Under consideration for publication in Formal Aspects of Computing

Scalable Context-dependent Analysis of Emergency Egress Models

M. Massink[1] and D. Latella[1] and A. Bracciali[2] and M. D. Harrison[3] and J. Hillston[4]

[1]Consiglio Nazionale delle Ricerche, Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", Pisa, Italy,
[2]Department of Computing Science and Mathematics, University of Stirling, U.K.
[3]School of Computing Science, Newcastle University, Newcastle upon Tyne, UK
[4]School of Informatics, University of Edinburgh, U.K.

Abstract. Pervasive environments offer an increasing number of services to a large number of people moving within these environments, including timely information about where to go and when, and contextual information about the surrounding environment. This information may be conveyed to people through public displays or direct to a person's mobile phone. People using these services interact with the system but they are also meeting other people and performing other activities as relevant opportunities arise. The design of such systems and the analysis of collective dynamic behaviour of people within them is a challenging problem. We present results on a novel usage of a scalable analysis technique in this context. We show the validity of an approach based on stochastic process-algebraic models by focussing on a representative example, i.e. emergency egress. The chosen case study has the advantage that detailed data is available from studies employing alternative analysis methods, making cross-methodology comparison possible. We also illustrate how realistic, context-dependent human behaviour, often observed in emergency egress, can naturally be embedded in the models, and how the effect of such behaviour on evacuation can be analysed in an efficient and scalable way. The proposed approach encompasses both the agent modelling viewpoint, as system behaviour emerges from specific (discrete) agent interaction, and the population viewpoint, when classes of homogeneous individuals are considered for a (continuous) approximation of overall system behaviour.

Keywords: collective behaviour; validation; stochastic process algebra; fluid flow; context dependency;

1. Introduction

Smart signage systems [LMBM07, KKK05], intended as pervasive computing systems designed to support various kinds of user such as travellers, patients and tourists in physical environments, are an important area of research. These systems, embedded within physical spatial settings, combine the use of sensors, displays and handheld mobile devices to provide timely information and services relevant to the people in a particular environment or situation. Envisaged environments include airports, hospitals, museums and open air public events. In such systems users interpret provided information and are intended to carry out activities as a

Correspondence and offprint requests to: M. Massink, CNR-ISTI, Via Moruzzi 1, I56124, Pisa, Italy. email: M.Massink@isti.cnr.it result of what has been read. There are many interesting issues that need to be addressed in the design of such systems. These include the identification of congestion and interference when different groups of users move through the same common physical spaces; evacuation times under different assumptions about the presence of other people 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. Modelling such issues to provide predictive information about system design alternatives not only requires adequate user and environment models but also needs to be highly scalable, domain oriented, easy to develop and efficient before it can be adopted by practitioners in the early phases of design. The formal modelling and analysis of human flows and collective behaviour in such smart environments is still a challenging and largely unexplored problem.

In the past, stochastic models have been used to capture various aspects of human behaviour. In [DMF01] performance aspects of continuous interaction with a finger tracking system have been modelled and analysed and in [CJT01] stochastic models have been used to compare the usability of different interface designs. More recently, stochastic models of user and system behaviour have been used to analyse and predict user performance in the presence of external interruptions [tBFM$^+$09]. The interest in the use of stochastic models to address aspects of human behaviour may not come as a complete surprise because human behaviour is neither completely deterministic, nor is it completely random or irrational. Empirical research shows that temporal aspects of human behaviour and the likelihood of error follow stochastic probability distributions [SG83].

However, though there is experience with stochastic modelling in a traditional HCI setting, the issue of scalability has remained problematic. A fundamental problem with most formal modelling approaches and the analysis of large collections of processes modelling the behaviour of many independent people and services, is how to deal with the state space explosion due to the interleaving of such processes. Approaches based on simulation can handle, in principle, a large number of entities and rather complex behaviour, but these techniques are computationally expensive. These costs may be justified when a mature system design is analysed, but are often prohibitive when used to explore many options during early phases of design. Furthermore, any technique used during design should facilitate remodelling of the system, working with alternative assumptions and allow comparison of results. Other techniques such as stochastic model-checking can only be used for models with a number of states that is far below what would be needed to analyse collective behaviour in an airport or a realistic smart signage system. Though techniques such as model checking may be useful to explore the model for a small number of entities, other methods of analysis are needed to be able to analyse the effect of collective behaviour.

A promising alternative analysis method is fluid flow analysis. This analysis is tailored to the Performance Evaluation Process Algebra (PEPA) and Bio-PEPA, a variant of PEPA originally designed to model biomolecular interaction, both proposed by Hillston et al. [Hil05, CH08]. The method is based on the abstraction of the identity of similar processes. In this approach only the *number* of processes that are in a certain state at any time is recorded. For each collection of processes this number is represented as a continuous function of time. Underlying ordinary differential equations (ODE) describe the evolution of these functions over time and can be derived automatically from the high level process algebraic model and then used for a form of transient analysis. Such analysis provides a view of the quantitative aspects of the collective behaviour over time. Tool support exists [Tri07] for the automatic derivation of ODEs from a particular class of PEPA specifications, consisting of cooperating groups of independent copies of components, and for the provision of various forms of analysis including fluid flow analysis and stochastic simulation [Gil77]. Also for Bio-PEPA similar tool support exists [CDG$^+$09, Bio]. A first exploration of the application of this approach to the modelling of a generic smart signage system with PEPA can be found in [HML09] and [MHL10]. These works focus on feasibility and scalability of the approach.

In this article two main issues are addressed. The first concerns the *validation* of the quantitative aspects of a representative example of the aforementioned class of models of smart service/signage systems. The particular instance chosen is that of emergency egress. Models of emergency egress typically involve physical locations through which people move to one or more predefined exits following particular routes. Such models have been extensively studied in the literature and detailed information is available on realistic case studies. One of the most widely known approaches, called Evacnet4, is based on a capacitated network flow transhipment algorithm and used for the generation of *optimal* building evacuation plans. The user's guide of this method [KFN98] provides a detailed case study of emergency evacuation of a three storey building. This will be used here as a basis for the validation of the prediction of results. In particular, we compare their results with egress times and room occupancy obtained by a fluid flow analysis of a process algebraic model of the same system.

The second issue concerns the inclusion of typical, *context dependent*, aspects of realistic evacuee behaviour in the egress model. A number of such aspects have been mentioned in recent critical reviews of emergency evacuation simulation models [SA05, Sti00]. Examples are evacuees that get wounded or intoxicated and as a consequence change their speed of movement or obstruct part of a room or corridor. Evacuees may also change direction when they observe danger in front of them and some may start searching for missing friends, colleagues or family members. There are many such observed social behaviours and they may have significant impact on the success of an evacuation. This kind of behaviour, in general, cannot be taken into consideration in models suitable for optimal flow analysis, such as Evacnet4, because of the particular way the underlying graph algorithms work. It is also observed in [SA05] that current modelling techniques are either not suitable to take such dynamic behaviour into account, or are based on detailed simulation techniques, requiring time and computational resources. Our contribution is specifically centered on the combination of two aspects. One is the use of process calculi that provides expressive and abstract linguistic composition operators for description of behaviour. The other one is a mathematically sound and efficient analysis technique. This framework integrates an *agent view*, where the behaviour of single agents is specified explicitly and the overall behaviour of the system emerges from the interaction of agents over time, and a *population view*, where system dynamics is observed in terms of classes of agents homogeneously following the same patterns of behaviour. In particular, we show how PEPA models can be used to model certain aspects of social dynamic human behaviour and to what extent a fluid flow analysis may contribute to obtaining efficiently realistic approximations of the results provided traditionally by approaches based on stochastic simulation for this class of models. It is shown that when the overall mean behaviour of the system is of interest, the fluid flow approximation provides a quick and appropriately accurate approximation of the limit behaviour. Of course, simulation may provide more detailed information, at much higher computational costs, which are in general not affordable during early phases of design.

Bio-PEPA gives us a further dimension. It makes it possible to describe how agents can change egress behaviour such as adding the possibility for agents to change the provided evacuation plan based on information they might sense from the environment via system communication. The Bio-PEPA process algebra 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 which can be used to describe the building topology and node capacity and locate agents within it. Also, the agent behaviour can be expressed as a function of the current state of the system, like level of congestion in given rooms, an abstraction of sensing the environment, common to standard agent models. Thanks to these features, simple planning behaviour can be embedded in agent models: whenever an agent gets to know that one of the two alternative paths to exit the building 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 occupancy of the two exit paths is kept balanced.

The comparison of the complexity/performance of the optimal flow algorithms underlying Evacnet4 versus fluid flow approximation is not addressed in this article. The two approaches are complementary and incomparable. Optimal flow algorithms are used to calculate *optimal* egress plans abstracting from social dynamic behaviour, whereas fluid flow analysis can be used to analyse models of different *given* egress plans possibly *including* aspects of social dynamic behaviour.

The outline of the article is as follows. First the emergency egress case study described in [KFN98] is presented in Section 2. This is followed by a brief introduction to the fluid flow semantics of the considered class of PEPA specifications in Section 3. In Section 4 a PEPA model of the case study is introduced and a comparison of the results concerning egress time and presence of persons in the building over time is presented in Section 5. Subsequently, in Section 6 an overview of commonly observed adaptive behaviour in emergency egress situations is given and an example is shown of how such behaviour can be included in the overall model and what contribution fluid flow analysis may provide to the analysis of the effect of such behaviour. In Section 7 a scaled version of the model is analysed dealing with a scenario with 21,200 evacuees. The issue of context dependency is discussed in Section 8, where Bio-PEPA is introduced together with a Bio-PEPA model concerning dynamic, context dependent route planning by agents and its analysis. Finally, in Section 9 a summary and discussion of the results and directions for future research are provided.

This article collects and extends earlier results that appeared in [MBLM10], mainly regarding the validation of the developed PEPA egress models, and in the informal participant proceedings of the Third Int. Workshop on Logics, Agents and Mobility, 2010 [BHLM10], focussing on the treatment of contextual information in Bio-PEPA models.

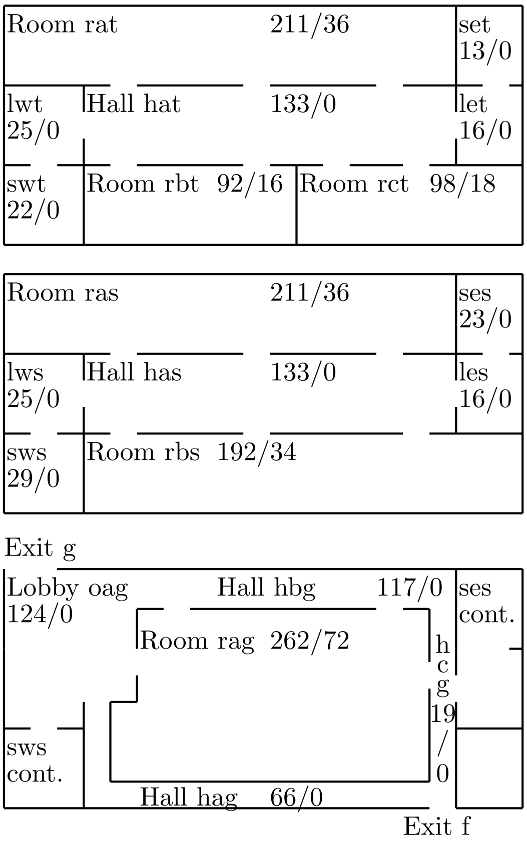


Fig. 1. Structure of example building from EVACNET4 manual (NC/IC)

2. The emergency egress case study

The case study used to validate the fluid flow models of smart signage systems presented in [HML09, MHL10] is a representative example of the emergency evacuation of a three storey building, originally introduced in [KFN98]. The case study description includes human factors details such as how fast people cover a certain distance on average and how many people may pass on average through a standard door in a given time period. In Figure 1 the three storey building of the case study is shown. Rooms and other building sections referred to as ‘locations’, ‘sections’ or ‘nodes’ in this article, are identified by a three letter acronym. The first letter indicates whether it is a room (r), a hall (h), a stairwell (s), a landing (l) or a lobby (o). The second letter indicates which room, hall etc. it is, by a letter ranging from a to z. The third letter indicates the floor level; ground floor (g), second floor (s) and third floor (t). The numbers NC/IC denote the *node capacity* (NC) and the number of people *initially present* (IC). Doors between building sections are indicated by spaces. Doors are also assumed between adjacent halls and between the hall and the lobby on the ground floor.

When an alarm occurs it is assumed that evacuees are going to the nearest of the two exits situated on the ground floor of the building, indicated as ‘Exit g’ and ‘Exit f’ in Figure 1, following the shortest way out.

3. PEPA and fluid flow semantics

Systems are described in PEPA as interactions of components that may engage in activities. Components reflect the behaviour of relevant parts of the system. Activities capture the behaviours that the components perform. A component may itself be composed of components. A PEPA activity consists of a pair *(action type, rate)* in which *action type* (sometimes abbreviated by simply *action*) denotes the type of the action, while *rate* characterises the negative exponential distribution of the activity duration. In this article durations will be measured in minutes. A positive real-valued random variable X is exponentially distributed with rate $r > 0$ if the probability of X being at most t, i.e. $Prob\{X \leq t\}$, is $1 - e^{-r \cdot t}$ if $t \geq 0$ and is 0 otherwise, where t is a real number. The expected value of X is $1/r$. The PEPA expressions considered in this article may be formally specified using the following grammar:

$$S \;::=\; (\alpha, r).S \;\big|\; S \;+\; S \;\big|\; C \qquad P \;::=\; P \underset{L}{\bowtie} P \;\big|\; S$$

where S denotes a *sequential component* and P denotes either a *model component* or a sequential component. C stands for a constant which denotes a sequential component. A sequential component $(\alpha, r).S$ carries out activity (α, r). After performing the activity, the component behaves as S. Component $S_1 + S_2$ models a system that may behave either as S_1 or as S_2, representing a race condition between the two components. The model component $P_1 \underset{L}{\bowtie} P_2$, where $\bowtie$ is the cooperation operator, defines the set of action types L on which model components P_1 and P_2 must synchronise (or cooperate); P_1 and P_2 proceed independently with any activity not occurring in L. The expected duration of a cooperation of activities of action type a belonging to L is a function of the expected durations of the corresponding activities in the components. Essentially, it corresponds to the longest expected duration (i.e. the slowest rate). A formal semantics of PEPA based on Continuous Time Markov Chains (CTMC) can be found in [Hil96]. Two shorthand notations are introduced: If the set L is empty $P_1 \underset{L}{\bowtie} P_2$ is written as $P_1 \parallel P_2$; if there are n copies of P_1 in parallel cooperating over L with m parallel copies of P_2 this is written as: $P_1[n] \underset{L}{\bowtie} P_2[m]$. Constants are defined by means of proper defining equations. In this way legal PEPA components can be described as cooperations of sequential processes.

