Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

A MODEL-DRIVEN AND SIMULATION-BASED METHOD TO ANALYZE BUILDING EVACUATION PLANS

Daniele Gianni

Dept. Applied Sciences and Technologies
Guglielmo Marconi University
Rome, Italy

Paolo Bocciarelli, Andrea D'Ambrogio

Dept. Enterprise Engineering
University of Rome Tor Vergata
Rome, Italy

Giuseppe Iazeolla

Dept. Applied Sciences and Technologies
Guglielmo Marconi University
Rome, Italy

ABSTRACT

Modern buildings are often expected to satisfy minimum safety requirements to define upper bounds for safety metrics, such as evacuation time. The building design must therefore consider prediction of these metrics for a set of representative evacuation scenarios. These scenarios can be rather complex, and often can be investigated only using building evacuation simulators. However, these simulators might require considerable development effort, and their use might therefore become less convenient, for time and cost issues. In this respect, this paper introduces a model-driven method to automatically develop building evacuation simulators from informal specifications of building evacuation scenarios, i.e., building plans and behavioral descriptions of evacuees. Specifically, the paper shows how a floor plan develops in the structural characteristics of an Extended Queueing Network (EQN) model and how the behavioral description can be used to parameterize the EQN model. The paper also presents an example application along with preliminary validation issues.

1 INTRODUCTION

System design is often supported by simulation techniques and tools that enable designers to investigate system alternatives on virtual representations, avoiding the costs of physical prototypes and overcoming the limitations of analytical approaches. Modern buildings are often required to satisfy large sets of requirements, from purely structural ones, concerning the mechanical engineering, to energy and safety ones. For the inherent variety and complexity of requirements, simulators and design techniques are often used manually to solve ad-hoc issues.

In the wider engineering community, model-driven approaches have proven to be effective to support systems engineering activities (Bocciarelli and D'Ambrogio 2014). These approaches offer several advantages, including increased formality of design models specified by use of standard languages, e.g., UML (OMG 2011) or SysML (OMG 2010), so as to enable the automated generation of design documentation or simulation code ready to be executed, for example. Applying these approaches to building design, while integrating with available formats of building design specifications, allows one to obtain similar advantages and more conveniently integrate simulation-based validation activities. This can be particularly valuable to support the design for safety requirements, such as the number of exit ways or stairwells to meet a specified total evacuation time. Indeed, evacuation scenarios might present diverse resulting situations depending on many parameters, including building occupation, building plans and evacuees behavioral models (Pan et al. 2006).

In line with the aforementioned approaches, this paper introduces a model-driven method to semi-automatically derive building evacuation simulation models from informal specifications of building design.

The method takes as input a building plan and a set of behavioral descriptions, and yields as output a simulation model in the form of an Extended Queueing Network (EQN) model (Bolch et al. 2006). The EQN model is then mapped to the corresponding implementation code specified by use of jEQN (D'Ambrogio et al. 2006, Gianni and D'Ambrogio 2007), a Java-based language to implement EQN models that can be transparently executed either as a sequential simulation or as a distributed simulation. In addition, maintaining a linked traceability between the building specification and the simulation code allows one to promptly deal with any subsequent modification of the building design and rapidly enact further investigations. The proposed method is almost entirely automated, thus potentially contributing to effectively integrate simulation techniques in the building design for the relevant safety requirements.

The paper is organized as follows: Section 2 reviews related works, while Section 3 and Section 4 give the details of the building model and the simulation model, respectively. Section 5 illustrates the proposed model-driven method and, finally, Section 6 presents an example application and a preliminary validation with complementary simulation and modeling approaches.

2 RELATED WORK

Several contributions can be found in literature that deal with building evacuation issues. Due to the widespread and crosscut characterization of this topic, the several available contributions face the building evacuation problem from a different perspective, emphasizing a subset of the several issues which characterize the general problem. As stated in Section 1, this paper proposes a methodological contribution which exploits model-driven techniques and distributed simulation to automate and ease the analysis of evacuation plans during the preliminary building design. In this respect, this literature review specifically takes into account: i) relevant contributions dealing with the building evacuation problem from a general point of view, ii) contributions that specifically focus on modeling and methodological issues and iii) contributions that take into account the adoption of distributed simulation approaches.

The analysis of building evacuation scenarios has been widely studied in the last decades to support the design of evacuation procedures at building design time. As identified in (Pursals and Garzón 2009), one of the first and most relevant study that proposes the adoption of *queueing networks models* can be found in (Smith and Towsley 1981). Recent and valuable contributions that propose extensive comparative analysis of the most relevant tools and modeling approaches can be found in (Santos and Aguirre 2004, Zheng et al. 2009). Such studies recognize the effectiveness of using evacuation simulation tools, e.g., EXODUS (Group 2003), to support the building evacuation analysis. Nevertheless, such approaches are affected by limitations in terms of weak integration into the design life cycle, failure to properly account for resources contention and a very low degree of customizability (most commercial tools do not provide any customization feature). Differently, this paper adopts EQN models as an effective formalism to capture resources contention and exploits model-driven and simulation-based techniques to analyze building evacuation plans starting from building preliminary design models. The use of model-driven techniques allows to obtain an high degree of customizability in terms of the required input model and the adopted simulation formalism.

