A Scalable Fluid Flow Process Algebraic Approach to Emergency Egress Analysis

M. Massink, D. Latella, A. Bracciali
CNR-ISTI
Pisa, Italy
Email: {M.Massink, D.Latella, A.Bracciali}@isti.cnr.it

M. D. Harrison
School of Computing Science
Newcastle University
Newcastle upon Tyne, UK
Michael.Harrison@ncl.ac.uk

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. 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. In previous work we have successfully explored a scalable analysis of stochastic process algebraic models of smart signage systems. In this paper we focus on the validation of a representative example of this class of models in the context of emergency egress. This context has the advantage that detailed data is available from studies with alternative analysis methods. A second aim is to show how realistic human behaviour, often observed in emergency egress, can be embedded in the model and how the effect of this behaviour on building evacuation can be analysed in an efficient and scalable way.**

***Keywords*-collective behaviour; validation; process algebra; fluid flow;**

I. Introduction

Smart signage systems, 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. Common recurring elements of smart signage systems are: physical spaces; (handheld) displays; sensors; services and users. Users will interpret information on displays and, ideally, carry out activities as a 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; arrival 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 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 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 [4] performance aspects of continuous interaction with a finger tracking system have been successfully modelled and analysed and in [2] 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 [1]. 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 can be described by using stochastic probability distributions [15].

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 is how to deal with the state space explosion that arises as a consequence of interleaving the behaviour of the many independent people and services that such processes model. 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 final 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 allowing 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 such techniques in this case 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, called fluid flow analysis tailored to the Performance Evaluation Process Algebra (PEPA) and proposed by Hillston [9], 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 [16] 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 [5]. A first exploration of the application of this approach to the modelling of a generic smart signage system can be found in [6] and [12]. These works focus on feasibility and scalability of the approach.

In this paper the main focus is on *validation* of the quantitative aspects of a representative example of this 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 [10] provides a detailed case study of emergency evacuation of a three storey building which will be used here as a basis for the validation of the prediction of results, in particular egress times and room occupancy, of a fluid flow analysis of a process algebraic model of the same system. In addition to the validation of the approach using data from the Evacnet4 study, this paper takes the further step of extending the PEPA model to include typical dynamic aspects of *realistic* evacuee behaviour. A number of such aspects have been mentioned in recent critical reviews of emergency evacuation simulation models [13], [14]. 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 can in general not be taken into consideration in models suitable for optimal flow analysis, such as Evacnet4, because of the particular way the underlying graph algorithms work. This is also observed in [13] remarking 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.

This work is a first step to explore the extent to which PEPA models can be used to model aspects of social dynamic human behaviour and whether a fluid flow analysis may also be helpful in obtaining efficient and realistic approximations in comparison to approaches based on simulation. In fact, a further validation of the fluid flow approximation for this class of models is provided by comparing its results to those obtained via stochastic simulation of the same model. It is shown that when the main interest is in the overall mean behaviour of the system, the fluid flow approximation provides a very quick and quite accurate approximation of the limit behaviour of its simulation. Of course, simulation may provide more detailed information, but this may come at much higher computational costs, which are in general not affordable during early phases of design. The comparison of the complexity/performance of the optimal flow algorithms underlying Evacnet4 versus fluid flow approximation is not addressed in this article. This is because of the complementary nature of the approaches. Optimal flow algorithms are used to calculate *optimal* egress plans abstracting from social dynamic behaviour and fluid flow analysis can be used to analyse models of different *given* egress plans possibly *including* aspects of social dynamic behaviour. Furthermore, in [10] performance figures on the algorithms for the particular case study are unfortunately not provided.

The outline of the paper is the following. First the emergency egress case study described in [10] is presented in Sect. II. This is followed by a brief introduction to the fluid flow semantics of the considered class of PEPA specifications in Sect. III. In Sect. IV 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 Sect. V. Subsequently, in Sect. VI 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 Sect. VII a scaled version of the model is analysed dealing with a scenario with 21,200 evacuees. Sect. VIII finally provides a summary of the results obtained and directions for future research.

