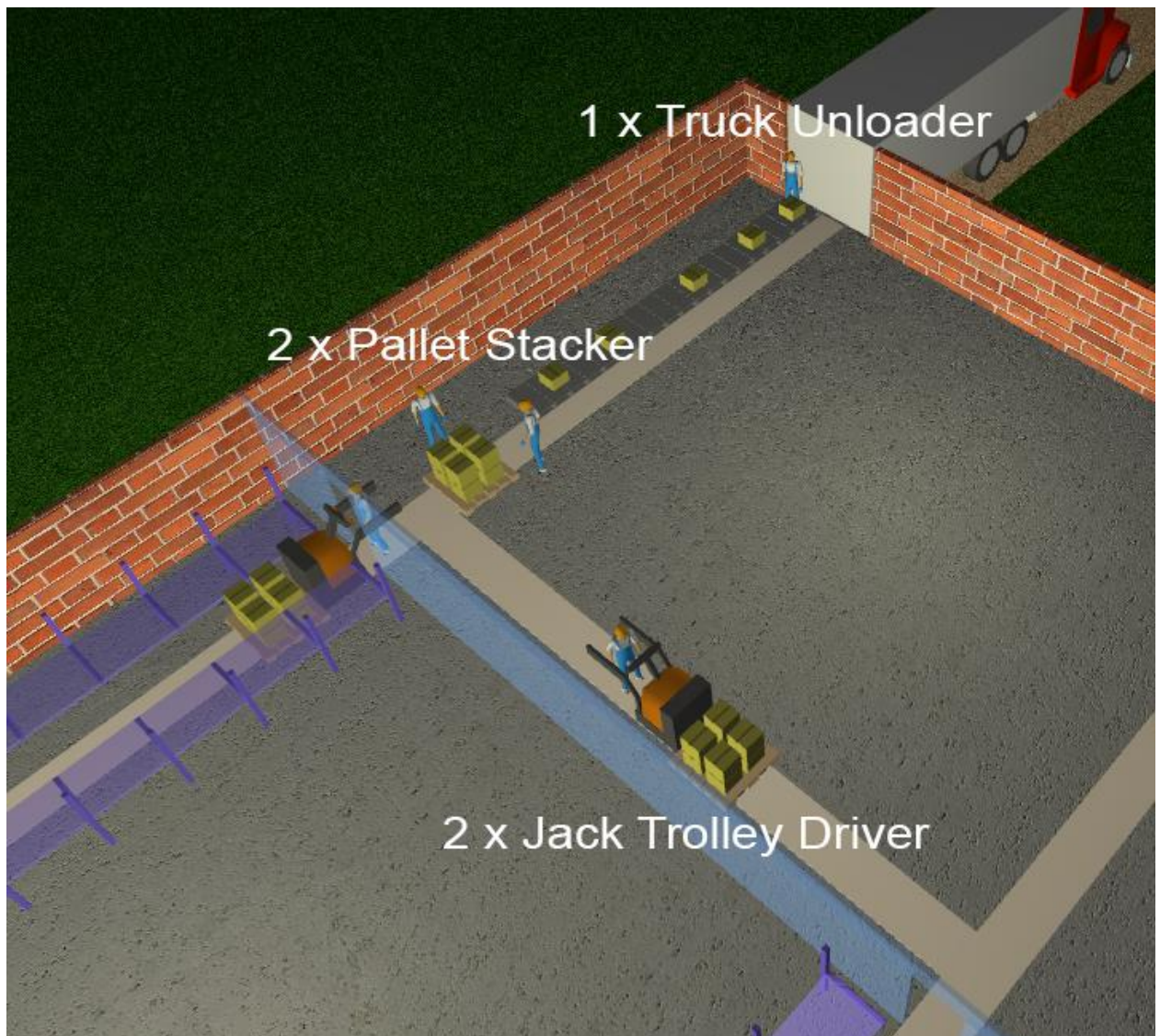


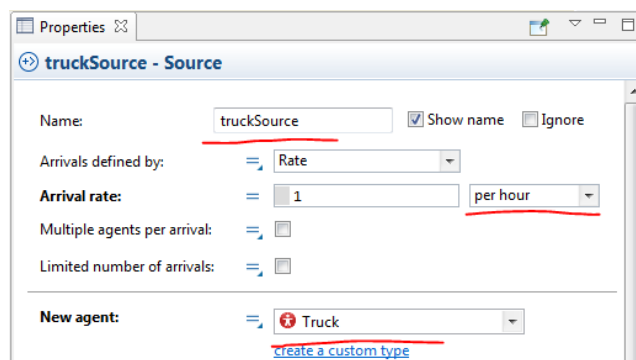
Simple unloading process in a warehouse



This tutorial shows how to create and visualize a simple unloading scenario in a warehouse. Let us read the description of the scenario and do the corresponding actions in AnyLogic on the fly:

Trucks. Trucks arrive at the warehouse at a continuous rate of 1 per hour.

- [create a new agent type Truck](#)
- drop a **Source** block and change its properties:

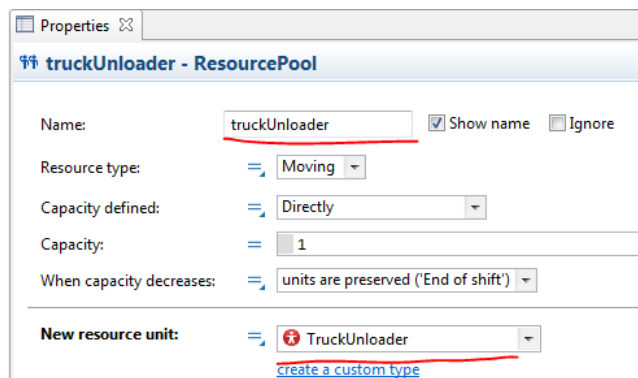