As regards modeling and methodological issues, relevant contributions can be found in (Wang et al. 2013, Wang et al. 2013, Wang et al. 2012, Sayed Ahmed and Wainer 2010). In (Wang et al. 2013) a model-driven approach is used to design a scalable *building information modeling and simulation* framework, which allows one to conduct a simulation-based analysis of evacuation plans. The paper also includes an example application to study the evacuation model of a multi-floor building. Finally, as a relevant on-going work, the paper highlights the need of a model-driven flow of transformations, which should be provided to automate the entire proposed simulation-based process. In (Wang et al. 2013, Wang et al.

2012, Sayed Ahmed and Wainer 2010) the Building Information Modeling (BIM) and its open standard Industry Foundation Classes (IFC) are adopted to model the building under study, while the cell-DEVS formalism is used to specify the simulation model. Similarly to such contributions, this paper proposes a model-driven method that is specifically focused on the analysis of building evacuation plans at design time (thus not dealing with real-time crowd behavior during an emergency evacuation). Moreover, in this paper we provide a complete flow of automated transformations that enable an easy integration of simulation-based techniques in the traditional building design process. On the other hand, differently from such papers, we use a different notation to represent both the (input) building and the (output) simulation models.

Contributions that specifically deal with the use of distributed simulation can be found in (Filippoupolitis et al. 2012, Dimakis et al. 2010, Filippoupolitis and Gelenbe 2009). Such contributions identify the main limitations of the traditional sequential simulation approaches, which are exacerbated in context where a real-time emergency response is needed, i.e.: i) fault tolerance, as the sequential application is a unique point of failure, ii) performance limitations, as the performance is tied to the computational capabilities of a single host, iii) limitations to support cooperation and collaboration in complex scenarios. In this respect, this paper contribution exploits the use of the jEQN language, which is implemented on top of SimArch (Gianni et al. 2011), a layered architecture that makes it possible to transparently execute simulation models in either a sequential or distributed environment. For validation purposes, this paper considers the case study introduced in (Filippoupolitis et al. 2012), which adopts SimArch to implement an agent-based distributed simulation system.

3 BUILDING MODEL

As discussed in Section 1, the goal of the method proposed in this paper is to automate the simulation-based analysis of evacuation plans from the early stages of the building design process. The method is based on a chain of automated model transformations that take as input an abstract model of the building, in terms of both building structure and occupants behavior, and yield as output an executable simulation model that gives a prediction about the effectiveness of the given building evacuation plan. A relevant issue is the formalism adopted for representing the building model. This paper adopts a neutral and graph-based formalism (Filippoupolitis et al. 2012), which is general enough not to be tied to any industrial/commercial notation and is able to capture the information required to automatically obtain an appropriate simulation model. It is to be remarked that the adopted notation does not influence the validity of the proposed method, as the required graph-based model can be derived by any other building model notation (e.g., an IFC-compliant model).

The adopted modeling approach includes both i) the formal description of the structural characteristics of the building, provided by use of the set theory, and ii) the mathematical representation of the occupants behavior. The following two subsections describe the structural and the behavioral building model, respectively.

3.1 Structural model

The structural model consists of an undirected graph in which the nodes represent the relevant points on the plan and the edges define the space of movements among the nodes. Nodes are thus associated to doors, desks, exits, fire extinguishers, etc. In addition, nodes are also introduced in the areas that are likely to be overpopulated and in the intersection of two or more civilian evacuation flows, where a competition for the use of physical space can occur (e.g., a corridor, where intersection nodes are introduced near each door node, to consider the joining flow between the civilians leaving the rooms and the civilians walking along the corridor). Figure 1 shows an example floor plan and an overlapped related graph-based model.

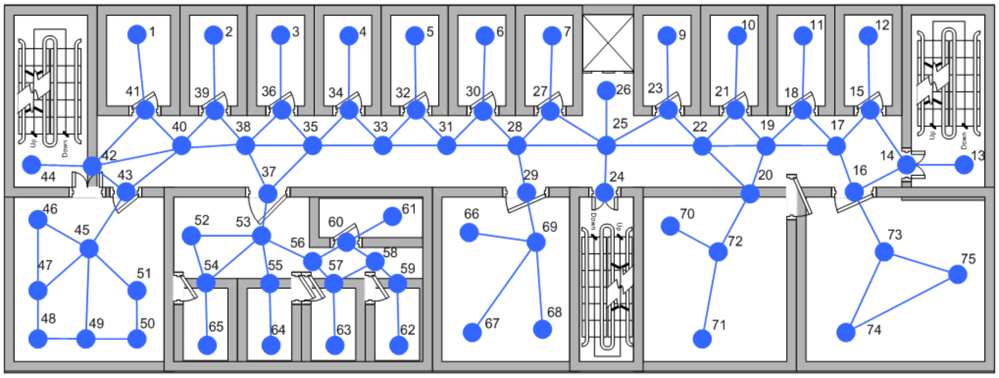

Figure 1: Example floor plan and graph-based structural model

The graph contains seventy-five nodes and ninety-four edges and can be more formally described using set theory concepts. Let us denote with N the set of nodes, being n_i the *i-th* node, with E the set of edges, being e_{ij} the edge connecting nodes n_i and n_j, and with E_i the set of edges that reach node n_i.