PEPA semantics allows for the application of different analysis and evaluation techniques including fluid flow analysis. This is illustrated by means of a small example taken from [TGH10], for details see [Hil05]. Consider the following model:

$$\begin{array}{rcl}
P & = & (\alpha, p).P' \\
P' & = & (\beta, p').P \\
Q & = & (\alpha, q).Q' \\
Q' & = & (\gamma, q').Q \\
System & = & P[N_P] \underset{\{\alpha\}}{\bowtie} Q[N_Q]
\end{array}$$

This model comprises two classes of components; one class with N_P instances of process P and one class with N_Q instances of process Q. Each pair of components (P, Q) can cooperate over action type α. Processes P' and Q' carry out independent actions β and γ, respectively, before returning to the state in which α may be performed. In a fluid flow interpretation of the model a function can be associated to each state occurring in the specification, i.e. to P, P', Q and Q'. Each function yields a continuous approximation of the total mean number of components which are in the related state at a given time instant. So, for the above example we would obtain four functions $\mathcal{P}(t)$, $\mathcal{P}'(t)$, $\mathcal{Q}(t)$, $\mathcal{Q}'(t)$ with the following initial values: $\mathcal{P}(0) = N_P$, $\mathcal{Q}(0) = N_Q$, and $\mathcal{P}'(0) = \mathcal{Q}'(0) = 0$. Let us focus on the dynamics of $\mathcal{P}(t)$, i.e. the (approximation of the) average number of processes in state P at time t. This number *increases* whenever a β-labelled transition takes place. It *decreases* whenever an α-labelled transition occurs. Intuitively, in the first case, the 'average amount of change' is proportional to the number $\mathcal{P}'(t)$ of processes that are in state P' with a factor equal to the rate p' of β-labelled transitions. The 'average amount of change' when $\mathcal{P}(t)$ decreases is given by $min(p\mathcal{P}(t), q\mathcal{Q}(t))$ because in this case one needs to consider that (i) α-labelled transitions are cooperation transitions and that (ii) the above mentioned PEPA design choice for cooperation requires the *slowest* rate to be chosen. Thus the following differential equation can be derived, which relates $\frac{d\mathcal{P}(t)}{dt}$ to $\mathcal{P}(t)$, $\mathcal{P}'(t)$ and $\mathcal{Q}(t)$:

$$\frac{d\mathcal{P}(t)}{dt} \;=\; p'\mathcal{P}'(t) - min(p\mathcal{P}(t), q\mathcal{Q}(t))$$

By repeating the same reasoning for P', Q and Q' one obtains the following system of differential equations:

$$\begin{array}{rcl}\frac{d\mathcal{P}(t)}{dt} & = & p'\mathcal{P}'(t) - min(p\mathcal{P}(t), q\mathcal{Q}(t)) \\ \frac{d\mathcal{P}'(t)}{dt} & = & -p'\mathcal{P}'(t) + min(p\mathcal{P}(t), q\mathcal{Q}(t)) \\ \frac{d\mathcal{Q}(t)}{dt} & = & q'\mathcal{Q}'(t) - min(p\mathcal{P}(t), q\mathcal{Q}(t)) \\ \frac{d\mathcal{Q}'(t)}{dt} & = & -q'\mathcal{Q}'(t) + min(p\mathcal{P}(t), q\mathcal{Q}(t))\end{array}$$

to be solved with initial conditions as above. Such systems of differential equations can be solved numerically. In this work we use the PEPA-plugin analysis tool [Tri07] that provides, among others, fluid flow analysis and stochastic simulation. Stochastic simulation is based on the original CTMC semantics of PEPA.

Several approaches have been developed to investigate the relationship between the solution of the derived differential equations and stochastic simulation. We give an informal hint of the approach by Kurtz [Kur70] taken from [TGH10] and mention the results obtained by Hayden and Bradley [HB10]. In order to sketch the approach by Kurtz it is useful to consider an alternative representation of the CTMC associated with PEPA models by PEPA's standard formal semantics [Hil96] as informally recalled above. This representation abstracts from component identities and records only the number[1] of components in each class that are in the same state at a given time. We illustrate this for the example above and we denote the relative aggregated CTMC by $CTMC(P[N_P] \underset{\{\alpha\}}{\bowtie} Q[N_Q])$. The generic state of such a CTMC corresponds to a global *System* state, which is represented by a 4-tuple of natural numbers (n_P, n'_P, n_Q, n'_Q), where n_X is the number of processes that are in state X in such a global state. Such a representation is also called a *numerical vector representation.* For instance, the initial state in the example is $(N_P, 0, N_Q, 0)$, and the execution of one α-labelled transition in this state corresponds to a transition from $(N_P, 0, N_Q, 0)$ to $(N_P - 1, 1, N_Q - 1, 1)$. The relevant rates can be easily derived using the PEPA semantics rules. Now, let us consider the family of CTMCs $CTMC_1, CTMC_2, \ldots$ where

$$CTMC_j = CTMC(P[j \cdot N_P] \underset{\{\alpha\}}{\bowtie} Q[j \cdot N_Q]).$$

In such a sequence of CTMCs the size of each of the cooperating classes of processes is multiplied by a factor $j \in \mathbb{N}^+$. It can be shown that, under certain hypotheses, asymptotically, a sample simulation path in $CTMC_n(t)$ may be closely approximated by $n \cdot (\mathcal{P}(t), \mathcal{P}'(t), \mathcal{Q}(t), \mathcal{Q}'(t))$ where $(\mathcal{P}(t), \mathcal{P}'(t), \mathcal{Q}(t), \mathcal{Q}'(t))$ is the solution to the initial value problem of the above set of ODEs with initial value $(N_P, 0, N_Q, 0)$. Although formally this correspondence between stochastic simulation and the numeric solution of the set of ODEs of a model occurs in the limit, in practise, often a good correspondence can be observed also when populations are relatively small. This is shown to be the case for the models presented in the following sections. We refer the reader interested in further details of the theory developed by Kurtz to [TGH10] and [Kur70].

In [HB10] a direct link is established between the aggregated CTMC interpretation of the class of PEPA models used in this article and the solution of the related derived differential equations. It is shown that for simple models *without* synchronisation the solution of the differential equations captures *precisely* the mean number of instances that are in a particular state at a given time. It is also shown that this solution is a *close approximation* of the *mean* of the transient distribution of components in a system for models *with* synchronisation.

4. PEPA model of emergency egress

In this section a concise description of the model is provided. It is composed of: (1) processes describing the behaviour of evacuees; (2) processes modelling doors between building sections; (3) processes modelling the handling of requests for information by the evacuees; and (4) processes modelling the occupancy of space by evacuees in the various building sections. In the following we describe each of the above components presenting excerpts of the PEPA model. For realistic studies such as the three storey building presented in Section 2 the complete PEPA models become very large and it is not feasible or at least easy to develop an error free version of the model by hand. However, given domain input that characterises the structure of the building, escape routes, initial location of evacuees at the time of alarm, etc., it is not difficult to

[1] Note that here the real number is intended and not the expected number.

generate such a PEPA specification automatically. This is also facilitated by the modular structure of the specifications.

4.1. Evacuees

Evacuees are assumed to be somewhere in the building when an alarm occurs, after which they move towards an exit (see Figure 1). Each evacuee is informed about (it is assumed) which building exit is the nearest and about where to go next every time they enter a new location. This information is conveyed for example via a smart badge or a handheld device.

As an example we illustrate the behaviour of evacuees located in Room *rat* on the third floor heading to exit g. The process name *EvcEgoprat* reads as "**Ev**a**c**uee going towards **E**xit g **o**ccupies a **p**lace in Room *rat*", from which location a request is made for information to go to exit g by means of synchronisation with a request handler on action *lrateg*. The name of this action reads as "when in **l**ocation *rat* and heading to **e**xit g where should I go next?". This request is sent implicitly without involvement of the evacuee e.g. by a handheld device, but modelled explicitly. Assuming that the request takes on average 1 second to be emitted, and choosing 1 minute as unit of time for the model, the rate of request can then be characterised by $a = 60$, that is, the expected time value at which the action occurs is $1/a$, i.e. $1/60 = 1$ second:

$$\mathit{EvcEgoprat} \;=\; (\mathit{lrateg}, a).\mathit{EvcEgRecrat}$$

After the request the evacuee awaits a response. This is modelled by *EvcEgRecrat*:

$$\begin{aligned} \mathit{EvcEgRecrat} &= (\mathit{egratlhat}, r).\mathit{MveEgFratThat} \\ \mathit{MveEgFratThat} &= (\mathit{nop}, \mathit{ustep}/d).\mathit{EvcEgFratThat} \end{aligned}$$

The response depends in general on the current location and the route available to the required exit. In the above case there is only one possibility. The evacuee moves from Room *rat* to Hall *hat* (third floor). In the general case, for example in Hall A on the third floor the specification would look like:

$$\begin{aligned} \mathit{EvcEgRechat} \;=\; & (\mathit{eghatlrat}, r).\mathit{MveEgFhatTrat} + (\mathit{eghatlrbt}, r).\mathit{MveEgFhatTrbt} \\ & +(\mathit{eghatlrct}, r).\mathit{MveEgFhatTrct} + (\mathit{eghatllwt}, r).\mathit{MveEgFhatTlwt} \\ & +(\mathit{eghatllet}, r).\mathit{MveEgFhatTlet} \end{aligned}$$

This specifies that the evacuee may, in principle, receive different indications of how to proceed because this location is connected to several other neighbouring locations. However, the request handlers used in this model are deterministic and therefore, upon a request from an evacuee, situated in a particular location heading to a particular exit, only *one* of the above 5 possible actions will be enabled corresponding to the deterministic routing information supplied by the request handler, as specified in Section 4.3. On average, it is estimated that the response takes 1 second to arrive. Hence $r = 60$. In the PEPA fragment above *MveEgFratThat* reads as "**M**o**ve** to go towards **E**xit g **F**rom *rat* **T**o *hat*". The moving itself takes time as well, which depends on the distance (d, in metres) that the evacuee must cover and the speed at which the evacuee moves. In [KFN98] an average speed of circa 60 m/min is used. We let *ustep* denote such a unit speed. The average time for covering d metres is thus d/ustep and the corresponding rate is ustep/d.

The distances considered in our model are those specified in [KFN98][2]. These distances model the average distance a person in a room needs to cover to reach the nearest door to the indicated adjacent room. Notice that, according to this definition, the distance covered for moving from a room to an adjacent one may be different from the distance from the latter to the former, i.e. distance depends on direction. This dependency is irrelevant in our validation experiment, since, as in [KFN98], the evacuation flow is pre-determined, but we have to take it into account in the extensions of the model discussed in Section 6.

When moving from Room *rat* to Hall *hat* the evacuee has to pass a door connecting the two locations, *rathat*, and does so with rate *udr*, which characterises the average time a person needs to pass through a single standard door. Upon entering the Hall the evacuee **o**ccupies a **p**lace in **l**ocation *hat* (action *lhatop*),

[2] They can be found in Table C-3 of that document.

and ***f***rees a ***p***lace in ***l***ocation *rat* (action *lratfp*). From this point the behaviour repeats, but now with the evacuee positioned in Hall *hat*:

$$\mathit{EvcEgFratThat} \quad = \quad (\mathit{rathat}, \mathit{udr}).(\mathit{lhatop}, s).(\mathit{lratfp}, s).\mathit{EvcEgophat}$$

At some point the evacuee reaches an exit, for example exit g, via the lobby on the ground floor *oag* (see Figure 1) passing through a door with rate *udr* (activity $(\mathit{oagg}, \mathit{udr})$). After ***o***ccupying a ***p***lace at exit ***l***ocation g (action *lgop*) and releasing a place in the lounge (*loagfp*), the evacuee also ***f***rees the ***p***lace at exit g (*lgfp*). The time it takes an evacuee to occupy or free a place is characterised by rate s. It is assumed that it takes on average one second, so again rate $s = 60$:

$$\begin{aligned} \mathit{EvcEgFoagTg} \quad &= \quad (\mathit{oagg}, \mathit{udr}).(\mathit{lgop}, s).(\mathit{loagfp}, s).(\mathit{lgfp}, s).\mathit{EvcEgArrived} \\ \mathit{EvcEgArrived} \quad &= \quad (\mathit{nop}, a).\mathit{EvcEgArrived} \end{aligned}$$

After these activities the evacuee has officially 'arrived' at the destination exit. This latter behaviour is modelled as an infinite loop by process *EvcEgArrived*, performing only activity *nop* (no-operation) with rate a without synchronisation. Modelling the status of the evacuee in this way permits visualisation of the number of evacuees that have arrived over time. The value of the rate at the self-loop at *EvcEgArrived* has no influence on the overall performance.

4.2. Doors

Doors are very simple processes. They model the average time needed to let one person pass through a standard door. The process *DUrathat* models a standard door from Room *rat* to Hall *hat* on the third floor:

$$\mathit{DUrathat} \quad = \quad (\mathit{rathat}, \mathit{udr}).\mathit{DUrathat}$$

The parameter *udr* characterises the average time it takes a person to pass through such a door. Based on experimental data from [KFN98] a rate of $\mathit{udr} = 30$ seems a good approximation.

When a door is non-standard, say twice as large, or when there are several doors from one section to another, this is modelled by specifying the number of door-processes of a certain type in the final process composition as will be explained later. Door processes synchronise with the evacuee processes modelling evacuees passing through doors. Note that the rate *udr* is specified both in the process modelling the door and in the process modelling the evacuee. This is because when there are many evacuees that need to pass by a limited number of doors, the bottleneck is formed by the number of doors. This is reflected by the semantics of the cooperation operator of PEPA which says that the effective rate of this synchronisation is the minimum of the number of evacuees times the rate at which each of them may pass a door and the number of doors times the rate at which they make an evacuee pass. Note that this also covers the case in which there are more doors than evacuees. In that case the effective rate is given by the number of evacuees times *udr*.

