A MESOSCOPIC APPROACH TO MODELING AND SIMULATION OF PEDESTRIAN TRAFFIC FLOWS

Juri Tolujew and Felix Alcalá
Fraunhofer Institute for Factory Operation and Automation
Sandtorstr. 22, 39106 Magdeburg, Germany
Juri.Tolujew@iff.fraunhofer.de

KEYWORDS
Pedestrian traffic flows, mesoscopic simulation, online simulation, early-warning system

ABSTRACT

The paper discusses the simulation of pedestrian traffic flows in public buildings from a practical point of view: The models developed are intended for use in an operative early-warning system. They must be able to forecast how the situation will evolve in each room of the building for the near future. By evaluating possible measures they can be used as a tool that pro-actively avoids critical system states. A mesoscopic approach to creating these models is described and partially realized. It envisions groups of persons as modeling and simulation objects and thereby distinguishes between physical and logical (virtual) groups. The paper presents novel solutions to the modeling of both types of person groups. An example of a model described in XML and implemented with Java is demonstrated.

INTRODUCTION

Models for pedestrian traffic flows are often categorized by the type of moving objects used to represent the persons to be simulated. On the one hand there is *macroscopic simulation* used to model relatively large numbers of persons that are distributed in an abstract space. Analogous to gases and liquids, physical laws and therefore differential equations can describe their movement (Helbing 1997). In *microscopic simulation* on the other hand every moving object represents exactly one person. Thus even the smallest geometric detail of their movement can be considered. The most important members of the category of microscopic models are models based on cellular automata (Schreckenberg and Sharma 2002) and queuing systems in the form of analytical models or simulation models (Løvås 1994).

The authors of this paper deal with the simulation of pedestrian traffic flows in conjunction with the development of a concept for an *operative early-warning system*. This system monitors and controls pedestrian traffic flows in a public building which is a traffic junction for large quantities of travelers (Hanisch et al. 2003). Since the early-warning system operates *online*, the model must have specific properties:

- The forecast interval is approximately one hour.
- The results of the simulation contain information on the distribution of the *amounts of persons* in predefined rooms of the building.
- The model is started at regular intervals, usually around 5 minutes.
- Since it is based on *direct measurements* of pedestrian traffic flows in each room, the input data is an exact image of the current situation in the building.
- All *real and planned events* are considered in every simulation run. This includes the arrival and departure of passengers as well as actions of the personnel controlling the pedestrian traffic flows.

Critical analysis of the aforementioned categories concluded that none of them can be used for the development of models for an early-warning system. Macroscopic models based on differential equations seem to be too imprecise in this situation since normally they can only be used to describe the average, collective behavior of persons. The two types of microscopic models have a downside of relatively long computing times. Additionally, their level of detail does not match the level of detail at which processes in rooms of a building can be observed and described.

A CLASS OF MESOSCOPIC MODELS

The main idea behind the approach presented in this paper is that *person groups* are regarded as modeling and simulation objects. This approach can be called *mesoscopic modeling and simulation* because it is situated between the microscopic and the macroscopic methodology (Lerner et al. 2000). In this class of modeling, two kinds of person groups can be distinguished: *physical groups* (also real groups) correspond to person quantities actually in particular rooms. Adequate sensors can estimate the number of persons in a physical group and this number is used as input data for the simulation. *Virtual* or *logical groups* represent person groups whose members show similar behavior because they have the same intention (e.g. to board a specific train). The members of a logical group are normally divided among several physical groups. Consequently, every physical group typically consists of members of several logical groups.

Proceedings 18th European Simulation Multiconference
Graham Horton (c) SCS Europe, 2004
ISBN 3-936150-35-4 (book) / ISBN 3-936150-36-2 (CD)

The simulation uses a discrete time with a step size of Δt, which in principle can also be variable. Two models have to be calculated in every time step, one for each kind of groups. Position changes of persons are mapped in the model for physical groups in such a way that new physical groups are formed at the end of each time step Δt. Individual persons' behavior such as choosing a new path is calculated in the model for logical groups and is statistically correlated with reality. The model of logical groups can be updated at the end of time step Δt if a new group is needed or if an existing group can be discarded because it has no more members.

IMPLEMENTATION OF MODELS FOR PEDESTRIAN TRAFFIC FLOWS

Models based on differential equations are usually developed by applying simulators for continuous processes (e.g. MATLAB/Simulink). Any commercial simulator (e.g. Enterprise Dynamics or eM-Plant) is suitable for executing microscopic simulations in the form of queuing systems. Special tools like JCASim are used for microscopic models based on cellular automata.