In addition to nodes and edges, the graph presents also the following properties: 1) the nodes are grouped depending on the physical area (room or corridor); 2) each edge is characterized by a weight defining a distance; and 3) a subset of nodes identifies the exit nodes. These properties can be represented as attributes of the graph elements, nodes and edges, and can be referenced with the syntax *graphElement.property*, where *graphElement* can be any node or edge, and *property* is the name of the property. For example, the group of nodes n_{20} is identified by $n_{20}.group$. Similarly the exit nodes and the weights are identified with the following syntax: $n_i.isExit$, where *isExit* is of boolean type, *true* on exit nodes (e.g.: n_{24}, n_{44}, and n_{13}), *false* on the remaining one, and $e_{ij}.weight$, where *weight* is the arc length in *cm*.

The following sets can then be introduced:

$$G_j := \{n_i \in N : n_i.group = j\}, G := \{G_i, \forall i \in [0; N_{group}]\}$$

$$W := \{w_{ij}, \forall (i,j) : e_{ij} \in E \wedge w_{ij} = e_{ij}.weight\}$$

$$N_{exit} := \{n_i : n_i \in N \wedge n_j.isExit\}$$

where G_j is the set of nodes in the $j-th$ group, which is also characterized by the property $G_j.maxCapacity$, e.g., the maximum number of civilians who can stand in the group, and G is the set of groups. W is the set of edge weights, and N_{exit} is the set of exit nodes.

Besides the above defined structural properties the graph can be analogously decorated with other properties that can be useful to describe the behavioral model. For example, properties such as hazard level for fire or smoke, or the number of civilians standing on a group of nodes and edges, can be used to define behaviors that depend on the less dangerous path or on the estimated individual evacuation time, respectively.

The following subsection introduces a metamodel formalization of the adopted graph-based notation, so as to specify building models that are instances of such a metamodel.

3.1.1 Building Metamodel

According to model-driven standards introduced by the OMG's MDA - Model Driven Architecture (OMG 2003), the graph-based model structure formally described from an abstract perspective is specified by a MOF-compliant metamodel, in order to be effectively handled by the model transformations that are part of the proposed method. In this respect, Figure 2 shows the MOF-compliant metamodel introduced in this paper for the specification of building models.

According to the description provided in the previous section, a building model is a collection of nodes and edges, represented by the metaclasses `Node` and `Edge`, respectively. As a building may be constituted by several floors, each one divided into well defined areas (e.g., stairs, corridors, rooms, etc.), the building model provides the metaclasses `Floor` and `Area` to group nodes according to the building topology. Finally, the metaclass `ExitNode` is introduced to represent nodes acting as a way out from a given area.

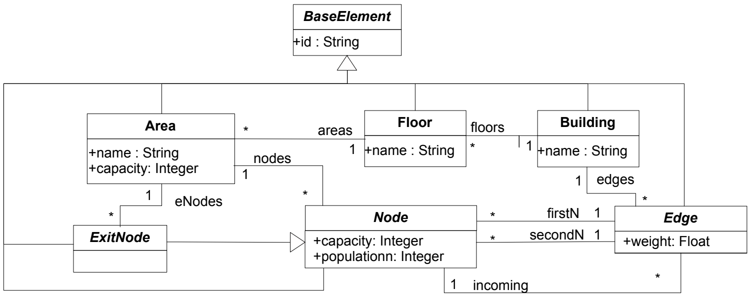

Figure 2: MOF metamodel of building models

3.1.2 Behavioral model

The behavioral model defines how the civilians move as constrained by the structural model. Civilians are grouped in categories, which define the classes of statistical equivalence of civilians, and are described by a motion and a decision model for each category. The *motion model* defines the pace of the movement on the structural model, while the *decision model* defines the path chosen in the evacuation process. For the sake of simplicity, this paper only considers a single category of civilians. Further categories can be introduced using the same schema to represent, e.g., hazard and panic scenarios.

