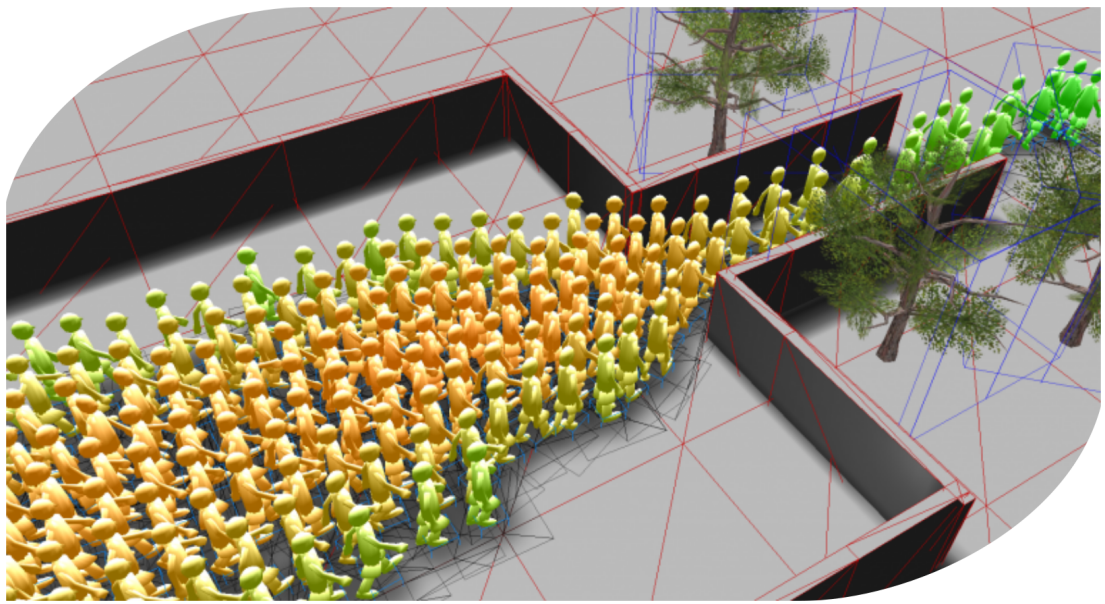
How to simulate crowds and behavioral changes

In a nutshell

Benedikt Kleinmeier

April 14, 2021



Outline

About me

I am a computer scientist

- Firmware development for different MCUs
- Application development

⇒ I.e., a strong focus on software engineering, test-driven development and CI/CD.

I am not an electrical engineer / UX designer

- No deep hardware knowledge!

⇒ I am not an expert, but I am very interested in hardware and data visualization.

Curriculum

- 2008–2014: Computer science at Hochschule München HM
  - BA thesis: Firmware to get data from a pressure sensor and a GUI to visualize this data: C C# .NET
  - MA thesis: Linux driver to acquire data from a GPS sensor: C C++ Real-Time Linux
- 2015–2017: Firmware developer at Infineon
  - Bootstrap loader (ARM-based MCUs): C
  - Firmware/Startup software (Aurix 2G): C Python TDD CI/CD
- Since 12/2017: PhD at HM and TUM HM & TUM
  - Modeling and simulation of behavioral changes
  - Using open source simulator Vadere (`http://www.vadere.org/`)
  - Pedestrian dynamics research group (Prof. Dr. Köster, Hochschule München)
    Java Python matplotlib TDD CI/CD

Curriculum

- 2008–2014: Computer science at Hochschule München HM
  - BA thesis: Firmware to get data from a pressure sensor and a GUI to visualize this data: C C# .NET
  - MA thesis: Linux driver to acquire data from a GPS sensor: C C++ Real-Time Linux
- 2015–2017: Firmware developer at Infineon
  - Bootstrap loader (ARM-based MCUs): C
  - Firmware/Startup software (Aurix 2G): C Python TDD CI/CD
- Since 12/2017: PhD at HM and TUM HM & TUM
  - Modeling and simulation of behavioral changes
  - Using open source simulator Vadere (`http://www.vadere.org/`)
  - Pedestrian dynamics research group (Prof. Dr. Köster, Hochschule München)

    Java Python matplotlib TDD CI/CD

Curriculum

- 2008–2014: Computer science at Hochschule München HM
  - BA thesis: Firmware to get data from a pressure sensor and a GUI to visualize this data: C C# .NET
  - MA thesis: Linux driver to acquire data from a GPS sensor: C C++ Real-Time Linux
- 2015–2017: Firmware developer at Infineon
  - Bootstrap loader (ARM-based MCUs): C
  - Firmware/Startup software (Aurix 2G): C Python TDD CI/CD
- Since 12/2017: PhD at HM and TUM HM & TUM
  - Modeling and simulation of behavioral changes
  - Using open source simulator Vadere (`http://www.vadere.org/`)
  - Pedestrian dynamics research group (Prof. Dr. Köster, Hochschule München)
    Java Python matplotlib TDD CI/CD

Why do we need pedestrian dynamics?

Stampede at Love Parade, Duisburg, 2010, photo: Wiffers 2010

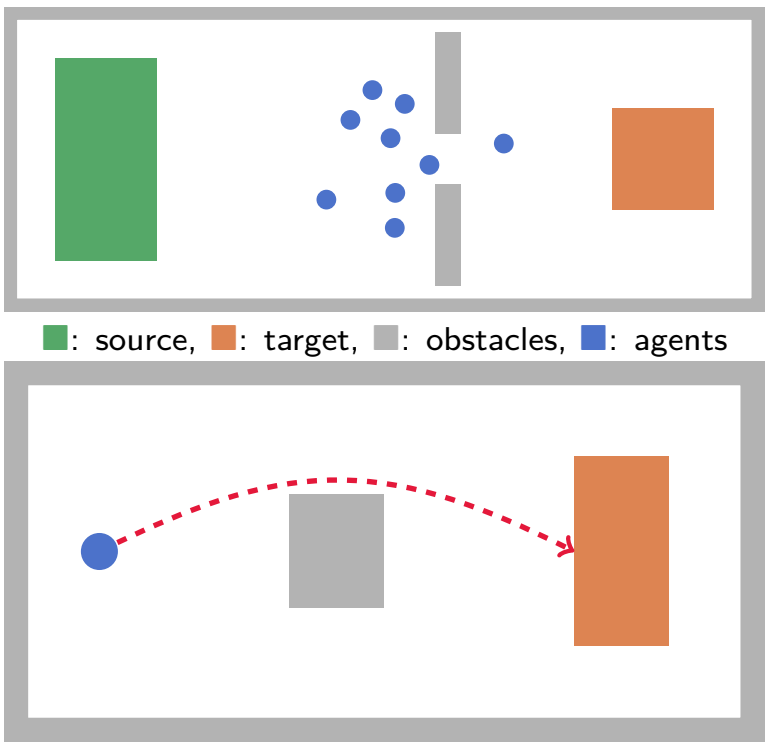
Goal: Reduce risks for life and limb wherever crowds gather by getting insights into pedestrian streams with simulations

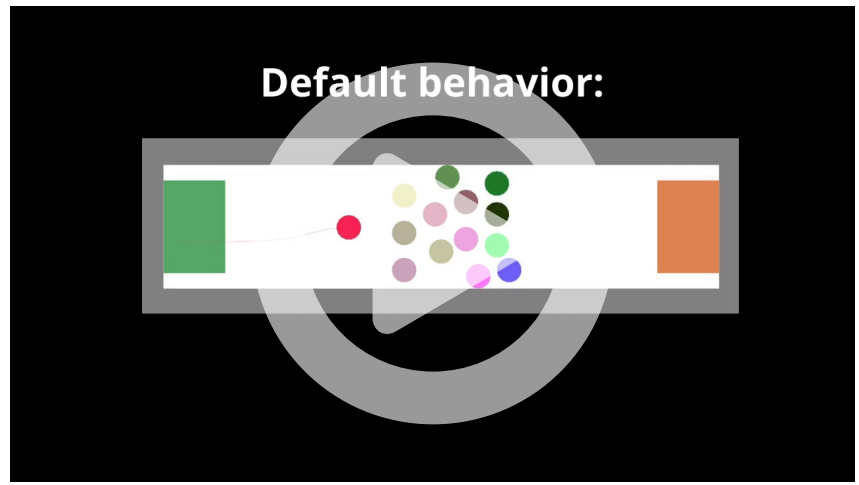
How to simulate crowds

Two main problems:

1. Navigation and wayfinding (solved → next slide)

2. Behavioral changes (unresolved → my PhD)



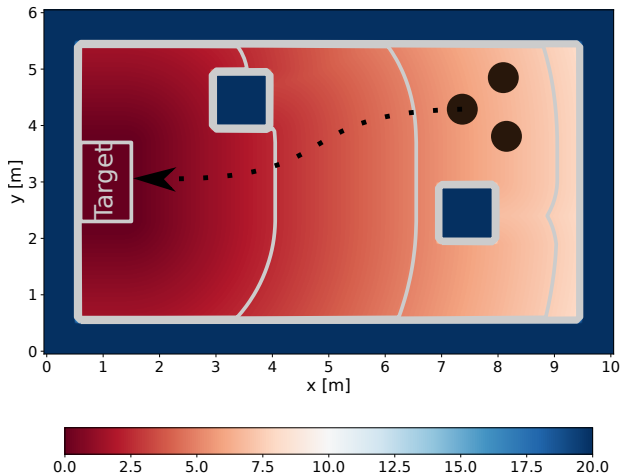


Navigation and wayfinding: Our approach

1. A floor field based on the eikonal equation:

$$|\nabla u(x)| = \frac{1}{f(x)}, x \in \mathbb{R}^2$$

- Solution $u(x)$: encodes minimal distance from x to target curvature $\partial\Omega$
- Given: speed $f(x)$
- Fast marching method (finite differences)



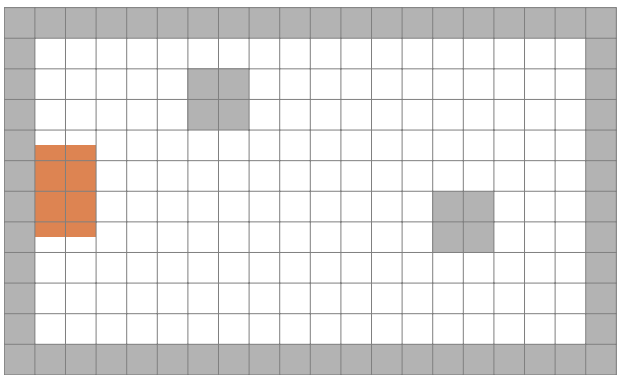
2. An optimization for each agent in each simulation step (goal: reduce travel time)