None of the features of these universal or special simulators constitutes an advantage when working with the category of mesoscopic models presented here. In fact, mesoscopic simulation requires the development of very specific algorithms. They are best implemented using a universal programming language, which leads to relatively high model performance, something extremely important for online simulations.

As discussed below, another aspect of the development of mesoscopic models is the necessary flexibility of the model. This can be achieved by following the principle of data-based model creation and using XML as the modeling language. Among other things, this paper discusses the results of a data-based, Java-based implementation of a mesoscopic model for physical person groups based on the principle of dynamic route segmentation.

EVENT-BASED GENERATION AND MODELING OF LOGICAL GROUPS

A simple example is treated, in which one physical group of persons (passengers on a platform) consists of a simple sum of several logical person groups since all persons to be modeled are located in one room. The arrival and departure of two trains is represented by four events shown in table 1.

Table 1: Description of Four Main Events

Event name	Event time	Event description
Ev1	13 min	arrival of train 1
Ev2	20 min	departure of train 1
Ev3	23 min	arrival of train 2
Ev4	31 min	departure of train 2

In systems where the inflow and outflow of person quantities are caused by means of transportation, most logical groups are formed by concrete events. Table 2, figures 1 and 2 show the eight processes that can be inferred from the events of table 1 (Pax is an abbreviation for the word "passengers"). Groups of departing persons are set up and populated before the corresponding events take place (processes Pr1 and Pr5). Likewise, every new group of arriving persons appears immediately after the occurrence of the corresponding event (processes Pr2 and Pr6).

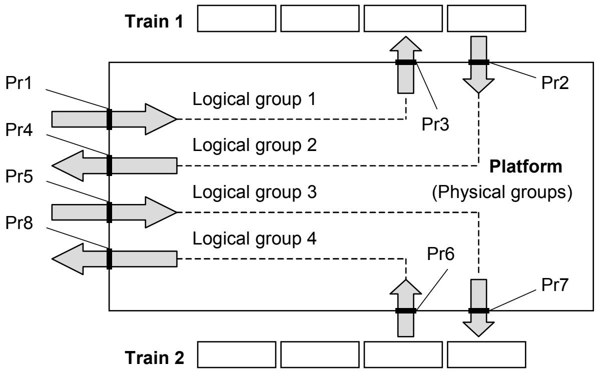

Figure 1: Observation Points for Eight Processes on a Platform

The model of behavior shown in figure 2 for four logical groups is simple but not trivial. For example, it takes into account that while a part of the group "boarding train 1" is already inside the wagon, the rest of this group may still be on its way to the train (the processes Pr1 and Pr3 overlap for two minutes). Moreover, the model contains 20 persons who change trains: 100 persons arrive with train 1 (process Pr2) but only 80 of them leave the platform (process Pr4) because the remaining 20 persons are waiting for train 2. The length of the platform is assumed to be such that the first person who has left a particular train can leave the platform after a three minute walk (offset between the beginnings of Pr2 and Pr4).

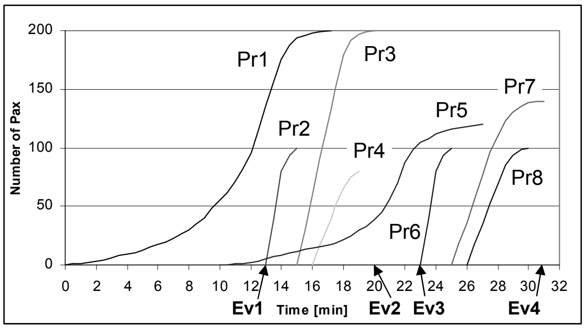

Figure 2: Cumulative Pax Quantities for Eight Processes on a Platform

Table 2: Description of Eight Processes on the Platform

Process name	Start time	End time	Number of pax	Flow direction	Process description
Pr1	0	17	200	input	The pax are coming onto the platform to board train 1
Pr2	13	15	100	input	The pax are deboarding train 1
Pr3	15	20	200	output	The pax are boarding train 1
Pr4	16	19	80	output	The pax who have arrived with train 1 are leaving the platform
Pr5	10	27	120	input	The pax are coming onto the platform to board train 2
Pr6	23	25	100	input	The pax are deboarding train 2
Pr7	25	31	140	output	The pax are boarding train 2
Pr8	26	30	100	output	The pax who have arrived with train 2 are leaving the platform