4.3. Request handling

Request handling models the responses given to an evacuee in a certain section of the building to requests for information about where to go next to reach a certain exit. For example, process *RqHLrat* can handle requests from evacuees in Room *rat* heading for exit g or exit f. In case the exit is g, as we have seen above, synchronisation on *lrateg* takes place and the response *egratlhat* is given to the evacuee. This means that in order to reach exit g the evacuee first has to go to location *hat*. After the response, the request handler is again available to handle further requests. The request and response rates are the same as defined earlier:

$$\mathit{RqHLrat} \quad = \quad (\mathit{lrateg}, a).\mathit{RqHLratRespEg} + (\mathit{lratef}, a).\mathit{RqHLratRespEf}$$

$$\begin{aligned} \mathit{RqHLratRespEg} \quad &= \quad (\mathit{egratlhat}, r).\mathit{RqHLrat} \\ \mathit{RqHLratRespEf} \quad &= \quad (\mathit{efratlhat}, r).\mathit{RqHLrat} \end{aligned}$$

In this model the router is static and deterministic. In other words, it always enables exactly one action that indicates where the evacuee should go next, also when the evacuee could in principle go to different neighbouring rooms. The process algebraic specification style however makes it easy to extend the behaviour with forms of stochastic routing, distributing evacuees over different possible routes.

The request handling services are not part of the original example [KFN98]. In our model they provide sufficient service level not to cause any significant delay to the evacuees. The time that evacuees need to orient themselves and follow indications to the exit in the original case study is assumed to be comparable with the display guides assumed in our model. This assumption has been made to be able to compare the overall timing results of the two models for validation purposes.

4.4. Places

When a place is free in a room (for example, *rat*) the process that models an evacuee entering the room can synchronise on action type *lratop* and ***o***ccupy a ***p***lace in ***l***ocation *rat*. The place can be released (***f***ree ***p***lace) by synchronising on action type *lratfp*. A PEPA fragment of a process modelling a place in Room *rat* is defined below. The value of rate s is the same as that defined in the process modelling the behaviour of evacuees.

$$\begin{aligned} \mathit{PlaceFreeLrat} &= (\mathit{lratop}, s).\mathit{PlaceFullLrat} \\ \mathit{PlaceFullLrat} &= (\mathit{lratfp}, s).\mathit{PlaceFreeLrat} \end{aligned}$$

4.5. Overall model architecture

The overall model consists of two main sets of independent processes. The first one contains the processes relative to the evacuees initially present in the building. For instance, $\mathit{EvcEgoprat}[18]$ represents 18 ***Ev***a***c***uees with destination ***E***xit g that initially ***o***ccupy a ***p***lace in ***r***oom ***a*** on the ***t***hird floor. The processes in this set do not synchronise with one another on any action (operator $\|$). The second set contains the processes relative to doors, request handlers and free and occupied places. These processes also do not synchronise with one another on any action.

The two sets are composed by means of a cooperation operator $\underset{\{lrateg,\ldots,lgfp\}}{\bowtie}$, forcing the processes in the two sets to synchronise on relevant actions. For instance, the presence of $lrateg$ requires that evacuees in room *rat* heading for exit g synchronise with the appropriate request handler in order to be routed along the correct path. The number of evacuees that are present in each room initially is given by the number between square brackets. This is also done for the available resources, such as free/occupied places, doors etc. The total number of evacuees in the model is 212.

$$\begin{array}{l} (\mathit{EvcEgoprat}[18] \parallel \mathit{EvcEfoprat}[18] \parallel \mathit{EvcEgoprbt}[16] \parallel \mathit{EvcEfoprct}[18] \parallel \mathit{EvcEgoprag}[36] \parallel \mathit{EvcEfoprag}[36] \parallel \\ \mathit{EvcEfopras}[18] \parallel \mathit{EvcEgopras}[18] \parallel \mathit{EvcEfoprbs}[17] \parallel \mathit{EvcEgoprbs}[17]) \\ \quad \underset{\{lrateg,\ldots,lgfp\}}{\bowtie} \\ (\mathit{DUrathat}[3] \parallel \ldots \parallel \mathit{DUhagf}[1] \parallel \mathit{RqHLrat}[4] \parallel \ldots \parallel \mathit{RqHLg}[4] \parallel \\ \mathit{PlaceFreeLrat}[175] \parallel \mathit{PlaceFullLrat}[36] \parallel \ldots \parallel \mathit{PlaceFreeLg}[13]) \end{array}$$

In this model it is assumed that evacuees go to the exit nearest (in distance) to the location in which they are initially. For rooms that are at the same distance from both exits it is assumed that the evacuees are equally distributed over the two exits. This routing, though plausible and realistic, differs somewhat from the optimal routing on which the evacuation times obtained with Evacnet4 are based. The latter aims at minimising the egress time, but implicitly assumes that evacuees are able to find the shortest (in time), less congested route to any of the available exits.

The complete model for the three storey building is composed of many kinds of processes, each of which with many instances in parallel as indicated in the final composition of the model. The resulting state space is far beyond what could be analysed currently with, for example, stochastic model-checkers. Therefore in the next sections we consider the application of the scalable fluid flow approximation to assess its applicability to the class of models presented in the current section.

from	to	n	d	from	to	n	d	from	to	n	d
rat	hat	3	7	hat	rct	2	15	let	hat	1	5
rbt	hat	2	7	hat	lwt	1	15	let	set	1	10
rct	hat	2	7	hat	let	1	15	swt	lwt	1	10
hat	rat	3	15	lwt	hat	1	5	set	let	1	10
hat	rbt	2	15	lwt	swt	1	10				

Table 1. Kind and number of doors and distances (from, to, number, distance).

loc	NC	nr. rqh	loc	NC	nr. rqh	loc	NC	nr. rqh
rat	211	4	hat	133	4	set	13	4
rbt	92	4	let	16	4	lwt	25	4
rct	98	4	swt	22	4			

Table 2. Node capacity (NC) and nr. of guidance services (nr. rqh) for each location (loc).

4.6. Automatic generation of the model

A prototype of an automatic generator for the PEPA model described above has been implemented in the functional programming language Haskell. As an example, the Tables 1, 2, 3 and 4 present the domain specific input data for the three storey building, restricted to those for the third floor for reasons of space, except for the data presented in Table 3. The translation from the tabular format of the tables above to PEPA is straightforward; consequently its correctness has been verified just by inspection. The full specification of the model can be found in the PEPA web site, in the examples section[3].

5. Validation of the model

The PEPA model presented in the previous section captures a number of important aspects of emergency egress. Agents model independent behaviour of typical evacuees that are guided to an exit following predefined routes through the various building elements. The size of the building elements, the presence of doors that may slow down people moving through the building, the average speed with which people move, the particular routes to the nearest exits, are all elements that are captured by the model. The automatic generation of PEPA models from domain orientated input makes it possible to study and analyse series of models considering different building structures, routing, evacuee behaviour and position of exits. In this section we show that fluid flow analysis of the model can provide information about evacuation times and the population of evacuees present over time in a particular building section (i.e. node profiles) as well as on congestion of escape routes for the routing considered. However, it is important to know whether such results

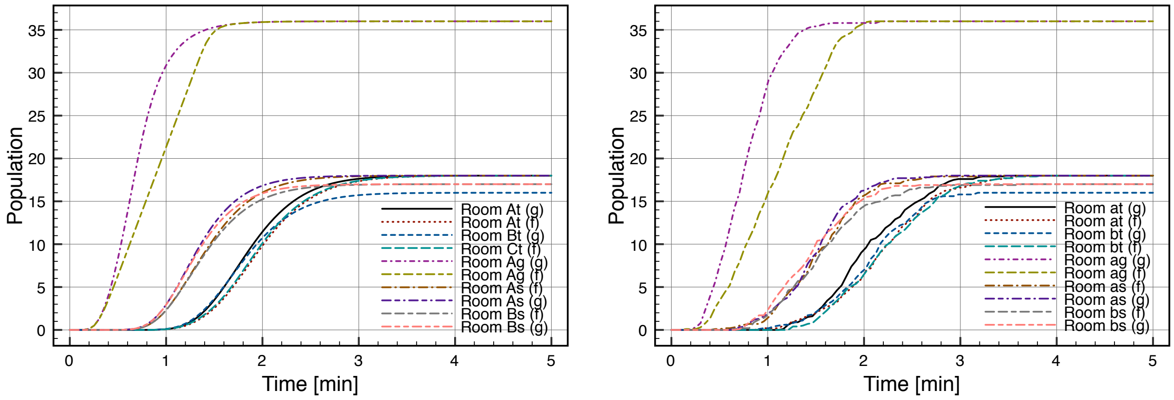
[3] http://www.dcs.ed.ac.uk/pepa/examples/

loc	exit	pres.	loc	exit	pres.	loc	exit	pres.
rat	g	18	rag	f	36	rbs	g	17
rat	f	18	rbt	g	16	rag	g	36
ras	f	18	rct	f	18			
rbs	f	17	ras	g	18			

Table 3. Evacuees initially present (pres) and related exit for locations (loc) at each floor.

from	exit	next	from	exit	next	from	exit	next
rat	g	hat	hat	g	lwt	set	g	les
rat	f	hat	hat	f	let	set	f	les
rbt	g	hat	lwt	f	hat	swt	g	lws
rbt	f	hat	lwt	g	swt	swt	f	lws
rct	g	hat	let	g	hat			
rct	f	hat	let	f	set			

Table 4. Routing of evacuees for floor 3.



(a) Arrival of evacuees (ODE)

(b) Arrival of evacuees (GIL)

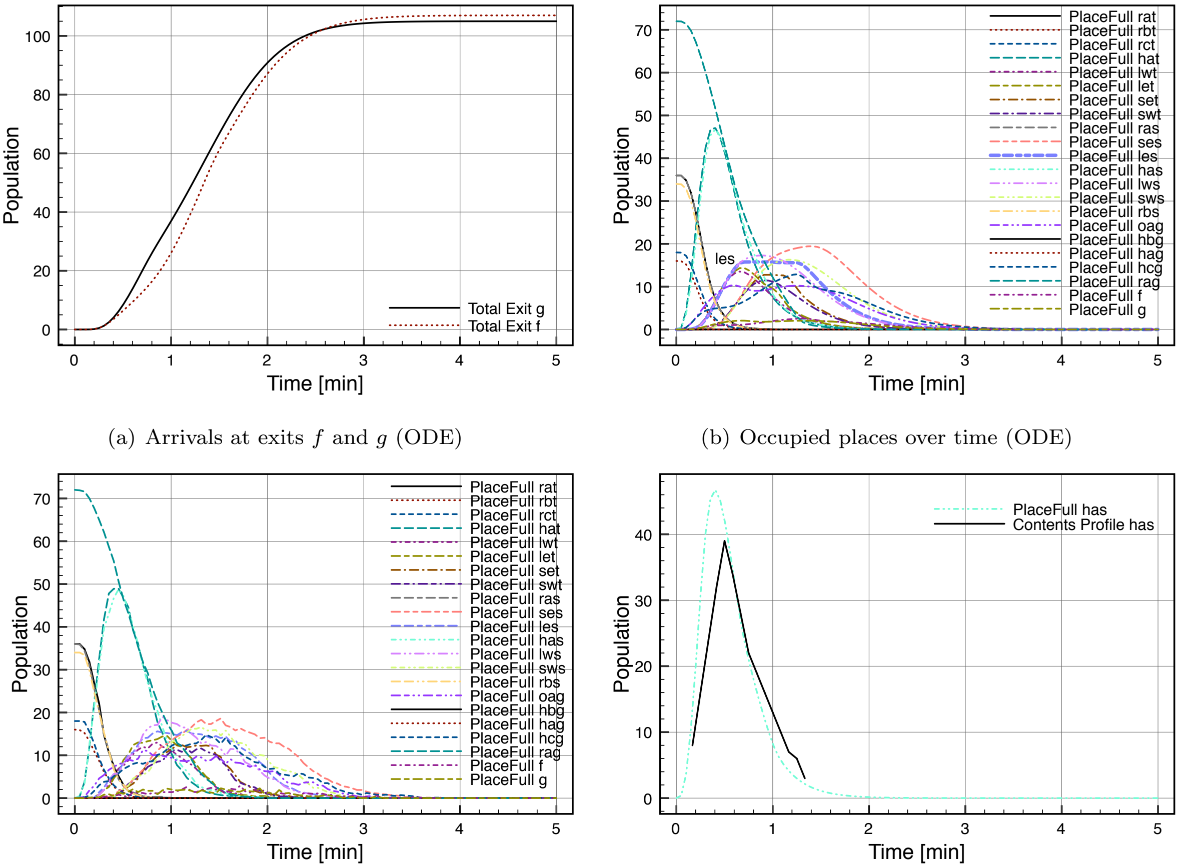
Fig. 2. Arrivals of evacuees

are compatible with those obtained with well-established approaches of emergency egress analysis such as Evacnet4. We therefore compare the fluid flow results for evacuation times and node profiles, obtained with the PEPA model for the three storey building, with the results of the optimal flow analysis obtained with Evacnet4 and presented in [KFN98]. The comparison takes both the predicted mean evacuation time of the three storey building and the population of evacuees present over time in a particular building section (i.e. its node profile) into consideration.

All stochastic simulation and fluid flow analyses concerning the PEPA egress models have been performed with the mentioned parameters on an Apple iMac 2.66GHz Intel Core i5 using the PEPA-plugin analysis tool [Tri07] unless otherwise stated. The PEPA-plugin relies on Gillespie's algorithm [Gil77] for stochastic simulation and on an adaptive step-size 5th order Dormand-Prince ODE solver for fluid flow analysis.

5.1. Predicted building evacuation time

In Figure 2(a) the arrivals of all evacuees from the different rooms are shown, over the time interval [0, 5] minutes using a fluid flow analysis (ODE). The labels in the legend are "Room xy (z)", where x indicates the name of the room, y indicates the floor and (z) indicates from which exit the evacuees left the building. For example, the leftmost curve in Figure 2(a) shows the evacuees arriving from Room a on the ground floor leaving the building through exit g. It is clear that evacuees on the ground floor are those nearest to the exits and therefore they reach these exits first. It is also clear that *all* evacuees from Room a on the ground floor reach their exit. In Figure 2(b) results are shown for a stochastic simulation (GIL) of the same model for the same time interval, with 10 independent runs. As can be observed, the curves in Figure 2(a) and Figure 2(b) correspond well despite the relatively small number of replicated runs of the stochastic simulation. The total



(a) Arrivals at exits f and g (ODE)

(b) Occupied places over time (ODE)

(c) Occupied places over time (GIL)

(d) Node profile Hall A second floor (ODE vs Evacnet4)

Fig. 3. Total arrivals and node profiles

(mean) time for all evacuees to leave the building is close to 3 minutes. This corresponds well to the 170 seconds predicted by Evacnet4 for the *optimal* evacuation time of the same three storey building.