The motion model is easily defined with the motion time function on the nodes and the edges. In the case of edges, the motion time is easily defined as:

$$t_{e_{ij}} = \frac{w_{ij}}{S}(s), e_{ij} \in E$$

where $t_{e_{ij}}$ is the time function for the motion along edge e_{ij}, w_{ij} represents the physical length of the edge e_{ij} and S is the motion speed in walking conditions. The speed is available from several studies and it is commonly modeled as a stochastic variable U uniformly distributed in $[145;155]$ cm/s (British Standard Institute 1997). Differently, the motion time on the nodes requires further considerations because nodes have no physical dimensions by definition. This issue can however be easily overcome. For the determination of the service times, the nodes can be associated to a short length l, which is to be subtracted for half of its value to the weights of the incident edges representing opposite evacuation flows. We shall choose l equal to 50 cm and the service time results $50/S$, which can be approximated without considerable error to $50/150 = 0.3s$, considering the speed uniform and equal to the average value of U. The motion time on the nodes is thus defined as:

$$t_{n_i} = 0.3(s)$$

It is important to remark how such functions can be made more general to model also other phenomena. For example, in panic conditions the motion along the edges might be faster and the motion across the nodes might be subjected to collision phenomena, which reduce the physical resource utilization by increasing the motion time in function of the number of queued civilians. This can be formulated as:

$$t_{e_{ij}} = \frac{w_{ij}}{S_{panic}}(s), e_{ij} \in E$$

$$t_{n_i} = 0.3 + k \cdot n_i.queueLength(s), k = 0.5(s)$$

The decision model defines the evacuation strategy for each category of civilians. The model is defined by a formal expression that indicates which of the exit nodes the civilians reach, and the path they use. Several different models can be considered to represent the civilian behavior. Possible example models are the *shortest distance* model, according to which a civilian at node n follows the path to the closest exit, the *stochastic distance-based* model, in which, at each step, the chosen path is selected depending on a probability related to the distance of each exit compared to the others, and the *fastest evacuation time* model, in which civilians follow the path identified as the fastest by previous experiments. The flexibility of the proposed approach allows to easily manage the chosen decision model, by transforming such a model into an appropriate parametrization of the related simulation model. This issue is further discussed in Section 5.1.

In the rest of the paper, the *shortest distance model* is considered. Such a model can be formulated with the following expression:

$$n_k = \min_{n_k \in N_{exit}} \{d_k(n)\}$$

where $d_k(n)$ is the shortest distance between the civilian current position and node n, and is therefore calculated on the weights of the graph edges.

4 SIMULATION MODEL

Beside the specification of the building model that constitutes the input of the proposed method, the second relevant issue to deal with is the adopted simulation approach, which in turn affects the kind of simulation model to be used. As aforementioned, this paper adopts the EQN formalism to represent the civilians behavior and the contention for accessing the related resources (e.g., the physical spaces in the building).

This approach is in line with well-established and widely used modeling techniques in other domains, such as traffic systems and computer networks. The network approach operates at higher level of abstraction compared to other ones (e.g., cellular automata and agent-based) which fail to represent the resource contention and/or in which the individual intelligence is simulated by a stochastic process over several simulation runs. In addition, the simulation of EQN models requires less computational resources achieving analogous accuracy levels. This implies a better manageability of the simulation experiments, which for this non stationary system can be easily automated within the context of the same program execution. On the other hand, the EQN approach fails to provide the scalability level of agent-based or cellular automata approaches and is thus not appropriate for the analysis of crowd evacuation scenarios.

4.1 EQN metamodel

According to the proposed model-driven approach, and similarly to the building model, the EQN-based simulation model is built as an instance of a MOF metamodel. In this respect, this work adopts the EQN metamodel proposed in (Bocciarelli et al. 2012) that, for the sake of completeness, is depicted in Figure 3 and herein briefly outlined.

The metamodel include a metaclass for each EQN entity (e.g., Source, Service Centers, Waiting Systems) and shows the relationships which may hold among them.

The proposed model-driven method exploits the jEQN language to implement the corresponding executable simulation. As jEQN allows to transparently and effortlessly handle the execution of simulations over distributed infrastructures, the EQN model must include information needed to drive the generation of the several federates composing the distributed simulation. To this purpose, in order to model the partitioning of an EQN model into several subnetworks, the `Subnetwork` metaclass has been added to the metamodel. Moreover, as a *subnetwork* is constituted by a collection of *centers* and *links*, the metamodel

