SyDEVS Library

Tutorial

Autodesk Research

March 2018

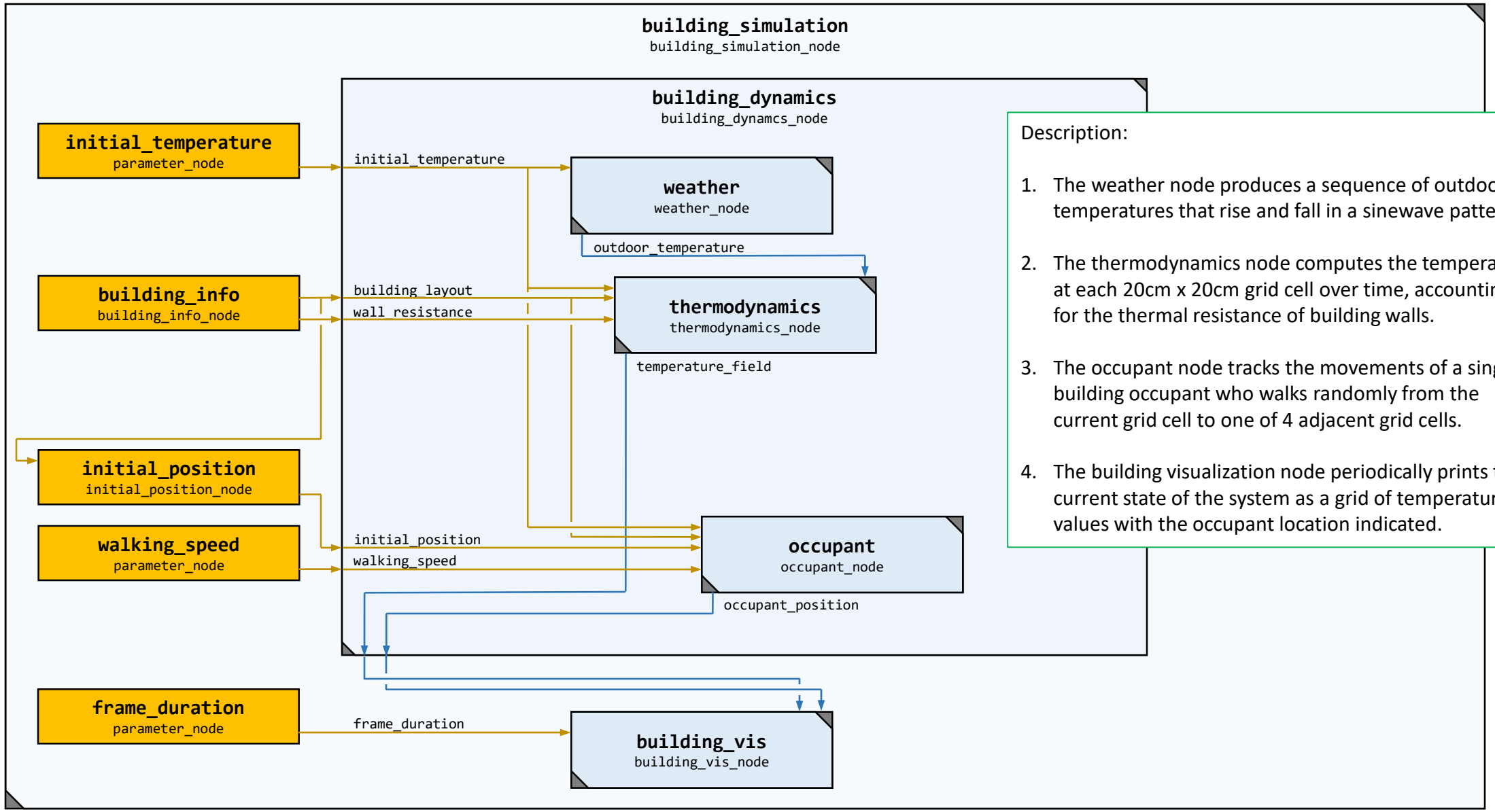
Basic Objective

Start with the **Example Simulation Model**.