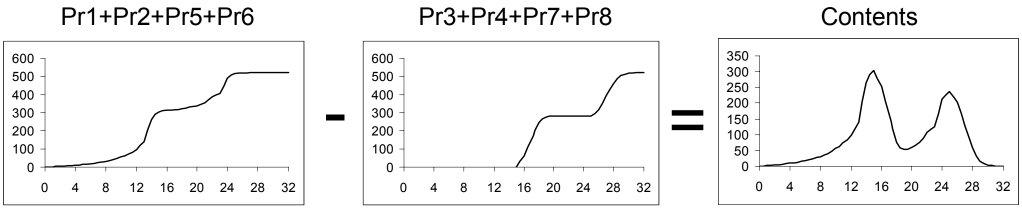

Figure 3: Calculation of the Entire Pax Quantity on the Platform

The characteristic of the process description in figure 2 is the use of cumulative person quantities as indicators. It has been demonstrated in (Tolujew 2003) that these dynamic indicators can be used as operands in arithmetic and algebraic operations. Figure 3 illustrates the last operation, which calculates the number of persons on the platform during the entire duration of the process. The four input and output processes were accumulated beforehand. The dynamic of the person quantity on the platform shown in the diagram “contents” is a precise result directly inferred from the specified processes Pr1 – Pr8.

MESOSCOPIC SIMULATION OF THE MOVEMENT OF PERSON GROUPS

The mesoscopic model consists of four basic components shown in table 3. They can be linked freely, i.e. each component can be reached from and may lead to any other component. Components can also be linked to themselves.

It is assumed that, in one time step, every component is capable of accepting any input quantity. The movement of persons is modeled using the component passageway, a unidirectional connection between two points in the real room. Using passageways as a model for pedestrian movement presupposes three assumptions:

- The time progresses in equidistant time intervals Δt.
- The situation depicted in the passageway is the one at the *end* of each time interval.
- A passageway is composed of several parallel “virtual tracks”, each of which is assigned a specific velocity. Thus, any velocity distribution can be approximated by defining several tracks with different velocities. The number of tracks depends on the model precision required.

Table 3: Basic Components of the Mesoscopic Model

Component class	Component function	Examples
Source	Fills the model with persons	An arriving train; Main entrance of the building
Sink	Removes persons from the model	A departing train; Door to parking lot
Service station	Delays persons	Ticket counters; Security checkpoints
Passageway	Models distances between components. Persons are delayed until they have covered the distance.	Transfer from platform A to platform B

A track is characterized by the length l of the passageway and the specific velocity v of the persons on this track. All persons on the track advance a distance of $v{\cdot}\Delta t$ during the time Δt. This distance defines the length of the so-called segments, into which every track is divided. The number of segments in a track is $m = l/(v{\cdot}\Delta t)$; the result of the division is rounded up to the next integer. As a result, the total length of all m segments ($m{\cdot}v{\cdot}\Delta t$) may be longer than the length of the passageway *l*. In this case, a part of the last segment (segment number *m*) is outside the track.

Figure 4 shows an example of a track with three segments (*m*=3). Additionally, a virtual forth segment is defined (segment number *m+1*) to calculate the output quantity of this track at each time interval. The terms s1,…,s4 are used to name the length of each segment. The terms n1,...,n3 represent the number of persons in the respective segment at the end of the time step Δt.

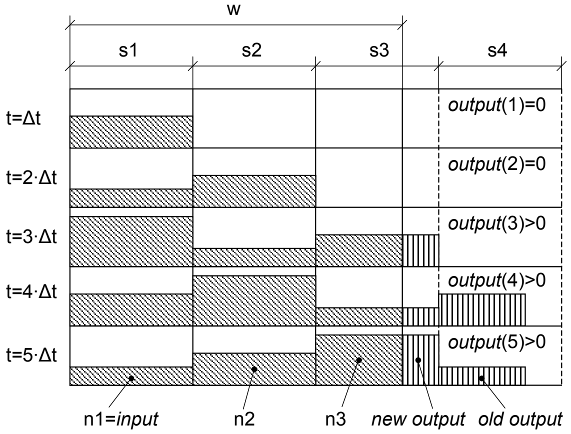

Figure 4: Basic model of one track of a tracked route in the mesoscopic approach using the method of dynamic route segmentation

Figure 4 illustrates the operation of one track during five time steps Δt. During the first time step, n1 persons enter the track, indicated by the hatched area. The first few representatives of this quantity leave the track during the third time step; their number equals the hatched area behind the boundary of the passageway. During the time units 4 and 5 the output quantity of the track is the sum of two segment quantities (old output and new output).

