A Framework for A Graph- and Queuing System-Based Pedestrian Simulation

Srihari Narasimhan
IPVS
Universität Stuttgart
Stuttgart, Germany

Hans-Joachim Bungartz
Institut für Informatik
Technische Universität München
Garching bei München, Germany

Abstract - *With a growing trend in usage of mobile navigation devices, there is also a need for an intelligent pedestrian navigation system. As an example, one might think of a visit to a large commercial center or a theme park, where the visitor must be efficiently guided to accomplish his tasks. In order to efficiently schedule a visit in such a scenario, the system not only needs to simulate and determine the waiting times across each destination, but also to integrate the entire system into the geometry of the scenario, to provide spatial context to non-spatial data. In this paper, we present a framework for a pedestrian simulation that models the movement of pedestrians by embedding the simulation into a geometric context. The resulting model considers the dimensions and architectural constraints of the scenario, simulates the pedestrian movement over a 3D graph network and gives an overall view of the congestions along the paths, waiting times at the destinations, etc.*

Keywords: discrete-event simulation, queuing systems, pedestrian simulation, graph algorithms

1 Introduction

A typical activity of a pedestrian involves finding his way from his current position, through a path to his destination and getting his job done at the destination. During this process, the pedestrian not only interacts with other pedestrians along his way, but also with the layout and geometry of the environment that surrounds him. It is therefore important to consider the geometric parameters when modeling the behavior of pedestrians. In this paper, we present an approach to model the pedestrian behavior in a 3D space such as a building by representing the movements as queuing systems and embedding the pedestrian simulation model into a geometric scenario. The whole idea is to extract architectural information from the geometry such as the dimensions of pathways, capacity of the rooms, exact location of the destinations, etc., and to integrate the queuing system-based simulation model into this data. As a result, the exact movement of pedestrians along with the congestions and waiting times can be recorded across the entire scenario.

Before explaining the core implementation part, this paper starts with introducing how queuing systems are used to model pedestrian movements and how they are embedded into a geometric context. Since the pedestrian behavior is analyzed here, the stochastics on pedestrian behavior are discussed together with the modeling and simulation part. Following that, an analysis is made with the sample data that is collected, and the performance of the simulation is studied. The paper concludes with an outlook into the possible use of this kind of approach and further directions of research.

2 Queuing System-Based Modeling of Pedestrian Flow

Queues are present almost everywhere, right from supermarkets and cafeterias to trains entering and departing the railway station to even in data that is sent over the network. In our scenario, the main target are pedestrians (hereafter also called entities), who execute their tasks at their destination. Since it is usual that there are more entities wishing to execute a specific task and not so many service counters are available to serve these entities, a queue is formed. The queue is modeled using the discrete-event simulation method where the arrival of entities into the queue, the service and the departure of entities are recorded as events. Similarly, the movement of an entity along the path can also be thought of as a queue [1] where the entities arrive into a path, wait if there are congestions and transit the path. The fundamental parameters needed for modeling a queuing system are the arrival time and service time. In case of modeling a queue along the path, the arrival rate is exponentially distributed. The arrival time to each path is influenced by the departure from the previous paths and the service time is the walking speed of the pedestrian which is determined based on the pedestrian properties and the actual path situation. Several statistics are available (for example as in [2]), which can be used to derive a precise time needed to walk across the path. A stochastic function models the walking speed using these statistics and by considering the actual path situation, the walking speed of a specific pedestrian is generated. In case of modeling a queue at the destination, the arrival time is clearly the time when the entity leaves its path to enter the destination and the service time depends on the activity of the entity. The service time in this case is generated based on the type of the destination and the profile of the customer. A histogram model is used here. Other parameters include the capacity of the queue and the availability of service counters which are determined from the architectural model.

