# SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE

NAAC ACCREDITED 'A' GRADE COLLEGE

(ISO 9001:2015 CERTIFIED INSTITUTION)

NERUL, NAVI MUMBAI - 400706


**Seat No:** 01

**Certified that** B Joshnasagari Vijayasarathi Deepika

**Of Class** Msc(CS) Part - 2 **has duly completed the practical**

**course in the subject** Simulation &Modelling

**during the academic year** 2021-22 **as per the syllabus**

**prescribed by the University of Mumbai.**

**Subject Teacher** **External Examiner**

**Head of Department** **Principal**

# INDEX

| SR.NO | TITLE | SIGN |
|---|---|---|
| 1 | **Design and develop agent based model by**<br>• **Creating the agent population**<br>• **Defining the agent behavior**<br>• **Add a chart to visualize the model output.** | |
| 2 | **Design and develop agent based model by**<br>• **Creating the agent population**<br>• **Defining the agent behavior**<br>• **Adding a chart to visualize the model output**<br>• **Adding word of mouth effect**<br>• **Considering product discards**<br>• **Considering delivery time** | |
| 3 | **Design and develop agent based model by**<br>• **Creating the agent population**<br>• **Defining the agent behavior**<br>• **Adding a chart to visualize the model output**<br>• **Adding word of mouth effect**<br>• **Considering product discards**<br>• **Consider delivery time**<br>• **Simulating agent/Consumer impatience**<br>• **Comparing model runs with different parameter values** | |
| 4 | **Design and develop System Dynamic model by**<br>• **Creating a stock and flow diagram**<br>• **Adding a plot to visualize dynamics**<br>• **Parameter Variation**<br>• **Calibration**<br>**[ Use a case scenario like spread of contagious disease for the purpose]** | |
| 5 | **Design and develop a discrete-event model that will simulate process by**<br>• **Creating a simple model**<br>• **Adding resources**<br>• **Creating 3D animation**<br>• **Modeling delivery**<br>**[Use a case situation like a company's manufacturing and shipping].** | |
| 6 | **Design and develop time-slice simulation for a scenario like airport model to design how passengers move within a small airport that hosts two airlines, each with their own gate. Passengers arrive at the airport, check in, pass the security checkpoint and then go to the waiting area. After boarding starts, each airline's representatives check their passengers' tickets before they allow them to board.** | |

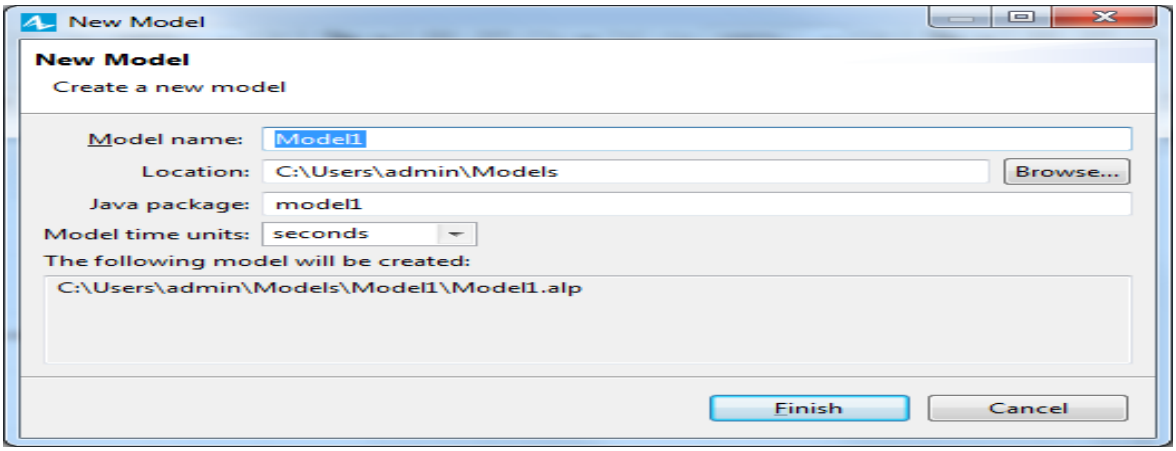
# Practical No. 1

Design and develop agent based model by

- Creating the agent population
- Defining the agent behavior
- Add a chart to visualize the model output.

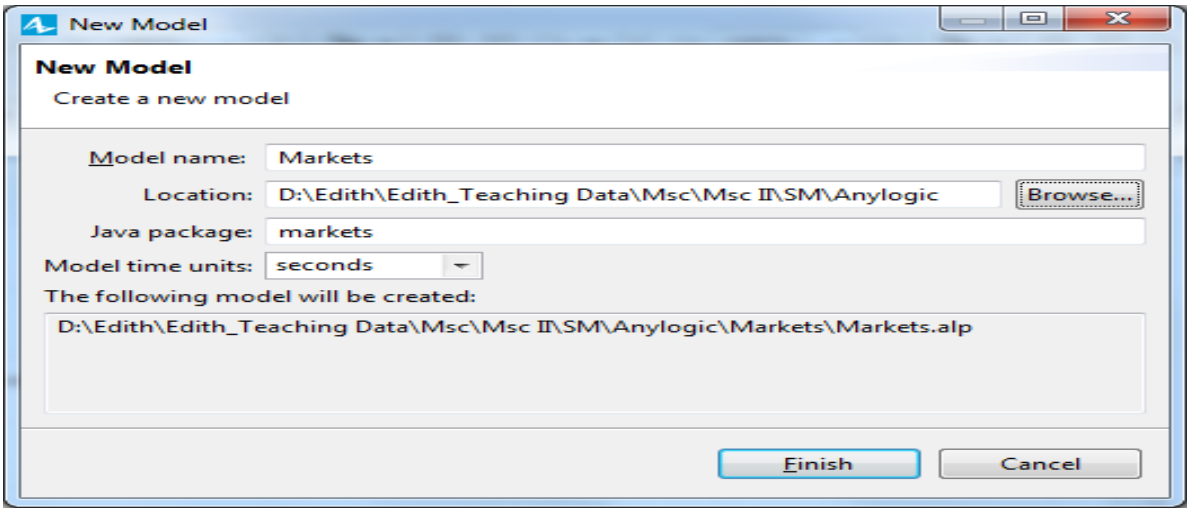
[Use a case scenario like grocery store, telephone call center etc for the purpose].

Close the Welcome page, and create a new model by selecting **File** > **New** >**Model** from AnyLogic's main menu. The **New Model** wizard will open.