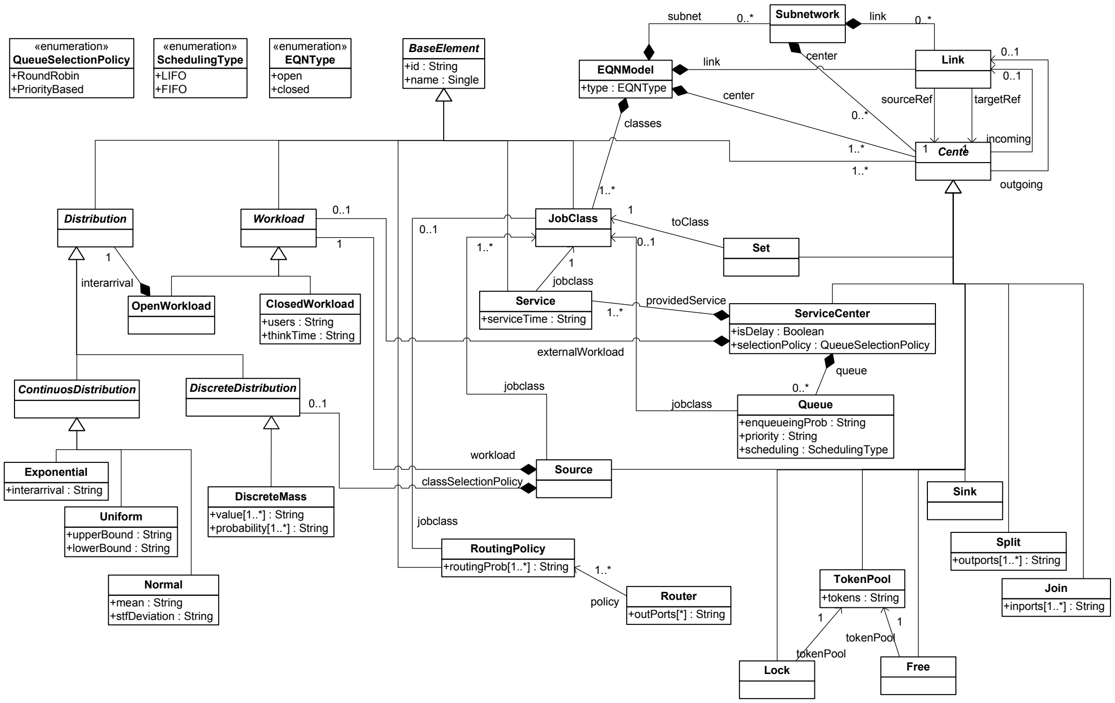

Figure 3: MOF metamodel of EQN models

also includes the aggregation relationships `center` and `link` between the `Subnetwork` metaclass and the `Center` and the `Link` metaclasses, respectively.

For a complete description of the EQN metamodel, the reader is referred to (Bocciarelli et al. 2012).

5 MODEL-DRIVEN METHOD FOR BUILDING EVACUATION ANALYSIS

Figure 4 illustrates the model-driven method that enables the simulation-based analysis of evacuation plans of buildings, from the early stages of the building design and construction process. Specifically, during the building preliminary design, engineers may want to assess the validity of the defined evacuation plans and/or evaluate, among several alternatives, the one that ensures a safe evacuation in case of an emergency. At this initial step, a graph-based building model that can be assumed compliant to the metamodel discussed in Section 3.1.1 is available[1]. At the second step, the *evacuation plan analysis* is carried out by generating and executing the EQN-based simulation model illustrated in Section 4. Specifically, this step is structured as follows:

- the `graph-to-EQN` *model-to-model* transformation is carried out. Such a transformation takes as input the graph-based model and yields as output an EQN model of the building under study;
- the `EQN-to-jEQN` *model-to-text* transformation is carried-out, in order to generate the jEQN executable code from the EQN model;
- The jEQN code is then deployed onto a SimArch-enabled platform and finally executed to yield the results of the evacuation plan analysis. This task is marked as both manual and automated, to highlight the fact that the output of the *model-to-text* transformation requires a minimal manual effort to be configured and executed over the computational platform.

[1] The graph-based model of the building under study can be obtained either manually or by use of automated or semi-automated model transformations from the building model formats introduced by given domain-specific standards and/or tools.

The simulation results allow engineers to assess at design time the validity of the designed evacuation plan and, ultimately, either confirm the effectiveness of the building design or suggest a revision of the project in order to improve the efficiency and safety of emergency evacuation.

The following two sections give additional details about the `graph-to-EQN` and the `EQN-to-jEQN` transformations.

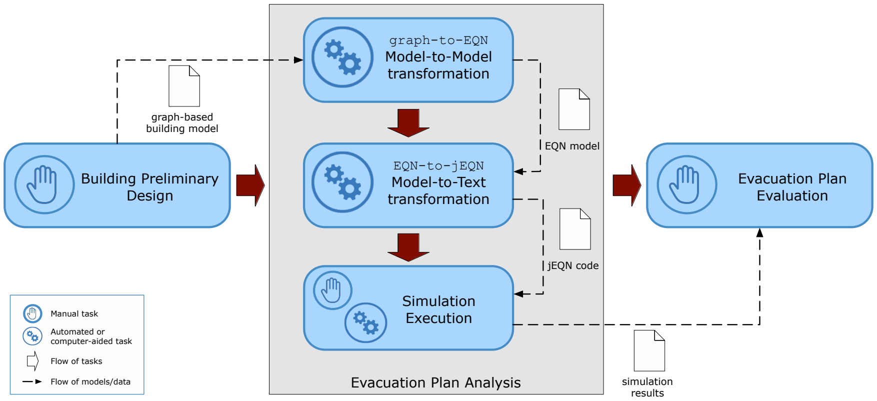
graph-based building model
graph-to-EQN Model-to-Model transformation
EQN model
Building Preliminary Design
EQN-to-jEQN Model-to-Text transformation
jEQN code
Evacuation Plan Evaluation
Simulation Execution
Evacuation Plan Analysis
simulation results
Manual task
Automated or computer-aided task
Flow of tasks
Flow of models/data

Figure 4: Model-driven method for the simulation-based analysis of evacuation plans

5.1 Graph-to-EQN model-to-model transformation

The `graph-to-EQN` model transformation yields as output an EQN simulation model from a graph-based building design model taken as input. Figure 5 depicts the basic structure of the output EQN model, which includes the following elements: a source, a fork, a router (implicitly shown in figure) and a join node. The source injects a single job in the system. The job is automatically routed to the fork node, which instantiates the initial building population and includes a delay center to simulate the evacuation reaction time. Then the jobs are spread over the plan model by the router, which directs them towards the configured access points. All the jobs traverse the network and are collected by the join node. This node waits for the arrival of all jobs before triggering a new job into the fork node, thus starting another experiment.