Figure 3(a) shows the cumulative arrivals of evacuees from all floors for each of the exits. There are in total 105 evacuees for exit g and 107 for exit f. In the original case study these numbers differ resulting in 148 for exit g and 64 for exit f. This difference can be explained by the fact that in [KFN98] an *optimal* flow is calculated whereas in our model a *likely* flow is modelled in the sense that people go to the nearest (in distance) exit and not necessarily to the one that is fastest to reach.

5.2. Node profiles

Figure 3(b) shows the number of places occupied by evacuees over time for each of the building sections (rooms, halls, landings, stairwells etc.) on every floor as a result of a fluid flow analysis (ODE), over the time interval [0, 5] minutes. In Figure 3(c) the results are shown for a stochastic simulation (GIL) of the same model. Again both figures show a good correspondence despite the relatively few independent runs in the stochastic simulation. The leftmost curves show that all evacuees leave the rooms in which they are initially situated. The curve labelled *PlaceFull les* shows that after one minute the maximum capacity of the Eastern Landing, i.e. 16 places, of the 2nd floor staircase is reached indicating a bottleneck in the escape routes for the specified number of people present in the building at the time the signal for evacuation is given.

Figure 3(d) shows the comparison of the original node profile, at intervals of 5 seconds, for Hall *has* 2nd floor as reported in [KFN98] (solid line) and that of fluid flow analysis of our model (dotted line). The curve labelled *PlaceFull has* shows the number of evacuees in the hall with a maximum reached within half a minute, according to our model. This maximum is reached after about 30 seconds in the original node profile, showing close correspondence. The number of evacuees in the Hall at peak time is 39 in the original profile and 47 with the fluid flow model. The last evacuees leave hall *has* after about 1.5 minutes in the fluid flow model. In the Evacnet4 case the curve ends after circa 80 seconds, which is a little less than 1.5 minutes.

5.3. Summary of results

The results obtained within a fluid flow framework are consistent with those in [KFN98] for normative evacuee behaviour, that is evacuees that simply follow their way to the exit. The consistency between the results in [KFN98] and the fluid flow results are not exact because the results in [KFN98] are given in terms of *optimal fluxes*. The algorithm used in [KFN98] calculates the optimal schedule for evacuees to leave the building. In the fluid flow framework we have used a reasonably likely schedule instead, namely that evacuees go to the nearest exit to leave the building. This may not be identical to the optimal schedule because the optimal schedule also takes into account where congestion may occur and re-schedules evacuees to a different exit than the nearest one to minimize such congestion. This explains why the evacuation time reported in [KFN98] for the particular building studied is indeed slightly shorter than that obtained with the fluid flow approach. Nevertheless, given that the likely schedule of evacuees going to the nearest exit creates only minimal congestion, the case study in [KFN98] and its results establish a good calibration point for the fluid flow approach at least when evacuee behaviour is kept simple (as in [KFN98]).

However, the fluid flow framework also allows consideration of a richer evacuee behaviour and more realistic scenarios. In this sense, the fluid flow approach *complements* the search for optimal routing solutions. The model can be easily adapted to reflect scenarios in which evacuees get hurt or dynamically change their way choosing a different exit, as will be illustrated in the next Section.

The results also show a good correspondence between the fluid flow analysis and the average behaviour obtained via stochastic simulation. In some cases stochastic simulation can be more informative than fluid flow approximation, especially when an "individual or agent view" of the problem appears more suitable than a "population view", for instance when one is interested in observing the occurrence of exceptional individual behaviour or when the emerging behaviour is determined by a restricted number of individuals that does not justify a continuous approach. The formal relationship between the fluid flow approximation and stochastic simulation is one of convergence, when certain conditions are fulfilled, as discussed briefly in Section 3. The developed egress model is one of the cases in which we can observe a clear correspondence between stochastic simulation and fluid flow results, even for a relatively small number of stochastic samples and perhaps not so large population numbers.

6. Social aspects of evacuee behaviour

One of the observations made in the review of emergency evacuation models [SA05, Sti00] is that current models incorporate only limited assumptions about human social behaviour. The incorporation of such aspects in the models could render them much more accurate and realistic and useful to analyse various scenarios. Although approaches such as Evacnet4 do take certain aspects of human factors into account, such as average inter-person spacing, average speed in walkways or staircases and the effect of doorways on the dynamic capacity of a flow, no provision is made for social (group) interaction or agent modelling. Examples of the kind of issues that would make such models more realistic are the following.

Agent homogeneity. Most approaches assume that the persons moving through the building have similar physical abilities. In reality, evacuating a primary school, a hospital or an office would require a consideration of people with varying abilities. It would be useful to be able to model explicitly certain classes of persons depending on their ability and evaluate how this would influence the overall evacuation.

Agent behaviour. In an emergency, the physical situation may change dynamically. Part of a building may fill with toxic gas or smoke that causes some people to slow down or even force them to stop. This, in turn, changes the capacity of the routes to an exit and probably causes people to move to another exit or follow a different route. Another consequence could be an increase in the likelihood that guidance instructions will be misinterpreted.

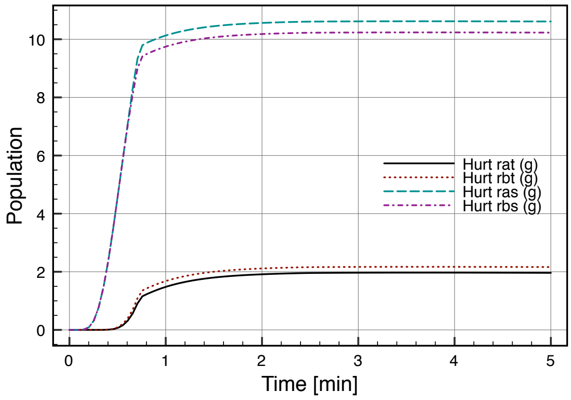


Fig. 4. Hurt evacuees in *lws* from different rooms (ODE)

Group behaviour. Some people decide to follow the direction of a leader or the flow of the majority, moving as a group rather than as individuals. Others may decide to help people moving about in small groups. Also the effects of group decision-making processes, mixed flow directions, group integration, conflict, panic and emergent behaviour have been mentioned as factors of potential risk in emergency egress.

In the following, models of some of the above social behaviours are illustrated indicating how these affect the time to evacuate a building.

6.1. Hurt evacuees

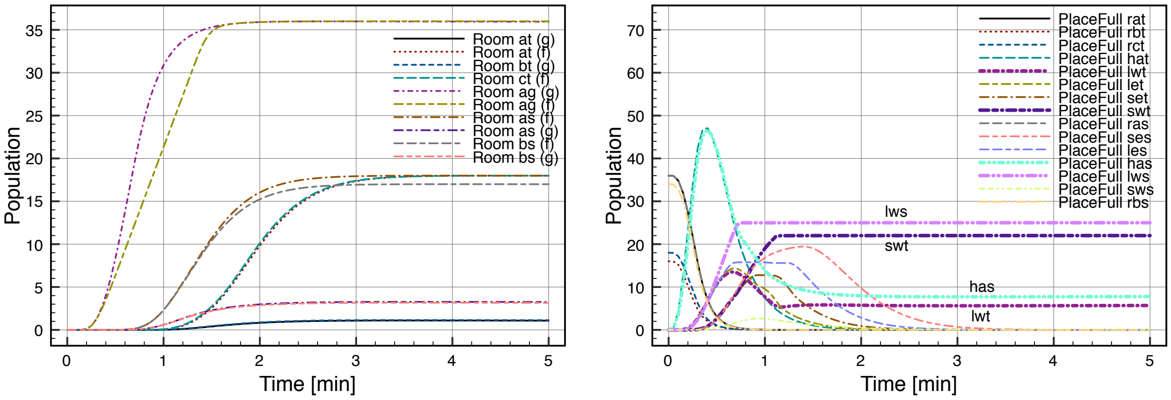
An example of a more realistic scenario would be the event in which during an evacuation a certain part of the building fills with smoke or other toxic gases. The presence of such gases could intoxicate evacuees who, as a result, are unable to proceed to the exits and their presence may hinder other evacuees. This would change the flow capacity of building sections dynamically.

The following excerpt of an extension of the model presented in Section 4 reflects the situation described above assuming that the ***l***anding of the ***W***estern staircase on the ***s***econd floor (*lws*) is containing toxic gases. After entering the landing, it is assumed that evacuees have a relative probability of 20% to pass through without problems and 80% to get hurt thereby preventing them from continuing hence keeping a place in *lws* occupied. This holds for evacuees entering the landing from Hall *a* on the second floor (*has*), as well as for those entering from the Western stairwell on the third floor (*swt*). The excerpt below shows the extension for *swt*, that for *has* being similar.

$$\begin{aligned}
EvcEgInLlwsFswt &= (llwsop, 0.2 * s).EvcEgInLlwsFswtfp + (llwsop, 0.8 * s).EvcEgswtHurt \\
EvcEgswtHurt &= (lswtfp, s).EvcEgHurtSTOP \\
EvcEgInLlwsFswtfp &= (lswtfp, s).EvcEgoplws \\
\\
EvcEgHurtSTOP &= (nop, a).EvcEgHurtSTOP
\end{aligned}$$

Figure 4 shows from which rooms the wounded evacuees in the Western Landing arrive. Since the total capacity of the Western Landing on the second floor is 25 it is clear that after about one minute it has become impossible for any evacuee to pass through the Western Landing to reach the staircase.

Figure 5(a) shows the effect of the presence of toxic gases in the Western Landing on the arrival at the exits of evacuees from the various rooms. For example, on average, only one evacuee from room *rat* on the third floor arrives via exit *g* and the same for evacuees from room *rbt* on the third floor. All the others get blocked. Figure 5(b) shows in which other building areas on the third and second floor people get blocked due to the toxic gases in *lws*. Note that not only is the Western Landing full after about half a minute (curve PlaceFull lws) but also the Western staircase (curve PlaceFull swt) connecting the third and the second floor, the Western landing on the third floor (curve PlaceFull lwt) and, to some extent, the Hall on the second



(a) Arrivals of evacuees in case of toxic gases in *lws* (ODE) (b) Node occupancy in case of toxic gases in *lws* (ODE)

Fig. 5. Node occupancy and arrivals in case of toxic gases in *lws*

floor (curve PlaceFull has). This is due in part to the toxic gases hurting people in the Western Landing, but also due to the specific static routing assumptions. In this scenario it is assumed that evacuees continue to receive instructions about where to go next. In reality, when people see that the way in front of them is blocked they try to go to another exit requesting new directions to the request handler. This is modelled in the next section.

6.2. Hurt evacuees and dynamic selection of alternative exits

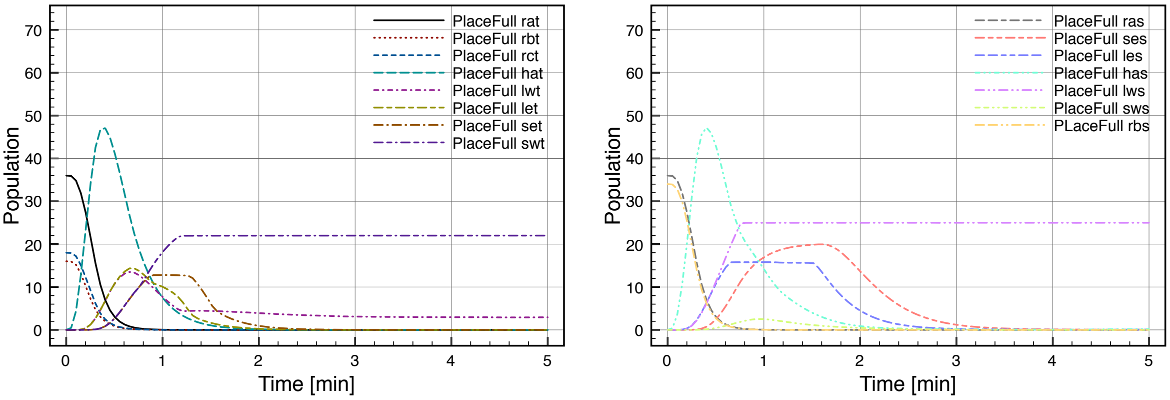
In this section a further extension is considered in which evacuees decide to move to another exit when the entrance to the Western Landing on the second floor is increasingly congested. This is modelled as a third option in the evacuee behaviour process below. This option may occur with a smaller rate (6 instead of 60). This reflects the fact that this option is less likely to be taken when the landing is still relatively free (race condition principle). The excerpt below shows only the extension for *swt*, that for *has* being similar. Note the change from *EvcEg*... to *EvcEf*... in the third element of the choice in the expression below modelling the change of exit.

$$\begin{aligned} \mathit{EvcEgInLlwsFswt} \quad &= \quad (\mathit{llwsop}, 0.2 * s).\mathit{EvcEgInLlwsFswtfp} + (\mathit{llwsop}, 0.8 * s).\mathit{EvcEgswtHurt} \\ &\quad\;\; +(\mathit{change}, 6.0).\mathit{EvcEfopswt} \end{aligned}$$

The evaluation results in Figure 6(a) and Figure 6(b) show that in this case almost all evacuees that have not been hurt and that were first heading to exit g but that could not pass through the Western landing, now follow indications to exit f instead and safely leave the building. We can also observe how this behaviour affects the total evacuation time.

In particular, Figure 6(a) shows the occupancy of the building sections on the third floor. The figure shows that evacuees from all building sections on this floor manage to leave the floor except those in the Western stairwell and some on the Western landing. This may at first seem surprising because in the new model evacuees that do not manage to enter the Western landing on the second floor can now choose another exit and receive related instructions. However, a closer look at the routing table shown in Table 4 (last two lines in the right-hand part of the table) shows, that evacuees in the Western stairwell on the third floor are routed to the Western landing on the second floor *both* in case they are going to exit g *and* in case they are going to exit f. This explains why those evacuees remain blocked in that stairwell in any case. This analysis illustrates how such routing problems may be detected. The origin of the problem is due to the fact that the routing policy is such that once people are in a stairwell going down they should not change direction otherwise a serious bottleneck on the stairs will occur.

Figure 6(b) shows the node occupancy on the second floor. Clearly, the Western landing on the second floor fills up quickly and then remains full. This represents hurt evacuees that are not able to continue. It



(a) Full places third floor

(b) Full places second floor

Fig. 6. Full places on second and third floor model with alternative exits (ODE)

Model	Interval	Analysis time ODE	Analysis time GIL[nr. repl.]
Plain	[0-5]	90,803 ms.	166,887 ms.[10]
Change	[0-5]	966,461 ms.	206,558 ms.[10]
Large	[0-5]	64,087 ms.	1,350,981 ms.[10]
Large	[0-100]	1,475,469 ms.	27,207,695 ms. [1]

Table 5. Overview of analysis times

can also be observed that all the evacuees in Hall *a* on the second floor now manage to leave the floor. In this case choosing another exit results in a different route that does not include the Western landing. All other building sections on the second floor are eventually evacuated.

