

Assignment 2

Building modelling in AToMPM

Zohra Akbari & Randy Paredis

In this report, accompanied by the **AToMPM** source (model) files, we will describe what we did for each task, how we approached them and which issues we experienced.

1 Accompanied Files

Before we go further into detail about the implementation of the assignment, let's first discuss which files have which purpose.

assignment2.report.pdf Normally, this is the file you're reading at the moment. It contains a detailed description of each task.

BmodMM.model This is the abstract syntax model file of **Bmod**.

Bmod.defaultIcons.model This file is the concrete syntax file of our abstract syntax model.

Bmod.defaultIcons.metamodel This is the compiled concrete syntax, compiled from **Bmod.defaultIcons.model**.

task4.model This file is the non-trivial floor, modelled for task 4.

2 Task 1

For this task, we based ourselves on the **Bmod.mdepth** file from the last assignment. Although, everything seems to be the same, there were a few changes made, which we've listed below.

- The **Behaviour** (visually represented with a pentagon in our model) holds the information about the action profile and the perception levels, which is the same as in our **Bmod.mdepth** file, but this time, each

possibility was set using an `ENUM`, instead of a hardcoded value from a string.

- The `Movement Edge` changed into four `Associations`: `A_left`, `A_right`, `A_top` and `A_bottom`. This allowed the *snapping* that was required in task 3 (section 4).
- The `Fire Node` changed into a boolean field (called `On_fire`) in the `Cell Class`.

3 Task 2

As an addition to our current functionality, we added a `name` attribute to the `Room Class`, so we could display it for this task. It will be displayed on top of the room, if it is set.

As said before, the action profile and perception levels are encoded in the `Behaviour Class`.

4 Task 3

So here, we added some *snapping* functionality, as required. A `Person` will be placed roughly in the middle of a cell and using the valid `associations`, you can snap the cells validly together.

Also, the color coding, we used is as follows:

`grey` is used for a normal `Occupant`.

`blue` is used when the `Occupant` is running (and thus is aware of the `Fire`).

`cyan` is used to denote the `Occupant` has been told there is a `Fire`.

5 Task 4

We created a single floor and a rather easy one for that matter. The reason we did not make a more detailed floor is because the software was in our opinion way to finicky to do things without losing our patience.

The dangerous condition we modelled was the `OccupancyCondition` from last assignment, e.g. to warn the user if, during a simulation, it occurs that there are way to many people inside a room. It is displayed using a star, with the maximal number of occupants, denoted inside; and connected to a `Room`.

6 Man-Hours

In total, we believe we've spent 30 man-hours on this project.