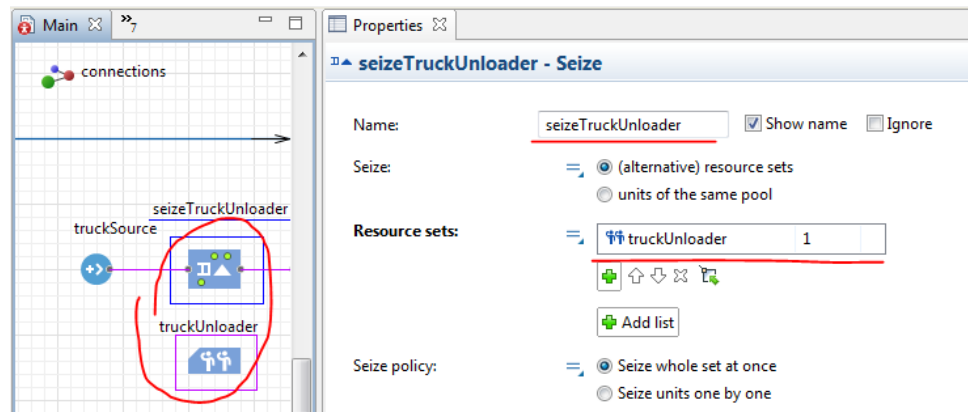


The first truck arrives at simulation time 0.

- add the following code to the **On startup** field of the **Main** agent:
`truckSource.inject(1);`

Only 1 truck can be seized at a time by a Truck Unloader.

- create a new resource type **TruckUnloader**
- drop **Seize** and **Resource Pool** blocks for the truck unloader and change their properties:

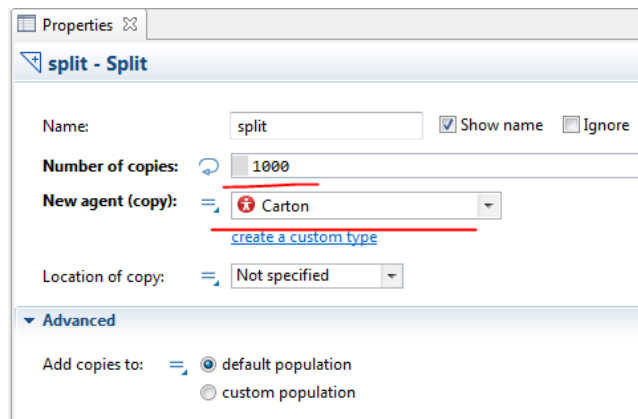


A maximum of 4 trucks are allowed to queue at the warehouse.