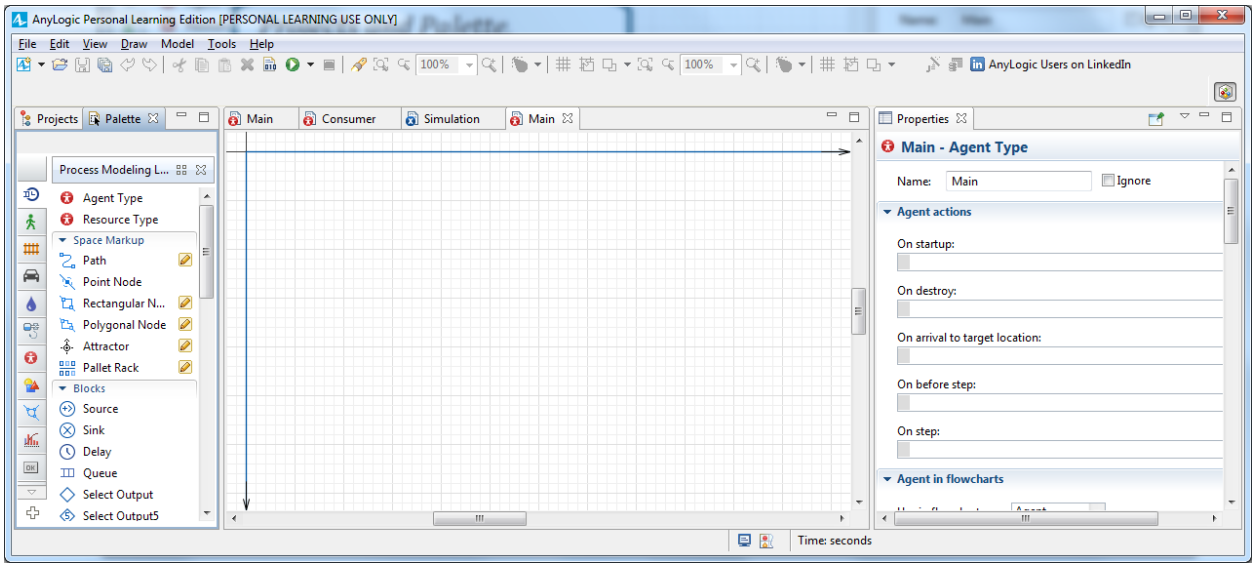



In the **Model name** box, enter the new model's name: Market.

**3.** In the **Location** box, select the folder where you want to create the model. You can browse for a folder by clicking **Browse** or type the name of the folder you want to create in the **Location** box.

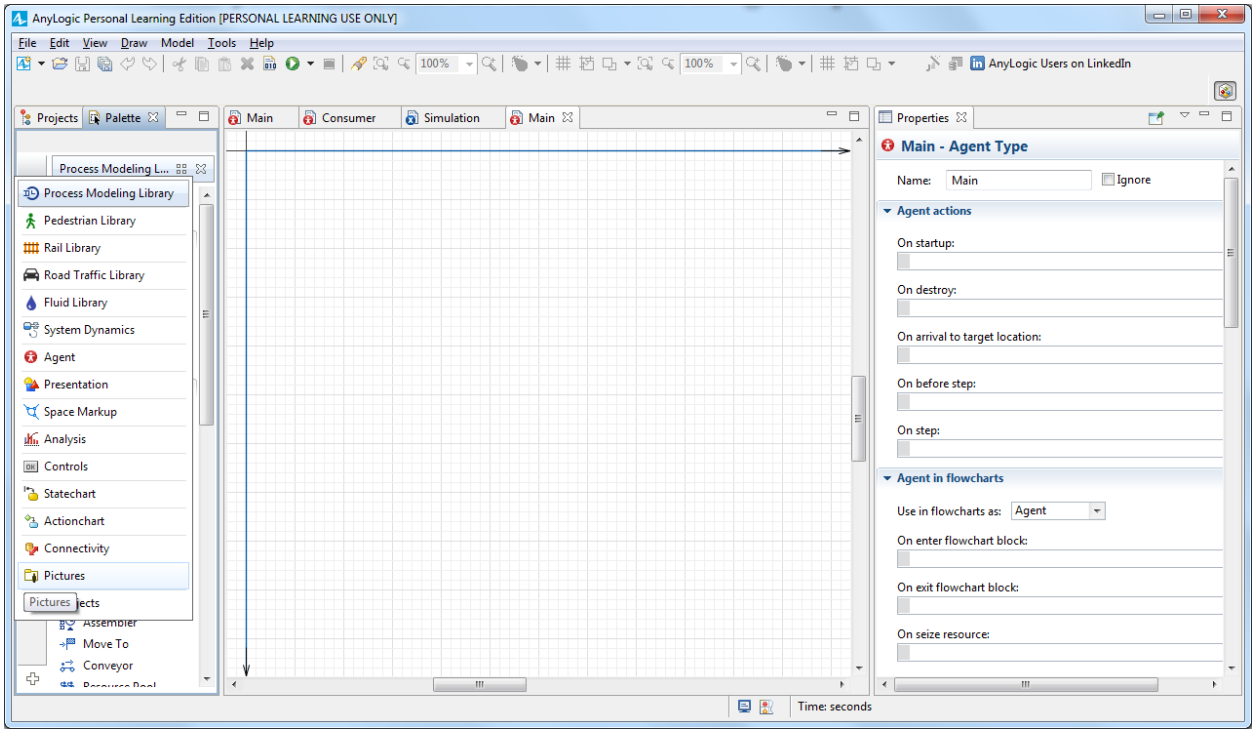


Click **Finish**.

