done by the PRISM model checker [14].

Acknowledgments. The authors would like to thank Stephen Gilmore, Maria Luisa Gorriero and Allan Clark (The University of Edinburgh) for their support with the Bio-PEPA plug-in, and Michael D. Harrison (Newcastle University) for his collaboration on precursor models of the one here presented. This research has been partially funded by the CNR project RSTL-XXL and by the Italian PRIN MIUR project PaCO. Jane Hillston is supported by the EPSRC ARF EP/c543696/01.

References

- [1] Bio-PEPA Home Page. <http://www.biopepa.org/>.
- [2] F. Ciocchetta, A. Duguid, S. Gilmore, M. L. Guerriero, and Hillston J. The Bio-PEPA Tool Suite. In *Proc. of the 6th Int. Conf. on Quantitative Evaluation of SysTems (QEST 2009)*, 2009.
- [3] F. Ciocchetta and M. L. Guerriero. Modelling biological compartments in Bio-PEPA. *ENTCS*, 227:77–95, 2009.
- [4] F. Ciocchetta and J. Hillston. Bio-PEPA: An extension of the process algebra PEPA for biochemical networks. *ENTCS*, 194(3):103–117, 2008.
- [5] F. Ciocchetta and J. Hillston. Bio-PEPA: A framework for the modelling and analysis of biological systems. *TCS*, 410(33-34):3065–3084, 2009.
- [6] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.
- [7] M. D. Harrison, M. Massink, and D. Latella. Engineering crowd interaction within smart environments. In *EICS 2009 - ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pages 117 – 122. ACM, 2009.
- [8] R. A. Hayden and J. T. Bradley. A fluid analysis framework for a Markovian process algebra. *TCS*, 411(22-24):2260–2297, 2010.
- [9] T. A. Henzinger. The theory of hybrid automata. In *LICS*, pages 278–292, 1996.
- [10] J. Hillston. A compositional approach to performance modelling, 1996. Distinguished Dissertation in Computer Science. Cambridge University Press.

- [11] J. Hillston. Fluid flow approximation of PEPA models. In *Proceedings of QEST'05*, pages 33–43. IEEE Computer Society, 2005.
- [12] T. M. Kisko, R. L. Francis, and C. R. Nobel. Evacnet4 user's guide, 1998. <http://www.ise.ufl.edu/kisko/files/evacnet/EVAC4UG.HTM>.
- [13] R.A. Kosińska and A. Grabowski. Langevin Equations for Modelling Evacuation Processes. *Acta Physica Polonica B Proceedings Supplement*, 3(2):365–376, 2010.
- [14] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic Model Checking for Performance and Reliability analysis. *ACM SIGMETRICS Performance Evaluation Review*, 2009.
- [15] M. Massink and M. Harrison. Modelling interactive experience, function and performance in ubiquitous systems. In *ENTCS*, volume 261, pages 23–42. Elsevier, 2010.
- [16] M. Massink, M. Harrison, and D. Latella. Scalable analysis of collective behaviour in smart service systems. In *Proceedings of the 25th Annual ACM Symposium on Applied Computing*, volume 2, pages 1173–1180. ACM, 2010.
- [17] M. Massink, D. Latella, A. Bracciali, and M. Harrison. A process algebraic fluid flow model of emergency egress. Technical Report 2010-TR-01, CNR-ISTI, 2010.
- [18] Joris Slegers. A Langevin Interpretation of PEPA Models. *Electr. Notes Theor. Comput. Sci.*, 261:71–89, 2010.