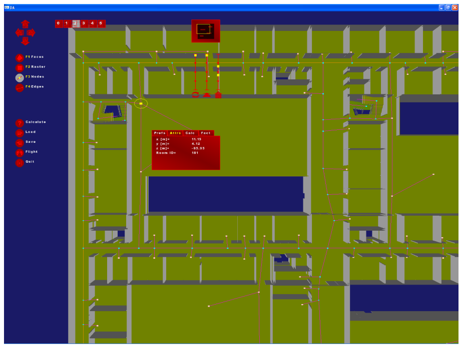
3 Geometry Embedding of Queuing Systems

Now that we use the same queuing model for modeling the paths and the service areas, we therefore need to create several queuing systems for each possible path and each destination, each with different properties and parameters suiting the paths and rooms. For this purpose, the CAD model of the building is parsed through using the Pathscan [3] program to automatically derive a graph of paths and to define the destinations across the building. Pathscan is a flexible tool that reads a CAD model of an architectural building, generates slices of the model in upright direction, identifies all floors where a pedestrian can move and exports the paths on to a graph. The tool also allows the user to define rooms and their properties and to make changes to the final graph, too. Once all adjustments are made, the tool exports the data into an XML file that can be easily parsed into the simulation code. The resulting data shows the co-ordinates of each node, the edges connecting them, the type of edges (staircase, pathway, private, etc.), the rooms, and the nodes that lie in the rooms. Figure 1 depicts a snapshot of the output created by the Pathscan tool that is used to define the graph and the rooms.

For test purposes, the existing CAD-Model of the computer science building at the Universität Stuttgart [4] is used. The Pathscan tool is used here to extract the graphs and to define hypothetical destinations.

4 Modeling and Simulation

As mentioned before, the pedestrian movement is modeled as a queuing system. The major key that drives the discrete-event simulation is the stochastic function that acts as the source of input to the



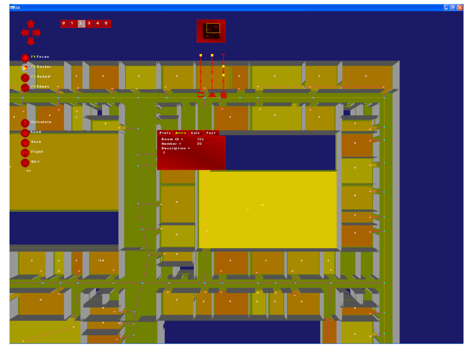


Figure 1: Pathscan tool used to create and modify the graph (left), and to define the rooms and their specifications (right).

model. In modeling a queuing system, the two most important input values needed are the arrival rate and the service time. The arrival rate in this case is chosen to be an exponentially distributed value. The service time along the paths depends on the walking speed of the entity. The service time at the destination depends on the task executed by the entity which is in turn determined from the type of the destination and the customer profile. In the case of our pedestrian simulation, the geometry data is more or less static, meaning not many stochastics are involved, whereas it is the customer data that drives the simulation. Since the stochastic functions need information about the customer and also the destination type, different profiles are defined for the same. All profiles are defined using the XML format, which makes it easy to translate available statistical data into input profiles needed for our simulation.

4.1 Customer Profile

Typically, it can be noted that each customer behaves differently. Some of them are active visitors such as people who work in the scenario or who tend to stay there for most of the time. Other customers just make short visits. In order to distinguish these types of customers, the duration of average stay is defined in the profile. Furthermore, each customer has a specific list of destinations to visit and each destination type could be visited more than once. It must also be noted that certain restrictions may apply to the destination types such that not all customers are allowed to visit that particular destination type. For example, a temporary visitor may not be allowed to enter a high security zone or a private office. Apart from that, the choice of destinations to visit depends on different times during the day: a restaurant is more crowded during noon than any other time of the day. The resulting profile includes

- duration of stay,
- choice of destination types and repeated visits to the respective types,
- distribution of visits to the destinations during different times of the day,
- and distribution of different customer types visiting the scenario.

Since our intention to build a framework for a pedestrian simulation, the customer profile is filled up with hypothetical values. Due to the flexibility of the profile, one can also study the bottlenecks and waiting times for several scenarios by just changing the profile values.

4.2 Geometry Layout

As mentioned above, the geometry data is more or less static. The only stochastic parameter considered within the geometry layout are the service times that are attached to different destination types. The average service time of each destination type is defined in the destination profile. The service time of a customer at the destination is derived from this profile.

4.3 Implementation

Apart from the above mentioned profiles, parameters such as maximum number of customers, termination conditions, etc., are defined to control the simulation. As in a discrete-event simulation, the queuing system is modeled as arrival, service and departure events. Once the simulation starts and a new arrival is scheduled, the arrival event appends the entity into the queue and schedules the next arrival. The service event determines if there are available resources and schedules the departure event after the entity is served. The departure event removes the entity from the queuing system, and if necessary, schedules the entity into the next queuing system. The discrete-event simulation always scans the next scheduled events and executes them. The most important aspect is the scheduling of events. The events are scheduled by using a "three-phase" approach or the "ABC" approach as explained in [5]. The current location of each entity is continuously monitored and this data helps in deciding the next activity of the entity.

