A COMBINED CELLULAR AUTOMATA - DEVS SIMULATION FOR ROOM MANAGEMENT WITH VACATION TIMES

M. Bruckner[1], S. Tauböck[1], N. Popper[2], D. Wiegand[1], S. Emrich[1], S. Zerlauth[1]

[1]Vienna University of Technology, Vienna, Austria;
[2]"die Drahtwarenhandlung" - Simulation Services, Vienna, Austria

bruckner.martin@gmx.at (Martin Bruckner)

Abstract

In autumn of 2006, at the Vienna University of Technology the project TU Univercity 2015 was launched, which includes the modernization of the buildings and the concentration from the faculties of maximum two locations. In the context of this project, the Institute for Analysis and Scientific Computing intended to create a model to optimize the classroom assignment and simulate the processes in place at the university. This model, which is called MoreSpace, was realized mostly in the simulation language Enterprise Dynamics (ED), with exception of the part which calculates the times which the students need to change the lecture rooms. This outsourced part of the simulation is now the topic of this paper. The simulator is implemented in object-oriented programming language Java and connected via TCP / IP with Enterprise Dynamics. To model the dynamic behavior of single individuals an Agent-based system was chosen in which the individuals move on a discrete grid. The cell size is 0.125 x 0.125m, so that 1m2 consists of 64 cells. Each student takes 4 x 4 cells, or 0.5m x 0.5m. The move forward of people in a building is depending on several related factors. Some of these items are the density of the people in a group, the maximum speed which varies for each individual. These factors of course are in strong relation to the environment. In this case the difference in moving forward in a roomily area or a staircase will be mentioned. It is also of crucial importance, whether a person is facing multiple other individuals. All this is relevant for the speed and thus for the required time which the students need to switch from location A to another location B, in our case.

Keywords: Eurosim, Cellular Automaton, Pedestrian flow

Presenting Author's biography

Martin Bruckner was born in Austria in 1976. He passed his mathematic study on the Vienna University of Technology from 2002-2009. His field of activity is pedestrian flow and evacuation simulations.

1. The Model

Basically, this simulation consists of two simulators. On the one hand Enterprise Dynamics, a commercial Software from *INCONTROL* based on the DEVS (Discrete Event System Specification) formalism in which the optimization of the room utilization is modeled. On the other hand a proprietary development in *JAVA* which provides the times who are needed from the students to changed the lecture rooms. This part shall ensure that the time between two teaching units, who are maybe in different buildings, is sufficient dimensioned.

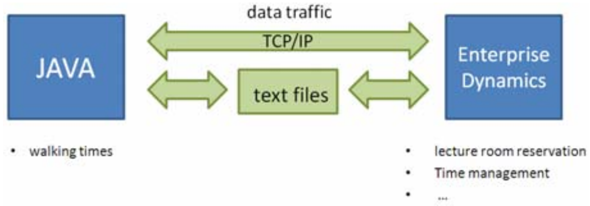

Fig. 1 data traffic

In order to implement the task a cellular automats (CAs) model was used [6, 7]. The main components of this model is a discrete plane and the individual agents, or as in our case, the individual students who are moving on this grid, and whose decision on the further action depends on the surrounding agents. Because of the fact that the university area is too big to display to only one of such grids, the buildings are divided in logically coherent parts that are connected at several points to give the agent the possibilities to change these discrete planes. In order to allow the students to move through the huge number of planes in the shortest possible way a combination of graph theory and a kind of scalar field is used

2. The Building plans:

The import of the building structures is via special edited planes. The editing is explained in the following shortly.

- Unnecessary information like dimensioning and many more are removed because it has no importance for the simulation
- The particular parts of the building will be colorized in one of the ten defined colors. (corridor, up- downstairs, wall, outdoor private or class room, no pass, window, text)
- In the next step the complete colored image will be exported to an image file in the Portable Network Graphics (PNG) format. The resolution of the graphics is chosen such that each pixel represents just 0,125m x 0,125m in reality, what exactly corresponds to the simulation of a cell of the CAs. For each pixel, the three primary colors of the RGB color space (red-green-blue) and in addition the alpha value stored. The range of values for each of these four channels is 0-255, which is 8 bit consuming.
- The last working step is to insert, with a special program, the so called *Change Areas* and *Doors*. This software tool is furthermore important because here the declaration of the buildings, CAs, rooms and many more will performed and stored finally in the *project* file. This Comma-Separated Values (CSV) project file is in a way the "brain" of the data set.

