

StationSim GCS

Keiran Suchak

October 22, 2020

1 ODD Protocol

ODD (Overview, Design Concepts, Details) is a protocol for outlining and documenting agent-based models [1, 2, 3]. The aim of such a protocol is to allow researchers to share and reproduce their models via a standardised format. The protocol consists of three components, each of which cover a number of different topics:

1. Overview providing an overview of the model, touching upon each of the following topics outlined in Section 1.1.
2. Design Concepts: detailing which of the following design concepts are relevant to the model (detailed in Section 1.2).
3. Details: elaborating on the internal mechanics of the model, which will cover each of the topics outlined in Section 1.3.

1.1 Overview

The Overview section of ODD aims to provide a cursory overview of the model that is being documented. This includes subsections on each of the following topics:

1. Purpose and patterns: What can a user expect the model to do/achieve, and what patterns might the model be fit to simulation [4]?
2. Entities, state variables and scales: What types of agents and environments are we looking to model, what variables are we using to represent them, and what spatial and temporal scales are we working at?
3. Process overview and scheduling: What processes govern the interactions in the model, and in what order to they occur?

This information should enable others to develop and outline of the what the model looks like at a functional level.

1.2 Design Concepts

When designing an agent-based model, there are a number of design concepts that are often incorporated. The following is a non-exhaustive list of the design concepts that may be involved in an agent-based model:

1. Basic principles
2. Emergence
3. Adaptation
4. Objectives
5. Learning
6. Prediction
7. Sensing
8. Interaction
9. Stochasticity
10. Collectives
11. Observation

These design concepts are detailed in greater depth in [2].

1.3 Details

The Details section of ODD aims to build upon the above sections, providing the specifics required to build a fully functioning model. This takes the form of the following topics:

1. Initialisation: What information is required to start the model with respect to defining the initial model state and calibrating parameter values?
2. Input data: What information does the model require as it runs?
3. Submodels: How do the individual processes governing model behaviour work?

2 ODD for StationSim GCS

2.1 Overview

2.1.1 Purpose and patterns

StationSim GCS is an updated version of the StationSim model. The original StationSim aimed to simulate the motion of pedestrians across a hypothetical rectangular station with 3 entrances on one side and 2 exits on the opposite side as shown in Figure 1. The new StationSim GCS also aims to simulation the motion of pedestrians across a station; however, in this case the model is based on the real-world example of Grand Central Station in New York, focusing specifically on the concourse area highlighted in Figure 2. This is reflected in the simulation environment shown in Figure 3. The environment consists of X gates which act simultaneously as both entrances and exits. Each pedestrian in the simulation is assigned an entrance and an exit and, upon entering the environment, seeks to move as directly as possible towards their assigned exit without colliding with other pedestrians. Where collisions are more likely to occur (e.g. close to entrances/exits and around solid obstacles), we typically observe crowding as population densities increase.

What
are the
patterns

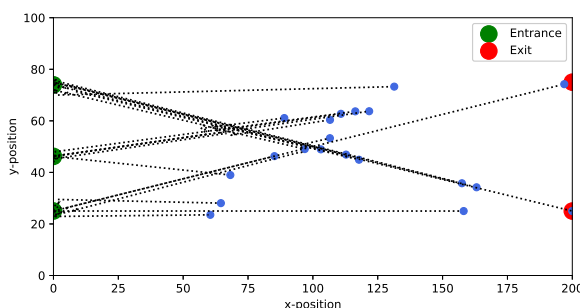


Figure 1: Layout of environment in original StationSim model.

2.1.2 Entities, state variables and scales

The StationSim GCS model is made up of 4 different types of entities:

1. Agents,
2. The environment,
3. Gates around the edge of the environment, and
4. Obstacles in the environment.

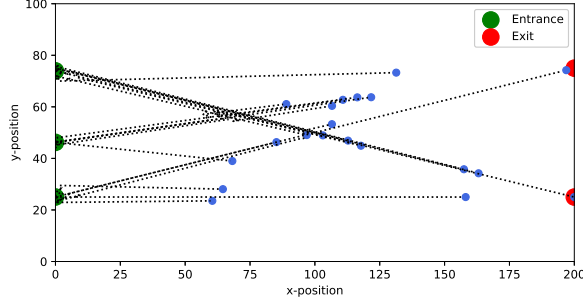


Figure 2: Layout of Grand Central Station concourse.

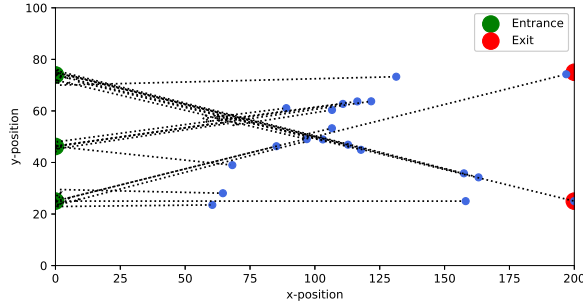


Figure 3: Layout of environment in StationSim GCS model.