4.4 Sample Data Collection and Analysis

Several simulation runs are made for different parameter settings such as the number of customers who visit the building, different customer profiles, changes in the capacity at the destination, etc., and the output data are collected for each simulation run.

A short study of the sample data collected is made for the following scenario:

- About 3000 customers visit the building, each with a specific list of tasks to execute.
- Utmost 1000 customers are present in the building at any given point of time.
- A restaurant, with a capacity for 50 people, is chosen for analysis, and simulations are run to study its occupation during different times of the day.
- The probability distributions in the customer profile suggest that the probability of a customer visiting the restaurant is highest at noon, followed by the time frame between 9am and 10am and then followed by the time frame between 3pm and 4pm.
- The simulation is run for a period of 8 hours (9am to 5pm).

The plot in Fig. 2 shows the number of customers (customers already present in the restaurant plus the customers waiting in the queue) during different times of the day.

From the plot, the maximum number of customers present in the restaurant can be determined. In this case, the number never exceeds the maximum capacity at any point of time. This implies that there is no queue formed and the customers gain immediate entry as soon as they reach the restaurant.

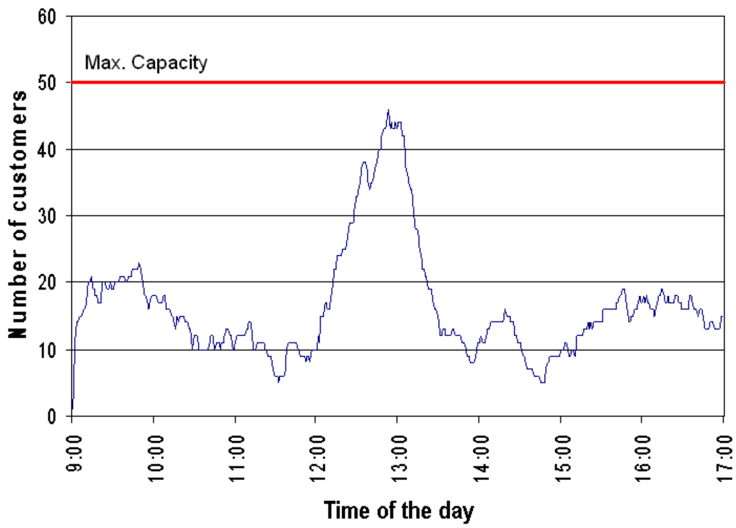


Figure 2: Number of customers recorded recorded within our simulation throughout a day for a sample restaurant with a capacity for 50 customers.

5 Performance Analysis of the Simulation Framework

The simulation framework is intended to be highly flexible. The idea is to accept a given geometry and a set of customer profiles, simulate the pedestrian model and collect data relating to the congestions that occur across the model and the waiting times at different destinations. This facilitates a full fledged pedestrian simulation in a 3D scenario to study the bottlenecks that occur in the architecture.

Since the pedestrian simulation is modeled using a discrete-event simulation method, the simulation proceeds only on the occurrence of a new event. The events are scheduled in the increasing order of their occurrence. The simulation calculates the time these events occur and list them out. Therefore, the number of computations performed is relatively low even for a large model and the speed of execution is rather high. The accuracy of the results depends on the quality of input modeling. In this scenario, a model simulated for about 3000 customers visiting the scenario with a graph size of about 600 nodes and edges each for a time period of 8 hours causes about 761,000 events to be activated. The simulation of such a model, without any output analysis, takes about 45 seconds to execute on a Pentium 4 processor (3 GHz). The memory utilization strongly depends on the customer profile and the tasks a customer wishes to execute. In this scenario, about 20 MB of main memory was needed to compute the movement of entities along the graph.

This framework was also tested for different problem sizes. However, there are memory and CPU restrictions involved in very large models. Handling of these issues are not discussed in this paper. The largest model simulated within this framework was for about 115,000 visitors, for a time period of 8 hours. This simulation takes about 31 minutes for computation and it needs almost 2 GB of main memory.