(src/example/demo/building7m)

Turn it into the **Example Simulation Model with Enhancements**.

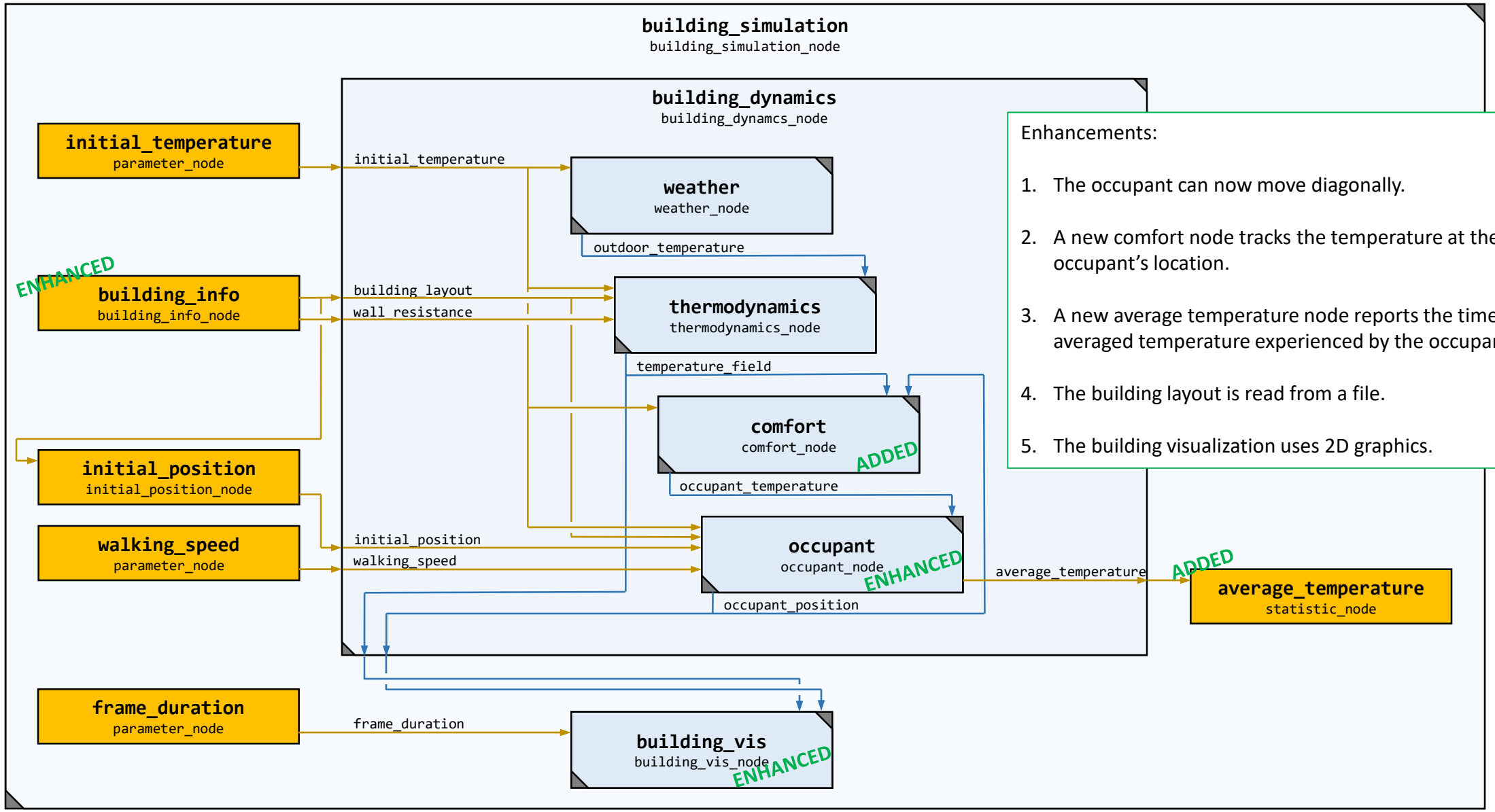
Example Simulation Model



Description:

1. The weather node produces a sequence of outdoor temperatures that rise and fall in a sinewave pattern.
2. The thermodynamics node computes the temperature at each 20cm x 20cm grid cell over time, accounting for the thermal resistance of building walls.
3. The occupant node tracks the movements of a single building occupant who walks randomly from the current grid cell to one of 4 adjacent grid cells.
4. The building visualization node periodically prints the current state of the system as a grid of temperature values with the occupant location indicated.

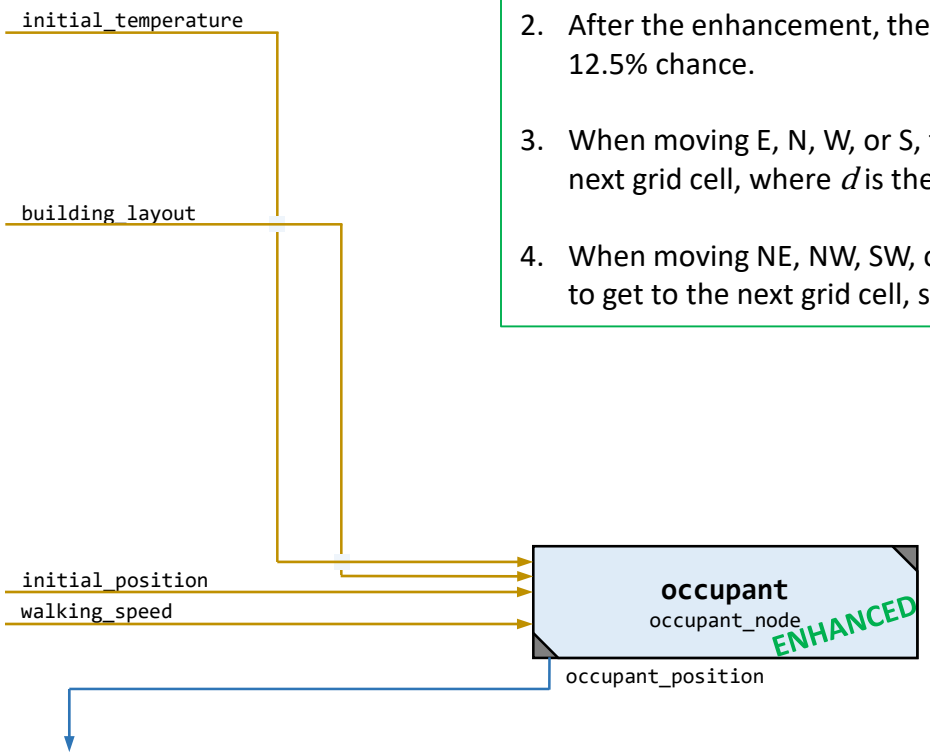
Example Simulation Model with Enhancements



Enhancements:

1. The occupant can now move diagonally.
2. A new comfort node tracks the temperature at the occupant's location.
3. A new average temperature node reports the time-averaged temperature experienced by the occupant.
4. The building layout is read from a file.
5. The building visualization uses 2D graphics.