6.3. Summary of results

We have shown an example of how more realistic agent behaviour and dynamic aspects of the status of building elements can be addressed with PEPA. In this version of the model agents can get hurt and change dynamically the capacity of a particular building element. This dynamic change can be noticed by other agents because of the time they need to wait when attempting to enter the congested building element. If this takes a relatively long time, the probability that another route is chosen by the agent increases. The race-condition construct of the PEPA choice-operator has been exploited to model this form of context-dependency in an indirect way. In Section 8 we explore the language constructs offered by Bio-PEPA to model context dependency in a more direct way. We expect to facilitate also the modelling of other context dependent social behaviour mentioned at the beginning of the current section.

The results in this section show very plausible outcomes for fluid flow analysis which closely correspond to the related simulation results obtained with the same model (their presentation has been omitted due to page limitations). In particular we have shown how the fluid flow analysis can be used to detect congestion on pre-defined routes, due to dynamic events that may occur during emergencies, and where critical routing problems may occur.

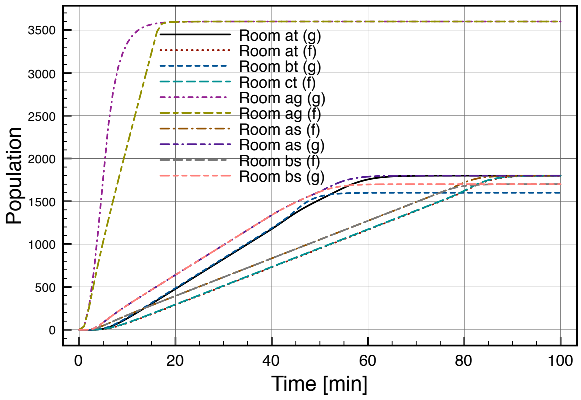


Fig. 7. Arrivals in a large building (ODE)

7. Scaling up

The model presented in Section 4 considers only a limited number of evacuees in order to compare results with the existing case study. To illustrate the scalability of the proposed fluid flow approach for the egress model we present some results in Figure 7. This model with a total of 21,200 evacuees, assumes a much larger though similar building (obtained by multiplication of all room capacities and distances by a factor of 10 and the number of evacuees by a factor of 100). Table 5 provides an overview of the execution time, in milliseconds, for fluid flow analysis (ODE) and stochastic simulation (GIL), of some of the models presented in this article where ‘Plain’ refers to the model introduced in Section 4, ‘Change’ to the model in Section 6.2 and ‘Large’ to the model discussed above. The ODE analysis time increases with the increasing complexity of the model. A similar increase can be observed for the stochastic simulation of the same models for 10 independent replications and the same time interval of 5 minutes. The ODE analysis of the large model shows an interesting phenomenon. For the same time interval the ODE analysis of the large model actually takes *less* time than that of the same model with far fewer evacuees and node capacities. This can be explained by the adaptive step-size used by the underlying numerical algorithm. For a time interval of 100 minutes the ODE analysis takes more time, about 24 minutes, but far less than the approximately 7.5 hours that are required for simulation of *one single* replication.

8. Context dependency

In Section 6 the importance of being able to take typical social aspects of evacuee behaviour into account in the models has been motivated. In a number of cases such behaviour is the result of a decision of the evacuee based on the locally perceived situation or information about the context of the evacuee. Examples are evacuees that follow the main flow of the evacuees that are directly surrounding them, or that are changing their mind on which route to follow when the way in front of them seems too crowded. The PEPA language offers only limited and somewhat indirect possibilities to take such context dependent behaviour into account. A variant of PEPA, called Bio-PEPA, is a process algebra in which transition rates, i.e. the parameters of the exponential distributions that govern the timing aspects of actions, can be defined as functions of the actual population levels. This provides strong support for the modelling of context dependent behaviour, even if it has to be handled with great care when non-linear functions are involved because the good correspondence between simulation results and ODE results of such models may not always be maintained.

In this section we briefly introduce Bio-PEPA and present alternative models of emergency egress, limited to the third floor of the three storey building for the sake of simplicity. The first model is shown to produce similar results to the equivalent PEPA model (restricted to the third floor). The second model, which is an extension of the first, consists of a scenario in which evacuees can be informed in real-time about congestion on their way to the exit, allowing them to decide to move to another exit, getting relevant guidance instructions.

This provides an example of how context dependent evacuee behaviour can be accounted for in Bio-PEPA models.

8.1. Bio-PEPA

Bio-PEPA [CH08, CH09, CG09] is a language originally developed for the modelling and analysis of biochemical systems. The analysis demonstrated in this section indicates that the notation has a much broader application. The main components of a Bio-PEPA system are the *"species" or "sequential" components*, describing the behaviour of individual entities, and the *model component*, describing the interactions between the various sequential components. 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:

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

where S is a *sequential component* and P is a *model component*. In the prefix term (α, κ) $\mathtt{op}$ S, κ is the *stoichiometry coefficient* of sequential component 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. In the egress models considered in the present article this coefficient is always 1, which models the fact that evacuees move individually through the building. The *prefix combinator* "op" represents the role of S in the action, or conversely the impact that the action has on the population of the sequential components. Specifically, $\downarrow$ indicates a *reactant* which will be consumed in the action, $\uparrow$ a *product* which is produced as a result of the action. For example, in $P = (\alpha, 2){\downarrow}P$ this means that when action α occurs, the population of P is decreased by 2 units. Note also the difference from PEPA expressions: P in the prefix combinator of Bio-PEPA must be the same as the name of the sequential component that is being defined, P in this case. The prefix operator only defines how, and due to which action, the population P changes. It should not be confused with the prefix combinator of PEPA, e.g. $P = (\alpha, r).P'$ where r denotes the rate of action α and P' the continuation of the behaviour of P after α has occurred. We provide a more detailed example to illustrate these differences below.

The remaining prefix combinators do play a role in the context of biochemistry, but are not used in the egress models presented in this section. They are $\oplus$ 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 $P \underset{L}{\bowtie} Q$ denotes synchronisation between components P and Q, the set L determines those activities on which the operands are forced to synchronise, with $\underset{*}{\bowtie}$ denoting a synchronisation on all common action types. If the set L is empty $P \underset{L}{\bowtie} Q$ is written as $P \parallel Q$. In the model component $S(x)$, the parameter $x \in I\!\!R$ represents the initial amount of the species.

As a small example we present a Bio-PEPA version of the PEPA model presented in Section 3. There are four sequential processes P, P', Q and Q' of which P and Q can perform action α on which they synchronise. In the case of PEPA, process P (Q respectively) would then become P' (Q' respectively) after α has been performed. In Bio-PEPA, instead, this is modelled by the fact that the number of instances of sequential processes P and Q are decreased by 1, and the number of instances of sequential processes P' and Q' are increased by 1. In a similar way, when process P' performs action β, the number of P' processes decreases by 1 and the number of P processes increases by 1 and similarly for Q' and action γ. This behaviour is expressed as follows:

$$\begin{array}{rcl} P & = & (\alpha, 1){\downarrow}P + (\beta, 1){\uparrow}P \\ P' & = & (\beta, 1){\downarrow}P' + (\alpha, 1){\uparrow}P' \\ Q & = & (\alpha, 1){\downarrow}Q + (\gamma, 1){\uparrow}Q \\ Q' & = & (\gamma, 1){\downarrow}Q' + (\alpha, 1){\uparrow}Q' \end{array}$$

To make sure that the increase and decrease of the number of entities happen simultaneously the processes need to synchronise in a proper way on shared actions. This is taken care of in the model component which is defined as:

$$System \;=\; (P(N_P) \underset{\{\alpha,\beta\}}{\bowtie} P'(0)) \underset{\{\alpha\}}{\bowtie} (Q(N_Q) \underset{\{\alpha,\gamma\}}{\bowtie} Q'(0))$$

What remains is to define the rates at which the actions occur. In Bio-PEPA for each action α 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 biochemical models 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. This rate function is predefined in Bio-PEPA and denoted by $\mathsf{fMA}(r)$ where r is the rate constant. Note that in Bio-PEPA the same action type can occur only with one particular associated rate, whereas in PEPA this rate may be different at each occurrence of the action in the model. This means that in Bio-PEPA we cannot define a rate function that captures the implicit rate function of PEPA in its full generality. Therefore, in the Bio-PEPA models that will be presented in this section we will make use of a combination of the rate function fMA and synchronisation to obtain a similar behaviour to that shown by the PEPA egress model.

At each system state 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 basic Bio-PEPA language is extended with the notion of *locations* also called *compartments.* A Bio-PEPA *system* with locations consists of a set of sequential components, a model component, and a context (locations, kinetics rates, parameters, etc.). The prefix term (α, κ) $\texttt{op}$ $S@l$ 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 $\langle \mathcal{L}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Comp, P\rangle$, 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.

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 [CH09, DGG$^+$09, CDG$^+$09, Bio]. These tools capture mappings from Bio-PEPA to differential equations for fluid flow analysis, stochastic simulation models [Gil77], continuous time Markov chains (CTMC) with levels [CH08] and PRISM models [KNP09]. The tool suite is available as a plug-in[4] for the Eclipse[5] open cross-platform development environment, see Figure 8.

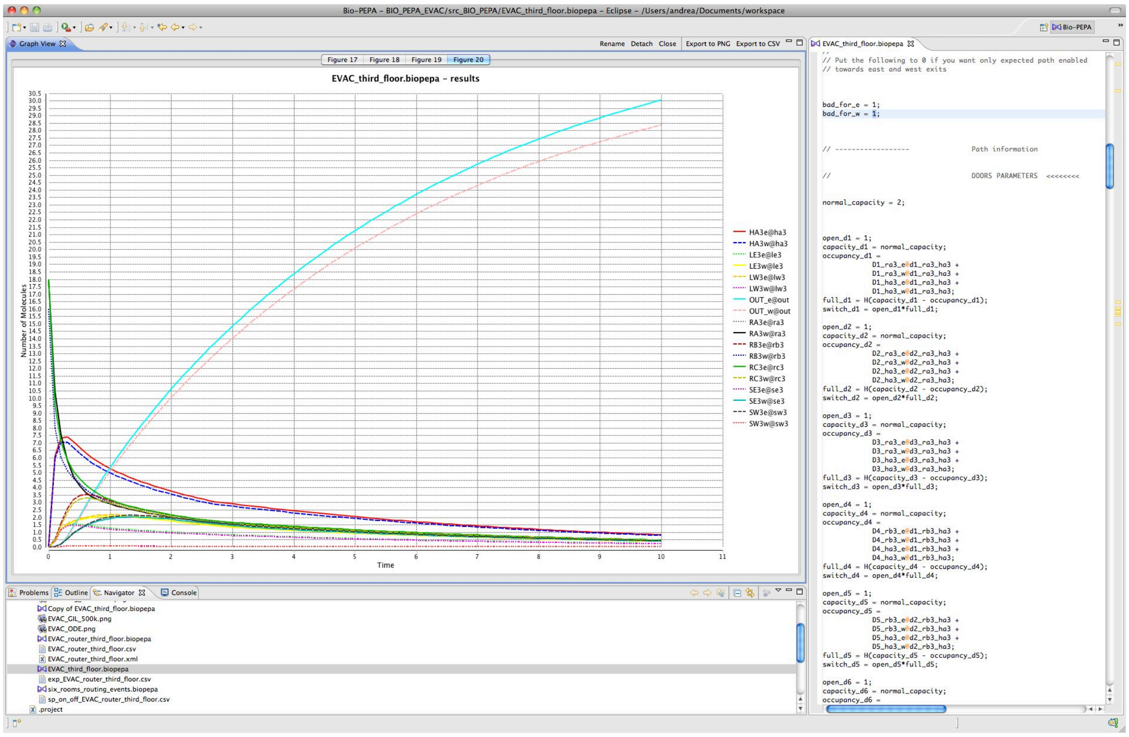
The plug-in, currently still under development, comprises an editor to develop the model specification and a tool bar reporting information about the model, errors and warning, the file systems and the like. It embeds both stochastic simulation engines based on the Gillespie algorithm and several ODE 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. Furthermore, the model specification can be exported in SBML and simulations can be represented in XML for Traviando, a simulation trace analysis tool [KT09].

8.2. Bio-PEPA model of emergency egress

The Bio-PEPA variant of the emergency egress model is illustrated by addressing only the third floor of the building. As before, there are three rooms (*rat*, *rbt*, *rct*), one hall (*hat*) and two staircases and landings, on the east side (*let*, *set*) and west side (*lwt*, *swt*) of the building, respectively, all connected by a number of doors and each with the same capacity as for the previous PEPA model. Staircase *set* directly leads to the eastern exit and staircase *swt* leads to the western exit. When the alarm goes off there are 36 agents in *rat*, half of which will be directed towards east and half to west. There are 16 agents in *rbt* heading west and 18 heading east. The same assumptions are made about average speed of movement, communication time and traversal time for each room according to room dimensions as in the the previous model.