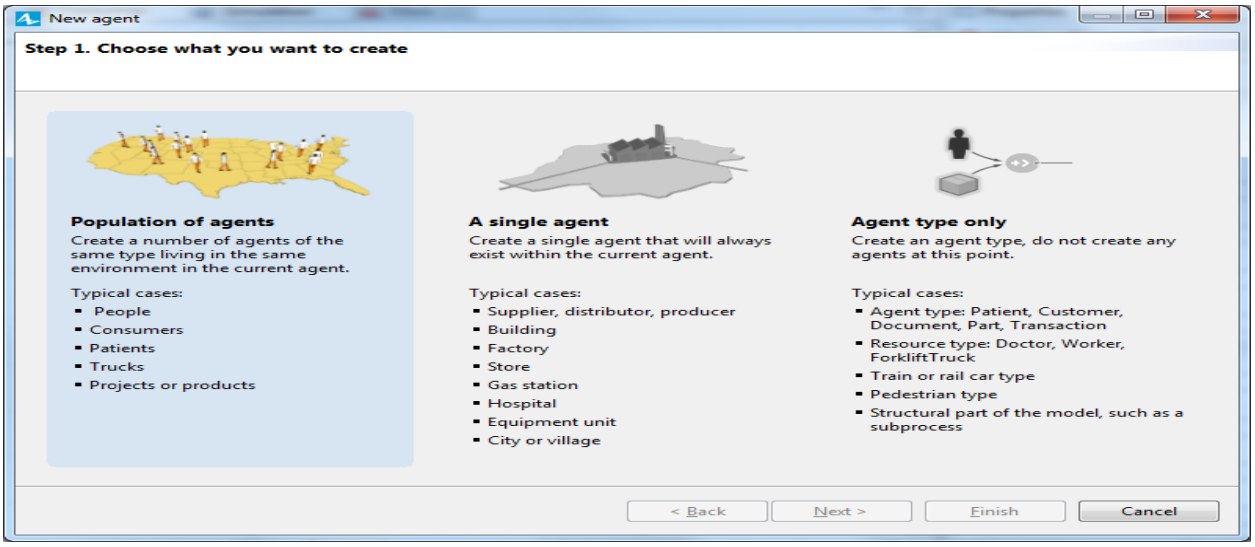



Our model has one agent type, Main. To add consumers, we'll need to create an agent type to represent consumers, and then create an agent population made up of instances of this consumer agent type. In AnyLogic 7, you can use the helpful **New agent** wizard to create agents.

**6.** We want to add a new model element, but we first need to switch to the **Palette** view by clicking the **Palette** tab.



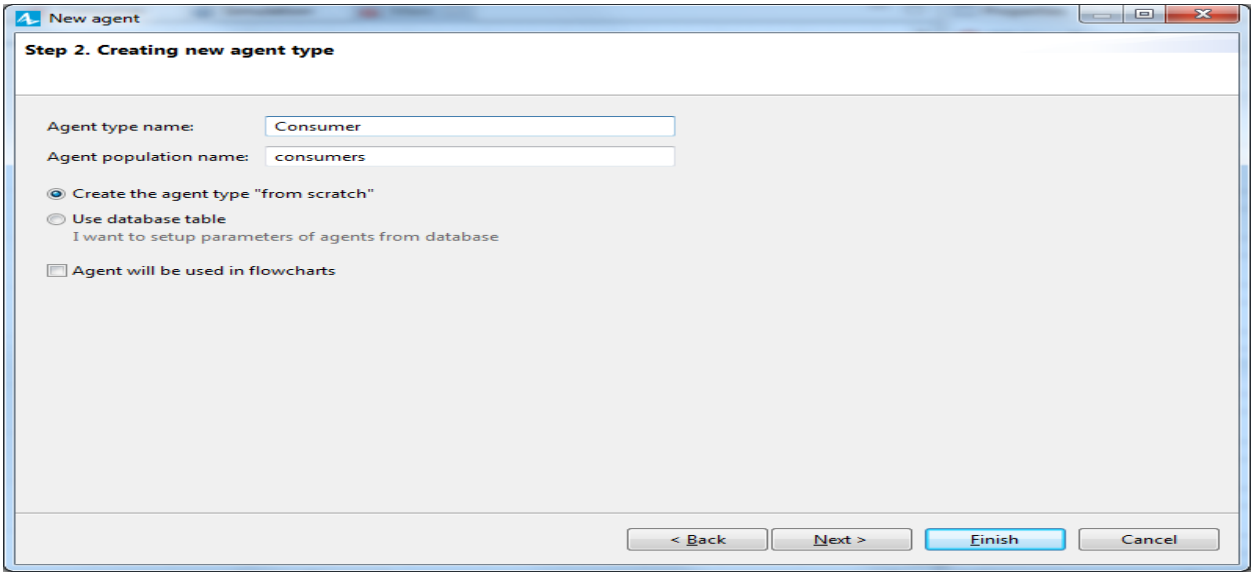
Drag the **Agent** from the **Agent** palette on to the Main diagram, and the **New agent** wizard will open.




On the **Step 1. Choose what you want to create** page, select the option that best meets your needs. Since we want to create multiple agents of the same type, select **Population of agents** and click **Next**.

10

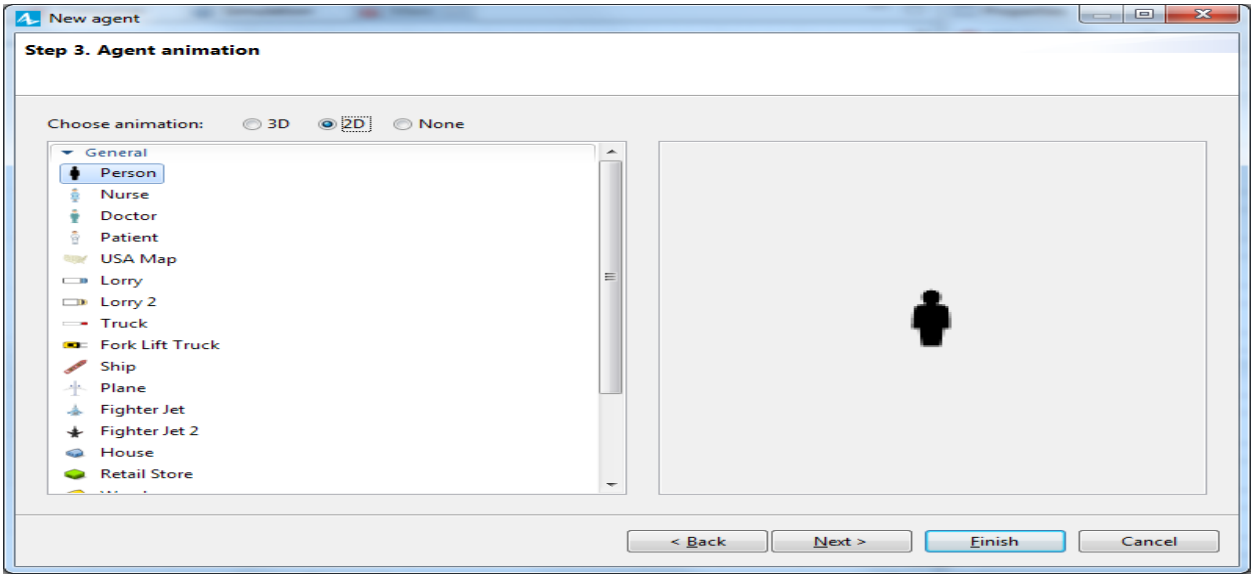
On the **Step 2. Creating new agent type** page, in **Agent type name** box, type Consumer. The information in the **Agent population name** box will automatically change to consumers