II. The emergency egress case study

The case study used for the validation of our fluid flow models of smart signage systems is a representative example of the emergency evacuation of a three storey building. The study is described in detail in [10]. 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. Open spaces between building elements represent doors. Rooms and other building sections referred to as 'locations', 'sections' or

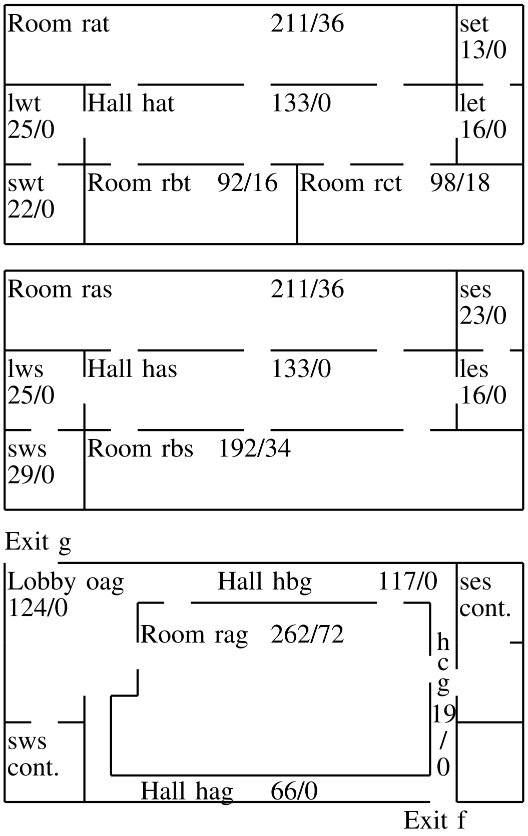

Fig. 1. Three floors of example building from EVACNET4 manual (NC/IC)

'nodes' in this paper, 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).

III. PEPA AND FLUID FLOW SEMANTICS

This section contains a very brief overview of the relevant dialect of the stochastic process algebra PEPA [8] for which a fluid flow semantics has been defined in [9] by Hillston.

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 paper 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 paper may be formally specified using the following grammar:

$$S ::= (\alpha, r).S \;\Big|\; S \,+\, S \;\Big|\; C \qquad P ::= P \bowtie_L P \;\Big|\; S$$

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 $P_1 \bowtie_L P_2$, 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); both model components proceed independently with any activity not occurring in L. The expected duration of a cooperation of activities a belonging to L is a function of the expected durations of the corresponding activities in the components. Typically, it corresponds to the longest one ([8] provides a detailed definition). Two shorthand notations are introduced. If the set L is empty $P_1 \bowtie_L P_2$ 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: $P_1[n] \bowtie_L P_2[m]$. Constants are defined by means of proper defining equations. This way legal PEPA components are cooperations of sequential processes.

PEPA semantics allows for the application of different analysis and evaluation techniques including fluid flow analysis which we illustrate by means of a small example taken from [17], for details see [9]. Consider the following model:

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

This model comprises two classes of components; one class with N_P instances of process P which are all in initial state P and one class with N_Q instances of process Q which are all in initial state Q. Each pair of components (P, Q) can cooperate over action type α. P' and Q' carry out independent actions β and γ, respectively, before returning to the state in which α may be performed. To each state occurring in the specification, i.e. to P, P', Q and Q', a function can be associated that yields a continuous approximation of the total mean number of components which are in that 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 and *decreases* whenever an α-labelled transition occurs. Intuitively, in the first case, the 'average amount of change' should be proportional to the rate p' of β-labelled transitions and the number $\mathcal{P}'(t)$ of processes in state P'; similarly, but considering (i) the fact that α-labelled transitions are cooperation transitions and (ii) the above mentioned PEPA design choice for cooperation which requires the *slowest* rate to be chosen, the 'average amount of change' when $\mathcal{P}(t)$ decreases is given by $min(p\mathcal{P}(t), q\mathcal{Q}(t))$. 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:

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

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 [16] that provides, among others, stochastic simulation and fluid flow analysis.