- set the **Queue capacity** of the **seizeTruckUnloader** to 4
- go to the properties of the **truckSource**, unselect the **Forced pushing** option and choose the **Wait in this block** mode from the drop-down list

Trucks arrive with a 1000 cartons each.

- create a new agent type **Carton**
- drop a **Split** block and change its settings:



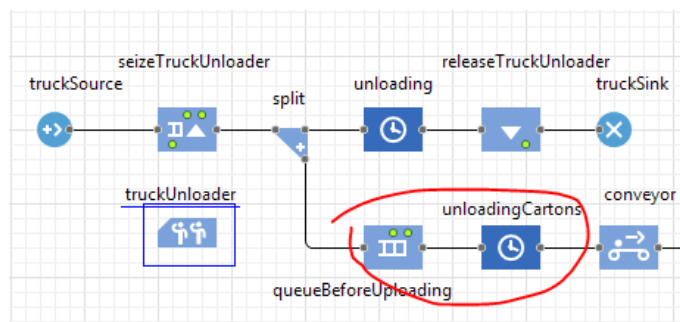
The cartons are split 500/500 between small and large cartons. The 500 large cartons are stored at the back of the truck and unloaded last (i.e. Truck driver side).

- add a parameter, e.g. **small**, of the type **Boolean** to the **Carton** diagram, set the **Initial value** to **true**
- add the following code to the **On exit copy** field of the **split**:

```
if(indexCopy > 500)
    agent.small = false;
```

Truck Unloader. Cartons are unloaded one size at a time. First the 500 small cartons and lastly the 500 large cartons are unloaded. Small cartons take 3 seconds and large cartons take 5 seconds to be unloaded by the Truck Unloader onto the conveyor belt.

- drop a **Queue** block, set the **Capacity** to 1000
- drop a **Delay** block, set the expression "agent.small ? 3 : 5" as the **Delay time**

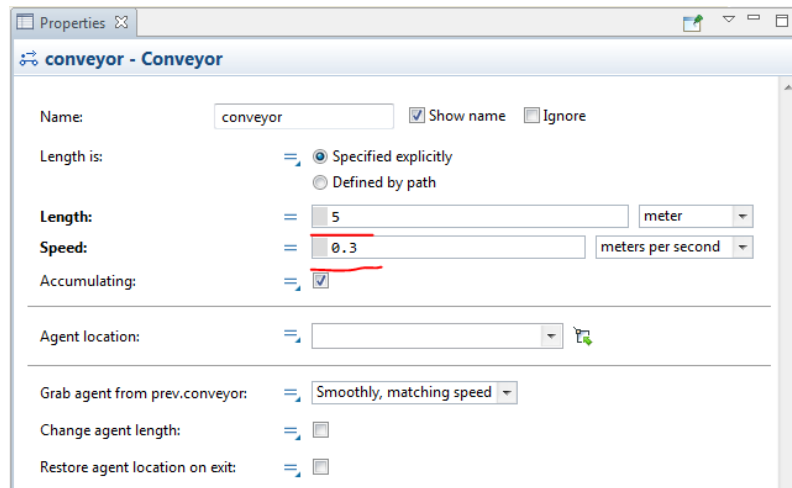


Note the **unloading** block was added as a buffer for trucks that are being unloaded by the truck unloader. A truck is released from the block when unloading process is finished. It is required to execute the code below as the **On exit** action of the **unloadingCartons**:

```
if(queueBeforeUploading.size() == 0)
    unloading.stopDelay(unloading.get(0));
```

Conveyor. The conveyor length is 5 meters. Conveyor runs at a speed of 0.3 m/s.