The topology of the EQN portion denoted as *System Model*, which identifies the variable portion of the EQN, depends on the structure of the given input building model and is generated according to the transformation rules discussed in this section.

The rationale of the transformation takes into consideration two different issues: the structural properties of the EQN, which describe the network topology (e.g., the set of deployed EQN elements and their interconnections), and the parameterization, which concerns the definition of the several model parameters affecting the behavior of jobs during the simulation (e.g., queueing policies, service times, routing probabilities, etc.).

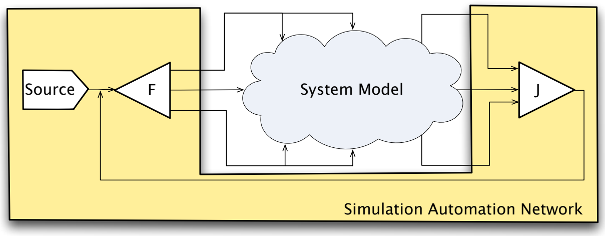

Figure 5: Structure of the EQN model for the evacuation analysis

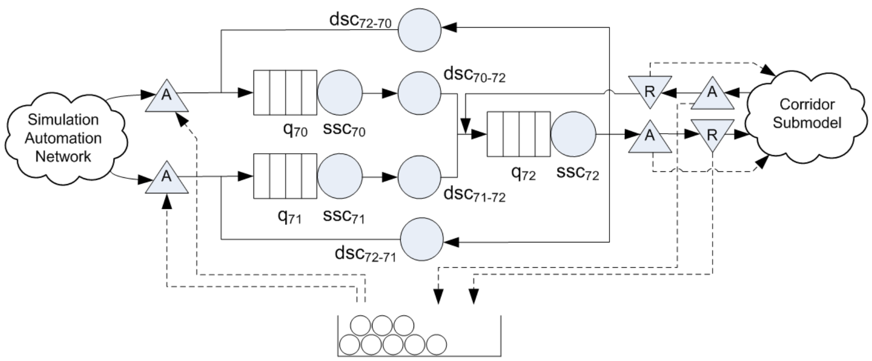

Figure 6: Detailed view of the EQN simulation model (portion)

5.1.1 Definition of the EQN topology

The procedure for the derivation of the EQN topology from the structural specification of the building model can be informally summarized as follows:

1. Definition of the following EQN elements:
   (a) a single server for each graph node;
   (b) a queue for each graph node;
   (c) two delay centers for each graph edge (both directions);
   (d) a passive queue system (Token Pool, Allocate and Release Nodes) for each access way to each group of nodes.
2. Definition of the following connections:
   (a) between queue and single server of each graph node;
   (b) between single server and delay centre following the graph structure;
   (c) between Allocate Nodes at the entry point of any group of nodes;
   (d) between Release Nodes at the exit point of any group of nodes;
   (e) between Allocate node, Pool of Tokens and Release node, which model the same group of nodes.

In addition, the simulation model includes the definition of the EQN elements for the simulation automation, and of their connections to the system model. These elements are shown in Figure 1 and their connections are established according to the possible initial configurations, i.e., the placement of the civilians over the floor model.

As an example, Figure 6 illustrates the EQN structural model for the room that contains nodes 70, 71 and 72, as shown in Figure 1.

5.1.2 Parameterization of the EQN model

The procedure for the derivation of the EQN parameters from the behavioral model specification can be informally summarized with the following steps:

1. identify all the categories of civilians
   (a) define a class of jobs for each category and calculate:
      i. queueing parameters at each single service center;
      ii. service time distributions at single service and delay centers;
      iii. routing parameters at each node.

where all the parameters of points i), ii) and iii) have to be calculated by applying a set of transformation rules that consider only a unique category of civilians, which in turn reflects in a unique class of jobs in the EQN model. For this class, the parameters identified at point i) can be determined by analyzing the specification with regard to the general civilian behavior in the simulated scenario. As aforementioned, the scenario does not consider panic or real emergency and therefore an orderly queueing at each node can be reasonably assumed. In addition, the presence of a single class makes it impossible the definition of priority-based queueing, such as multi-queueing. Finally, the type of service is non-preemptive since the interruption of the service for a higher priority job is not coherent with the physical behavior of the civilians in the modeled scenario.

Parameters identified in (ii) are the service times of service and delay centers. These can be directly retrieved from the formal specification of the motion model and defined as t_{n_i} and $t_{e_{ij}}$ respectively.