Navigation and wayfinding: Our approach

1. A floor field based on the eikonal equation:

$$|\nabla u(x)| = \frac{1}{f(x)}, x \in \mathbb{R}^2$$

- Solution $u(x)$: encodes minimal distance from x to target curvature $\partial\Omega$
- Given: speed $f(x)$
- Fast marching method (finite differences)



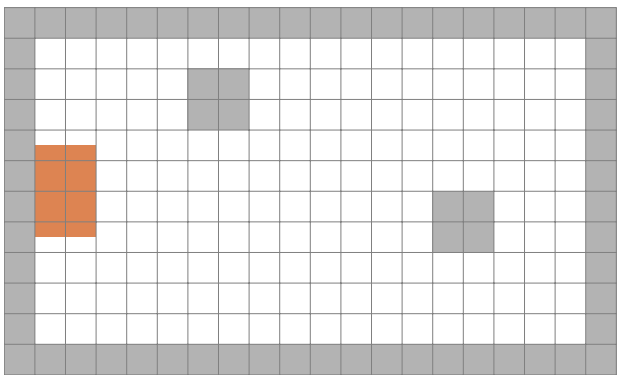
2. An optimization for each agent in each simulation step (goal: reduce travel time)

Navigation and wayfinding: Our approach

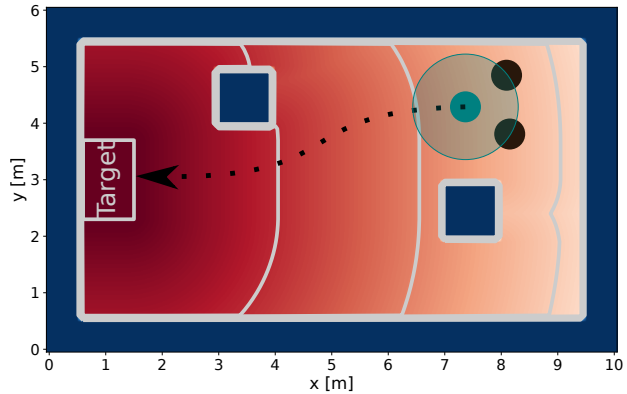
1. A floor field based on the eikonal equation:

$$|\nabla u(x)| = \frac{1}{f(x)}, x \in \mathbb{R}^2$$

- Solution $u(x)$: encodes minimal distance from x to target curvature $\partial\Omega$
- Given: speed $f(x)$
- Fast marching method (finite differences)



2. An optimization for each agent in each simulation step (goal: reduce travel time)

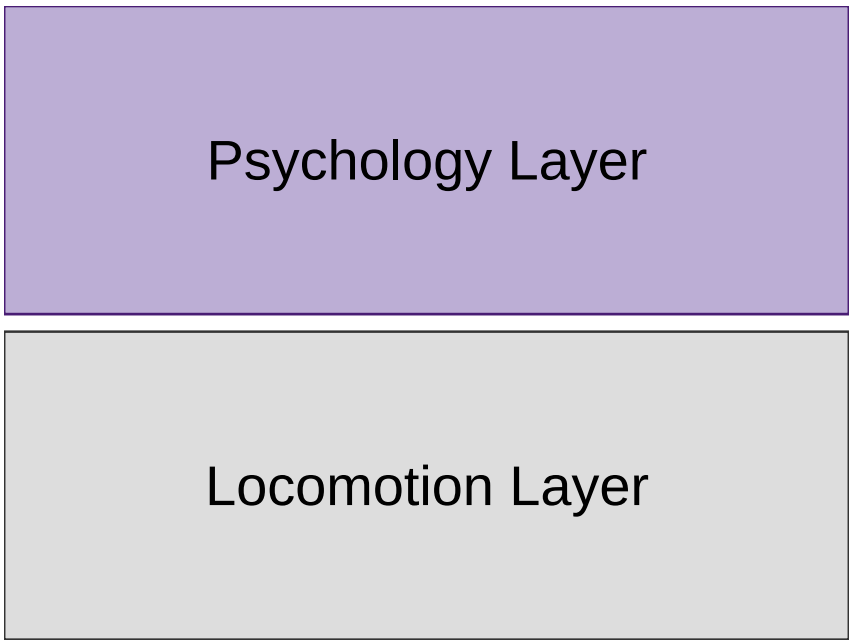


A crowd simulator

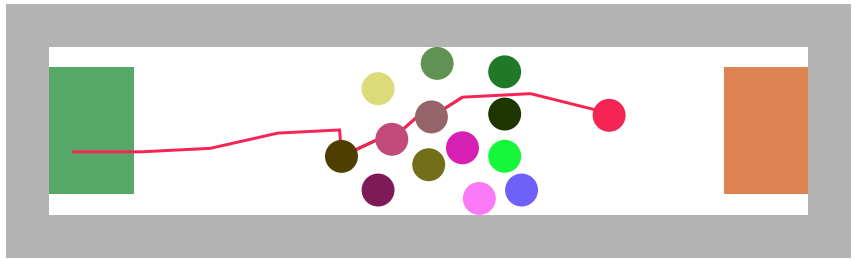
Bird's eye view onto a simulator:

```
while (simulationIsRunning) {
  ...
  // For each agent, search next agent
  // position that is closer to target
  locomotionModel.update(agents, time);
  ...
  time++;
}
```

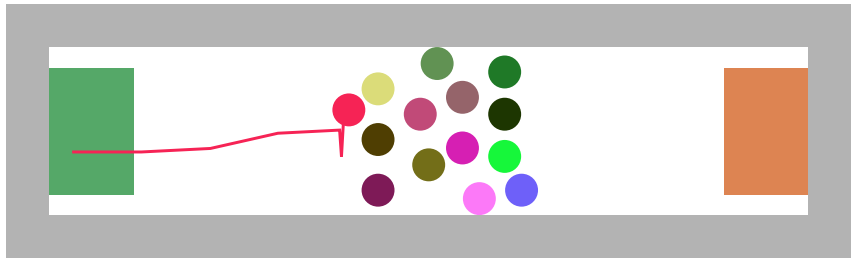
Psychology layer: Overview



Operationalization



With psychology



Without psychology

Simulator

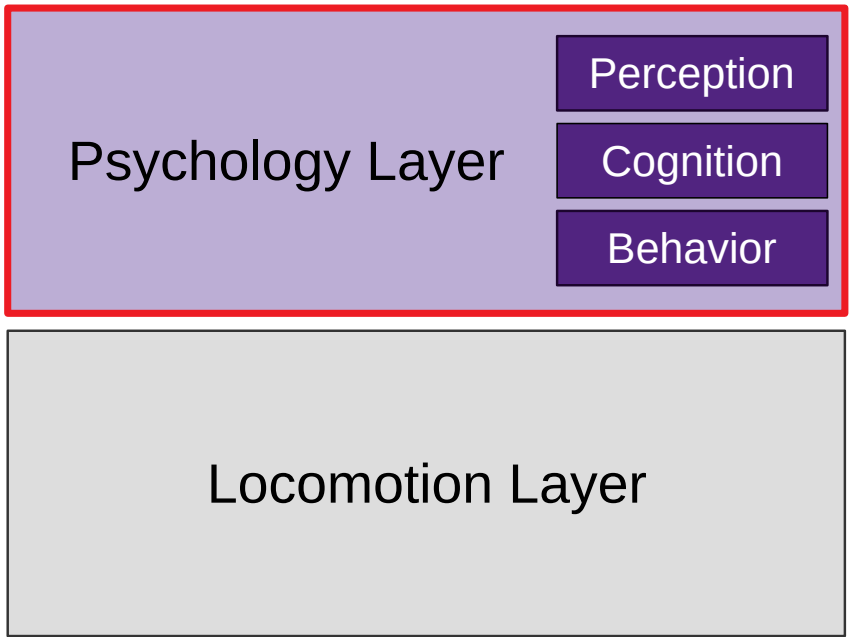
Bird's eye view onto a simulator:

```
while (simulationIsRunning) {
  ...
  // For each agent, search next agent
  // position that is closer to target
  locomotionModel.update(agents, time);
  ...
  time++;
}
```

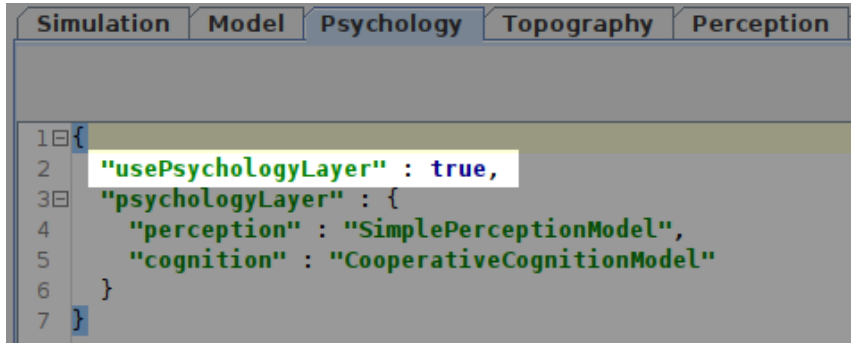
```
while (simulationIsRunning) {
  ...
  if (usePsychologyLayer()) {
    // Strategy pattern
    perceptionModel.update(agents, stimuli);
    cognitionModel.update(agents);
  }

  locomotionModel.update(agents, time);
  time++;
}
```

- A minimally invasive psychology layer
- that can easily be integrated in other crowd simulators