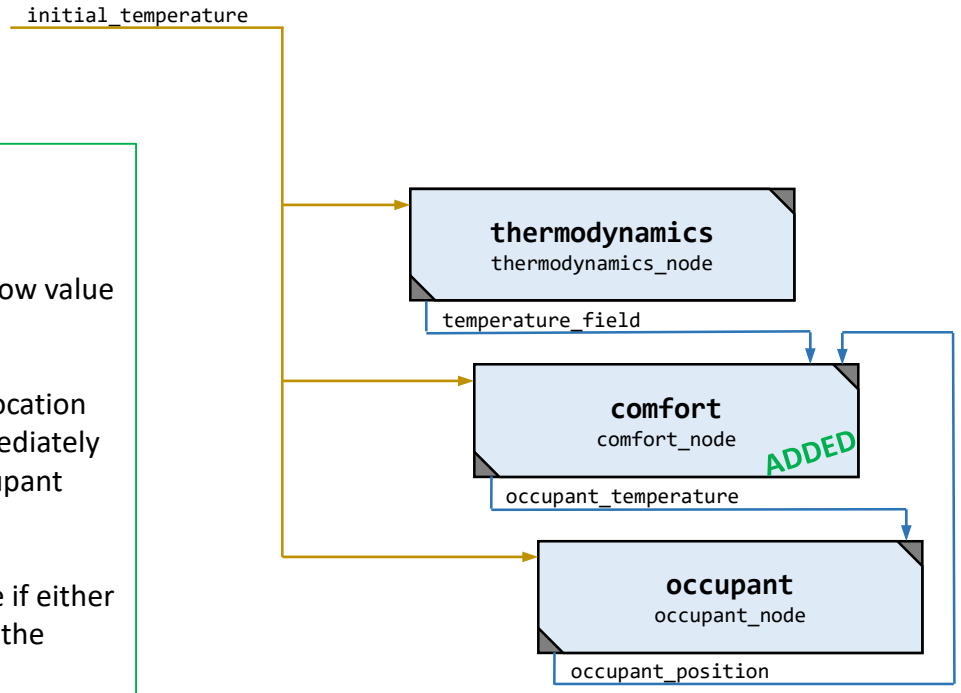
Enhancement #1: The occupant can now move diagonally.



Requirements:

1. Currently the occupant moves E, N, W, or S, each with a 25% chance.
2. After the enhancement, the occupant can move E, NE, N, NW, W, SW, S, or SE, each with a 12.5% chance.
3. When moving E, N, W, or S, the occupant takes a time of *d/walking_speed* to get to the next grid cell, where *d* is the distance between adjacent grid locations.
4. When moving NE, NW, SW, or SE, the occupant takes a time of *sqrt(2)*d/walking_speed* to get to the next grid cell, since the travel distance is that much longer.

Enhancement #2: A new comfort node tracks the temperature at the occupant's location.



Requirements:

1. The comfort node should assume the occupant's initial temperature is the flow value it takes as a parameter.
2. If the temperature at the occupant's location changes, the new temperature is immediately output and communicated to the occupant node.
3. The occupant temperature can change if either the temperature field is updated, or if the occupant's position is updated.
4. Note that if the temperature field or position changes, it does not necessarily mean that the occupant temperature has changed.

Enhancement #3: A new average temperature node reports the time-averaged temperature experienced by the occupant.