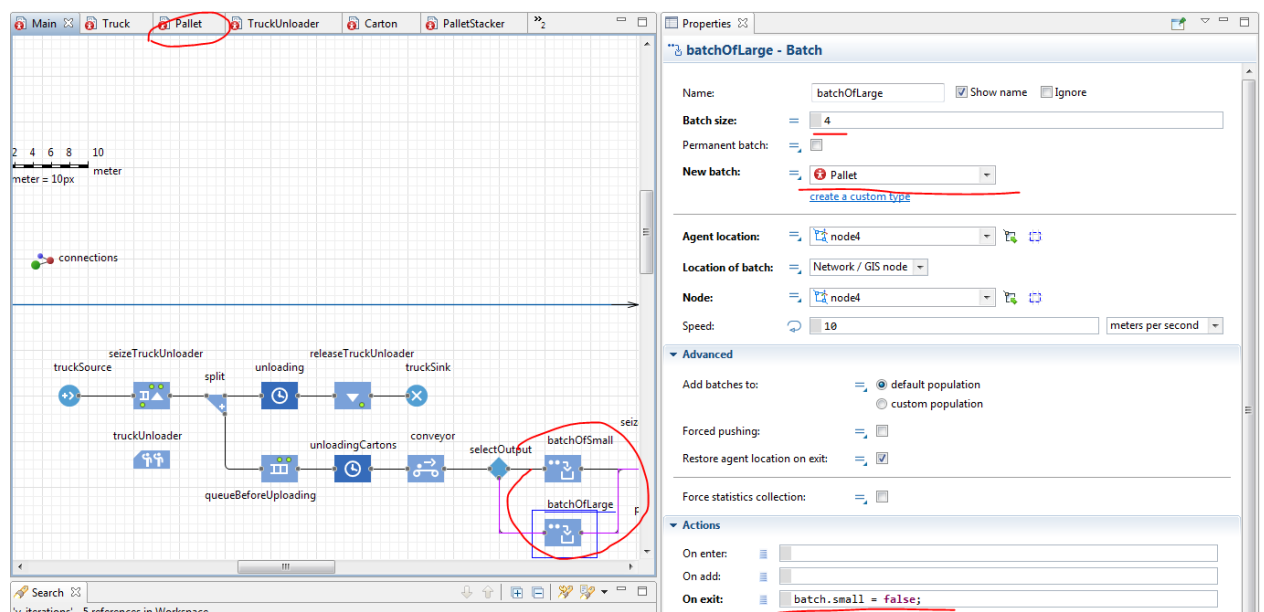
- drop a **Conveyor** block, change its properties as follows:



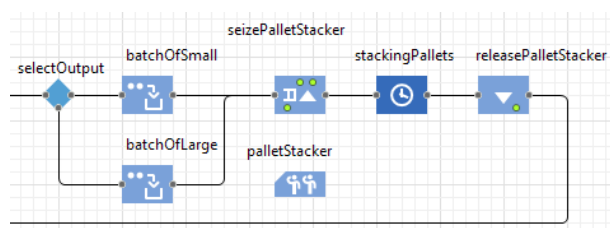
Pallet Stackers. Pallets are stacked with a maximum of 6 small cartons and 4 large cartons per pallet. Pallet Stackers stack the pallets at a rate of 5 seconds/small carton and 7 seconds/large carton.

Let's represent pallets as batches of cartons.

- drop two **Batch** blocks for small and large cartons
- create a new agent type **Pallet** with a parameter **small** (**true** by default). It can be changed to **false** in the **On exit** field of the **batchOfLarge**



- create a new resource type **PalletStacker**
- drop **Seize** and **ResourcePool** blocks for the pallet stackers
- drop a **Delay** block for stacking pallets, set the expression "agent.small ? 5 : 7" as the **Delay time**