[4] See `http://homepages.inf.ed.ac.uk/jeh/Bio-PEPA/biopepa.html`

[5] See `http://www.eclipse.org`



```
// Put the following to 0 if you want only expected path enabled
// towards east and west exits



bad_for_e = 1;
bad_for_w = 1;



// ------------------              Path information


//                                 DOORS PARAMETERS  <<<<<<<<


normal_capacity = 2;



open_d1 = 1;
capacity_d1 = normal_capacity;
occupancy_d1 =
            D1_ra3_e@d1_ra3_ha3 +
            D1_ra3_w@d1_ra3_ha3 +
            D1_ha3_e@d1_ra3_ha3 +
            D1_ha3_w@d1_ra3_ha3;
full_d1 = H(capacity_d1 - occupancy_d1);
switch_d1 = open_d1*full_d1;

open_d2 = 1;
capacity_d2 = normal_capacity;
occupancy_d2 =
            D2_ra3_e@d2_ra3_ha3 +
            D2_ra3_w@d2_ra3_ha3 +
            D2_ha3_e@d2_ra3_ha3 +
            D2_ha3_w@d2_ra3_ha3;
full_d2 = H(capacity_d2 - occupancy_d2);
switch_d2 = open_d2*full_d2;

open_d3 = 1;
capacity_d3 = normal_capacity;
occupancy_d3 =
            D3_ra3_e@d3_ra3_ha3 +
            D3_ra3_w@d3_ra3_ha3 +
            D3_ha3_e@d3_ra3_ha3 +
            D3_ha3_w@d3_ra3_ha3;
full_d3 = H(capacity_d3 - occupancy_d3);
switch_d3 = open_d3*full_d3;

open_d4 = 1;
capacity_d4 = normal_capacity;
occupancy_d4 =
            D4_rb3_e@d1_rb3_ha3 +
            D4_rb3_w@d1_rb3_ha3 +
            D4_ha3_e@d1_rb3_ha3 +
            D4_ha3_w@d1_rb3_ha3;
full_d4 = H(capacity_d4 - occupancy_d4);
switch_d4 = open_d4*full_d4;

open_d5 = 1;
capacity_d5 = normal_capacity;
occupancy_d5 =
            D5_rb3_e@d2_rb3_ha3 +
            D5_rb3_w@d2_rb3_ha3 +
            D5_ha3_e@d2_rb3_ha3 +
            D5_ha3_w@d2_rb3_ha3;
full_d5 = H(capacity_d5 - occupancy_d5);
switch_d5 = open_d5*full_d5;

open_d6 = 1;
capacity_d6 = normal_capacity;
occupancy_d6 =
```

Fig. 8. The Bio-PEPA Eclipse plug-in.

In Bio-PEPA the spatial structure of the building can be defined explicitly as a set of locations (compartments) without any nesting.

$$\begin{array}{lll}
\mathsf{location}\ rat: & \mathsf{size} = \mathsf{normal\_room}, & \mathsf{type} = \mathsf{compartment}; \\
\ldots & & \\
\mathsf{location}\ d1\_rat\_hat: & \mathsf{size} = \mathsf{normal\_door}, & \mathsf{type} = \mathsf{compartment}; \\
\ldots & &
\end{array}$$

These represent rooms, e.g. rat, and doors, e.g. $d1\_rat\_hat$, i.e. door 1 between room rat and hall hat. 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.

The behaviour of an agent in every location is modelled by means of a distinct sequential process. We also distinguish agents heading east from those heading west. For example, RA_e is the process modelling an agent in room rat, heading east. Such an agent can engage in several different kinds of interaction, amongst which is the action $(rat\_e\_in\_d1\_rat\_hat, 1)$ that models the agent heading east via door 1 moving from room rat to hall hat.

$$\begin{array}{lll}
RA\_e & = & (rat\_e\_in\_\_d1\_rat\_hat, 1)\downarrow RA\_e@rat + \ldots \\
 & & (hat\_e\_out\_\_d1\_rat\_hat, 1)\uparrow RA\_e@rat + \ldots \\
RA\_w & = & \ldots \\
HA\_e & = & \ldots \\
\ldots & &
\end{array}$$

The definition of the sequential process $D1\_rat\_e$ represents the class of agents who have entered door 1 from room rat. They can perform the action $(rat\_e\_in\_d1\_rat\_hat,\ 1)$, too. The cooperation on action $(rat\_e\_in\_d1\_rat\_hat,\ 1)$ between an agent in room rat and agent $D1\_rat\_e$ reduces the number of agents in the room by one, and that the number of agents at the door increases by one. This precisely reflects the movement of an agent from the room to the door.

$$\begin{array}{lcl} D1\_rat\_e & = & (rat\_e\_in\_\_d1\_rat\_hat, 1)\uparrow D1\_rat\_e@d1\_rat\_hat + \ldots \\ & & (hat\_e\_out\_\_d1\_rat\_hat, 1)\downarrow D1\_rat\_e@d1\_rat\_hat + \ldots \\ D1\_rat\_w & = & \ldots \\ \ldots & & \end{array}$$

The effects of these interactions are clearly located in a specific compartment of the model by the @ construct, so that $\downarrow RA\_e@rat$ specifies that an RA_e agent left room rat. It is important to remark that it is not necessary that an agent is already present at the door: an agent may be created as a result of the interaction. The synchronised increasing and decreasing of numbers of agents at locations model movement from one compartment to another. In the following, when we say that two processes synchronise we will also consider the case where one of the two "interacting" processes is created as an effect of the interaction.

The functional rates of agent actions are defined based on information on rates and contextual information. This is the place where context dependency of agent behaviour is modelled. Rates and contextual information are defined as variables and parameters. We first present the relevant variables and parameters and then show how they are used in the definition of the functional rates.

As before, time is measured in minutes. For instance, we have determined the rate $to\_rat\_d1 = 6$, accounting for 1/6 of a minute, i.e. ten seconds, to model the sum of the time for getting directions (around 3 seconds) and reaching door 1 in room rat (about 7 seconds to walk 7 meters at about $1m/s$). Analogously, rate $from\_d1$ accounts for the door exiting rate and has value 20. The average time leading to this rate is the sum of the time to cross the door (about 2 seconds) and the time to get a place in the new room (about 1 second). These rates are compatible with those in the PEPA model, except for the fact that some actions that were separate in the PEPA model, such as getting directions and moving towards the door, have now been amalgamated into a single action with an adjusted total rate.

Other variables and parameters have been defined that represent useful contextual information on the state of the model, also in terms of the population present in the various building elements, 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 $open\_d1 = 1$. Furthermore, variables are used to summarise relevant information about the dynamics of the system for the purpose of analysis and for defining context dependent behaviour. The variable *occupancy* represents the occupancy of door 1 as the sum of agents that are currently occupying the door:

$$\begin{array}{lcl} occupancy\_d1 & = & D1\_rat\_e@d1\_rat\_hat + D1\_rat\_w@d1\_rat\_hat + \\ & & D1\_hat\_e@d1\_rat\_hat + D1\_hat\_w@d1\_rat\_hat; \end{array}$$

The variable $not\_full\_d1 = \mathtt{H}(capacity\_d1 - occupancy\_d1)$ is a boolean test indicating whether the door is fully occupied by agents in transit, where $capacity\_d1 = normal\_capacity$ stands for the maximum number of agents that can occupy a door, 2 in our model. The function `H(_)` is a non-negative-test function with `H(x)=0` if `x=0` and `1` otherwise, and 0/1 arithmetic implements the boolean algebra; $switch\_d1 = open\_d1 * not\_full\_d1$ is a boolean flag accounting for a door being traversable, i.e. open and not full; $safeD1\_rat\_e = \mathtt{H}(D1\_rat\_e@d1\_rat\_hat)$ is a boolean value accounting for the existence of at least one $D1\_rat\_e$ agent ready to interact; and $allowance\_hat = \mathtt{H}(211 - RA\_e@rat - RA\_w@rat)$ represents the number of available free places in room hat.

The functional rates of the agent actions are defined by the Bio-PEPA construct kineticLawOf. For example, the action $rat\_e\_in\_d1\_rat\_hat$ is ruled by the mass action law fMA mentioned in Section 8.1:

$$\mathsf{kineticLawOf}\ rat\_e\_in\_d1\_rat\_hat \quad : \quad \mathsf{fMA}(to\_rat\_d1 \ * \ switch\_d1 \ * \ rat\_e\_in\_safe)$$

where to_rat_d1 is the rate, $switch\_d1$ the parameter discussed above, and $rat\_e\_in\_safe$ is a boolean flag guaranteeing that at least one agent RA_e is ready to synchronise on the action.

Other actions are ruled by the same kinetic law, such as $hat\_e\_out\_\_d1\_rat\_hat$ that regards the exiting from a door and it is constrained by the current occupancy of the arrival room, i.e. $allowance\_hat$.

$$\mathsf{kineticLawOf}\ \mathit{hat\_e\_out\_\_d1\_rat\_hat} \quad : \quad \mathsf{fMA}(\mathit{from\_d1} * \mathit{open\_d1} * \mathit{safeD1\_rat\_e} * \mathit{allowance\_hat})$$

Action $\mathit{hat\_e\_in\_\_d1\_rat\_hat}$ models agents entering a door from hall hat that leads to room rat. The two actions together, $\mathit{hat\_e\_out\_\_d1\_rat\_hat}$ and $\mathit{hat\_e\_in\_\_d1\_rat\_hat}$ model bi-directional movement of agents. By setting the involved parameters agents can be modelled that follow particular routes.

$$\mathsf{kineticLawOf}\ \mathit{hat\_e\_in\_\_d1\_rat\_hat} \quad : \quad \mathsf{fMA}(\mathit{to\_hat\_d1} * \mathit{back} * \mathit{switch\_d1} * \mathit{hat\_e\_in\_safe} * \mathit{bad\_for\_e})$$

where $\mathit{back} = 1$ indicates backward direction with respect to approaching an exit and $\mathit{hat\_e\_in\_safe} = \mathtt{H}(\mathit{RA\_e}@\mathit{rat})$ is an extra check to make sure that the action is only enabled when there are agents heading east present in hat. The parameter $\mathit{bad\_for\_e}$ is used to tune the routing. For example, when both $\mathit{bad\_for\_e} = 0$ and $\mathit{bad\_for\_w} = 0$ are selected the same routing policy is modelled as in the PEPA egress model.

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. This choice is justified pragmatically by the fact that the capacity control on doors, i.e. at most two agents can traverse a door together, makes the flux of evacuees through doors essentially linear in time and independent of the number of agents in a room. This leads to an emergent overall egress behaviour that is coherent with results in the 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 [KG10] where Langevin equations have been used.

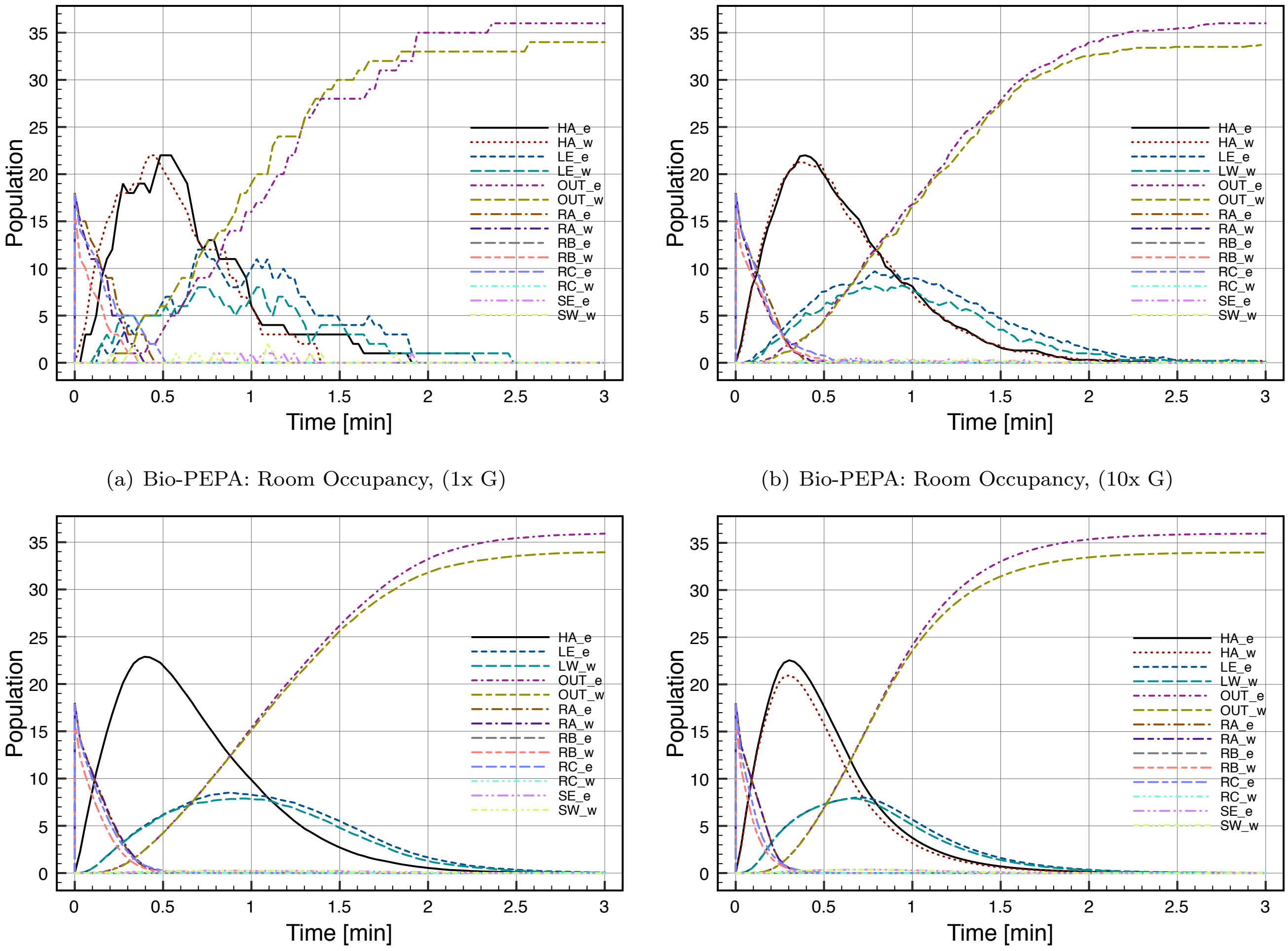
Finally, after the definition of each type of agent, their composition into the overall system is formalised. According to this model there are two sets of agents in the building: those representing evacuees in rooms and those representing evacuees that are traversing a door. The agents within these sets do not synchronise with each other on any action $\|$. Agents in the two different sets synchronise on every action they share $\underset{*}{\bowtie}$: whenever an agent moves from or into a room, it must do so together with the process that models the entrance or exit of an evacuee that is crossing the relevant door. This is guaranteed by a suitable matching of action names, as discussed for $\mathit{rat\_e\_in\_d1\_rat\_hat}$ above. The total number of evacuees that can be present at a door is governed by the variable switch which takes value 0 when the door is “full”, suspending further synchronisations related to agents entering the door until there is again space available. 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 $\mathit{RA\_e}$ and 18 $\mathit{RA\_w}$ in the building in room rat.

$$\begin{array}{lll} \mathit{third\_floor} & ::= & (\mathit{RA\_e}@\mathit{rat}(18) \parallel \mathit{RA\_w}@\mathit{rat}(18) \parallel \mathit{HA\_e}@\mathit{hat}(0) \parallel \ldots) \\ & & \underset{*}{\bowtie} \\ & & (\mathit{D1\_rat\_e}@\mathit{d1\_rat\_hat}(0) \parallel \mathit{D1\_rat\_w}@\mathit{d1\_rat\_hat}(0) \parallel \ldots) \end{array}$$

8.3. Principal differences between PEPA and Bio-PEPA egress models

There are a number of differences between the PEPA and Bio-PEPA egress models. They are mainly due to the differences in the particular set of available operators and concepts in both languages. The first important difference is the availability of the notion of location in Bio-PEPA, which is not available in PEPA. Locations (or compartments) can be addressed explicitly in the Bio-PEPA language whereas in PEPA they needed to be implicitly encoded into process names. This improves readability of the specification and is useful for the presentation of results.