Psychology layer: Optional

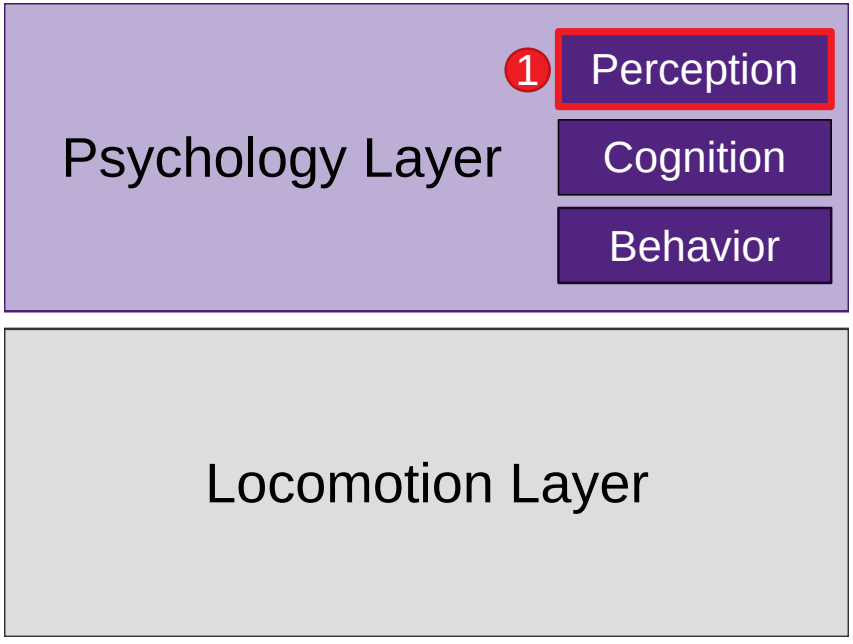


Operationalization

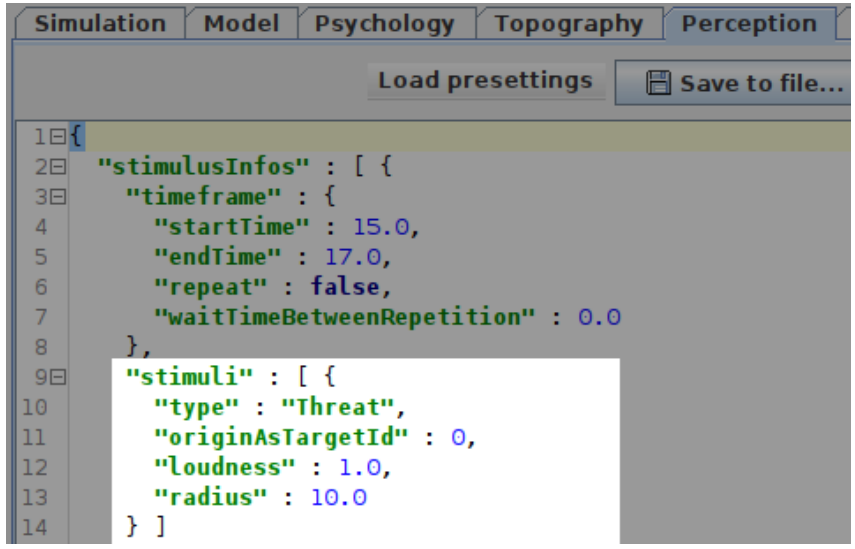


Simulator

Psychology layer: Perception

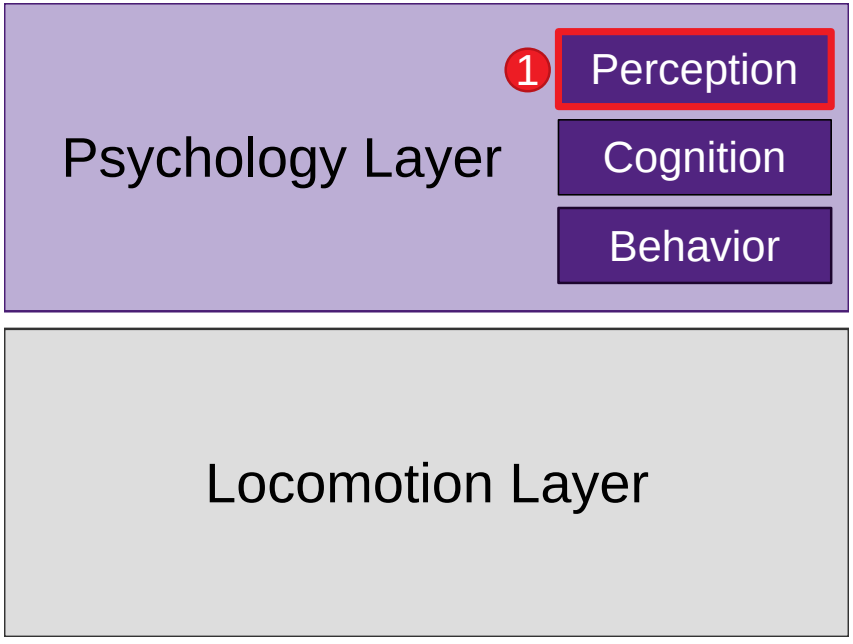


Operationalization

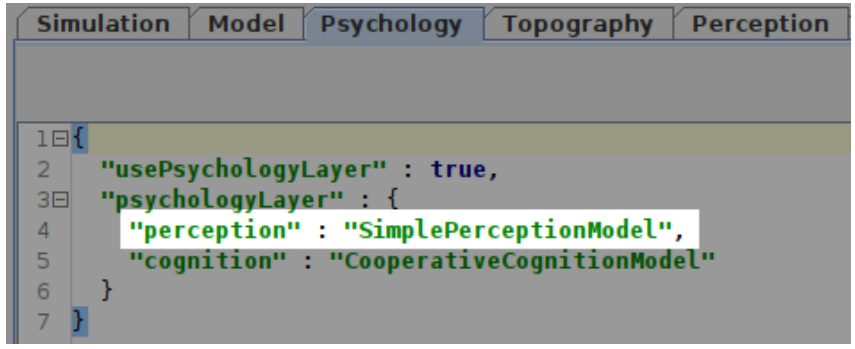


Simulator

Psychology layer: Perception

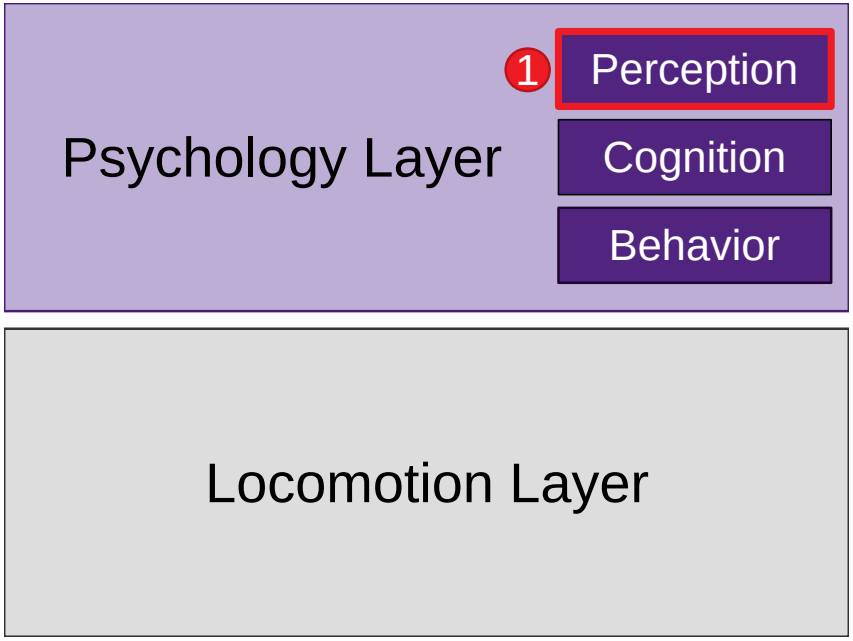


Operationalization



Simulator

Psychology layer: Perception

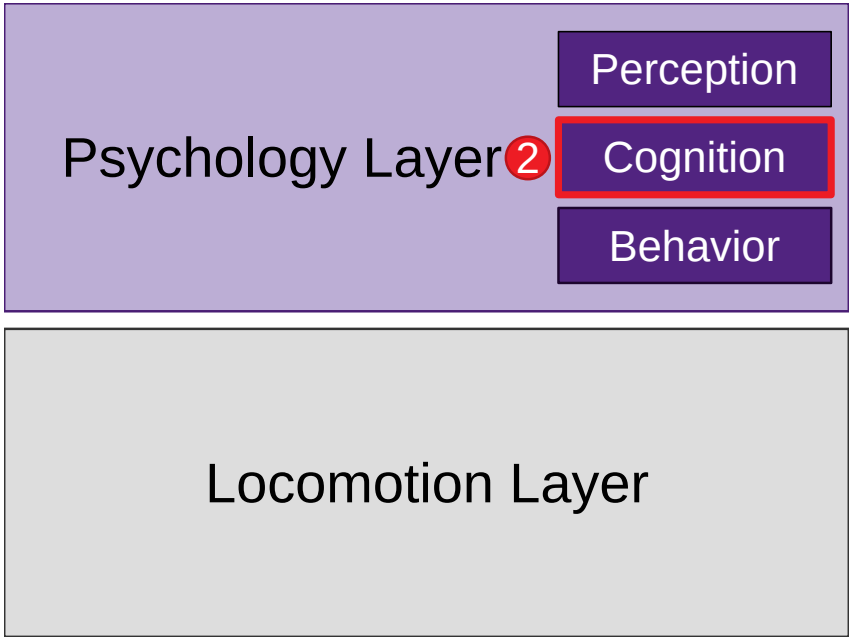



```
class SimplePerceptionModel
  ...
  void update(List<Pedestrian>
      pedestrians, List<Stimulus>
      stimuli) {

    for (Pedestrian ped : pedestrians) {
      mostImportantStimulus =
      prioritizeStimuli(stimuli, ped);
      ped.setMostImportantStimulus(
      mostImportantStimulus);
    }
  }
}
```