Finally, parameters identified in (iii) can be derived from the description of the decision model. In particular, the routing policies and probabilities must be determined by applying the formal description of the model to each of the nodes. For the considered shortest distance model, probabilities p_{ij} are obtained as:

$$p_{ij} = \begin{cases} 0 & for\,j : \exists k : n_k \in N_{exit} \wedge (d_k(n_j) + e_{ij}.weight) > \underset{n_i;n_l \in E_j, n_k \in N_{exit}}{min} \{d_k(n_l) + e_{il}.weight\} \\ 1 & for\,j : \underset{n_k \in N_{exit}}{min} \{d_k(n_j) + e_{ij}.weight\} = \underset{n_i;n_l \in E_j, n_k \in N_{exit}}{min} \{d_k(n_l) + e_{il}.weight\} \end{cases}$$

which can be read as follows: the case probability 1 is characterized by having the shortest distance (i.e., minimum of the distance function) to the closest exit. Differently, the probability is 0 when the calculated distance is not the minimum one.

5.2 EQN-to-jQN model-to-text transformation

The `EQN-to-jEQN` model-to-text transformation is executed to generate the Java-based jEQN code that implements the EQN model given as input. Such a transformation is reused from previous works and thus it is not discussed here. Interested readers are sent to (Bocciarelli et al. 2012) for the relevant details.

6 EXAMPLE APPLICATION AND VALIDATION ISSUES

The validation of emergency simulators is hardly possible with direct comparison of data from the real world, because emergency metrics for a specific building, such as the total or average individual evacuation time, often do not exist until some disasters occur. And in such cases collecting statistics is not a priority. Aside from this, this paper specifically addresses the generation of the simulation code from the informal model specification with the assumptions of the correctness and the accuracy of the modeling approach. For these reasons, the validation of the above simulators does not result to be fundamental in supporting the paper thesis. However, a preliminary validation of the proposed method has been carried out through a simplified analytical model and through comparison with analogous simulators (Gianni et al. 2008, Filippoupolitis et al. 2012). A simple analytical formula for the total evacuation time (i.e., the evacuation time of the last evacuees) can be derived for the above graph model by assuming that the queueing probability is very low (i.e. queueing time negligible), assumption reasonable for the scenario of fifty evacuees. In such conditions, with the initial uniform distribution of the evacuees, the following formula holds:

$$\bar{t}_{evac} = \bar{t}_{Node} \cdot \overline{nhops} + \frac{\sum_{n \in N} \frac{\underset{k \in OrdEx}{min} d(k)}{\#N}}{\overline{speed}}$$

which can be read as follows: the average evacuation total evacuation time is the sum of the average time spent on the nodes and the average time spent on the edges. The former factor can be easily computed as the product of the average time on a single node, which is a part of the model input data and is 0.3 s, multiplied by the average number of hops, which for the above graph is 4.4. The latter factor corresponds to the average distance from the closest exit divided by the average speed. The distance can be calculated as the sum of the shortest path lengths divided by the number of nodes where the civilians can initially be placed, and the speed is 150 cm/s. This distance can be calculated in about 1934 cm. Using the model data in this formula, the average total evacuation time is about 14.2 s. In the same scenario, with an initial population of fifty civilians, a run of 10000 simulation seconds gives an average total evacuation time of about 14.94 s, which is within reasonable error margin with respect to the analytical estimation. In addition, these data are also congruent with the agent-based simulation in (Gianni et al. 2008, Filippoupolitis et al. 2012), which for the identical scenario (area plan, motion and decision model) gives an average total evacuation time comparable with the obtained result.

7 CONCLUSIONS

Building design might benefit from an integrated approach in which model-driven and simulation techniques can be conveniently used to support the evaluation of design alternatives for safety requirements. This paper has introduced a model-driven method to automatically transform an informal model specification into a building evacuation model based on the EQN formalism. The method has been illustrated by use of an example building evacuation scenario consisting of one floor plan and an evacuee behavioral model. The method assumes the plan can be represented as a graph and then transformed into an EQN model topology, while the model parameters are obtained from the behavioral model. Once the model is obtained, the method executes an additional transformation to get to the jEQN-based simulation implementation. The paper has also presented a preliminary validation through comparison with simplified analytical methods and simulators of similar scenarios.

REFERENCES

Bocciarelli, P., and A. D'Ambrogio. 2014. "Model-driven method to enable simulation-based analysis of complex systems". *Modeling and Simulation-Based Systems Engineering Handbook* 3:119.

Bocciarelli, P., A. Pieroni, D. Gianni, and A. D'Ambrogio. 2012. "A model-driven method for building distributed simulation systems from business process models". In *Winter Simulation Conference*, edited by O. Rose and A. M. Uhrmacher, 227: WSC.

Bolch, G., S. Greiner, H. de Meer, and K. S. Trivedi. 2006. *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons.

British Standard Institute 1997. "Fire Safety Engineering in Buildings Part 1: Guide to the Application of Fire Safety in Engineering Principles".