In Bio-PEPA the language to define the behaviour of sequential components is closer to the definition of ordinary differential equations, whereas PEPA follows a more traditional process algebraic tradition. In PEPA there is a single pre-defined rate function for synchronising actions where the (constant) rate of an action may be different in each of the synchronising processes. Bio-PEPA offers a more flexible definition of the rates that may depend dynamically on the population of other processes not involved directly in the synchronisation. This provides a very powerful direct approach to define context dependent behaviour, but needs to be handled with care in case non-linear behaviour is introduced in which case the good correspondence between fluid flow and simulation can be lost easily. In the Bio-PEPA egress model the original PEPA synchronisation rate has been substituted by a combination of mass action law, which is known to be linear, and the modelling



(a) Bio-PEPA: Room Occupancy, (1x G)

(b) Bio-PEPA: Room Occupancy, (10x G)

(c) Bio-PEPA: Room Occupancy, (500x G)

(d) Bio-PEPA: Room Occupancy (ODE)

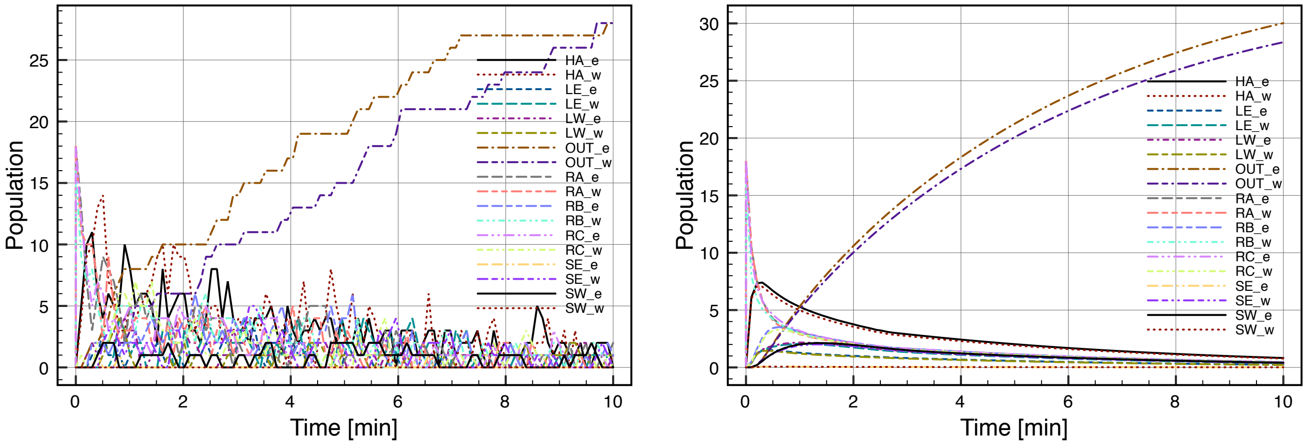
Fig. 9. Bio-PEPA Emergency Egress: Results.

of doors as particular locations. In the next section it is shown that this combination gives results that correspond closely to those obtained for the PEPA models.

A final difference between PEPA and Bio-PEPA egress models is the way routing is modelled. In PEPA the routing is encoded into the various request handler processes, whereas in the Bio-PEPA model this is modelled using the parameters *bad_for_e* and *bad_for_w* in the rate functions of actions modelling agents following the route to the eastern or western exit, respectively. By putting these parameters to 0 in actions that would lead them into the wrong direction, they take that wrong direction effectively with rate 0, thus never. These parameters are placed in such a way that they reflect the same routing as in the PEPA egress model for what concerns the third floor. Other routing patterns can be analysed by changing the values of these parameters.

8.4. Analysis

All analysis results are reported in figures over a suitable time interval using various analysis techniques, either continuous (ODE) by an adaptive step-size 5th order Dormand-Prince ODE solver, or discrete (1 x



(a) Bio-PEPA: Arrivals (10x G)

(b) Bio-PEPA: Arrivals (ODE)

Fig. 10. Bio-PEPA Emergency Egress: No Routing.

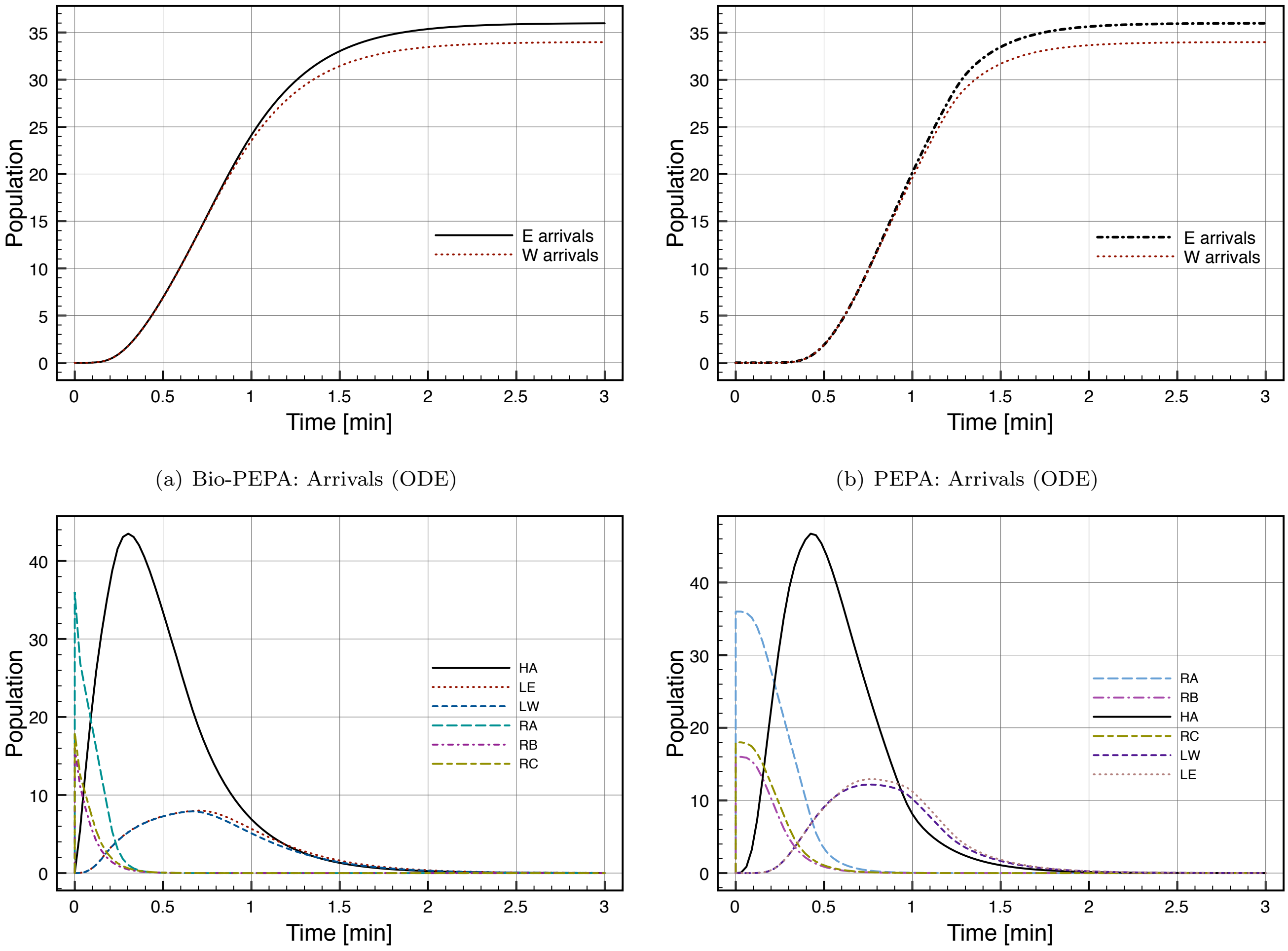
G) by Gillespie's Stochastic Simulation Algorithm, or averaged (N x G), where N > 1 indicates the number of independent simulation traces, of the numbers of relevant agents.

Figure 9 reports the occupancy of each room during evacuation. In Figure 9(a) a single stochastic simulation is depicted. It can be observed how the rooms where agents initially are (rat, rbt, rct, with populations RA_e and RA_w, RB_e and RB_w, RC_e and RC_w, respectively) become quickly empty within the first half a minute. Hall hat then becomes crowded (HA_e and HA_w), while other rooms, like the staircases (set, swt) have quite a low occupancy (SE_e and SW_w) during the entire time interval. This can be explained by the fact that the staircases directly and quickly lead to an exit. The number of agents that have left the building (OUT_e and OUT_w) grows linearly until the end of the evacuation, within the first two and a half minutes.

Several of these details, i.e. those relative to small numbers like staircase occupancy, tend to be blurred in Figure 9(b), where the average of 10 stochastic simulations is reported. Averaging over a larger number of stochastic simulations, 500 in Figure 9(c), tends rather precisely towards the results of the approximation obtained by means of fluid flow analysis and reported in Figure 9(d). This correspondence is expected for systems consisting of a large number of agents, but happens, as in this case, also for systems with relatively few agents, proving that fluid flow analysis can provide informative results for a wide range of cases.

Figure 10 reports the case in which no evacuation plan, or routing, is provided to agents, over a $[0, 10]$ minute time interval. Technically, this can be obtained easily by means of the switching variables associated with each door. As explained, preference degrees or boolean conditions can be associated with the traversal of each door in each room by each agent, according to its final destination. The lack of any such information corresponds to agents having no a priori fixed directions. Clearly, when agents move about randomly, evacuation times increase as expected. By observing the ODE approximation (Figure 10(b)), one could understand that all rooms become almost empty around minute 8. Observing the stochastic simulation, instead, it is easy to realise that several agents are still wandering 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 [HB10] for results on this topic.

Figure 11 reports a sanity check for the Bio-PEPA model. The results are compared with those obtained with the PEPA model in Section 4 considering the same static routing plan. It is worth recalling that the latter has been shown to be largely coherent with the results obtained from the original work in [KFN98], up to the expected differences, since the results therein are given in terms of optimal fluxes. Figures 11(a)-(b) report the agents' evacuation over the course of time of the PEPA model and the Bio-PEPA model respectively. They correspond well, although the Bio-PEPA model is slightly faster initially. Note that the two models are anyway based on different languages with different modelling abstractions, in particular different synchronisation-rate policies.



(a) Bio-PEPA: Arrivals (ODE)

(b) PEPA: Arrivals (ODE)

(c) Bio-PEPA: Room Occupancy (ODE)

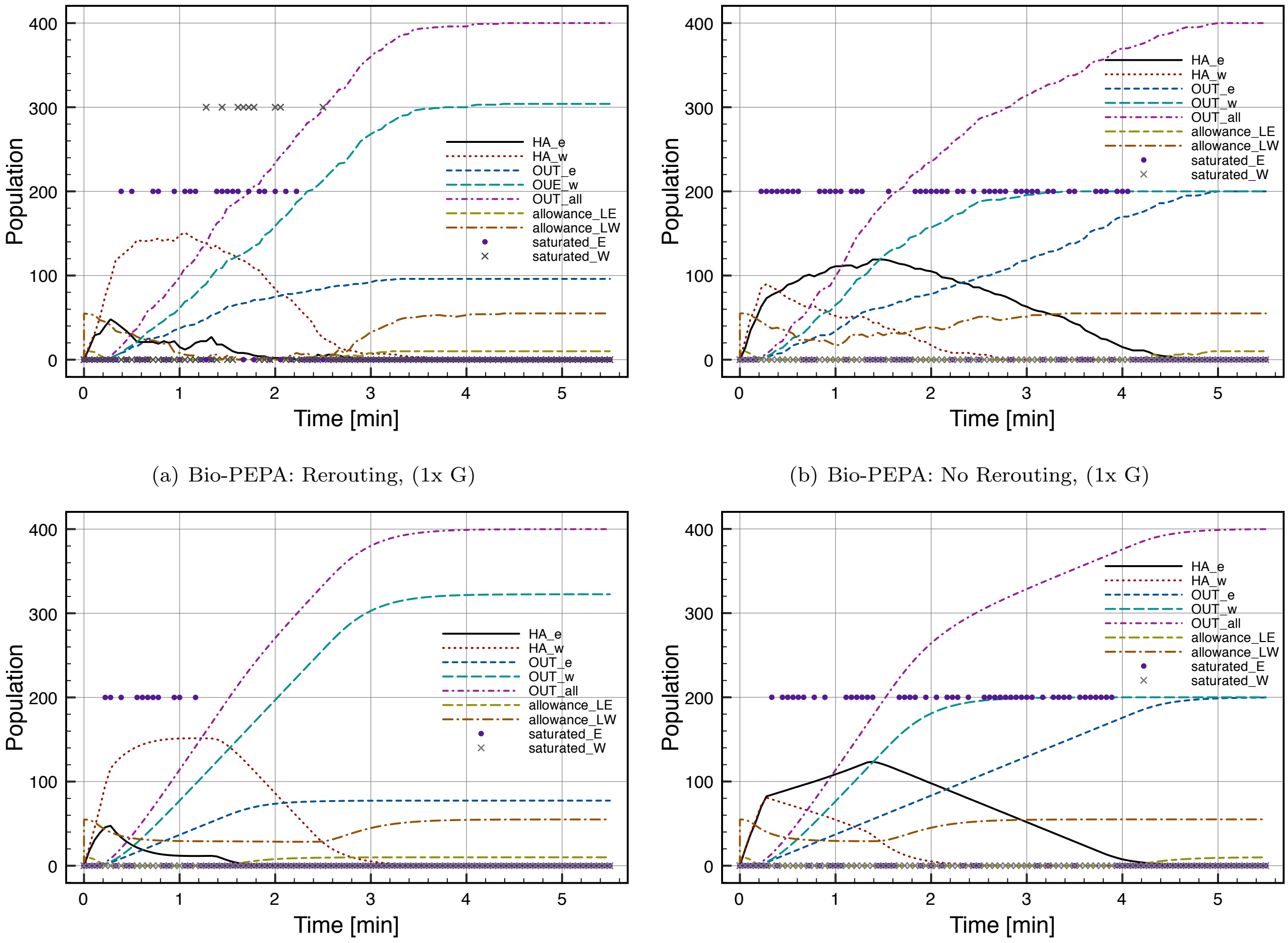
(d) PEPA: Room Occupancy (ODE)

Fig. 11. Bio-PEPA and PEPA Emergency Egress: Result Comparison (ODE).