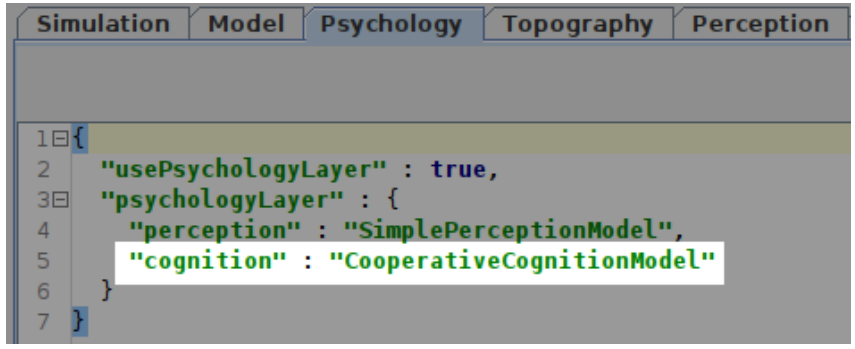
Operationalization

Simulator

Psychology layer: Cognition

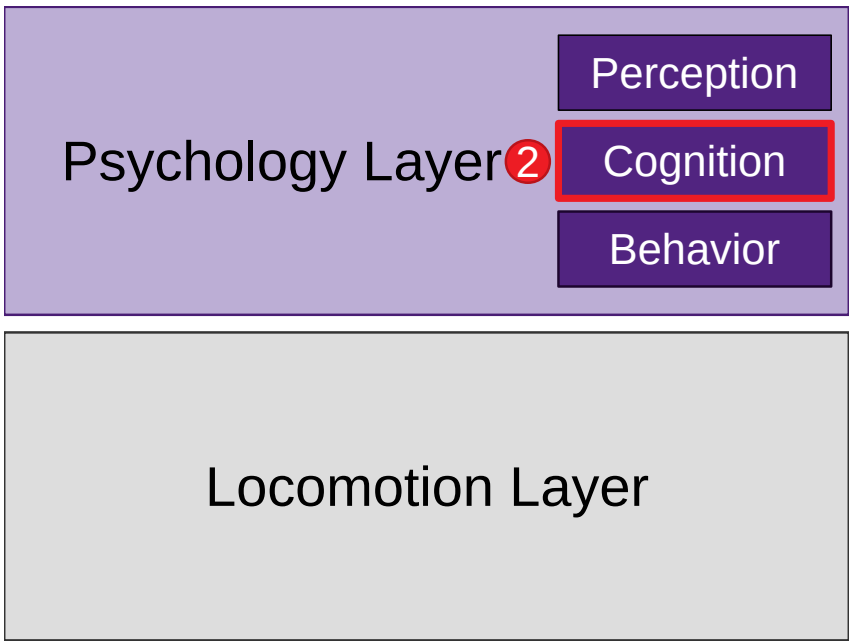


Operationalization



Simulator

Psychology layer: Cognition

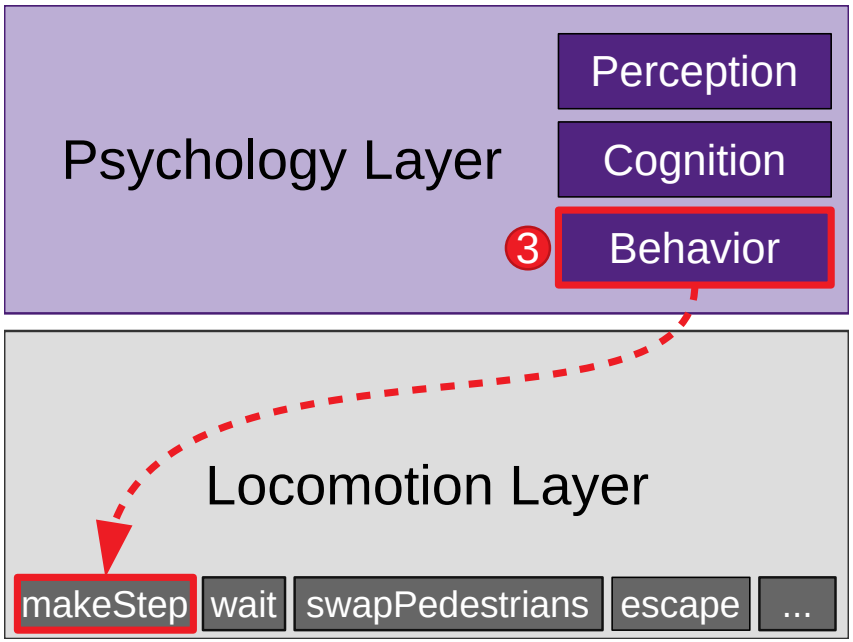



```
public class CooperativeCognitionModel {
  ...
  for (Agent agent : agents) {
    boolean cannotMove = agent.getSpeed(
     lastSteps) <= threshold;

    if (cannotMove) {
        agent.setSelfCategory(
     COOPERATIVE);
    } else {
        agent.setSelfCategory(
     TARGET_ORIENTED);
    }
  }

}
```

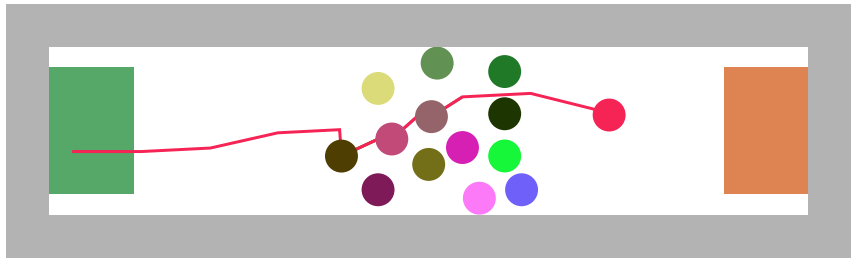
Operationalization

Simulator

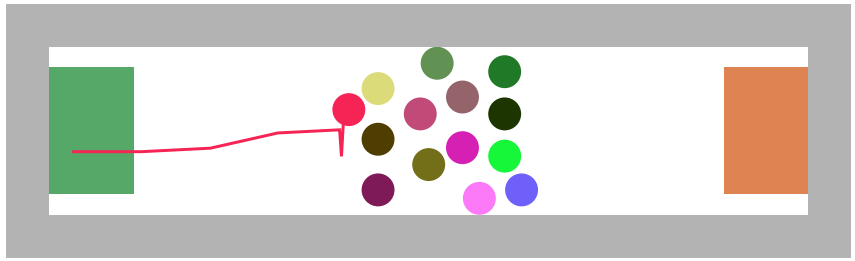
Psychology layer: Behavior



Operationalization



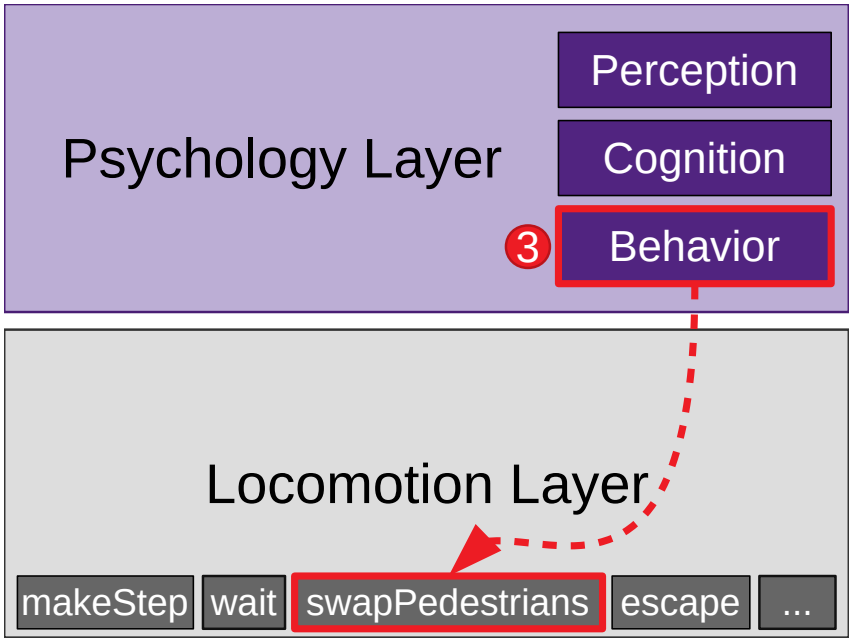
...towards cooperation.



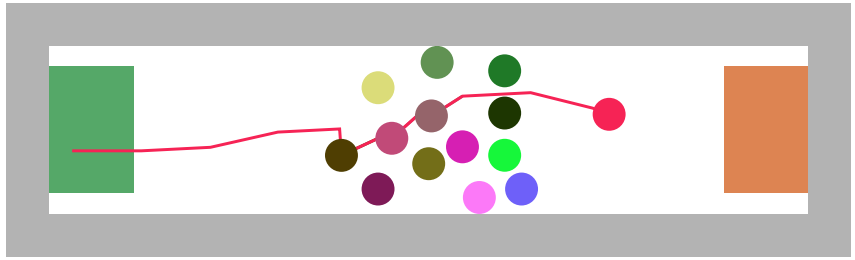
Behavioral change...

Simulator

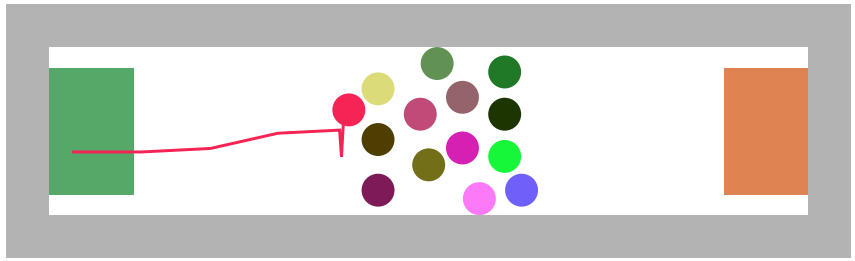
Psychology layer: Behavior



Operationalization



...towards cooperation.



Behavioral change...

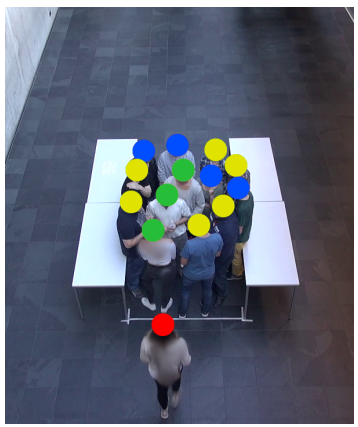
Simulator

Foundations

My research is based on three pillars:

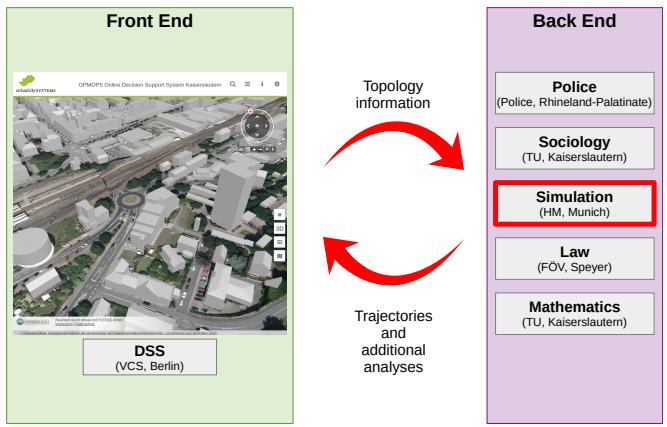
Experiment

- In Oct 2018 with 58 participants
- To re-enact "failing" simulation from previous slides



OPMOPS project

- Franco-German project about demonstrations in urban areas.
- Goal: Implement a "decision support system" to plan and handle events.