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 [11] taken from [17] and mention the results obtained by Hayden and Bradley [7]. In order to sketch the approach by Kurtz it is useful to consider an alternative representation of the CTMC associated to PEPA models by PEPA's semantics. This representation abstracts from component identities and records only the number of components in each class that are in the same state at a given time. We show this for the example above and we denote the relative aggregated CTMC by $CTMC(P[N_P] \bowtie_{\{\alpha\}} Q[N_Q])$. 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 an 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 $CTMC_j = CTMC(P[j \cdot N_P] \bowtie_{\{\alpha\}} Q[j \cdot N_Q])$. 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 over $CTMC_n(t)$ may be well 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 $(N_P, 0, N_Q, 0)$. We refer the interested reader to [17] and [11] for details.

In [7] a direct link is established between the aggregated CTMC interpretation of the class of PEPA models used in this paper 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.

IV. PEPA MODEL OF EMERGENCY EGRESS

In this section a concise description of the model is provided. It is composed of processes describing the behaviour of evacuees, processes modelling doors between building sections, processes modelling the handling of requests for information by the evacuees and processes modelling the occupancy of space by evacuees in the various building sections.

A. Evacuees

Evacuees are assumed to be somewhere in the building at the time of an alarm, after which they move towards an exit (see Figure 1). Each evacuee knows (it is assumed) which building exit is the nearest and that they get information about where to go next every time they enter a new location. As an example we illustrate the behaviour of evacuees located in Room rat at the third floor heading to *e*xit 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 $a = 60$:

$$EvcEgoprat = (lrateg, a).EvcEgRecrat$$

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

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

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 at the third floor the specification would look like:

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

This specifies that the evacuee may, in principle, receive different indications on 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 of 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 Subsection C. On average, it is estimated that the response takes 1 second to arrive. Hence $r = 60$. In the PEPA fragment above $MveEgFratThat$ reads as "***Mo**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 [10] 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 [10][1]. 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 [10], the evacuation flow is pre-determined, but we have to take it into account in the extensions of the model discussed in Sect. VI.

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 $lhatop$), 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:

$$\begin{aligned} EvcEgFratThat = {} & (rathat, udr).(lhatop, s). \\ & (lratfp, s).EvcEgophat \end{aligned}$$

At some point the evacuee reaches an exit, for example exit g, via the lobby at the ground floor oag (see Figure 1) passing through a door with rate udr (activity $(oagg, udr)$). 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 rate $s = 60$:

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

$$EvcEgArrived = (nop, a).EvcEgArrived$$

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.

B. 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 at the third floor:

$$DUrathat = (rathat, udr).DUrathat$$

The parameter udr characterises the average time it takes a person to pass through such a door. Based on experimental data from [10] a rate of $udr = 30$ seems a good approximation[2].

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 them. 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 (udr) and the number of doors times the rate at which they make an evacuee pass (udr). 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.

C. Request handling

Request handling models the responses given to an evacuee that is in a certain section of the building when requesting 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, which 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:

$$\begin{aligned} RqHLrat = {} & (lrateg, a).RqHLratRespEg + \\ & (lratef, a).RqHLratRespEf \end{aligned}$$

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

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 as well as when the evacuee

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

[2] Considering that in the example in [10], on average, 6.1 persons pass through 3 doors per time unit of 5 seconds to go from Room a to Hall h, we derive that it takes approximately 2 seconds for a person to pass a door. This brings to a rate of 30.

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 [10]. 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.

D. Places

When a place is free in a room (for example, rat) the process modelling 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.

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

E. 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, $EvcEgoprat[18]$ represents 18 ***Ev***a***c***uees with destination ***E***xit g that initially ***o***ccupy a ***p***lace in ***r***oom ***a*** at 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 $\bowtie_{\{lrateg,\ldots,lgfp\}}$, 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 proper 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.

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

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

TABLE I
KIND AND NUMBER OF DOORS AND DISTANCES.

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

TABLE II
NODE CAPACITY (NC) AND NR. OF GUIDANCE SERVICES.

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 having 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.

F. Automatic generation of the model

Given the size and scale of the system it is not feasible or at least easy to develop an error free version of the complete PEPA specification of the three storey building by hand. The modular structure of the model lends itself to automatic generation of the PEPA specification starting from simple domain oriented input data. This is the approach followed for the models in this paper. The functional programming language Haskell was used to develop a prototype of such a generator. As an example, the Tables I, II, III and IV present the domain oriented input data for the case study at hand (restricted to those for the third floor except for Table III).