6 Using the Simulation Data

As described in the previous section, the data collected is used to study the bottlenecks and waiting times of the underlying architecture at different times of a day. Now, assume that a new customer arrives with a specific list of tasks to execute. Using the simulation data, it is possible to prepare an efficient itinerary for the customer by optimally scheduling the tasks to be executed – either a priori or in real time with the use of some mobile device. For example, as in the scenario presented in section 4.4, when scheduling a visit to the restaurant, the customer need not wait at the restaurant, irrespective of his arrival time. Over a period of time, more and more customers receive an itinerary and these customers, in turn, cause a change in the existing congestions and waiting times. Therefore, necessary changes must be made in the simulation model to obtain new sets of data. Apart from that, it can be expected that a customer, with a mobile navigation device in hand, would also wish to execute certain tasks. With the help of location-awareness [6], it is possible to dynamically prepare an itinerary for such customers. The scheduling of tasks based on simulation data is a grand challenge and is beyond the scope of this paper, hence not dealt with here.

Furthermore, such a simulation framework can prove helpful during the architectural design process. Various scenarios can be simulated for the given architectural model to identify the possible bottlenecks in the model. By making iterative simulation runs, the plan of the architectural model can be optimized to suit a specific scenario. Since the movement of pedestrians are also identified here, informations such as where to place a shop for a maximum number of visitors, where to place emergency services, etc., can be determined. Evacuation of pedestrians is another topic that can be addressed using this framework. Since the simulation is embedded into the geometry, an evacuation scenario can be simulated to identify the congestions in the architectural model. A similar approach to detect bottlenecks in the architectural model by integrating evacuation planning is proposed in [7].

7 Conclusion and Outlook

The key aspect of this paper is the embedding of queuing systems into a geometry structure which gives rise to a new approach to pedestrian simulation. This approach closes the gap between pedestrian simulation tasks with a traditionally non-geometric background and the world of geometric modeling. The resulting simulation framework can model pedestrian behavior directly in a geometric scenario. One of the major objectives is to use this framework to identify the congestions and waiting times in a given scenario and with the resulting data, prepare an optimal schedule for a visit of a new customer. As more and more customers benefit from this scheduling, these customers in turn influence the status of the congestions and waiting times. Therefore, the simulation data needs to be constantly updated to improve the precision of the scheduling. The approach proposed in this paper provides a strong infrastructure to build such an intelligent pedestrian navigation system. Another interesting aspect to be considered, is the scalability of this approach. Work is also being done in simulating larger models within this framework. By combining the geometric models of several buildings and the pedestrian movement in each building, pedestrian simulation can be performed for a larger region. Furthermore, the pedestrian simulation model can be coupled together with the traditional traffic simulation, thereby provide an effective navigation system, spanning across many cities.

Acknowledgments

Special thanks to Ralf-Peter Mundani for his friendly review and valuable suggestions to this paper.

References

[1] Srihari Narasimhan and Hans-Joachim Bungartz, *Congestion-Aware Optimization of Pedestrian Paths*, In proceedings of the 18th ASIM Symposium in Simulation Technique, Erlangen, Germany, 2005.

[2] Kardi Teknomo, *Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model*, Ph.D. Dissertation, Tohoku University, March 2002.

[3] Thomas Drexl, *Development of Intelligent Shortest-path Algorithms for Architectural Models Regarding an Info Point System for the FMI in Garching*, Diploma Thesis, Technische Universität München, 2003.

[4] Stefan Giesecke, Matthias Stier and Sven Grumbein, *Path-Search in Architectural Models and Stereo Projection*, Softwarepraktikum, Universität Stuttgart, 2004.

[5] Jerzy Tyser, *Object-Oriented Computer Simulation of Discrete-Event Systems*, Kluwer Academic Publishers, 1999.

[6] Srihari Narasimhan, Ralf-Peter Mundani and Hans-Joachim Bungartz, *An Octree- and A Graph-Based Approach to Support Location Aware Navigation Services*, Accepted for the 2006 International Conference on Pervasive Systems and Computing (PSC '06), Las Vegas, USA, 2006.

[7] Ralf-Peter Mundani, Hans-Joachim Bungartz and Stefan Giesecke, *Integrating Evacuation Planning into an Octree-Based CSCW Framework for Structural Engineering*, In Proc. of the 22nd CIB-W78 Conference on Information Technology in Construction, Dresden, Germany, 2005.