Research stay in UK

- At University of Sussex
- Exchange with social psychology professor Dr. John Drury and his team

UNIVERSITY OF SUSSEX

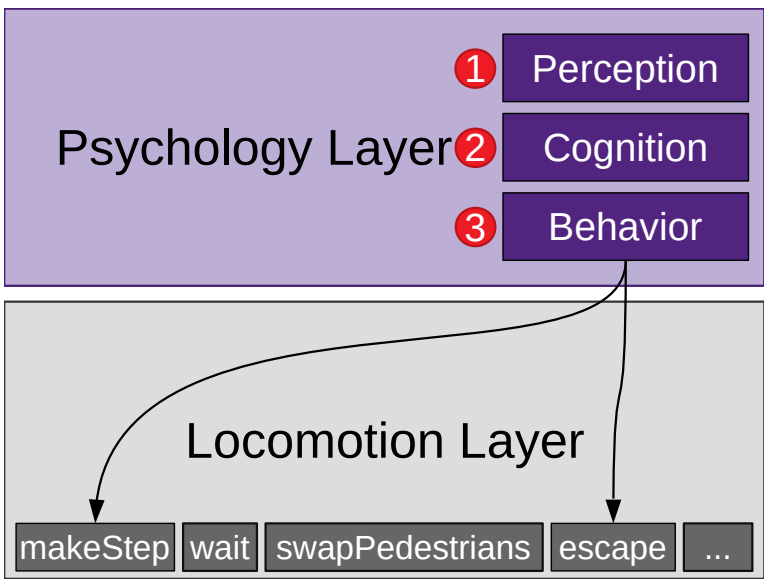
Summary

Research question:

How can changes in human behavior be operationalized for simulations?

Contributions:

My proposal:

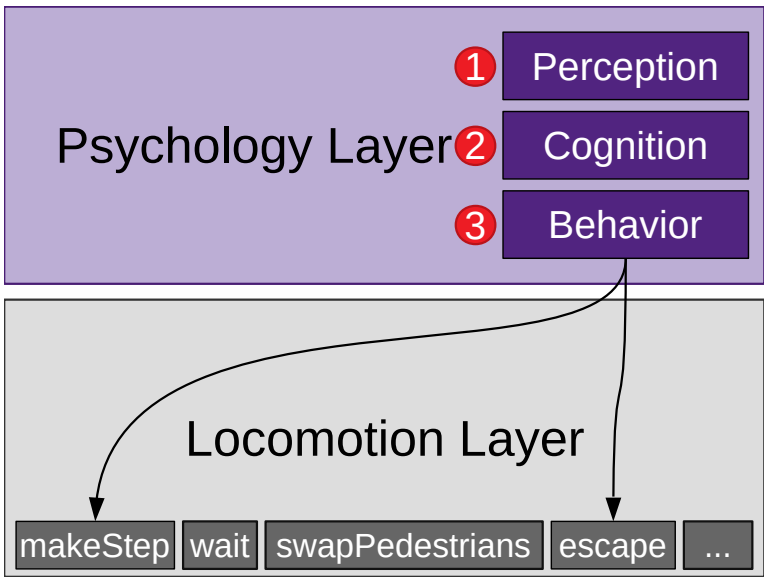


Summary

Research question:

How can changes in human behavior be operationalized for simulations?

My proposal:



Contributions:

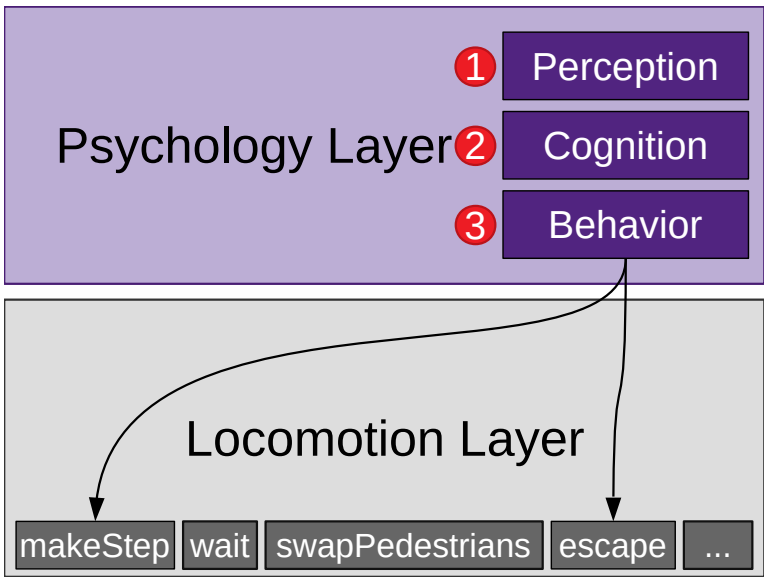
1. Integrated psychology layer
   - based on empirical data (own experiment)
   - tested by practitioners (police Rhineland-Palatinate)
   - operationalized behavioral changes with social psychologists

Summary

Research question:

How can changes in human behavior be operationalized for simulations?

My proposal:



Contributions:

1. Integrated psychology layer
   - based on empirical data (own experiment)
   - tested by practitioners (police Rhineland-Palatinate)
   - operationalized behavioral changes with social psychologists
2. Software engineering
   - I encapsulated my approach as reusable psychology layer (for other simulators)
   - Set up CI/CD to deploy Vadere simulator to website
   - Simulator maintenance (GUI, OpenStreetMap converter, ...)

All source code contributions are open source: `https://gitlab.lrz.de/vadere/vadere`

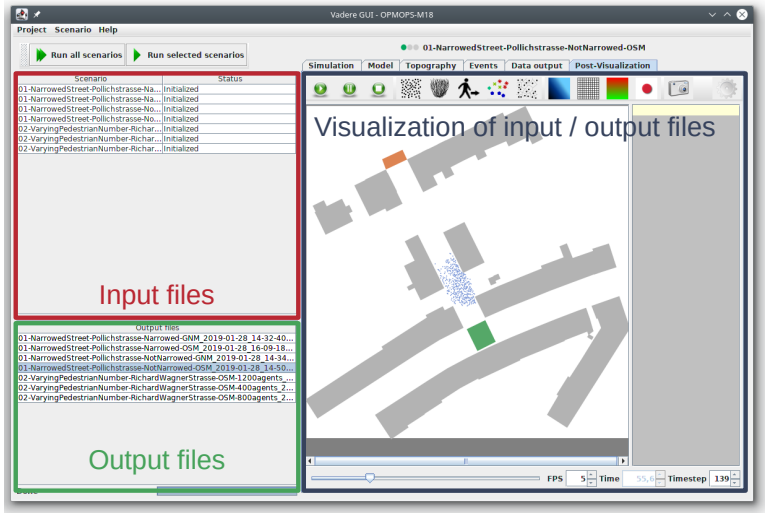
Backup slides

Simulator details . Simulation results . Navigation and wayfinding

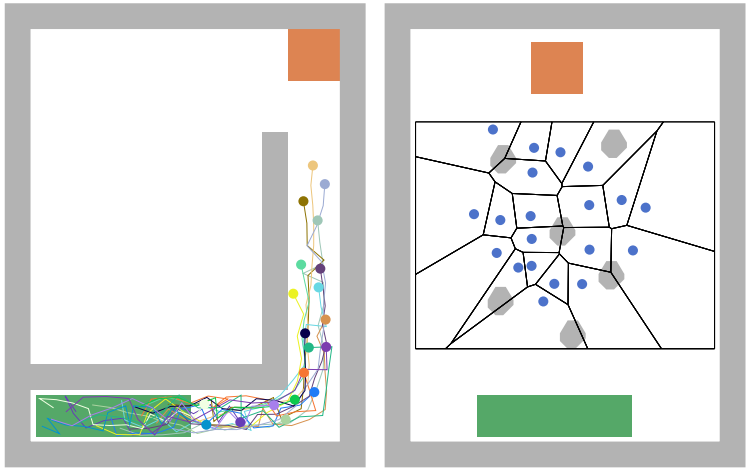
Vadere: Core features

Back

- Easy-to-use GUI
- CLI for automation
- Shipped with different locomotion models:
  - Mature: Optimal steps model (OSM), social force model (SFM), ...
  - Experimental: Behavioral heuristics model (BHM), Reynolds' steering, ...
- JSON-based input files
- Continuous integration / deployment pipeline



Vadere GUI features



Vadere: Core features

Back

- Easy-to-use GUI
- CLI for automation
- Shipped with different locomotion models:
  - Mature: Optimal steps model (OSM), social force model (SFM), ...
  - Experimental: Behavioral heuristics model (BHM), Reynolds' steering, ...
- JSON-based input files
- Continuous integration / deployment pipeline

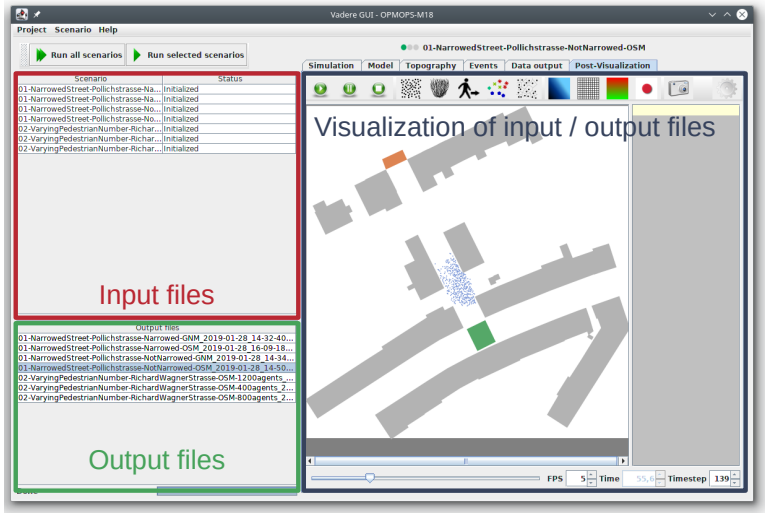
```
java -jar
vadere-console.jar \
scenario-run \
--scenario-file \
example.scenario
```