V. VALIDATION OF THE MODEL

In this section we validate the fluid flow analysis of the model described in Sect. IV by comparing our results with the results of the optimal flow analysis presented in [10]. The

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

TABLE III
EVACUEES INITIALLY PRESENT AND RELATED EXIT FOR ALL FLOORS.

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

TABLE IV
ROUTING OF EVACUEES FOR FLOOR 3.

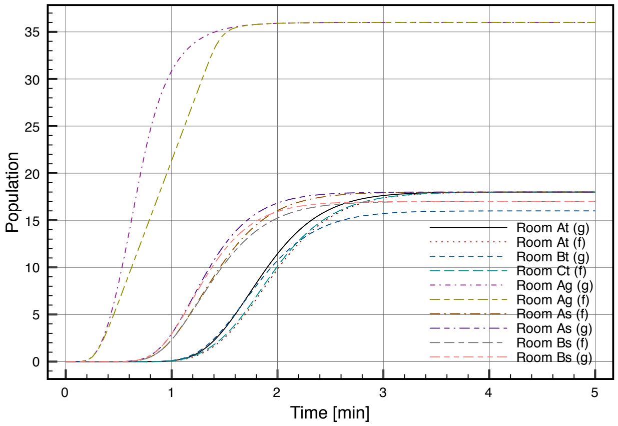

Fig. 2. Arrival of evacuees (ODE)

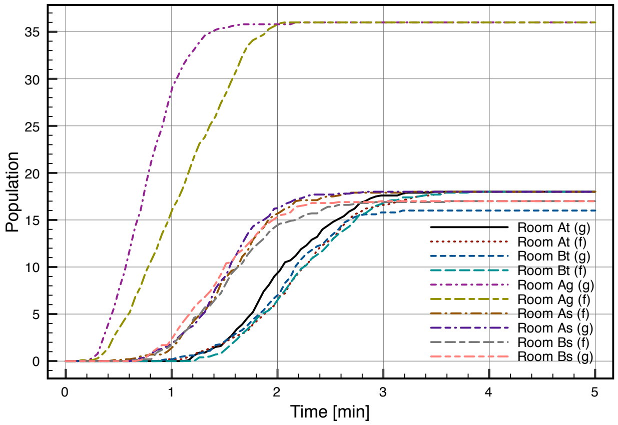

Fig. 3. Arrival of evacuees (GIL)

comparison takes into consideration 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).

The results we obtained within a fluid flow framework are coherent with those in [10], while the differences can often be explained by the different kinds of solution sought. The results in [10] are given in terms of optimal fluxes. Our modelling approach *complements* the search for optimal routing solutions by allowing other cases, sometimes closer to reality, to be considered. The model can be easily adapted to different scenarios, such as the presence of hurt evacuees or dynamic routing strategies, as illustrated in Sect. VI.

We also present averaged stochastic simulations of evacuation. Stochastic simulations 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". The relationship between the fluid flow approximation and stochastic simulation is one of convergence as discussed briefly in Sect. III. In this paper we limit ourselves to observe a clear coincidence of stochastic simulation and fluid flow results, even for a relatively small number of stochastic samples and perhaps not so large population numbers. We take this as a further confirmation of the proper construction of our model.

All stochastic simulation and fluid flow analyses in this paper have been performed with the mentioned parameters on an Apple iMac 2.66GHz Intel Core i5 using the PEPA-plugin analysis tool [16] unless otherwise stated.

A. Predicted building evacuation time

In Figure 2 the arrivals of all evacuees from the different rooms are shown, over the time interval [0, 5] minutes using an adaptive step-size 5th order Dormand-Prince ODE solver. 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 shows the evacuees arriving from Room A at the ground floor leaving the building through exit g. It is clear that evacuees at 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 at the ground floor reach their exit. In Figure 3 results are shown for a stochastic simulation of the same model using Gillespie's algorithm [5] for the same time interval, with 10 independent runs and confidence interval 0.05. As can be observed, the curves in Figure 2 and Figure 3 correspond reasonably well given the small number of replicated runs of the stochastic simulation.