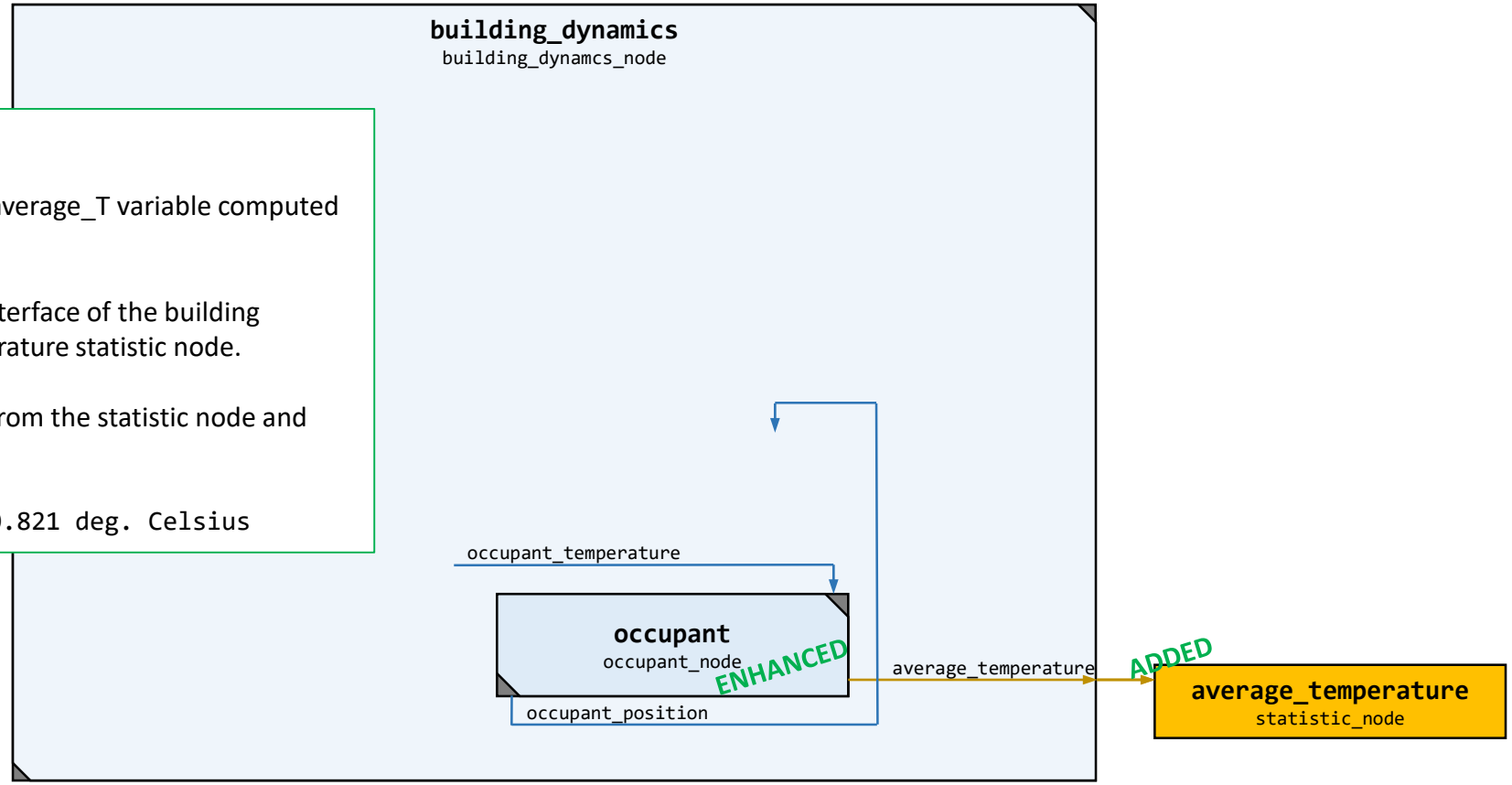
Requirements:

1. The occupant node should output the average_T variable computed in the finalization function.

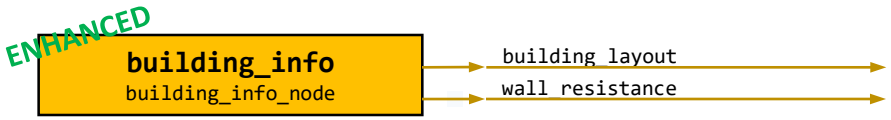
2. The flow value is passed through the interface of the building dynamics node into the average temperature statistic node.