Vadere's command line interface (CLI)

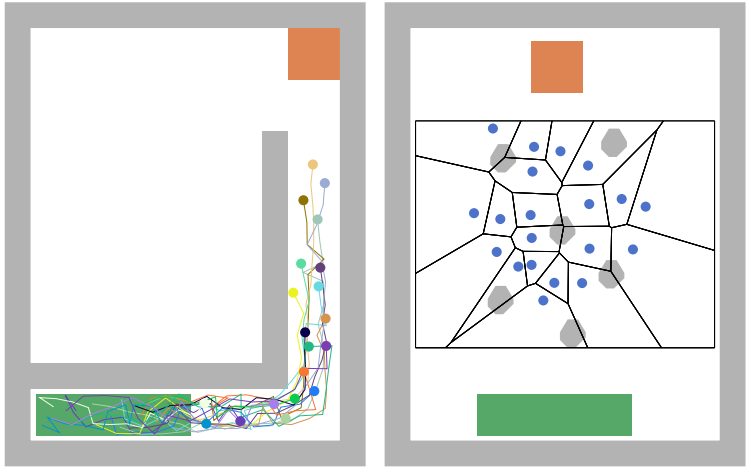
Vadere: Core features

Back

- Easy-to-use GUI
- CLI for automation
- Shipped with different locomotion models:
  - Mature: Optimal steps model (OSM), social force model (SFM), ...
  - Experimental: Behavioral heuristics model (BHM), Reynolds' steering, ...
- JSON-based input files
- Continuous integration / deployment pipeline



Vadere GUI features



Vadere: Core features

Back

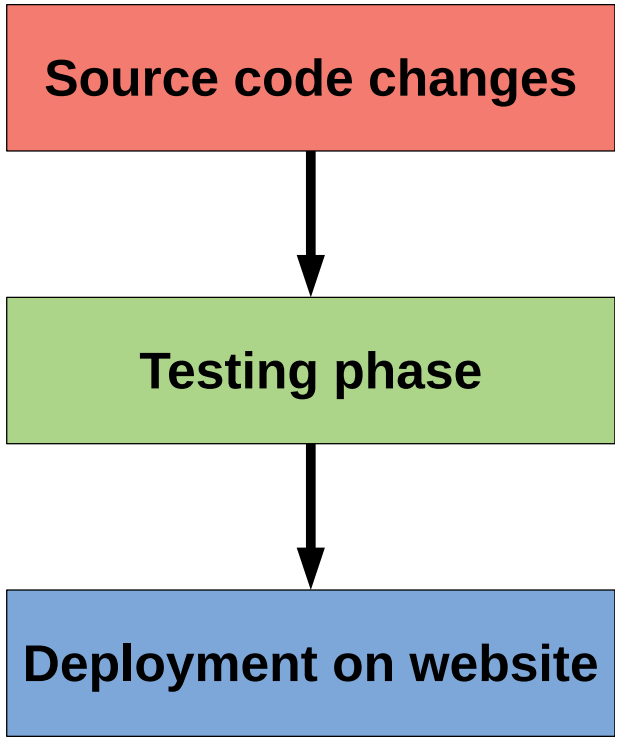
- Easy-to-use GUI
- CLI for automation
- Shipped with different locomotion models:
  - Mature: Optimal steps model (OSM), social force model (SFM), ...
  - Experimental: Behavioral heuristics model (BHM), Reynolds' steering, ...
- JSON-based input files
- Continuous integration / deployment pipeline

```
{
 "name" : "example",
 "description" : "",
 "release" : "0.10",
 ...
 "scenario" : {
  ...
  "topography" : {
    "obstacles" : [...],
    "stairs" : [...],
    ...
 }
}
```

Vadere: Core features

Back

- Easy-to-use GUI
- CLI for automation
- Shipped with different locomotion models:
  - Mature: Optimal steps model (OSM), social force model (SFM), ...
  - Experimental: Behavioral heuristics model (BHM), Reynolds' steering, ...
- JSON-based input files
- Continuous integration / deployment pipeline



Continuous integration pipeline

Getting started

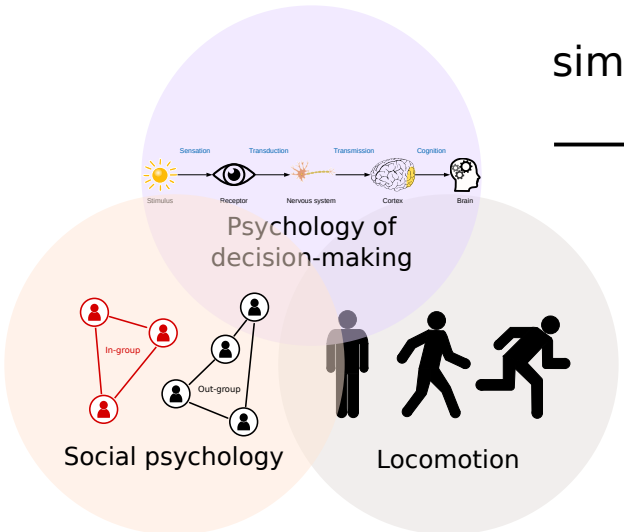
1. Download Vadere from: `http://www.vadere.org/releases/`
2. Unzip package from (1)
3. Run: `[path/to/java] -jar vadere-gui.jar`
4. In GUI, open one of the packaged demo scenarios

Simulation results

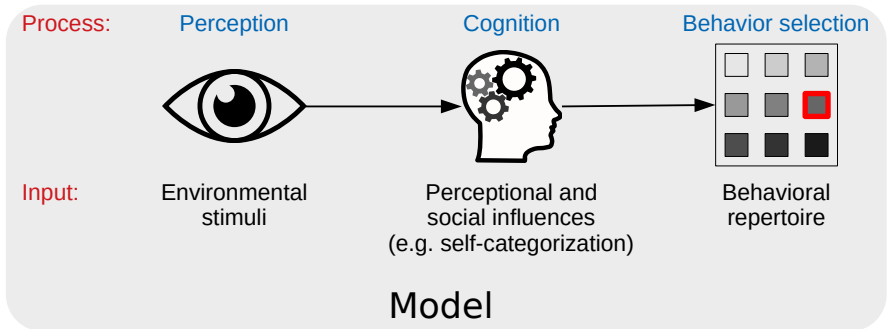
Back

I used three real-world use cases for validation:

1. Experiment: cooperative behavior
2. Perceived threat: fleeing and imitation behavior
3. Counterflowing agents: evasion behavior



simplifies to



Perceived threat: Simulation setup

- Real-world incident: stampede after false alarm
  1. People left underground station target-oriented
  2. After bang, people were fleeing
  3. Then, they were searching for a safe zone
  4. In-group members imitated fleeing behavior
     $\rightarrow$ Self-categorization theory (social psychology)
- Tools:
  - OpenStreetMap material
  - Vadere simulator with new psychology layer

Perceived threat: Simulation setup

- Real-world incident: stampede after false alarm
  1. People left underground station target-oriented
  2. After bang, people were fleeing
  3. Then, they were searching for a safe zone
  4. In-group members imitated fleeing behavior
     $\rightarrow$ Self-categorization theory (social psychology)
- Tools:
  - OpenStreetMap material
  - Vadere simulator with new psychology layer

Perceived threat: Simulation setup

- Real-world incident: stampede after false alarm
  1. People left underground station target-oriented
  2. After bang, people were fleeing
  3. Then, they were searching for a safe zone
  4. In-group members imitated fleeing behavior
     $\rightarrow$ Self-categorization theory (social psychology)
- Tools:
  - OpenStreetMap material
  - Vadere simulator with new psychology layer

Perceived threat: Simulation setup

- Real-world incident: stampede after false alarm
  1. People left underground station target-oriented
  2. After bang, people were fleeing
  3. Then, they were searching for a safe zone
  4. In-group members imitated fleeing behavior
     $\rightarrow$ Self-categorization theory (social psychology)
- Tools:
  - OpenStreetMap material
  - Vadere simulator with new psychology layer

Perceived threat: Simulation setup

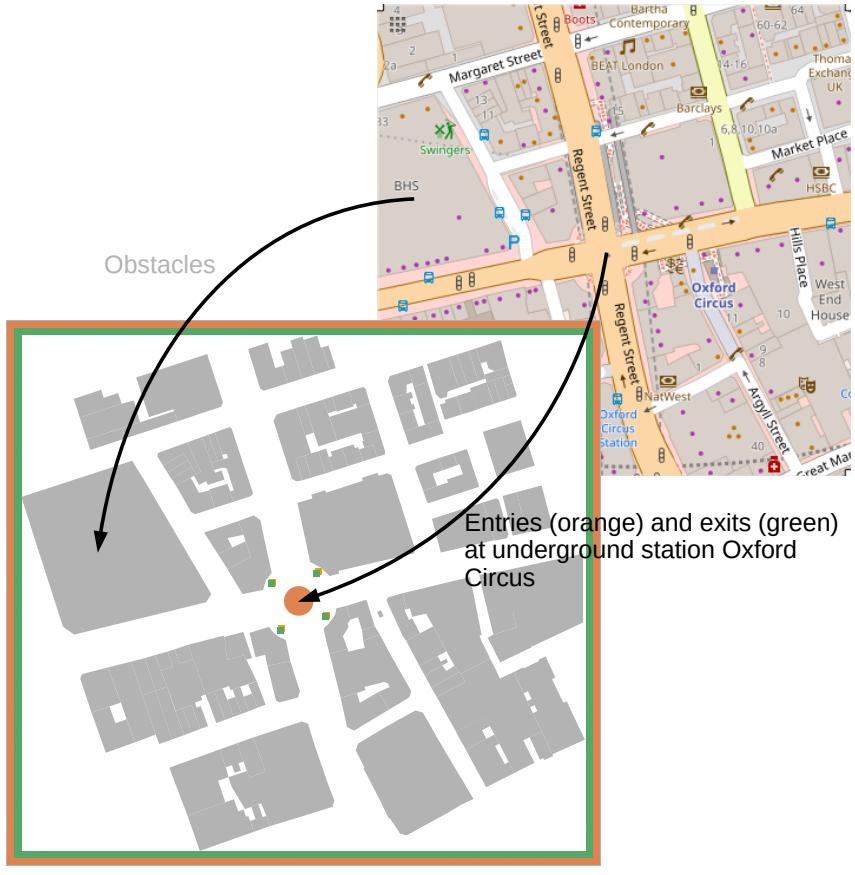
- Real-world incident: stampede after false alarm
  1. People left underground station target-oriented
  2. After bang, people were fleeing
  3. Then, they were searching for a safe zone
  4. In-group members imitated fleeing behavior
     $\rightarrow$ Self-categorization theory (social psychology)
- Tools:
  - OpenStreetMap material
  - Vadere simulator with new psychology layer