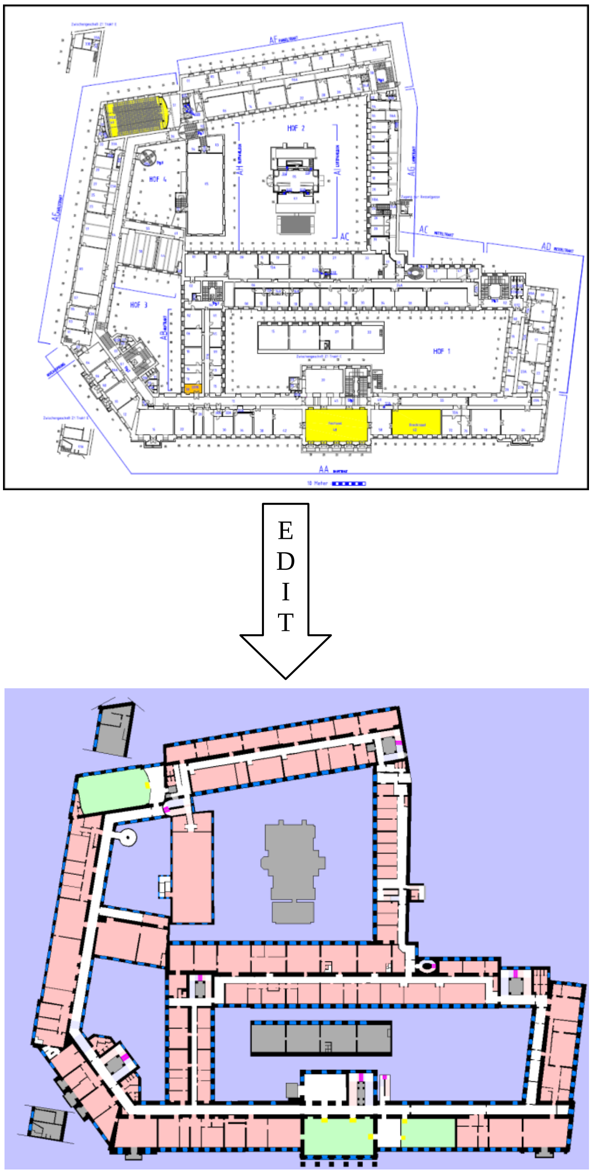

Fig. 2 example of a .png file (TU-Vienna, main building, first floor)

3. Routing

3.1 Global routing

To simplify student´s life they need a plan how to get from point A to point B and if possible in the shortest way. To ensure that this happens the graph theory was used. With aid of images and CSV file a connected undirected and weighted graph is generated.

- *Nodes:* Basically it exist only two types of nodes, *doors* and *change-areas*. In case of doors we consider only the entrance of the lecture-rooms and building-exits. Doors, such as fire doors, are not in use. Only a small narrowing in corridor handicapped the persons. The change-areas give the individuals the ability to change the CAs, for example at the end of a stair

- *Edges:* In every CA each node is connected via edges with each other one, whereby the edge weight is the distance in meter.

With aid of this nodes and edges and a slightly modified Dijkstra's algorithm it is now possible to find the path with the shortest weight of the edges and so the path with the shortest distance. This algorithm provides now a list which looks like the following.

Start Room
no CA change
Stair 1
CA change
Stair 2
CA change
Finish Room

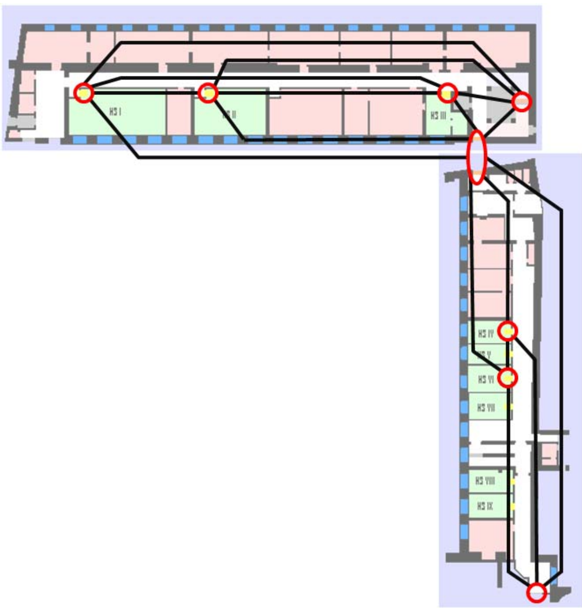

Fig. 3 example of a Graph

3.2 Local routing

The aim of the local routing is to move the individuals in the simulation to shortest way through the current Cellular automata. To implement this, the simulator used a Static Floor Field. As the name suggests, is this a field of static values, which contains the distances, to the various *nodes*, in cells. This field is embedded into the CAs. So it is possible to "*ask*" each cell "*how far is it to the door with name doorLectureRoom8*". To determine now the shortest path, for each direction the associated cells will be scanned and afterwards calculate the arithmetical average. So the direction with the smallest distance is to favor for the next step. To initialize this data structure, a modify variation of the Flood Fill algorithm is used. This is a simple procedure from Computer graphics, to colorize a bordered area.