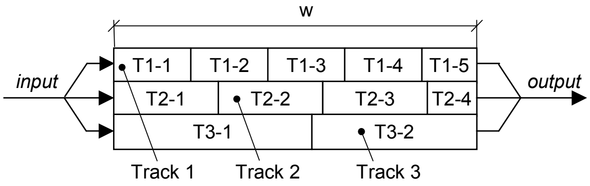

Figure 5: Basic model of a tracked route as a composition of several tracks

As stated before, each passageway consists of several tracks. Figure 5 illustrates the modeling of a passageway consisting of three tracks. The label "T3-1" denotes "Track 3, Segment 1". The input quantities of the passageway are split between the tracks according to the specified statistical distribution of movement speeds. The output quantities of the entire passageway are the sum of the outputs of the different tracks.

IMPLEMENTATION AND VERIFICATION OF A TEST MODEL

In order to test both the accuracy and performance of our mesoscopic approach, a *prototype simulator for the movement of physical groups* was designed and implemented. It is intended to be used in a control center of a train station or an airport to deliver information on expected system behavior to a decision support system. If the simulation predicts a critical state (e.g. check-in times at counter 4 will grow too long; room A5 will become too full), the decision support system proposes solutions and presents their simulated results to the control center users, who can then decide which alternative is best.

The data needed in order to run the simulation is distributed across several independent data sources:

- The *model structure* must be inferred from a database that contains information about both the building's structural design and its present state (e.g. malfunctions).
- The *current number of passengers* in each room, line and so on is measured by online sensors.
- *Future events* such as planned arrivals of passengers are stored in the transportation carriers' booking systems. Further, the building state database contains information on events that influence the model structure, e.g. the temporary shutdown of an elevator due to maintenance work.
- *Past events* may influence the simulation, for example, passengers deboarding a train that has recently entered the train station. Future consequences of past events are calculated by a coupled logical group simulator, which has yet to be implemented.
- The simulator also provides *routing information* for the different person groups since the physical group simulator does not dynamically assign targets to person groups. Until the logical group simulator is implemented, only static routing is possible.
- The external optimizer or decision support system initiates the simulation run and thus provides relevant *simulation control parameters*.

In view of the variety of data sources, data collection is done by independent programs or so-called *agents* that collect the necessary data and provide it to the simulator (Hanisch et al 2004). The strength of this design is its flexibility: The simulator remains unaffected by any change of data sources; only the data collecting agent has to be adjusted.

Data collection is triggered before each simulation run. While XML is the de-facto standard for cross-platform, cross-vendor information exchange, it is also a good format for describing simulation systems (Wiedemann 2002). The prototype uses XML to transfer the results of data collection from the agents to the simulator. XML is also used to describe all simulation control parameters, which leads to a completely data-driven simulation.

To illustrate this technique, consider the (very basic) model presented in figure 6 and its corresponding XML notation in listing 1. Note that no additional information is needed in order to run the simulation and that the results of the simulation can be saved in various data formats such as XML files and Microsoft Excel worksheets. Of course, additional input data (events, the current system state, etc.) can be supplied.

```
<simulation>
   <model>
      <metainfo>[…]</metainfo>
      <elements>
         <element name="entry">
            <source>
               <constOut>60</constOut>
            </source>
         </element>
         <element name="Segmented Way">
            <passageway>
               <length>200</length>
               <speed>1</speed>
               <speed>1.5</speed>
            </passageway>
         </element>
         <element name="exit">
            <sink/>
         </element>
      </elements>
      <links>
         <link from="entry" to="way"/>
         <link from="way" to="exit" weight="0.8"/>
         <link from="way" to="way" weight="0.2"/>
      </links>
      <units>
         <time>min</time>
         <distance>m</distance>
         <speed>m/s</speed>
         <persons>pax</persons>
      </units>
   </model>
   <simulationControl>
      <starttime>0</starttime>
      <endtime>120</endtime>
      <stepsize>1</stepsize>
   </simulationControl>
   <reporting>
      <report type="excel">[…]</report>
      <report type="xml">[…]</report>
    </reporting>
</simulation>
```

Listing 1: XML Source Code of a Simulation Run of the Simple Circle Example

As one can imagine, the XML simulation files for complicated model structures become relatively large. The model the authors used to measure the performance of the prototype consists of 70 service stations, 60 routes and 25,000 passengers, and resulted in 550 lines of XML code. Nevertheless, the simulation setup averaged out at less than one second on an 800 MHz machine. It took about 0.8 seconds to simulate one hour of real process. This time is almost independent of the amount of passengers that are simulated, because in the model passengers are represented by numbers rather than by individual objects. That means that the running time for 1,000 passengers and 100,000 passengers will be as good as equal, if one hour of real process is modeled in both cases. To compare this performance with conventional micro-simulation, an eM-Plant model was tested on the same machine. It took about three seconds to simulate 2,500 passengers; while a run with all 25,000 passengers usually needed more than 30 seconds.