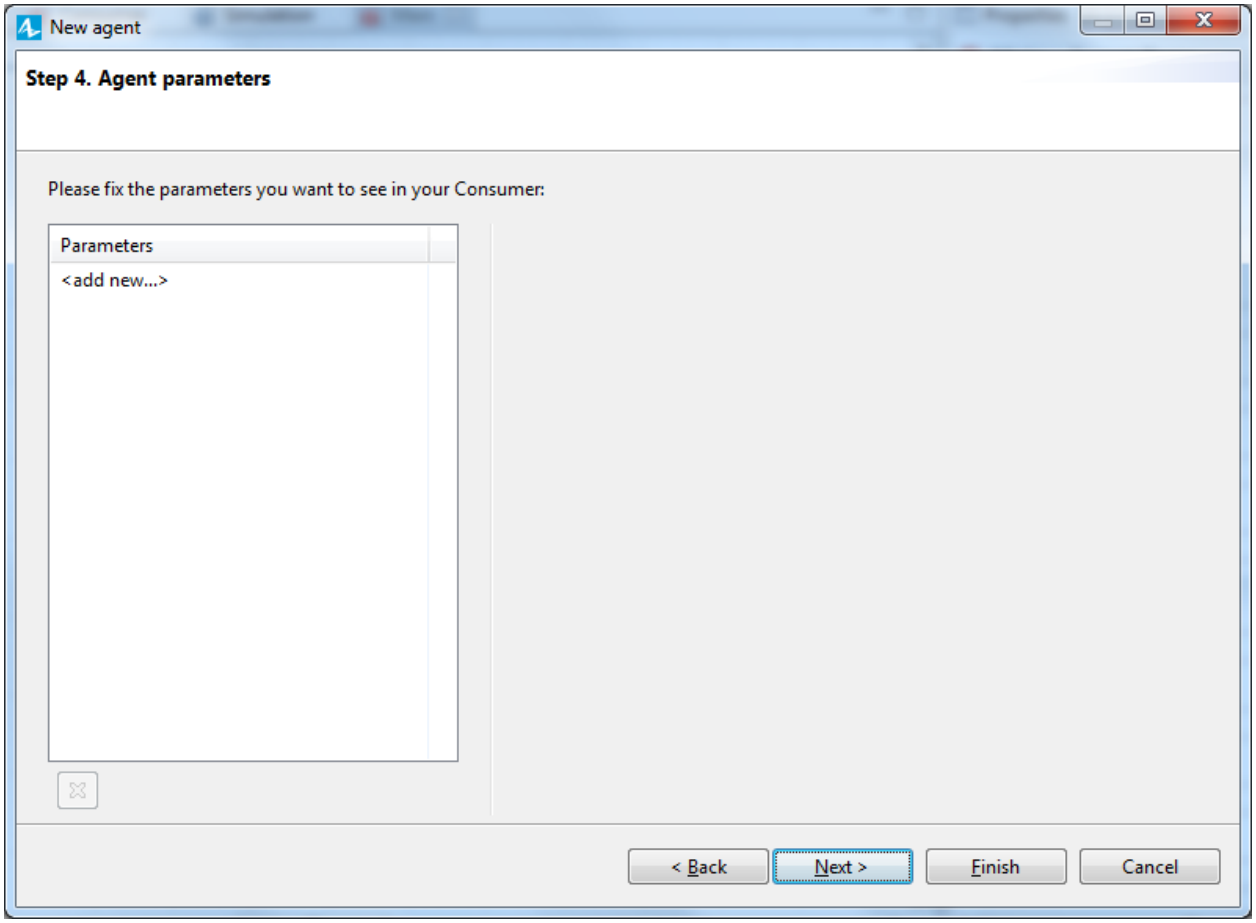



Click **Next**.

On the **Agent animation** page, choose the agent's animation shape. Since we're creating a simple model that uses 2D animation, choose **2D**, select the