The total (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 4 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 [10] 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.

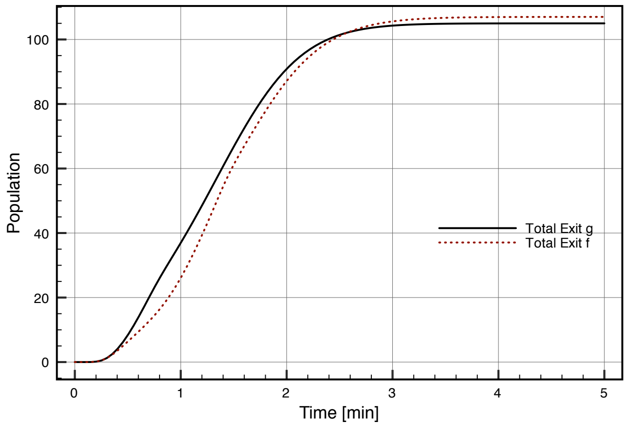

Fig. 4. Arrivals at exits f and g (ODE)

B. Node profiles

Figure 5 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 6 the results are shown for a stochastic simulation (GIL) of the same model. Again both figures show a good correspondence despite the low number of independent runs in the stochastic simulation.

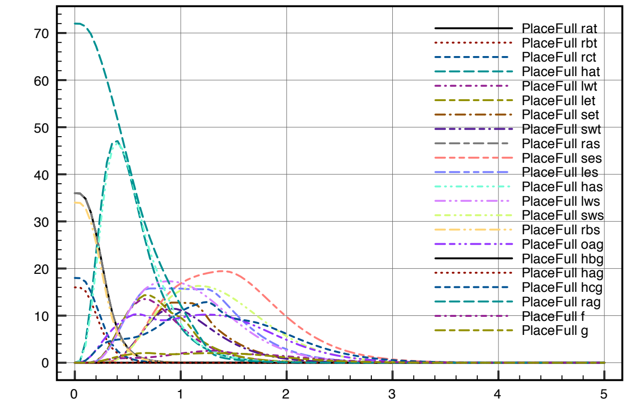

Fig. 5. Occupied places over time (ODE)

The leftmost curves show that 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 sign for evacuation is given.

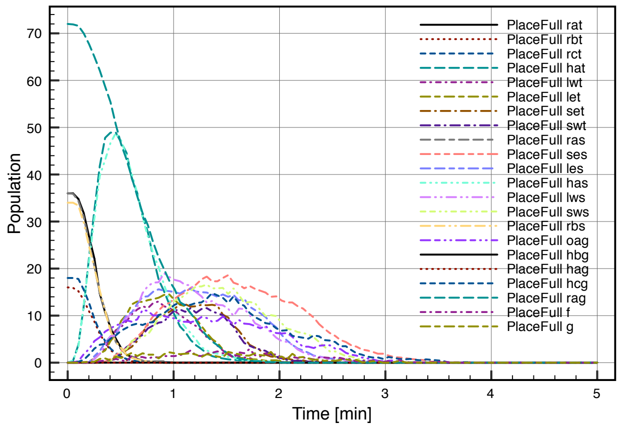

Fig. 6. Occupied places over time (GIL)

Figure 7 shows the comparison of the original node profile, at intervals of 5 seconds, for Hall has 2nd floor as reported in [10] (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 minute 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.

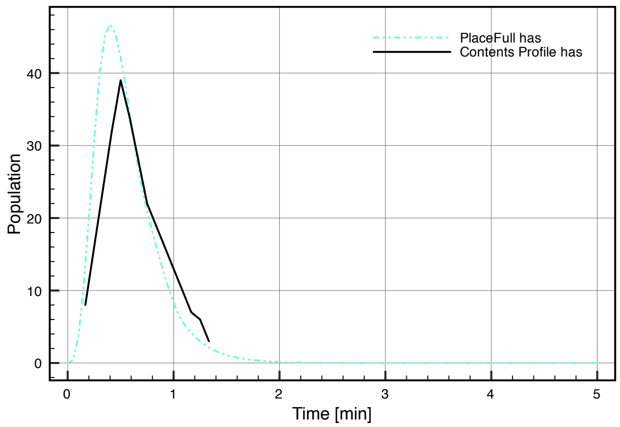

Fig. 7. Node profile Hall A second floor (ODE vs Evacnet4)

VI. Social aspects of evacuee behaviour

One of the observations made in the review of emergency evacuation models [13], [14] 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. 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 erroneous interpretation of guidance instructions.

