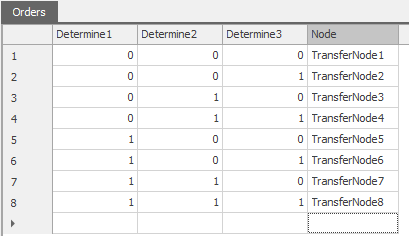
**UserDefineVehicleRequestMove Model Description**

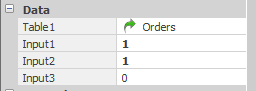
This model is going to show 3 methods to request a vehicle move to a transfernode and park there without a modelentity. The three methods all utilize the repeated group which can map the data between models.

The purpose of this example model is to let you to move the vehicle without a modelentity.

Below is the data table in this example. You can see there are three determine columns which are used to indicate which transfernode the vehicle should move to.



For the vehicle object, Under the ‘Data’ category, you can see three input variables. As shown below, the matching logic is Iuput1 – Determin1; Input2 – Determin2; Input3 – Determine3. Based on the inputs in the figure, the vehicle should move to TransferNode7

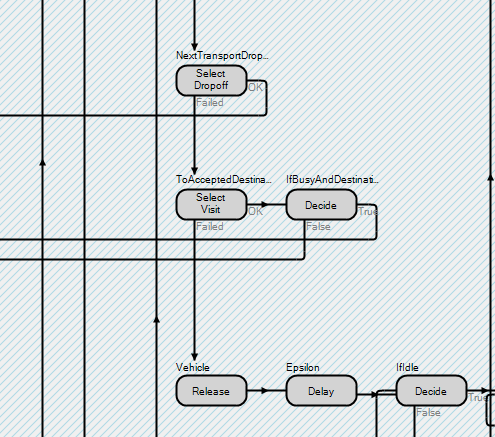


When you run the models, after clicking the button “Move” above the vehicle, you will see the vehicle move to TransferNode7 and then park there.

**Model1：**

In model1, we move the vehicle by **letting vehicle seize itself** and set a destination to it. When applying this method, you need to manually release the vehicle object in its VisitRequestQueue because unlike the modelentity, Simio currently does not provide a default mechanism to unload itself. ( I guess that’s because the vehicle is defined for loading and unloading modelentities but not itself)

So, if you do not manual release the vehicle, the vehicle will keep moving forever because the structure of **‘onVisitingNode’** process in Vehicle object. The exact logic is shown below:

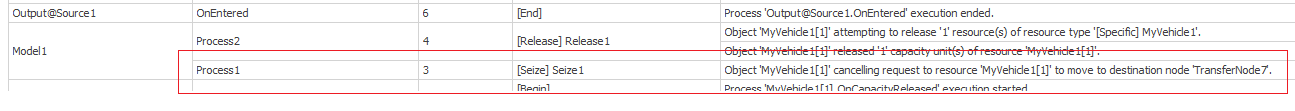


You can see, after unloading and loading(which is the previous logic but not shown in this figure), before release the vehicle, the **‘SelectVisit’** step will pick an entity in the vehicle visitrequestqueue and move to its destination. So, that’s why the vehicle keep moving.

Back to Model1, in Process banner, you will see two processes: process1 and process2. When you click the move, the process1 will be triggered and then before seizing the vehicle, process1 will execute process2. The process2 will wait a user defined amount of time to release the vehicle (when it moves).

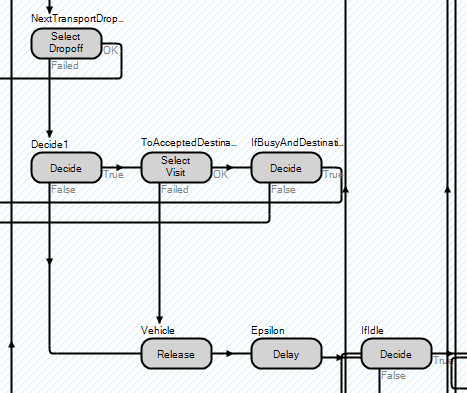
The reason I did not put a release step just after the seize step is because the seize step is not complete. Currently, I have no idea about what are the requirements for completing the seize step (Or more precisely, the move step )

The figure below shows what happen to the seize step. As you can see, it is cancelled rather than finished.



**Model2:**

In this model, I add a Decide step in the **onVisitingNode’** process in Vehicle object. The condition is **“(VisitRequestQueue.NumberWaiting != 1) || (VisitRequestQueue.LastItem.Is.ModelEntity)”.**  I show the part below.



This decide step will check whether the entity in the visitrequestqueue is modelentity or not. If there is only one entity and it is not modelentity in the visitrequestqueue, just release the vehicle. So when you run the model, you’ll find the vehicle behavior quite similar with Model1. The only difference is when to release the vehicle.

**Model3:**

Model3 provides an alternative way to let vehicle move. Rather than seizing the vehicle, I use transfer step to let the vehicle transfer to the freespace and use travel step to let the vehicle move to the destination. The only problem is I do not know how to fix the vehicle heading when you let the vehicle follow the network path. It seems the vehicle’s heading will be assigned to the nearest path.