Perceived threat: Simulation setup

- Real-world incident: stampede after false alarm
  1. People left underground station target-oriented
  2. After bang, people were fleeing
  3. Then, they were searching for a safe zone
  4. In-group members imitated fleeing behavior
     $\rightarrow$ Self-categorization theory (social psychology)
- Tools:
  - OpenStreetMap material
  - Vadere simulator with new psychology layer

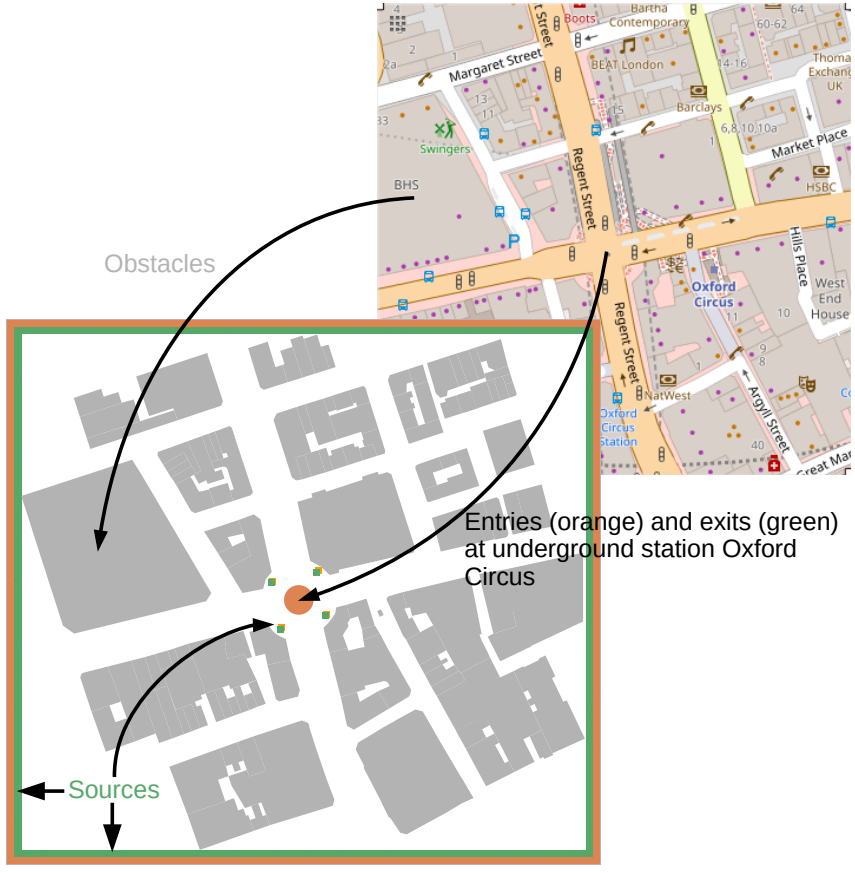
Perceived threat: Simulation setup

- Real-world incident: stampede after false alarm
  1. People left underground station target-oriented
  2. After bang, people were fleeing
  3. Then, they were searching for a safe zone
  4. In-group members imitated fleeing behavior
     $\rightarrow$ Self-categorization theory (social psychology)
- Tools:
  - OpenStreetMap material
  - Vadere simulator with new psychology layer



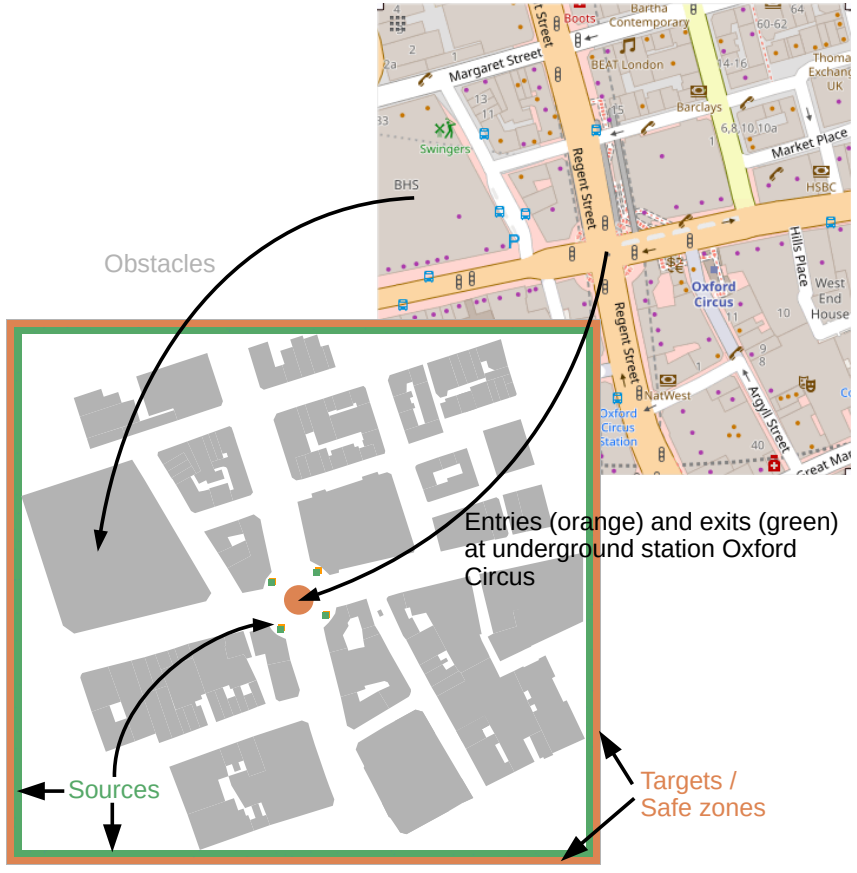
Perceived threat: Simulation setup

- Real-world incident: stampede after false alarm
  1. People left underground station target-oriented
  2. After bang, people were fleeing
  3. Then, they were searching for a safe zone
  4. In-group members imitated fleeing behavior
     $\rightarrow$ Self-categorization theory (social psychology)
- Tools:
  - OpenStreetMap material
  - Vadere simulator with new psychology layer



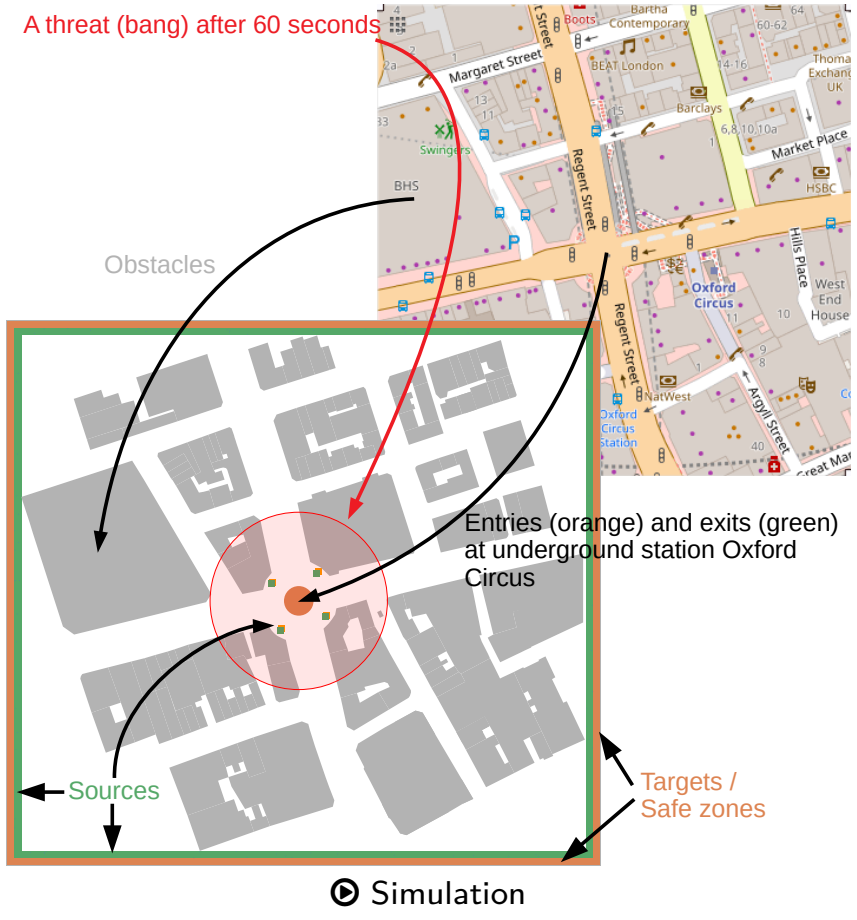
Perceived threat: Simulation setup

- Real-world incident: stampede after false alarm
  1. People left underground station target-oriented
  2. After bang, people were fleeing
  3. Then, they were searching for a safe zone
  4. In-group members imitated fleeing behavior
     $\rightarrow$ Self-categorization theory (social psychology)
- Tools:
  - OpenStreetMap material
  - Vadere simulator with new psychology layer