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.

A. Hurt Evacuees

A realistic event that may occur is that during 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 Sect. IV reflects the situation described above assuming that the ***l***anding of the ***W***estern staircase at 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 at the third floor (swt). The excerpt below shows the extension for swt, that for has being similar.

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

$$EvcEgHurtSTOP = (nop, a).EvcEgHurtSTOP$$

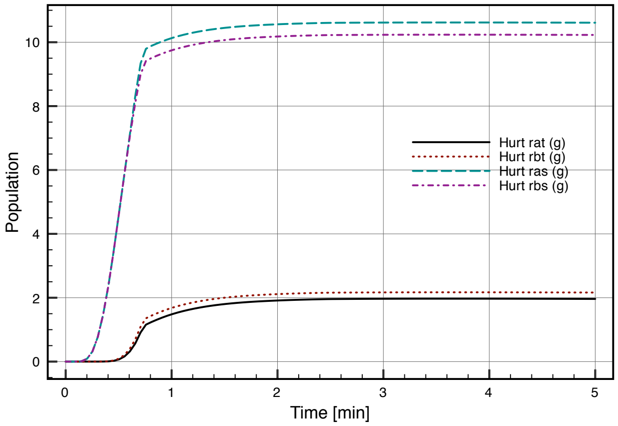

Fig. 8. Hurt evacuees in lws from different rooms (ODE)

Figure 8 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.

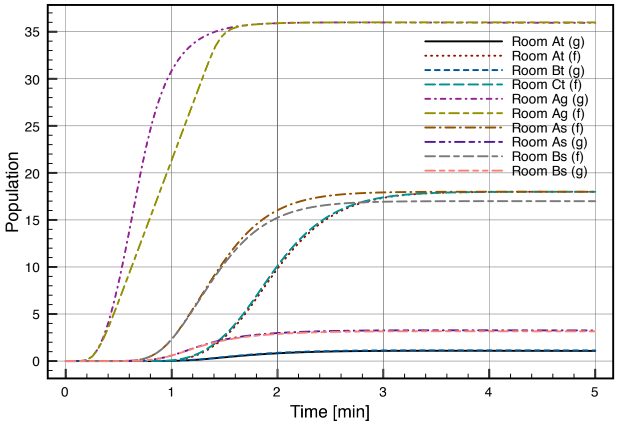

Fig. 9. Arrivals of evacuees in case of toxic gases in lws (ODE)

Figure 9 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 at the third floor arrives via exit g and the same for evacuees from room rbt at the third floor. All the others get blocked. Figure 10 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 the Western

Landing is 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 floor (curve PlaceFull has). This is in part due 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, though these instructions became obsolete given the circumstances. 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.

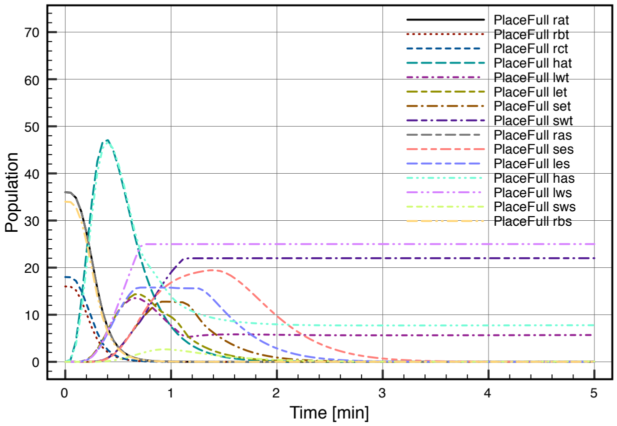

Fig. 10. Node occupancy in case of toxic gases in lws (ODE)

B. 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 blocked. 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.

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

The evaluation results in Figure 11 and Figure 12 show that in this case almost all non-hurt evacuees 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 11 shows the occupancy of the building sections at the third floor. The figure shows that evacuees from all building sections at 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, particularly so because the model has been changed in such a way that evacuees that do not manage to enter the Western landing at the second floor, both from the hall and from the Western stairwell at the third floor, are now able to go to another exit receiving directions for that exit. A closer look at the routing table shows however that evacuees in the Western stairwell at 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. In fact, this analysis shows 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 in order not to cause a serious bottleneck on the stairs.