D'Ambrogio, A., D. Gianni, and G. Iazeolla. 2006. "jEQN a Java-based Language for the Distributed Simulation of Queueing Networks". In *Proceedings of the 21st International Conference on Computer and Information Sciences*, ISCIS'06, 854–865. Berlin, Heidelberg: Springer-Verlag.

Dimakis, N., A. Filippoupolitis, and E. Gelenbe. 2010. "Distributed building evacuation simulator for smart emergency management". *The Computer Journal*:bxq012.

Filippoupolitis, A., and E. Gelenbe. 2009. "A distributed decision support system for building evacuation". In *Human System Interactions, 2009. HSI'09. 2nd Conference on*, 323–330. IEEE.

Filippoupolitis, A., E. Gelenbe, D. Gianni, L. Hey, G. Loukas, and S. Timotheou. 2012. "Distributed agent-based building evacuation simulator".

Gianni, D., and A. D'Ambrogio. 2007. "A language to enable distributed simulation of extended queueing networks". *Journal of Computers* 2 (4): 76–86.

Gianni, D., A. D'Ambrogio, and G. Iazeolla. 2011. "A software architecture to ease the development of distributed simulation systems". *Simulation* 87 (9): 819–836.

Gianni, D., G. Loukas, E. Gelenbe et al. 2008. "A simulation framework for the investigation of adaptive behaviours in largely populated building evacuation scenarios". In *Organised Adaptation in Multi-Agent Systems workshop at the 7th International Conference on Autonomous Agents and Multiagent Systems*.

Fire Safety Engineering Group 2003. "Exodus Introduction". http://fseg.gre.ac.uk/exodus/.

OMG 2003. "MDA Guide, version 1.0.1.". http://www.omg.org/mda/.

OMG 2010. "System Modeling Language, v.1.2". http://www.omg.org/spec/SysML/1.2/.

OMG 2011. "Unified Modeling Language, v.2.4.1". http://www.omg.org/spec/UML/.

Pan, X., C. S. Han, K. Dauber, and K. H. Law. 2006. "Human and social behavior in computational modeling and analysis of egress". *Automation in construction* 15 (4): 448–461.

Pursals, S. C., and F. G. Garzón. 2009, January. "Optimal building evacuation time considering evacuation routes". *European Journal of Operational Research* 192 (2): 692–699.

Santos, G., and B. E. Aguirre. 2004. "A critical review of emergency evacuation simulation models".

Sayed Ahmed, A., and G. A. Wainer. 2010, April. "Integrating Building Information Modeling & Cell-DEVS Simulation". In *Proceedings of 2010 Spring Simulation Conference (SpringSim10), SimAUD Symposium*, edited by M. S. Ahmed sayed Ahmed, Gabriel A. wainer, 45–52: SCS.

Smith, J., and D. Towsley. 1981. "The use of queuing networks in the evaluation of egress from buildings". *Environment and Planning B: Planning and Design* 8 (2): 125–139.

Wang, S., M. Van Schyndel, G. A. Wainer, V. Subashini Rajus, and R. Woodbury. 2012, Dec. "DEVS-based Building Information Modeling and Simulation for Emergency Evacuation". In *Proceedings of Winter Simulation Conference*, edited by IEEE, 1–12. Berlin, Germany.

Wang, S., G. A. Wainer, R. Goldstein, and A. Khan. 2013, April. "Solutions for Scalability in Building Information Modeling and Simulation-Based Design". In *Proceedings of the SCS/ACM Symposium on Simulation for Architecture and Urban Design*. San Diego, CA: SCS/ACM.

Wang, S., G. A. Wainer, V. Subashini Rajus, and R. Woodbury. 2013, April. "Occupancy Analysis using Building Information Modeling and Cell-DEVS Simulation". In *Proceedings of 2013 SCS/ACM/IEEE Symposium on Theory of Modeling and Simulation, TMS/DEVS13*. San Diego, CA: Society for Computer Simulation International.

Zheng, X., T. Zhong, and M. Liu. 2009. "Modeling crowd evacuation of a building based on seven methodological approaches". *Building and Environment* 44 (3): 437–445.

AUTHOR BIOGRAPHIES

DANIELE GIANNI is a requirements engineering consultant at EUMETSAT and assistant professor at the "Guglielmo Marconi" University (Italy). His research interests are in the areas of systems engineering, model-driven engineering and simulation. His email address is d.gianni@unimarconi.it.

PAOLO BOCCIARELLI is a postdoc researcher at the University of Rome "Tor Vergata" (Italy). His research interests include software and systems engineering, business process management and model-driven development. His email address is paolo.bocciarelli@uniroma2.it.

ANDREA D'AMBROGIO is associate professor at the University of Roma "Tor Vergata" (Italy). His research interests are in the fields of model-driven software engineering, dependability engineering, distributed and web-based simulation. His email address is dambro@uniroma2.it.

GIUSEPPE IAZEOLLA is full professor at the "Guglielmo Marconi" University, Roma (Italy). His research is in the areas of systems performance and reliability modeling and in the areas of parallel and distributed simulation. His email address is g.iazeolla@unimarconi.it.