**General** list's first item: **Person**, and click **Next**.

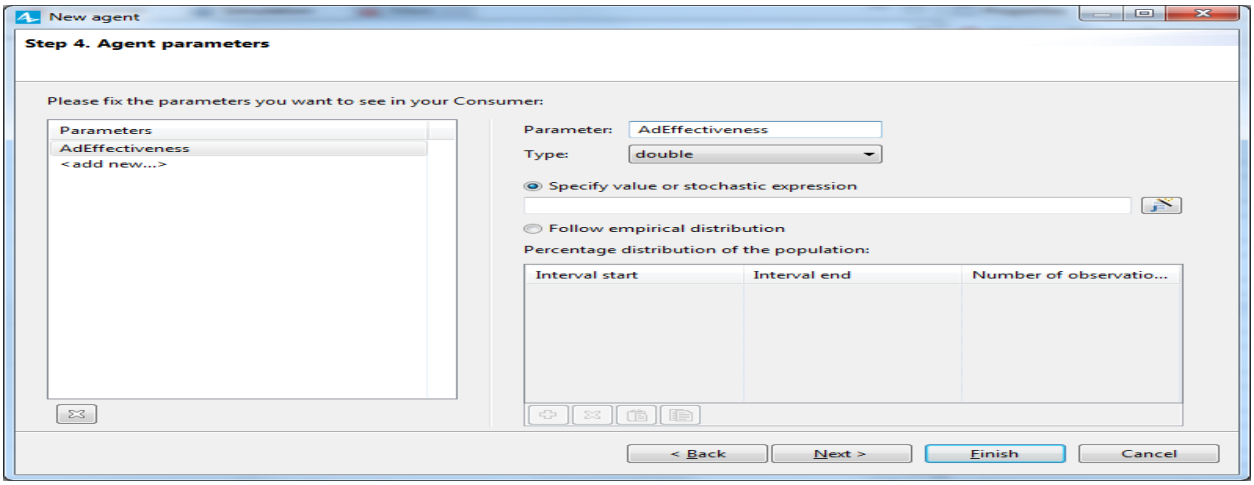


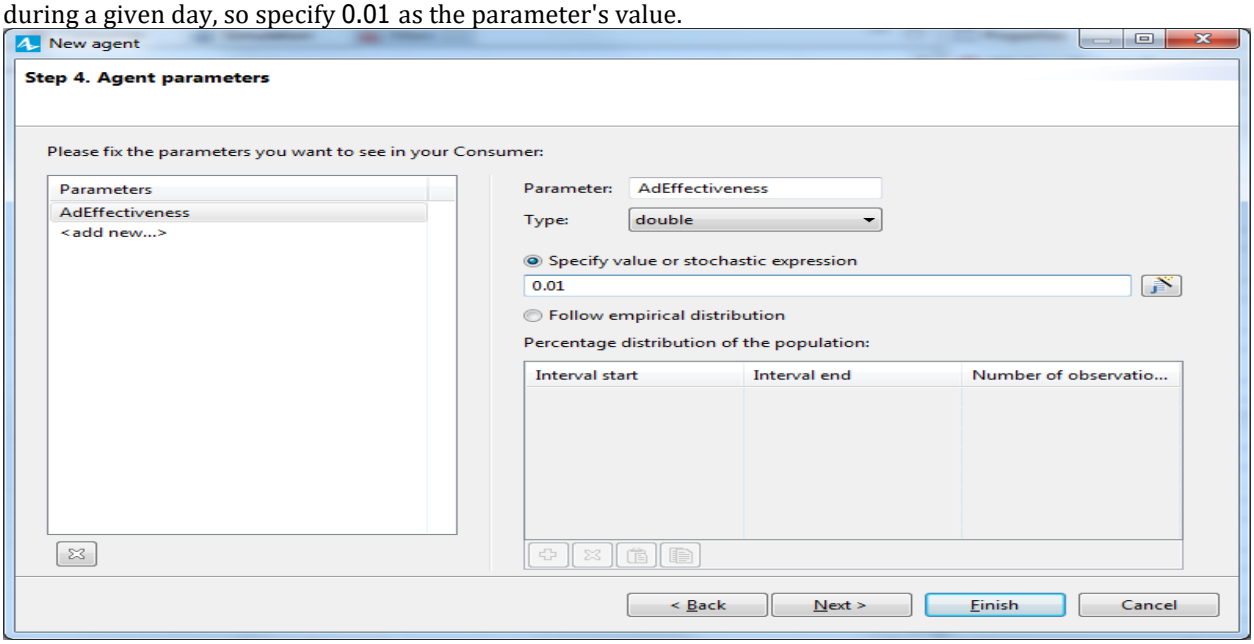

On the **Agent Parameters** page, define the agent's parameters or characteristics.
Since our model only considers advertising-related product purchases, we'll add a

parameter – AdEffectiveness – to define the percentage of potential users who become ready to buy the product during a given day.

On the left section, in the **Parameters** table, click **<add new...>** to create a parameter.



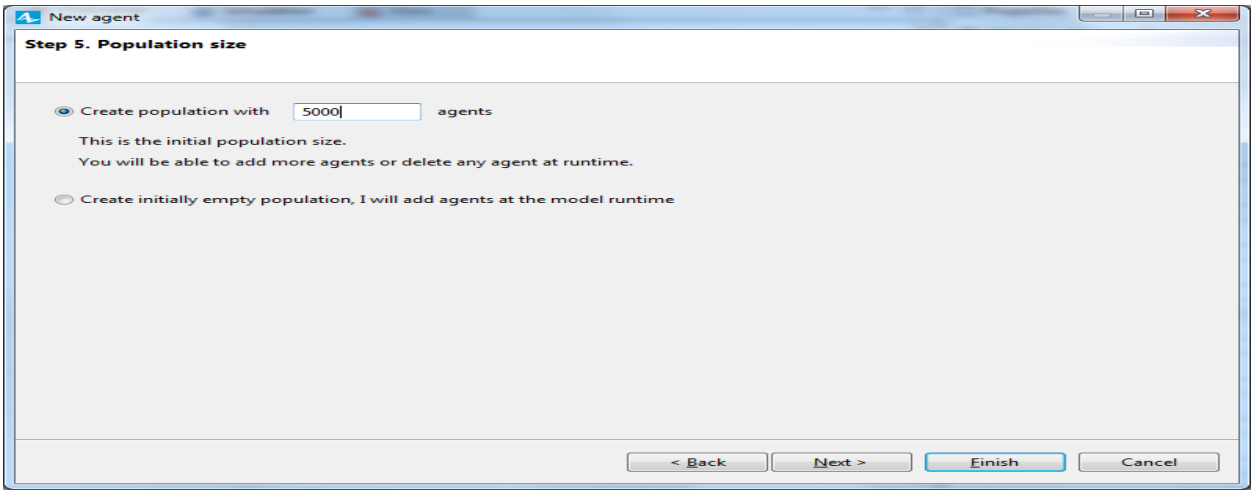

In the **Parameter** box, change the default parameter's name to AdEffectiveness, and choose **double** as the parameter **Type**. We'll assume an average of 1% of our model's potential users will want to buy the product during a given day, so specify 0.01 as the parameter's value.