Analogously, Figures 11(c)-(d) report room occupancy for both models and the Bio-PEPA model results show slightly faster room dynamics, but within an overall good correspondence.

To illustrate the context dependency aspects of the Bio-PEPA model, we consider a scenario in which evacuees can be informed, for example by means of a handheld device or intelligent badge, about congestion on their planned way out of the building and are given the opportunity to change their final destination exit. To model this scenario, 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. One way to do this would be by measuring the instantaneous congestion level of the *let* and *lwt* 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 *hat* 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). The chosen threshold in the model is a difference in allowance between *lwt* and *let*. This change of destination is modelled by adding two actions to the *HA_e* and *HA_w* agents, which now must synchronise on these swapping actions:

$$HA\_e@hat(n) \underset{\{swap\_to\_hat\_W, swap\_to\_hat\_E\}}{\bowtie} HA\_w@hat(n)$$



(a) Bio-PEPA: Rerouting, (1x G)

(b) Bio-PEPA: No Rerouting, (1x G)

(c) Bio-PEPA: Rerouting, (ODE)

(d) Bio-PEPA: No Rerouting (ODE)

Fig. 12. Bio-PEPA Emergency Egress: Rerouting.

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

$$\begin{aligned}&\mathit{swap\_on} * \mathtt{H}(\mathit{allowance\_let} - \mathit{allowance\_lwt} - \mathit{swap\_threshold\_WE\_hat}) *\\ &\quad \mathtt{H}(\mathit{enough\_HA\_w\_to\_HA\_e}) * \mathit{HA\_w@hat} * \mathit{swap\_speed\_WE\_hat};\end{aligned}$$

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

Figure 12 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 12(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.

In fact, this scenario reflects the situation in which the evacuees have the possibility to "look ahead" due to being better informed in real-time. This clearly provides an advantage in terms of faster evacuation.

Balanced occupancy can also be appreciated in Figure 12(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 12(b). Analogous results can be appreciated by means of fluid flow analysis and are reported in Figures 12(c)-(d). This 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 though they may do so occasionally in single simulation runs, while the more frequent ones relative to the east path are shown. The reason for this is that the ODE analysis provides an analysis of *average* behaviour. This experiment illustrates the complementarity of the two analysis techniques.

8.5. Summary of results

The results show three important things. First the Bio-PEPA egress model produces fluid flow and stochastic simulation (average over 500 independent runs) results that correspond well. Second the Bio-PEPA egress model results also correspond well with the PEPA model for the same scenario and for the third floor of the building. This is true both for the evacuation time and the node occupancy. The third observation concerns context dependency. The results show that agents can be modelled that use real-time information about congestion on their way to the selected exit to choose the best route to leave the building. Such information could be provided to them for example by means of their handheld devices. It has been shown that this indeed decreases evacuation time as expected due to a more balanced distribution of agents over the two routes to the exits.

In addition to all this, we have shown the effect on evacuation time and node occupancy when no routing information is provided to the agents. In this case they just walk randomly through the building and exit only when they happen to pass an exit. Although fluid flow results still correspond well to the simulation results in this case, the latter give a better impression of the variability of the node-occupancy and evacuation times, which get blurred in the average numbers provided by fluid flow analysis.

9. Conclusions and outlook

The application of formal modelling techniques, such as stochastic process algebras, to the scalable analysis of collective dynamics and systems containing a vast number of human and computational mobile agents appears to be an open research topic. This article contributes in several ways to address issues in this area. It explores the application of continuous and discrete quantitative analysis techniques to a realistic case study in the context of smart signage systems. The specific case study addressed has been drawn, adapted and expanded from an existing case study on emergency egress for a three storey building [KFN98]. The original analysis concerned optimal evacuation plans and node occupancy carried out by means of flow optimisation techniques. In this article we have developed alternative models of this case study using two different stochastic process algebras, PEPA and Bio-PEPA, and two different analysis techniques, fluid flow analysis and stochastic simulation. Fluid flow analysis relies on considering cooperating classes of large numbers of independent components describing the temporal evolution of such classes by approximating the average number of components that are in the same state at a given time. Such approximations are obtained by solving sets of ordinary differential equations that are automatically derived from the process algebraic models.

Both PEPA and Bio-PEPA provide an abstract, formally grounded, language for the description of behaviour of individuals and their interaction within the overall system. This combined approach is expressive, as the model can be easily adapted to describe different working hypotheses and scenarios, and efficient, since the continuous interpretation of fluid flow allows efficient numerical solvers to be used. We have shown that fluid flow analysis applied to the egress models produces results that correspond well to the results presented in [KFN98] for what concerns evacuation times and node occupancy of the case study on the three storey building, even for a relatively small number of evacuees. Furthermore, the fluid flow approach has been shown to be scalable with respect to the number of evacuees and the number of building elements.

A problem of the method for finding optimal evacuation plans in [KFN98] is that only very limited evacuee behaviour and dynamic aspects of emergency egress can be taken into account. In the proposed process algebraic approach more complex individual agent behaviour can be modelled within and dependent on a dynamic context. Examples of this have been provided both in PEPA and Bio-PEPA. In particular we have analysed a more realistic egress scenario in which evacuees get hurt and are occupying space in a building element that is filled with toxic gasses, blocking the way progressively to other evacuees. Blocked evacuees can change their mind dynamically about their preferred exit and decide to follow a different route. Fluid flow shows the consequences of this scenario for evacuation time, node occupancy and congestion given the routes, the location of the exits and the location of the evacuees in the building. Comparison with simulation based results for the same scenario provide a good correspondence. Furthermore, we have developed a Bio-PEPA model in which evacuees are informed about congestion levels on their way to the exit that they can use to change their route dynamically. It has been shown that the models in which evacuees were provided with such information indeed result in more balanced and thus faster evacuation, confirming expectations. This is another example of how context dependent behaviour of evacuees can be taken into account in process algebraic models. In Bio-PEPA context dependency can be addressed in a more direct way than in PEPA because of the concept of functional rates. In the definition of functional rates population levels of other processes, other than those directly involved in an action, can be taken into account. This, together with the explicit notion of locations facilitates the development of this kind of egress models and their analysis.

Both PEPA and Bio-PEPA are supported by a set of tools for fluid flow analysis and stochastic simulation. Stochastic simulation is well suited to address non-determinacy and emergent behaviour that can be triggered by a low number of agents, but it is generally computationally expensive. We have shown that fluid flow analysis can provide a good approximation of the average transient system behaviour, especially when agents are present in sufficiently high numbers as to make the frequency of actions high and the relative system change from single events small. Continuous techniques can be more computationally viable than discrete ones, making them more suitable for use in the system design phase, when many design options and parameter tuning are issues. Simulation techniques, on the other hand, may complement the analysis of average behaviour. Single simulation runs may for example provide insight in exceptional behaviour and variability of behaviour rather than average tendencies. A comparison of the computational costs of both techniques for the developed egress models underlines the efficiency of fluid flow. It suggests that fluid flow appears to be a valuable technique, especially whenever the overall behaviour of the system is not dramatically affected by stochastic noise. The experiments have been performed in collaboration with the team currently developing the Bio-PEPA toolkit, as a step towards the improvement of modelling in this domain.

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 particular aspects of crowd dynamics. The uptake of the fluid flow approach could furthermore be facilitated by the development of specification methods that are close to the best practises of domain experts. A preliminary example in this direction was shown in Sect. 4.6 where the PEPA specification is automatically generated from typical data that domain experts work with such as the architecture of a building, routing information, initial situation of the evacuees etc. This guarantees that domain experts work on the level of abstraction that they are familiar with while a process algebraic specification suitable for fluid flow analysis is correctly generated in an automatic way. This idea could be elaborated further by designing a graphical specification language for this domain.

The interesting fluid flow results obtained for the specific case study on emergency egress, both in terms of scalability of the analysis and in terms of the dynamic aspects of behaviour that can be modelled, are promising also for the analysis of other kinds of applications that share some similar aspects such as in the field of pervasive systems used by a large number of people, applications in social computing where information may influence people's behaviour and the analysis of particular randomised information distribution algorithms, also known as 'gossip protocols', in the area of telecommunication.

Furthermore, we plan to study the possibility of embedding other mathematical theories in a process algebraic context, which could be useful to better capture emergent behaviour. A starting point is represented by Langevin stochastic differential equations, used for crowd dynamics in [KG10] and compared against stochastic simulations of PEPA models in [Sle10]. Also, we intend to integrate existing mature verification techniques and develop new ones for this specific domain. A promising candidate is quantitative and statistical 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 [KNP09].

Acknowledgments

The authors would like to thank Stephen Gilmore, Maria Luisa Guerriero, Allan Clark and Adam Duguid (The University of Edinburgh) and Mirco Tribastone (LMU Munich) for their support with the PEPA and Bio-PEPA tools and discussion on precursor models of the one here presented. This research has been partially funded by the CNR project RSTL-XXL, by the Italian PRIN MIUR project PaCO, and by the EU-IP project ASCENS. Jane Hillston has been supported by the EPSRC ARF EP/c543696/01.

References

[BHLM10] A. Bracciali, J. Hillston, D. Latella, and M. Massink. Reconciling population and agent models for crowd dynamics. In *Preliminary Proceedings of the Third International Workshop on Logics, Agents, and Mobility (LAM2010)*, pages 3–17. Federated Logic Conferences (FLoC), 2010.

[Bio] Bio-PEPA Home Page. `http://www.biopepa.org/`.

[CDG+09] 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)*, pages 309–310. IEEE Computer Society, 2009.

[CG09] F. Ciocchetta and M. L. Guerriero. Modelling biological compartments in Bio-PEPA. *ENTCS*, 227:77–95, 2009.

[CH08] F. Ciocchetta and J. Hillston. Bio-PEPA: An extension of the process algebra PEPA for biochemical networks. *ENTCS*, 194(3):103–117, 2008.

[CH09] F. Ciocchetta and J. Hillston. Bio-PEPA: A framework for the modelling and analysis of biological systems. *TCS*, 410(33-34):3065–3084, 2009.

[CJT01] P. Cairns, M. Jones, and H. Thimbleby. Reusable usability analysis with Markov models. *ACM Transactions on Human-Computer Interaction (TOCHI)*, 8(2):99–132, 2001.

[DGG+09] A. Duguid, S. T. Gilmore, M. L. Guerriero, J. Hillston, and L. Loewe. Design and development of software tools for bio-pepa. In A. Dunkin, R. G. Ingalls, E. Yücesan, M. D. Rossetti, R. Hill, and B. Johansson, editors, *Winter Simulation Conference*, pages 956–967. WSC, 2009.

[DMF01] G. Doherty, M. Massink, and G. Faconti. Reasoning about interactive systems with stochastic models. *LNCS*, 2220:144 – 163, 2001.

[Gil77] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[HB10] R. A. Hayden and J. T. Bradley. A fluid analysis framework for a Markovian process algebra. *TCS*, 411(22-24):2260–2297, 2010.

[Hil96] J. Hillston. A compositional approach to performance modelling, 1996. Distinguished Dissertation in Computer Science. Cambridge University Press.

[Hil05] J. Hillston. Fluid flow approximation of PEPA models. In *Proceedings of QEST'05*, pages 33–43. IEEE Computer Society, 2005.

[HML09] 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.

[KFN98] 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.

[KG10] R.A. Kosińskia and A. Grabowskib. Langevin Equations for Modelling Evacuation Processes. *Acta Physica Polonica B Proceedings Supplement*, 3(2):365–376, 2010.

[KKK05] C. Kray, G. Kortuem, and A. Kruger. Adaptive navigation support with public displays. In *Proceedings of IUI 2005*, pages 326–328. ACM Press, New York, 2005.

[KNP09] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic Model Checking for Performance and Reliability analysis. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):40–45, 2009.

[KT09] P. Kemper and C. Tepper. Automated trace analysis of discrete-event system models. *IEEE Trans. Software Eng.*, 35(2):195–208, 2009.

[Kur70] Thomas G. Kurtz. Solutions of ordinary differential equations as limits of pure jump markov processes. *Journal of Applied Probability*, 7(1):49–58, 1970.

[LMBM07] M.E.M. Lijding, N. Meratnia, H.P. Benz, and A. Matysiak Szóstek. Smart signs show you the way. *I/O Vivat*, 22(4):35–38, 2007. ISSN 1389-0468.

[MBLM10] M. Massink, A. Bracciali, D. Latella, and Harrison M. A scalable fluid flow process algebraic approach to emergency egress analysis. In *Proceedings of the 8th IEEE International Conference on Software Engineering And Formal Methods (SEFM2010)*, pages 169–180. IEEE, 2010.

[MHL10] 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.

[SA05] G. Santos and B. E. Aguirre. A critical review of emergency evacuation simulation models. In *Proceedings of* *the NIST Workshop on Building Occupant Movement during Fire Emergencies, June 10-11, 2004*, pages 27–52. NIST/BFRL Publications Online, Gaithersburg, MD, USA, 2005.

[SG83] A.D. Swain and H.E. Guttman. Handbook of human reliability analysis with emphasis on nuclear power plant applications. Technical Report NUREG/CR-1278 SAND80-0200 RX, AN, U.S. Nuclear Regulatory Commission, 1983. Final Report.

[Sle10] J. Slegers. A Langevin Interpretation of PEPA Models. *ENTCS*, 261:71–89, 2010.

[Sti00] G. K. Still. Crowd dynamics, 2000. Dissertation. University of Warwick.

[tBFM$^+$09] M. H. ter Beek, G. Faconti, M. Massink, Ph. Palanque, and M. Winckler. Resilience of interaction techniques to interrupts. In *INTERACT 2009 - Human-Computer Interaction. 12th IFIP TC 13 International Conference. Proceedings, vol. I. LNCS, vol. 5726*, pages 494 – 509. Springer, 2009.

[TGH10] M. Tribastone, S. Gilmore, and J. Hillston. Scalable differential analysis of process algebra models. *To appear in Transactions on Software Engineering*, 2010.

[Tri07] M. Tribastone. The PEPA plug-in project. In *QEST '07: Proceedings of the Fourth International Conference on Quantitative Evaluation of Systems*, pages 53–54, Washington, DC, USA, 2007. IEEE Computer Society.