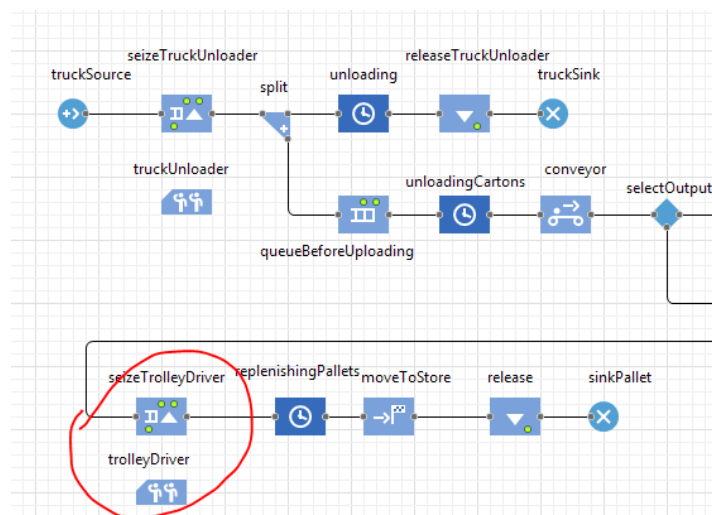
If the Pallet Stackers cannot keep up with the conveyor and Truck Unloader they will stop the conveyor belt.

The conveyor stops automatically if there is no space in front of it because of pull protocol. In case of pull protocol an agent is sent to the next block only when it can accept the agent.

Pallets are replenished by the Jack Trolley Drivers.

Jack Trolley Drivers. Once the pallet has been stacked by the Pallet Stacker, the Jack Trolley Driver moves the stacked pallet a short distance away, first replenishes the Pallet Stacker with a new pallet from a pallet pile (assume unlimited), moves the stacked pallet to storage and then returns with an empty jack trolley to the Pallet Stacker.

- create a new resource type **TrolleyDriver**
- drop **Seize** and **ResourcePool** blocks for the trolley drivers



Time to replenish pallet is 5 seconds.

- drop a **Delay** block, set the **Delay time** to 5 seconds

Time to move stacked pallet to storage is 10 seconds.

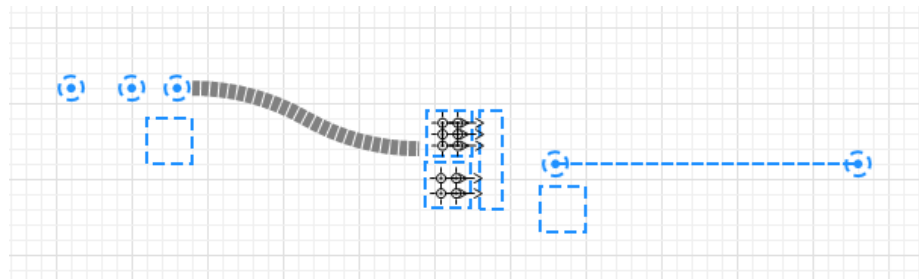
- drop a **MoveTo** block, set the **Trip time** to 10 seconds

Time to return with empty jack trolley to pallet stacker is 7 seconds.

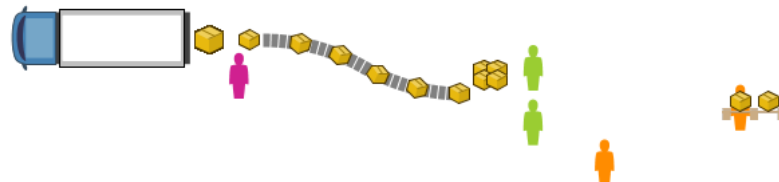
You can implement this by changing jack trolley speed or specifying jack trolleys' home location in 7 seconds drive from the store (the **Return to home location** option is selected by default in the properties of a **Release** block).

Note. Use animations.

Feel free to use any markup shapes for animating the unloading process. For example:



Design time



Runtime

Do not forget to specify markup elements in the **Agent location** parameter of the corresponding flowchart blocks.

Simulation run length = 8 hr's.

Or 480 minutes if model time units is set to minutes:

Scenario's To Be Tested. For each scenario measure change in the total number of cartons placed in storage by the Jack Trolley Drivers. Change the conveyor speed from 0.3 to 0.5 m/s.

The conveyor speed should be defined by a parameter to be able to test different scenarios.