These entities aim to simulate the scenario outlined in Section 2.1. The agents in this model represent pedestrians; these are portrayed as two-dimensional circular entities with finite radius. The variables pertaining to these agents can be found in Table 1. The environment in this model represents the concourse of Grand Central Station in New York; this is portrayed as two-dimensional continuous space bounded by rectangular walls within which agents may move. The model is designed such that the left-hand side of the environment represents the South side of the concourse, the right-hand side the North side, the top side the West side and the bottom side the East side. The variables pertaining to the environment entity can be found in Table 2.

Along the edge of these boundaries are located gates: one gate is located on the South side, five gates are located on the North side, two gates are located on the West side and two gates are located on the East side. The gates are points along the boundary of the environment at which agents may either enter or exit. These have specific fixed $x - y$ coordinates. Upon initialisation, each agent is provided with a start gate and end gate, from which it draws its initial location and target destination; in defining its target destination, the agent introduces some random noise to the $x - y$ coordinate

Variable Name	Description
location	Agent's x - y coordinates in 2-dimensional continuous space; bounded by the height and width of the environment
status	Agent's status; 0 indicates agent has not started, 1 indicates agent is active, 2 indicates agent has finished
size	Radius of agent's circular body
speed	Agent's speed; indicative of the distance covered by an agent in a single time-step
unique_id	Unique numerical identifier for a specific agent in a model
gate_in	Number of the gate through which of the gates the agent enters the environment ($0 \leq n \leq 9$)
gate_out	Number of the gate through which of the gates the agent exits the environment ($0 \leq n \leq 9$)
loc_desire	x - y coordinate of the agent's target destination; defined by taking the x - y coordinates of gate_out and adding some uniformly distributed random noise

Table 1: Table of state variables pertaining to gate entities.

Variable Name	Variable Type	Description
x		Agent's x-coordinate
y		Agent's y-coordinate

Table 2: Table of state variables pertaining to the environment entity.

Variable Name	Variable Type	Description
x		Agent's x-coordinate
y		Agent's y-coordinate

Table 3: Table of state variables pertaining to agent entities.

Variable Name	Variable Type	Description
x		Agent's x-coordinate
y		Agent's y-coordinate

Table 4: Table of state variables pertaining to the obstacle entity.

in order to emulate the non-zero width of the gate. The variables pertaining to the gate entities can be found in Table 3.

The environment also contains a single obstacle which represents a clock. As shown in Figure 2, this lies in the centre of the concourse. From a model architecture perspective, this obstacle is treated as a stationary agent; other agents therefore treat it as they would any other agent and make efforts to avoid colliding with it. The variables pertaining to the obstacle entity can be found in Table 4.

2.1.3 Process overview and scheduling

2.2 Design Concepts

1. Basic principles
2. Emergence
3. Adaptation
4. Objectives
5. Learning
6. Prediction
7. Sensing
8. Interaction
9. Stochasticity
10. Collectives
11. Observation

INSERT
FLOW
DIA-
GRAM
OF
OVER-
ARCH-
ING
PRO-
CESS
OF
MODEL

Which
of these
design
concepts
are in-
volved
in Sta-
tion-
Sim?

Give
a very
brief
of how
each
concept
is rele-
vant.

2.3 Details

2.3.1 Initialisation

2.3.2 Input data

2.3.3 Submodels

References

- [1] Volker Grimm, Uta Berger, Finn Bastiansen, Sigrunn Eliassen, Vincent Ginot, Jarl Giske, John Goss-Custard, Tamara Grand, Simone K Heinz, Geir Huse, et al. A standard protocol for describing individual-based and agent-based models. *Ecological modelling*, 198(1-2):115–126, 2006.
- [2] Volker Grimm, Uta Berger, Donald L DeAngelis, J Gary Polhill, Jarl Giske, and Steven F Railsback. The odd protocol: a review and first update. *Ecological modelling*, 221(23):2760–2768, 2010.
- [3] Volker Grimm, Steven F Railsback, Christian E Vincenot, Uta Berger, Cara Gallagher, Donald L DeAngelis, Bruce Edmonds, Jiaqi Ge, Jarl Giske, Juergen Groeneveld, et al. The odd protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation*, 23(2), 2020.
- [4] Volker Grimm and Steven F Railsback. Pattern-oriented modelling: a ‘multi-scope’ for predictive systems ecology. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1586):298–310, 2012.