Click next

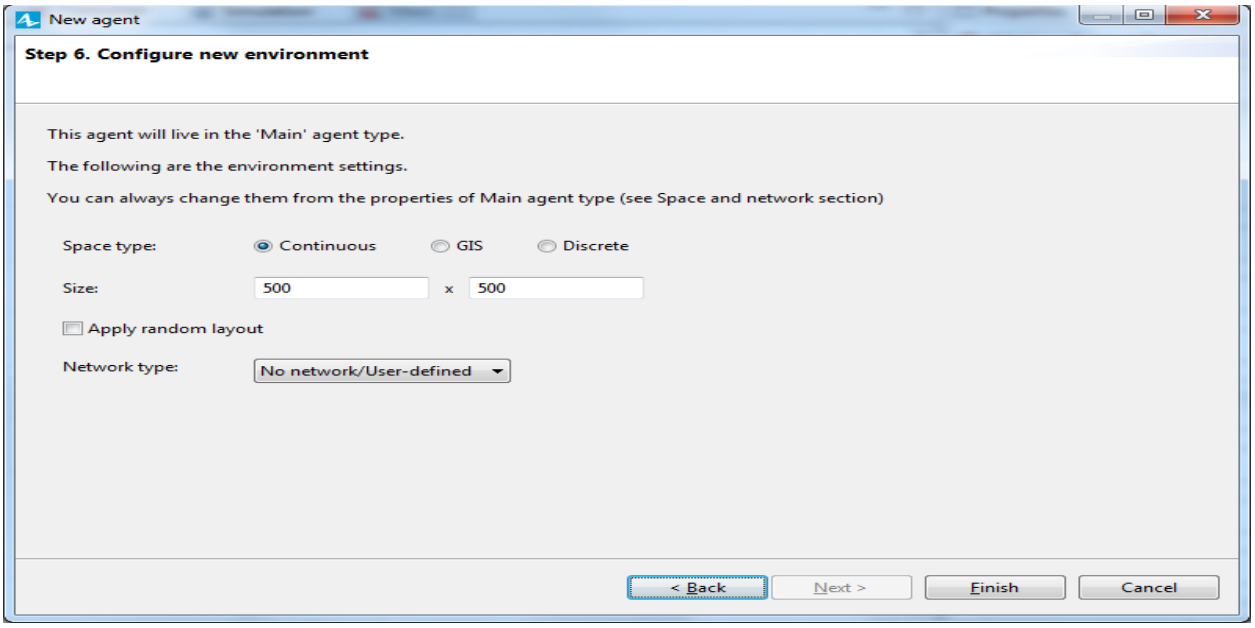
On the **Population size** page, type 5000 in the **Create population with ... agents** box to create 5000 instances of the Consumer type. Each instance in the population will model a specific agent-consumer.

While we've created our agent population, we won't see 5,000 Person animation figures on Main diagram. Instead, AnyLogic will use the 5000 agents in the population we've called consumers to simulate the market when we run our model.

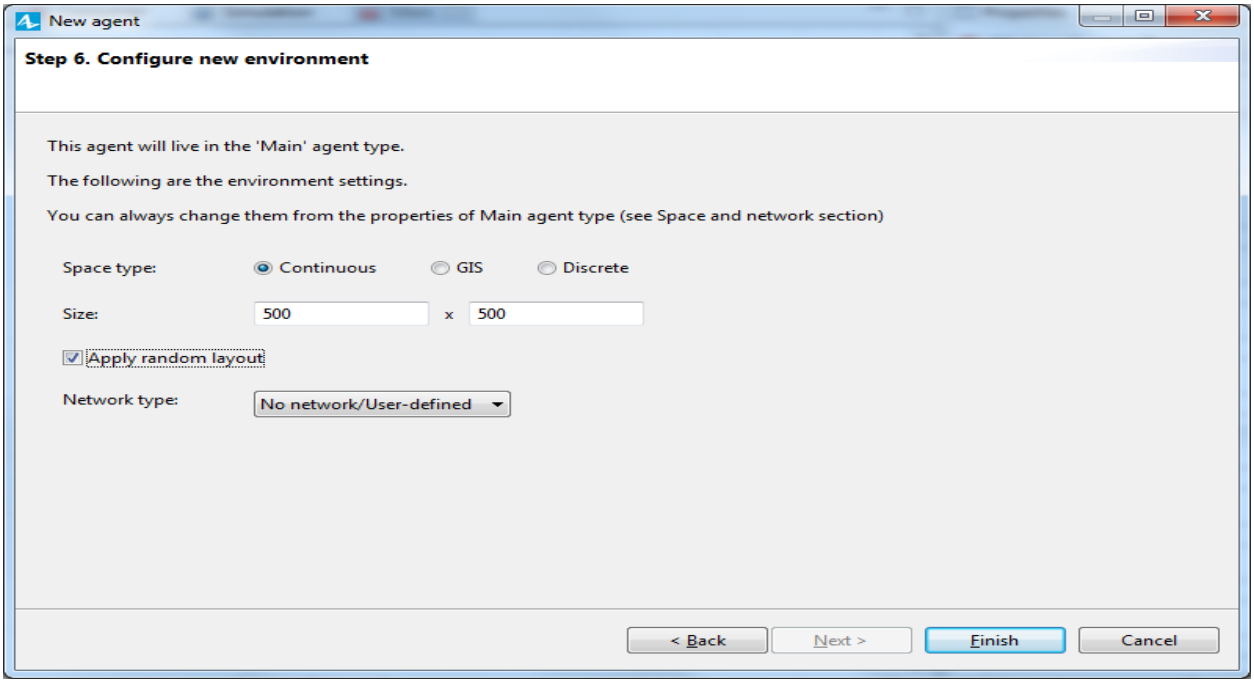



Click **Next**.

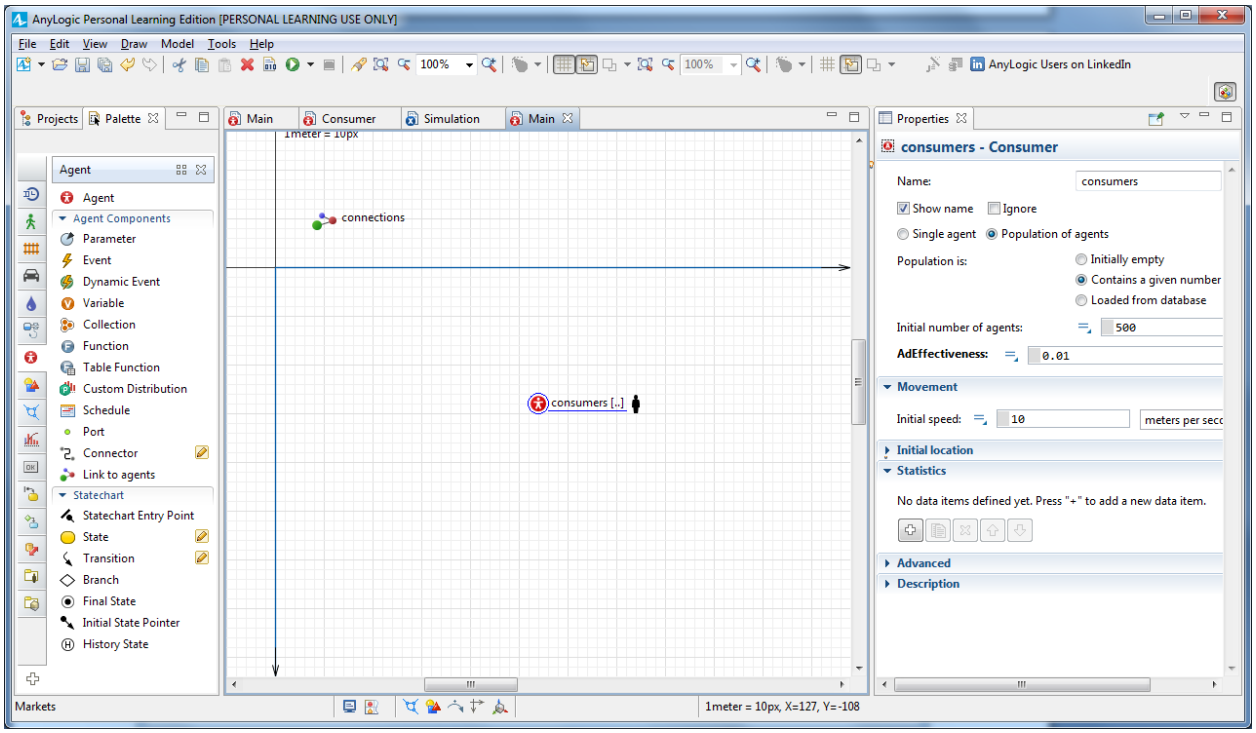
On the **Configure new environment** page, accept the default values for the environment's space type (**Continuous**) and both its **Width** and **Height** values (500). AnyLogic will display the agents in a 500x500 pixel rectangle.