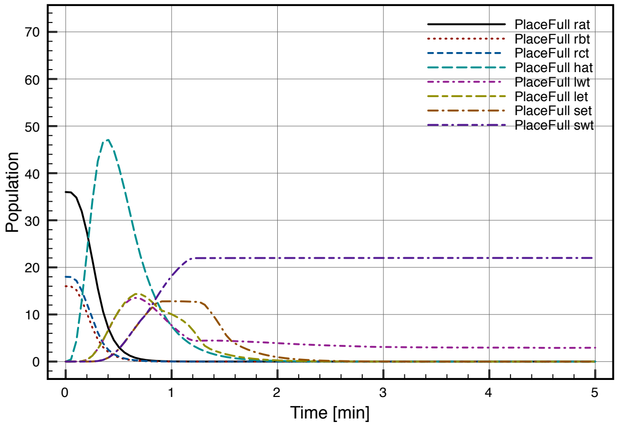

Fig. 11. Full places third floor model with alternative exits (ODE)

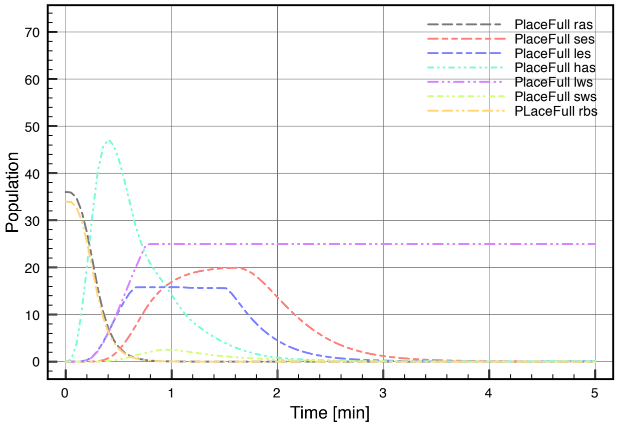

Fig. 12. Full places second floor model with alternative exits (ODE)

Figure 12 shows the node occupancy at the second floor. Clearly, the Western landing at the second floor fills up quickly

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

TABLE V
OVERVIEW OF ANALYSIS TIMES

and then remains full. This represents hurt evacuees that are not able to continue. It can also be observed that all the evacuees in Hall a at the second floor now manage to leave the floor. In this case going to another exit results in a different route that does not include the Western landing. All other building sections at the second floor are completely evacuated eventually.

VII. Scaling Up

The model presented in Sect. IV considers only a limited number of evacuees in order to compare results with the existing case study. Figure 13 shows fluid flow results for a model with a total of 21,200 evacuees, assuming a much larger though similar building (obtained by multiplication of all room capacities and distances by a factor 10 and the number of evacuees by a factor of 100). Table V 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 paper. 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 of Sect. VII 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. Unfortunately, we have no data on whether Evacnet4 could deal with examples of this size and how much time it would take to calculate an optimal evacuation plan.

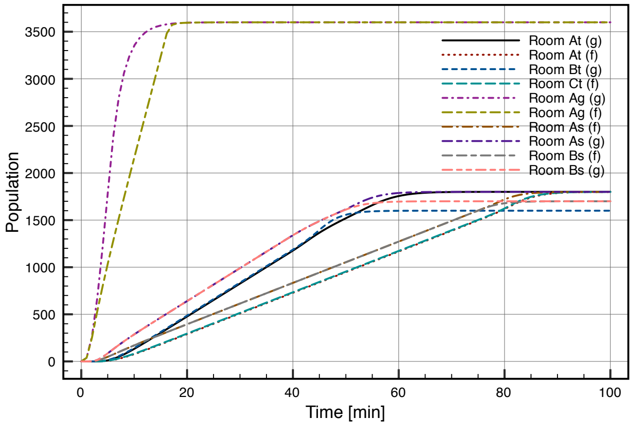

Fig. 13. Arrivals in a large building (ODE)