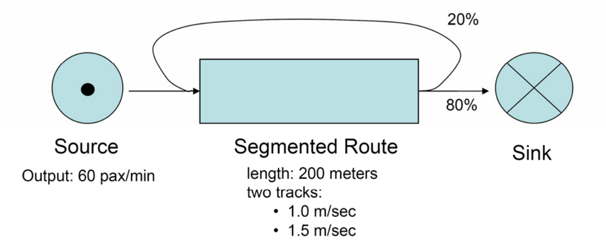

Figure 6: Simulation Model containing a Simple Circle

CONCLUSIONS AND FUTURE WORK

With its rather simple models, the mesoscopic approach presented outperforms classical microscopic simulation systems. Yet, it is fairly precise, allowing it to be used in any domain where both accuracy and speed are important.

One future step to improve the physical group simulator will be implementing the overcrowding of passageways. The main idea is to reduce the movement speed of the persons on an overcrowded route by a factor that is a function of the route's nominal capacity and its current load. Another step will be to implement the concept of rooms since they determine the amount of space available. This will make it possible to calculate the so-called *level of comfort*, indicating whether rooms are too empty or too full.

The most important task is to design and implement a simulator for logical groups and to couple it to the simulator for physical groups. The prototype will then be able to simulate pedestrian traffic flows with a dynamic model that changes during run time. Imagine, for example, the simulation is notified that an escalator is out of order. The simulation time is returned to the corresponding position and the simulation model is simply adjusted to the new conditions. Another example is the simulation of pedestrian traffic flows when a suspicious piece of luggage is found. In this case, some passageways or service stations have to be blocked while other, alternative routes are opened. In our model, the event can be placed anywhere in the future, resulting in immediate changes to the model at the precise time of the incident. As a result, together with its data-based model creation, the mesoscopic approach offers a level of flexibility that at the least is uncommon for classical industrial simulation systems.

REFERENCES

Hanisch, A., Tolujew, J., Richter, K., Schulze, T. "Online Simulation of Pedestrian Flow in Public Buildings". *Proceedings of the 2003 Winter Simulation Conference*, 1635-1641.

Hanisch, A., Tolujew, J., Meuschke, T., Schulze, T. “Datenkollektion zur online Simulation von Personenströmen”. *Simulation und Visualisation 2004*, Proceedings der Tagung in Magdeburg am 3. und 4. März 2004, 27-38.

Helbing, Dirk: Verkehrsdynamik: Neue physikalische Modellierungskonzepte. Springer-Verlag, Berlin Heidelberg New York, 1997.

Lerner, G., A. Hochstaedter, R. Kates, C. Demir, J. Meier, A. Poschinger. “The interplay of multiple scales in traffic flow: coupling of microscopic, mesoscopic, and macroscopic simulation”. *7th ITS World Congress on Intelligent Transport Systems*, Turin, Italy, 2000.

Løvås, G. “Modeling and simulation of pedestrian traffic flow”. *Transportation Res*. B. Vol. 28, No. 6, 1994, 429-443.

Schreckenberg, M. and S. D. Sharma. Pedestrian and Evacuation Dynamics. Springer-Verlag, Berlin Heidelberg New York, 2002.

Tolujew, J. “Prozessbilder als Grundlage für die Prozessanalyse in Materialflusssystemen.” *Simulationstechnik*, 17. ASIM-Symposium in Magdeburg, R. Hohmann (Hrsg.), SCS Int., Ghent 2003, 343-348.

Wiedemann, T. “XML-Standardschnittstellen für Simulations- und Animationsdaten”. *Simulation und Visualisierung 2002*, Proceedings der Tagung in Magdeburg am 28. Februar und 1. März 2002, 59-70.

AUTHOR BIOGRAPHIES

JURI TOLUJEW is a project manager in the Department for Logistics Systems and Networks at the Fraunhofer Institute for Factory Operation and Automation in Magdeburg (Germany). He received the Ph.D. in automation engineering from the University of Riga in 1976. He also received his habil. degree in computer science from the Otto von Guericke University Magdeburg in 2001. His research interests include simulation-based analysis of production and logistics systems, protocol-based methods for the analysis of processes in real and simulated systems as well as mesoscopic approaches to simulation.

FELIX ALCALA, born in Munich, is studying Commercial Information Technology at the Otto von Guericke University Magdeburg (Germany) where he received his intermediate Diplom in October 2003. His area of specialization is simulation. He has been working as a research engineer at the Fraunhofer Institute for Factory Operation and Automation in Magdeburg since January 2004. He can be contacted by e-mail at mail@felixalcala.de; his website is http://felixalcala.de.