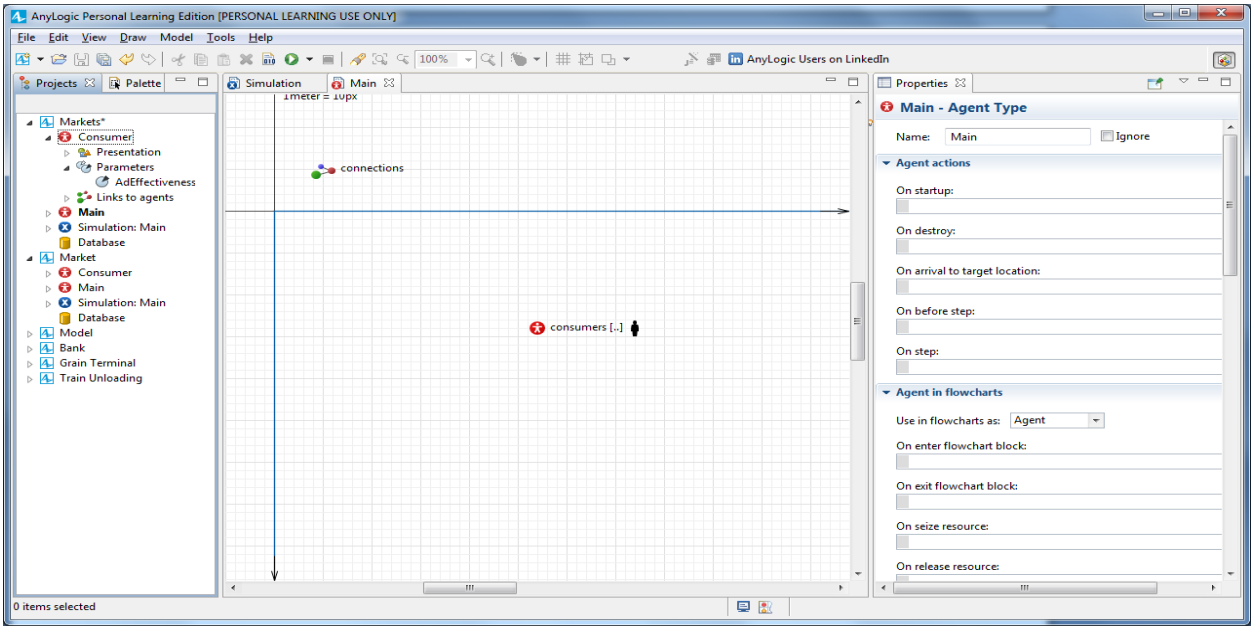

Select the **Apply random layout** box to randomly distribute the agents across the 500 pixel width and height we've defined. Since we don't want to create an agent network, we'll accept the default **No network/User-defined** network type.




Click **Finish**.



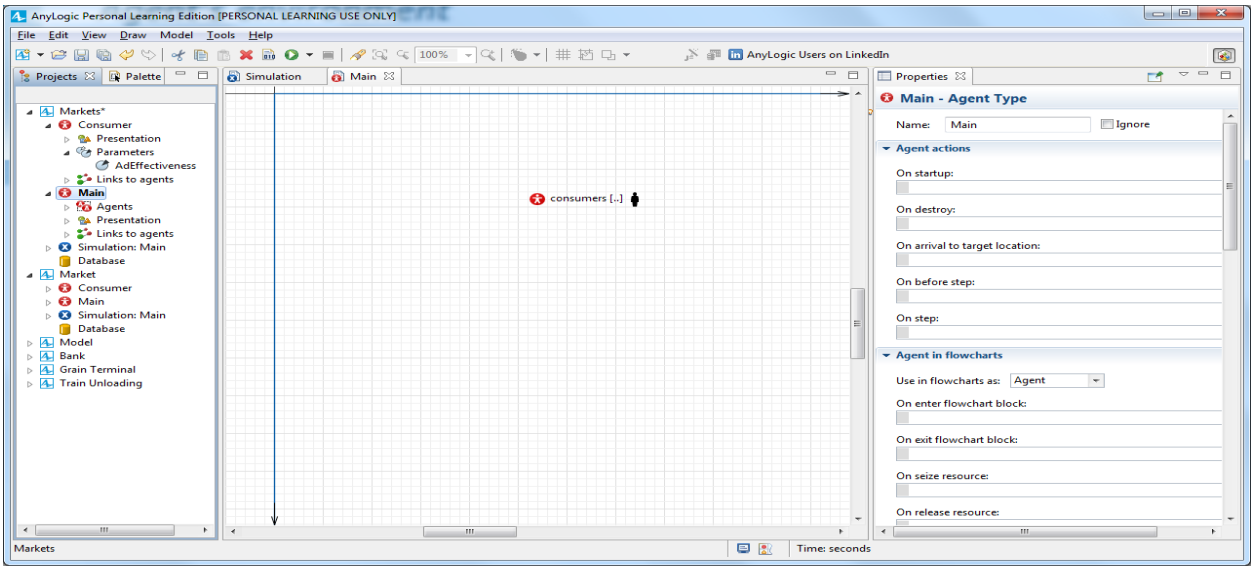

Let's use the **Projects** view to see the new elements that the wizard created. Expand the model tree branches to see the internals.