6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5
4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2
2	1	1	1	1	1	1	2
		0	0	0	0		
		-1	-1	-1	-1		

Fig. 4 Static floor field

Fig. 4 shows an example from a static floor field. In the below center yellow cells illustrate a door bordered left and right from a part of a wall. A cell with distance 0 indicate a start point of the door. The remaining numbers showed the distance of the associated cell to this start points.

4. Deadlock

One of the most difficult tasks in the implementation of the model was to dealing with the problem of deadlocks which occur when students interact with each other. Even a relatively small collision escalate often in a short time to a complete blockade, and thus to a deadlock. To handle this problem some mechanisms were implemented to detect collisions beforehand and trying to resolve deadlocks.

- **React to other students:** The first and perhaps most effective method is the timely reaction of contra flow or other barriers. To implement this, a search area of approximately 1.5 meters wide and 8 meters in length will be scanned and all founded students are divided in three groups according to their relative walking direction. (1) traffic, (2) cross traffic, (3) oncoming traffic. This subdivision will be considered by the selection of the next step. For example, when overtaking a student, or make room for oncoming traffic.

- **Jostle other persons:** In order to allow a necessary change of direction even if another student who walk parallel and in the same direction and prevent this, each individual has the possibility to make a request to the neighbor to make room. The student can afterward decide if he obeys this suggestion or not depending on if it possible for him.
- **Right-hand walking:** The next algorithm to avoid collision is to pass a contra flow on the right side will preferred. This behavior can be observed repeatedly in public and is also described in the literature by various experiments [4].
- **Shrink Student size:** Another very useful method is to imitate the behavior of people coming into the situation that the required space is too small for "normal" walk. We instinctively shrink our space, for example rotate our shoulders. This shrink is also implemented in this model. Each student normally required $0.25m^2$ or 4x4 cells in simulation. This space may, if the situation requires, shall be reduced to 3x3 or 2x2 cells.
- **Waiting in front of the auditorium:** Further potential sources of blockages are the entrances of the lecture rooms. So arrival Students are waiting outside of the room as long as the room and therefore the doors are free.
- **Second level:** The last algorithm to resolve deadlocks is a second emergency level. About this second level blockade can be reduced and it will ensure that the students reach their specified destination.

5. Walking speed

The walking speed of people is influenced by different factors. To simplify the simulation acceleration and deceleration will be ignore.

5.1 Depending on density

A crucial factor of move forward is the density of the traffic. The relation of density and speed is described with the so-called *Kladek* formula. [1]

$$v_h(\rho) = v_h^0(1 - exp^{\left(-\gamma\left(\frac{1}{\rho} - \frac{1}{\rho^{max}}\right)\right)}) \quad (1)$$

v_h^0 Max. speed, depending on the underground.

ρ^{max} Max. density, 5.4 P/m^2

γ calibrate constant (empirical determined), 1.913.

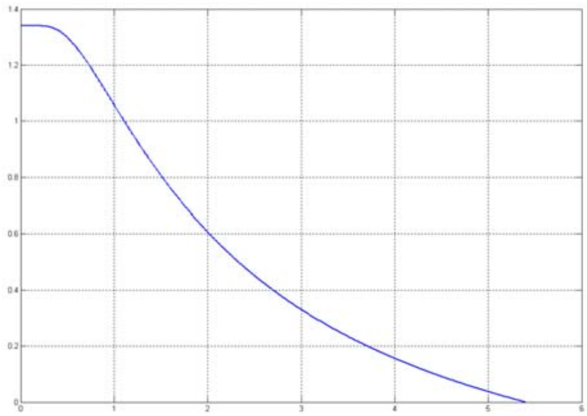

Fig. 5 Relation of Kladek v_h^0=1.34 m/s

5.2 Depending on underground

The ground where the persons are moving is also an important factor. To simplify this model only three different types are defined. These values are the v_h^0 for the mentioned Kladek formula.