VIII. Conclusions and Outlook

We have presented the validation of a modelling approach for collective dynamics by addressing a realistic case study in the context of smart signage systems. The problem has been drawn and adapted from an existing analysis of the emergency egress of a building, originally carried out by means of flow optimisation techniques. We have proposed the use of a fluid flow approach for the quantitative modeling of such systems, coupled with the expressiveness and versatility of formal method techniques based on the PEPA process algebra. 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. The PEPA language provides an abstract, formally grounded, description of the 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 efficient, since the continuous interpretation of fluid flow allows efficient numerical solvers to be used. Furthermore, other analysis techniques can be used, like stochastic simulations, directly provided by the PEPA toolkit, and stochastic model checking on reduced versions of such models.

Our validation relies on a strong coherence of the results obtained by fluid flow, by stochastic simulations and the results of the original optimal flux analysis. A comparison of the computational costs underlines the efficiency of fluid flow, which hence appears as a valuable technique, especially whenever the overall behaviour of the system is not dramatically affected by stochastic noise.

Examples relating to hurt evacuees or dynamic routing decisions have also been illustrated. The relevance of such non-optimal or socially influenced behaviour for the problem domain suggests interesting directions for future work. Contextual information, like the congestion of possible escape paths or the presence or obstacles in given areas of the building, also provide results that are particularly relevant. In order to take into account richer contextual information, we have started to consider BioPEPA [3], a specialisation of PEPA tailored to the modeling of biological systems. Quantities of classes of individuals and a notion of spatial location are first class objects in this language. This allows, for instance, the number of individuals of a given class in a room to be explicitly referred to when individuals or routers make decisions on preferred escape paths. We expect that a language like BioPEPA will enhance the expressiveness of the models. Experiments have been started in collaboration with the team currently developing the BioPEPA toolkit, as a step towards the improvement of modelling in the considered domain.

Acknowledgment

The authors would like to thank Stephen Gilmore, Mirco Tribastone and Jane Hillston (The University of Edinburgh) for their support with the PEPA plug-in and discussion on precursor models of the one here presented. This research has been partially funded by the CNR project RSTL-XXL and by the Italian PRIN MIUR project PaCO.

References

[1] M. H. ter Beek, G. Faconti, M. Massink, P. Palanque, and M. Winckler. Resilience of interaction techniques to interrupts. In: Proc. of INTER-ACT 2009, vol. I pp. 494–509, LNCS, vol. 5726, Springer, 2009.

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

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

[4] G. Doherty, M. Massink, and G. Faconti. Reasoning about interactive systems with stochastic models. In: LNCS, vol. 2220 pp. 144–163, Springer, 2001.

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

[6] M. D. Harrison, M. Massink, and D. Latella. Engineering crowd interaction within smart environments. In: Proc. of EICS09 - ACM SIGCHI Symp. on Eng. Interact. Comp. Sys.. pp. 117–122. ACM, 2009.

[7] R. A. Hayden and J. T. Bradley. A fluid analysis framework for a Markovian process algebra. *Theoretical Computer Science*, 411:2260–2297, Elsevier, 2010.

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

[9] J. Hillston. Fluid flow approximation of PEPA models. In *Proc. of QEST'05*, pages 33–43. IEEE Comp. Soc., 2005.

[10] 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.

[11] T. G. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *Journ. of Appl. Prob.*, 7(1):49–58, 1970.

[12] M. Massink, M. D. Harrison, and D. Latella. Scalable analysis of collective behaviour in smart service systems. In *Proc. of ACM-SAC, Volume 2, AI and Agents*. ACM, 2010.

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

[14] G. K. Still. Crowd Dynamics. Ph.D. Thesis, Univ. of Warwick, August 2000.

[15] A. Swain and H. 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. Final Report. 1983

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

[17] M. Tribastone and S. Gilmore and J. Hillston. Scalable Differential Analysis of Process Algebra Models. To appear in Transactions on Software Engineering, 2010.