3. In building7m.h, the value is captured from the statistic node and output as follows:

   ```
   Average Occupant Temperature: 20.821 deg. Celsius
   ```



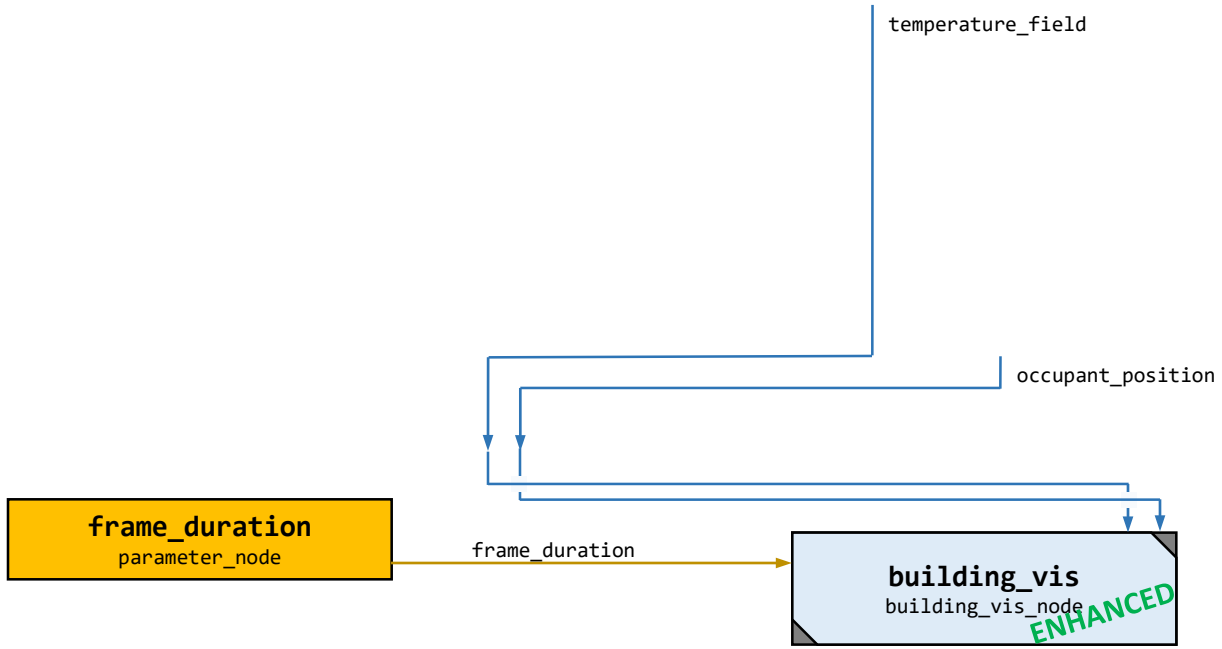
Enhancement #4: The building layout is read from a file.



Requirements:

1. Currently the building layout is just a square generated using a hard-coded routine in the building info node.
2. The layout is encoded in a 35 x 35 cell grid. With 20cm cells, the entire area is 7m x 7m.
3. Cell values of 0 represent indoor space; cell values of 1 represent walls; cell values of -1 represent outdoor space.
4. The hard-coded routine should be deleted, and the building layout should be read from the file “building7m.png”.
5. In the image, white pixels are indoor space; black pixels are walls; grey pixels are outdoor space.
6. Optional further enhancement: Add a new parameter node that supplies the filename to the building info node.

Enhancement #5: The building visualization uses 2D graphics.



Requirements:

1. Currently the building vis node periodically prints a grid of text indicating the temperature at every point except where the occupant is located.
2. The node should instead produce an animation of the results using 2D graphics.
3. For example, a 40-second video could be produced with a frame duration of 0.25 seconds (simulated time) and a frame rate of 30 fps.
4. Alternatively, the animation could be shown as the simulation is running. This would require simulated time to be synchronized with wallclock time, possibly using a sleep() instruction in the building vis node.

Tutorial Instructions

1. Run building7m.exe.

2. Open building7m.h.
   a) Re-run after inserting the following line:
   `sim.top.building_dynamics.thermodynamics.outdoor_temperature_input.print_on_use();`
   b) Re-run after inserting the following line:
   `sim.top.building_dynamics.occupant.print_on_event();`
   c) Re-run after inserting similar event notification and port value printing lines.

3. Undo changes above, and open weather_node.h.
   a) Find a location in the Planned Event Handler beneath the reassignment of the “rate” state variable.
   b) Re-run after inserting the following line:
   `print(tostring(rate));`

4. Undo changes above, and open building_simulation_node.h.
   a) Re-run after changing the frame duration parameter value from `duration::inf()` to `30_s`.

5. Implement at least the first 3 enhancements, and ideally all 5.