- Corridor 1.34 m/s [1]
- Upstairs 0.61 m/s (only horizontal) [2]
- Downstairs 0.71 m/s (only horizontal) [2]

Because the size of the students varies, but it is in any case bigger than one cell, it is possible that they occupy cells with different speed-values, for example when entering a stair. To deal with this problem, the maximum speed v_h^0 is the arithmetic average of all cells.

5.3 Depending on random speed

In addition to this speed calculation each student has a random speed factor from 0.8 .. 1.2. This factor will be multiply with the final speed value to bring the simulation closer to the reality.

6. Calculate next Step

Afterward is a brief overview of the calculation of one time step. In the simulation one time step is 250ms. Please note that a time step is not the same like a cell shift. One time step has usually a few cell shifts.

- **Data ascertainment:** density, traffic, …
- **Calculate next direction:** with aid of the collected data the next direction for the cell shift is chosen.
- **Calculate speed and time:** calculate the possible speed in this direction and the resultant time for this shift.
- **Repeat 1-3:** The first three points are repeated until the difference of the sum of the single shift time and the simulation time, in our case 250 ms, is minimal.

$$\left|\sum(shift\ times) - 250ms\right| = minimal$$

- **Calculate discretization error:** the difference of the above formula is the start value in the next iteration. This helps to minimize the discretization error.

  Following example show what's happens without consideration of this discretization error.

  Student speed = $1{,}34\ \frac{m}{s}$
  Step time Δt = 0,25 s
  From this it follows a distance of $1{,}34\frac{m}{s} * 0{,}25\ s = 0.335\ m$
  Possible discrete distance are $\frac{2}{8}m = 0{,}250\ m$ or $\frac{3}{8}m = 0{,}375\ m$
  To minimize the error we chose the value $0{,}375\ m$.
  Herewith the resultant absolute error is $0{,}375\ \mathrm{m} - 0{,}335\ \mathrm{m} = 0{,}04\ \mathrm{m}$ or a relative error from $\frac{0{,}04\ m}{0{,}335\ m} * 100 \approx 12\ \%$
- **Move:** In the last step the precalculate shifts for each student are performed. This happens with the *no crossing path* method [3]. This technique allows each student to move on its precalculated way until he has performed all shifts or cross a trajectory of another individual who has already moved. Because the order, for moving the students, in this algorithm is of importance a random process is implemented.

7. User Interface

The User Interface gives the user the ability to interact with the computer program and visualize the student trajectories.

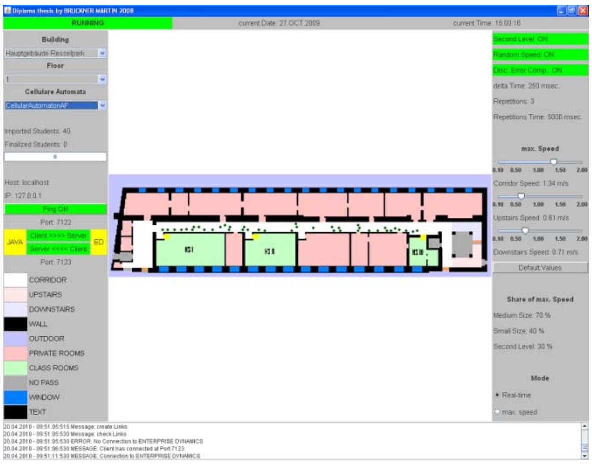

Fig. 6 User Interface

8. Scenarios

In these scenarios the dependence of the travel times from density and the presence of contra flow shall shown. The following experiments were performed 10 times in a row [5].

8.1. First scenario

The behavior of individuals shall evaluate with increasing number of people without contra flow and only two lecture rooms.

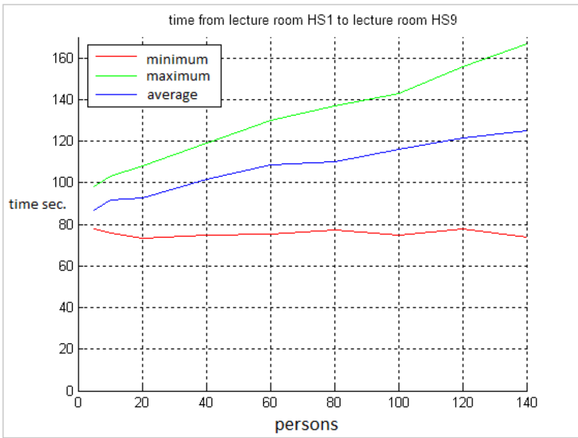

Fig. 7 Graph HS1 ➔ HS9

Unsurprisingly the average and maximum time is increasing steadily with rising density, while the minimum value with exception of slight variations is almost unchanged. This constant value is also not surprising, because the first students always find empty corridors and so need each time approximately the same periods for switching.