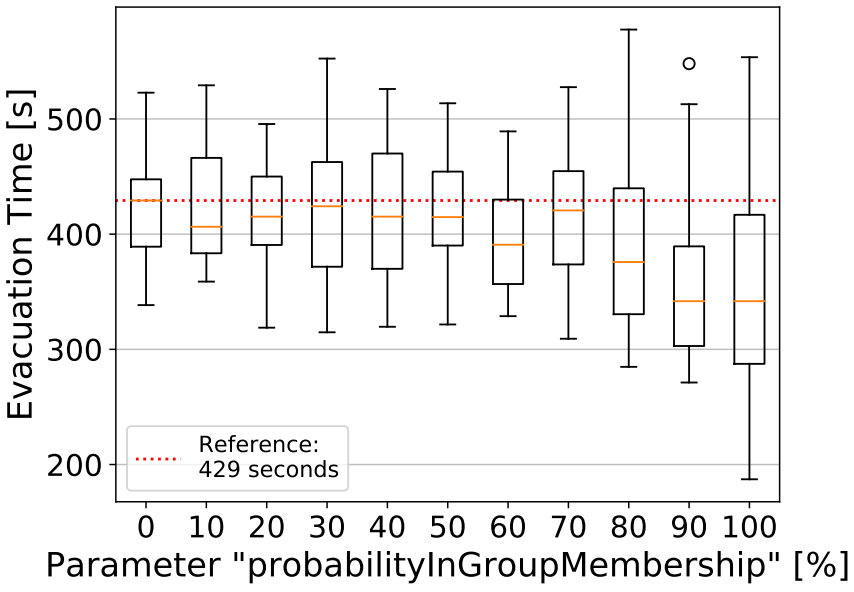


Perceived threat: Simulation setup

- Real-world incident: stampede after false alarm
  1. People left underground station target-oriented
  2. After bang, people were fleeing
  3. Then, they were searching for a safe zone
  4. In-group members imitated fleeing behavior
     $\rightarrow$ Self-categorization theory (social psychology)
- Tools:
  - OpenStreetMap material
  - Vadere simulator with new psychology layer

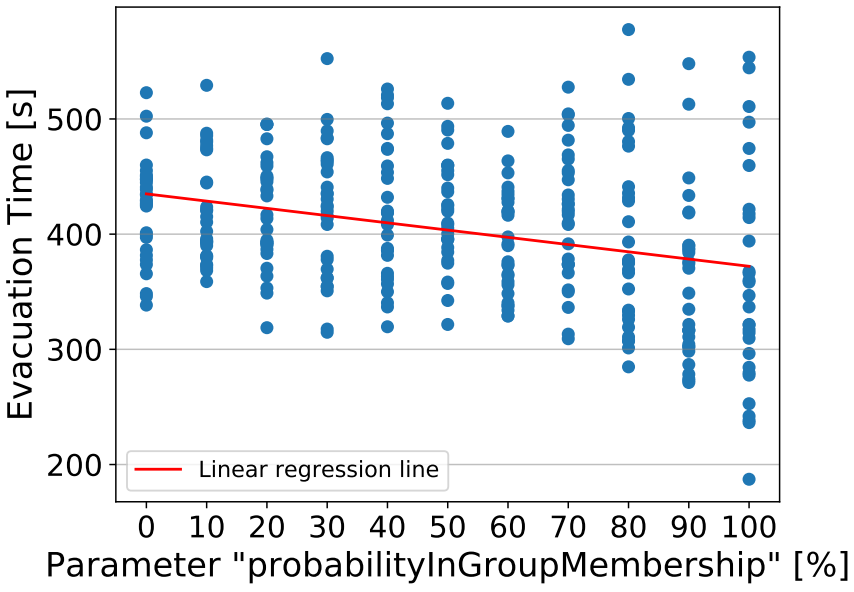


Perceived threat: Simulation results



- Quantified: Decreasing evacuation time
- The reusable psychology layer is a benefit for practitioners.
- They can scrutinize “collective flight” incidents.
  - When and how do people flee?
  - When do they follow (or ignore) others?
  - ...

Perceived threat: Simulation results

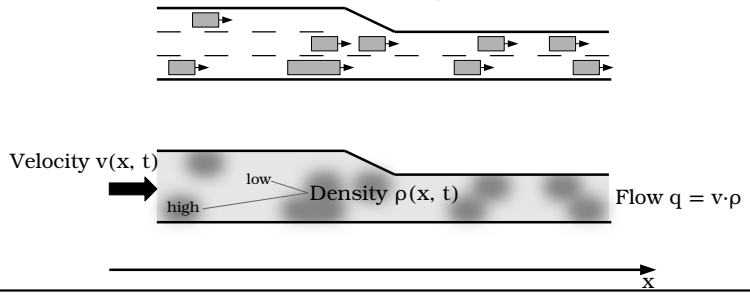


- Quantified: Decreasing evacuation time
- The reusable psychology layer is a benefit for practitioners.
- They can scrutinize "collective flight" incidents.
  - When and how do people flee?
  - When do they follow (or ignore) others?
  - ...

Modeling approaches: Pedestrian dynamics

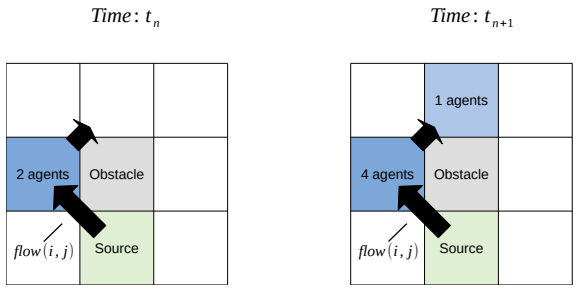
Back

Macroscopic



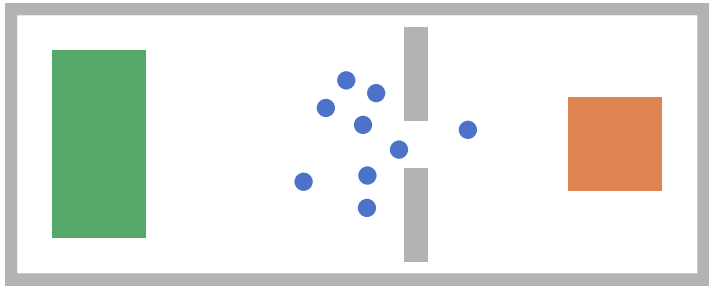
- Individuals are seen as continuum
- Usually, expressed as differential equations, i.e. density or concentration evolution over time

Mesoscopic (multi-scale)



- Simulation area is divided into cells where one cell can contain multiple agents
- Flow between cells is modeled based on properties of individuals

Microscopic

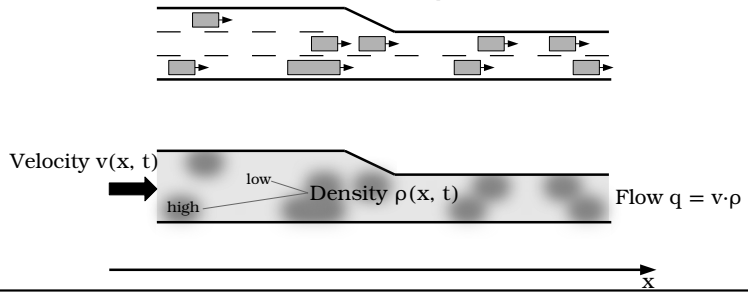


- Each agent has several attributes, e.g. preferred speed etc.
- The agent motion is modeled for each agent individually

Modeling approaches: Pedestrian dynamics

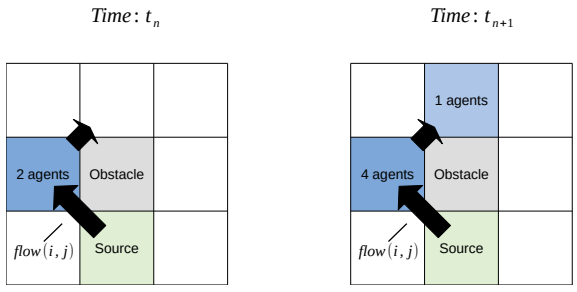
Back

Macroscopic



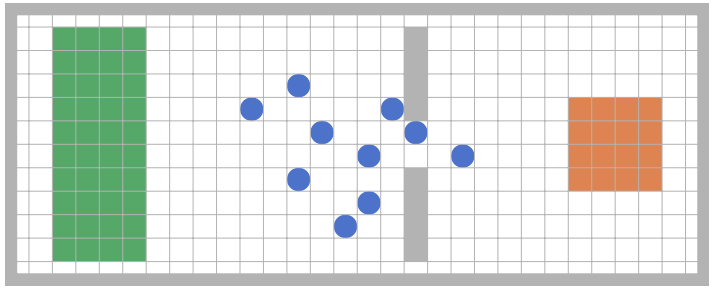
- Individuals are seen as continuum
- Usually, expressed as differential equations, i.e. density or concentration evolution over time

Mesoscopic (multi-scale)



- Simulation area is divided into cells where one cell can contain multiple agents
- Flow between cells is modeled based on properties of individuals

Microscopic



- Each agent has several attributes, e.g. preferred speed etc.
- The agent motion is modeled for each agent individually

References I

Kleinmeier, Benedikt and Gerta Köster (2020). "Experimental Setups to Observe Evasion Maneuvers in Low and High Densities". In: *Traffic and Granular Flow 2019*. Ed. by Iker Zuriguel, Ángel Garcimartín, and Raúl Cruz. Springer Proceedings in Physics. Springer. DOI: 10.1007/978-3-030-55973-1_15.

Kleinmeier, Benedikt, Gerta Köster, and John Drury (2020). "Agent-Based Simulation of Collective Cooperation: From Experiment to Model". In: *Journal of the Royal Society Interface* 17 (171), p. 20200396. ISSN: 1742-5662. DOI: 10.1098/rsif.2020.0396. URL: https://arxiv.org/abs/2005.12712.

Kleinmeier, Benedikt, Benedikt Zönnchen, et al. (2019). "Vadere: An Open-Source Simulation Framework to Promote Interdisciplinary Understanding". In: *Collective Dynamics* 4. DOI: 10.17815/CD.2019.21.

Sivers, Isabella von, Gerta Köster, and Benedikt Kleinmeier (2016). "Modelling stride length and stepping frequency". In: *Traffic and Granular Flow '15*. Ed. by Victor L. Knoop and Winnie Daamen. 27–30 October 2015. Springer International Publishing, pp. 113–120. DOI: 10.1007/978-3-319-33482-0.

Wiffers, Erik (2010). *Here, revellers flee out of the tunnel and the deadly stampede*. Agence France-Presse (AFP), Accessed: 10. December 2020. URL: https://www.spiegel.de/fotostrecke/photo-gallery-a-catastrophe-at-the-love-parade-fotostrecke-57501.html.

Zönnchen, Benedikt, Benedikt Kleinmeier, and Gerta Köster (2020). "Vadere—A simulation framework to compare locomotion models". In: *Traffic and Granular Flow 2019*. Ed. by Iker Zuriguel, Ángel Garcimartín, and Raúl Cruz. Springer Proceedings in Physics. Springer. DOI: 10.1007/978-3-030-55973-1_41.