8.2 Second scenario

The behavior of individuals shall evaluate with increasing number of people with contra flow in ten lecture rooms.

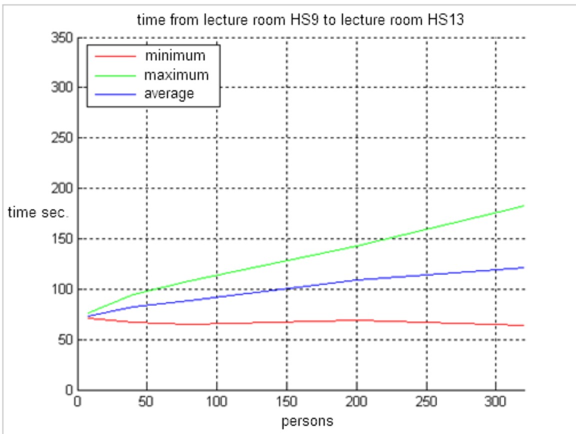

Fig. 8 Graph HS9 ➔ HS13

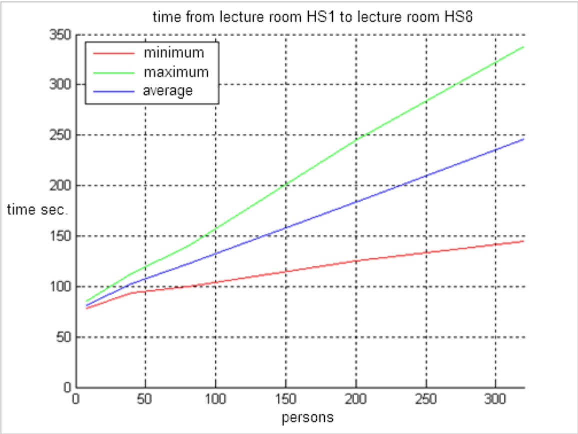

Fig. 9 Graph HS1 ➔ HS8

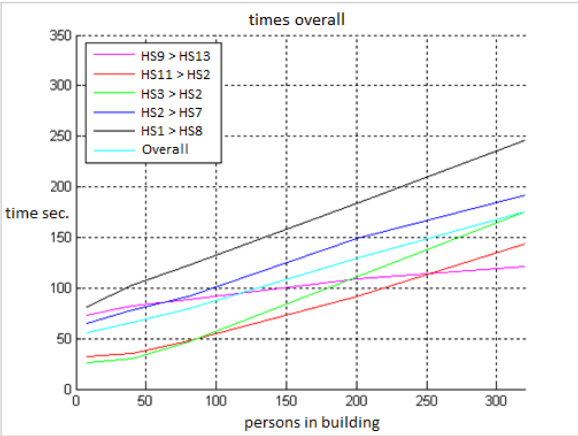

Fig. 10 times overall

Good to see is that the gradient of the single lines is depending on the length of the way. So the graph with the shortest distance HS9 > HS13 is quite flat, while the longest path HS1 > HS8 has a strong rise.

9 References

[1] Péter Molnár. *Modellierung und Simulation der Dynamik von Fußgängerströmen.* Shaker Verlag, Aachen 1996.

[2] Taku Fujiyama, Nick Tyler. *An Explicit Study on Wallking Speeds of Pedestrians on Stairs. Centre for Transport Studies.* University College London.

[3] Ansgar Kirchner, Huber Klüfel, Katsuhiro Nishinari, Andreas Schadschneider, Michale Schreckenberg, *Discretization effects and the influence of walking speed in cellular automata models for pedestrian dynamics.*

[4] Tobias Kretz, Anna Grünebohm, Maike Kaufman, Florian Mazur and Michael Schreckenberg. *Experimental study of pedestrian counterflow in a corridor.* Journal of Statistical Mechanics: Theory and Experiment. 2006

[5] Martin Bruckner. *Modellierung der Fußgängerdynamik im universitären Betriebmit Hilfe von Zellulären Automaten in der Programmiersprache JAVA.* Vienna University of Technology 2009

[6] Victor J. Blue, Jeffrey L. Adler. *Cellular automata microsimulation for modeling bi-directional pedestrian walkways.* Transporation Research Part B 2001

[7] Fang Weifeng, Yang Lizhong, Fan Weicheng. *Simulation of bi-direction pedestrian movement using a cellular automata model.* Physica A 2003