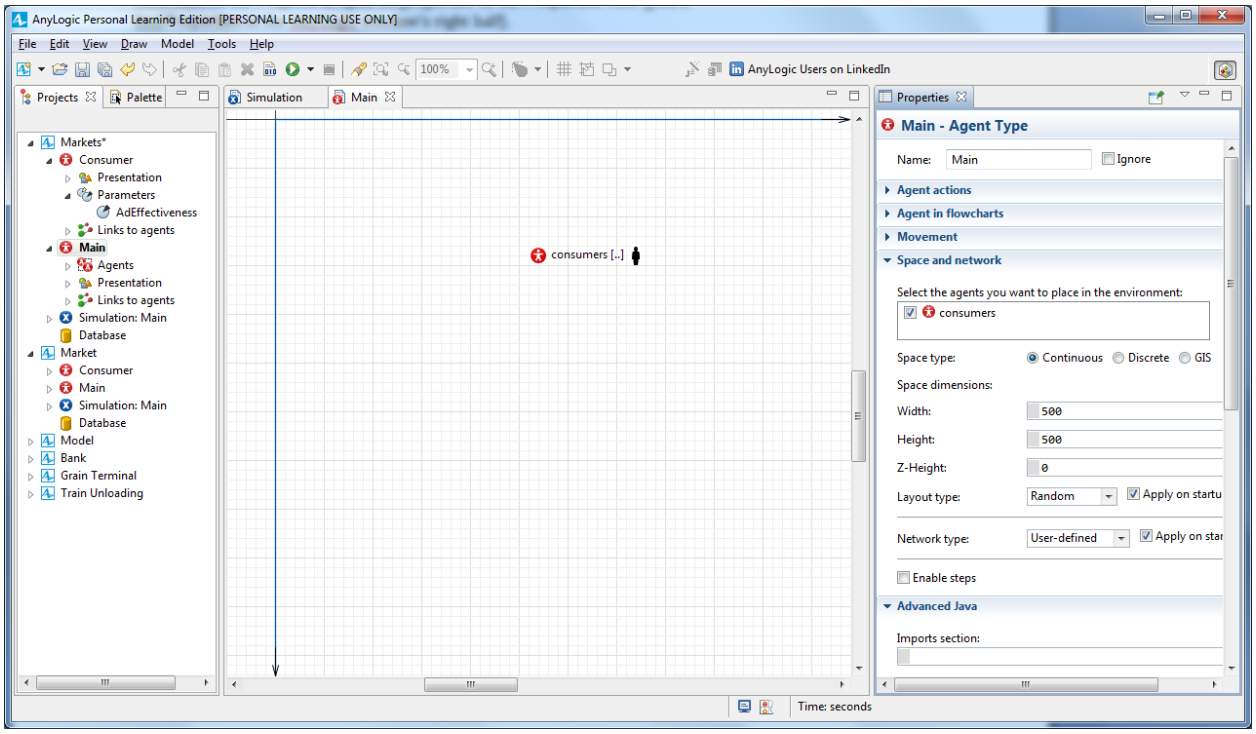

Our model now has two agent types: Main and Consumer.

- The Consumer agent type has the agent's animation shape (person, in the **Presentation** branch) and the parameter AdEffectiveness.
- The Main agent type contains the agent population consumers (a set of 5000 agents of type Consumer ).

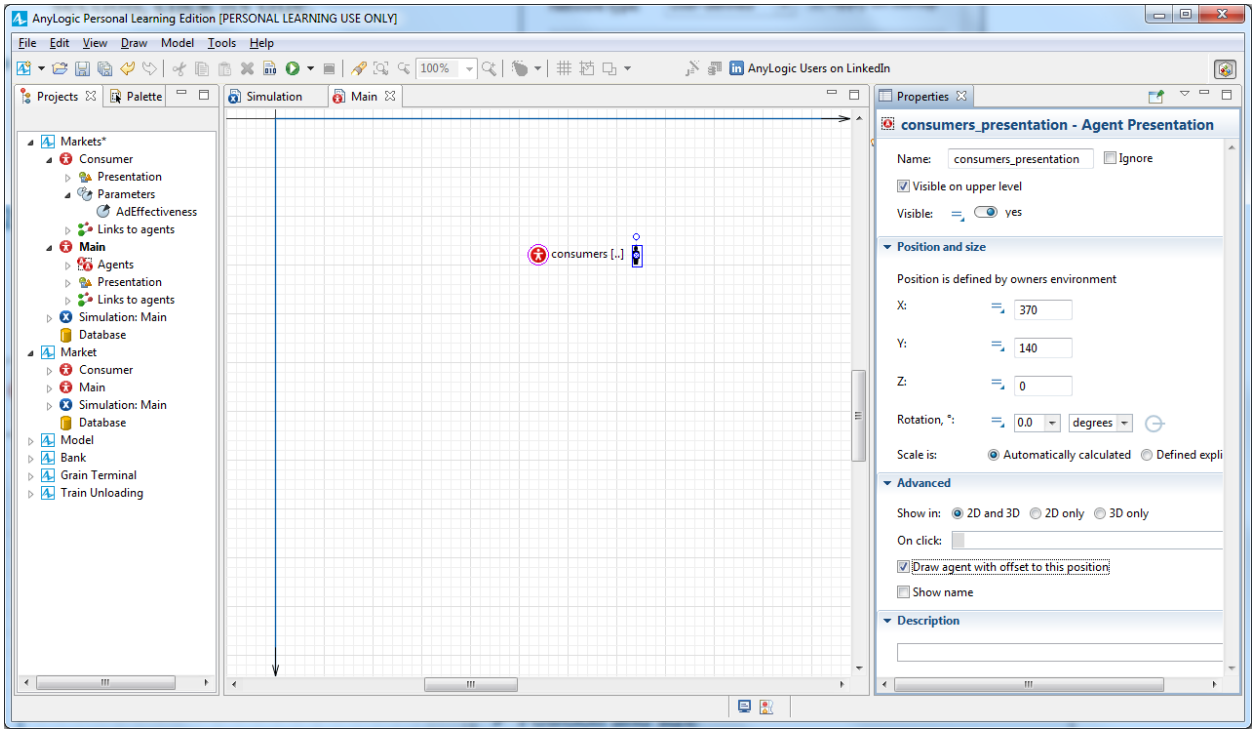
Click **Main** in the **Projects** to open its properties in the **Properties** view (you'll find **Properties** in the AnyLogic window's right half).

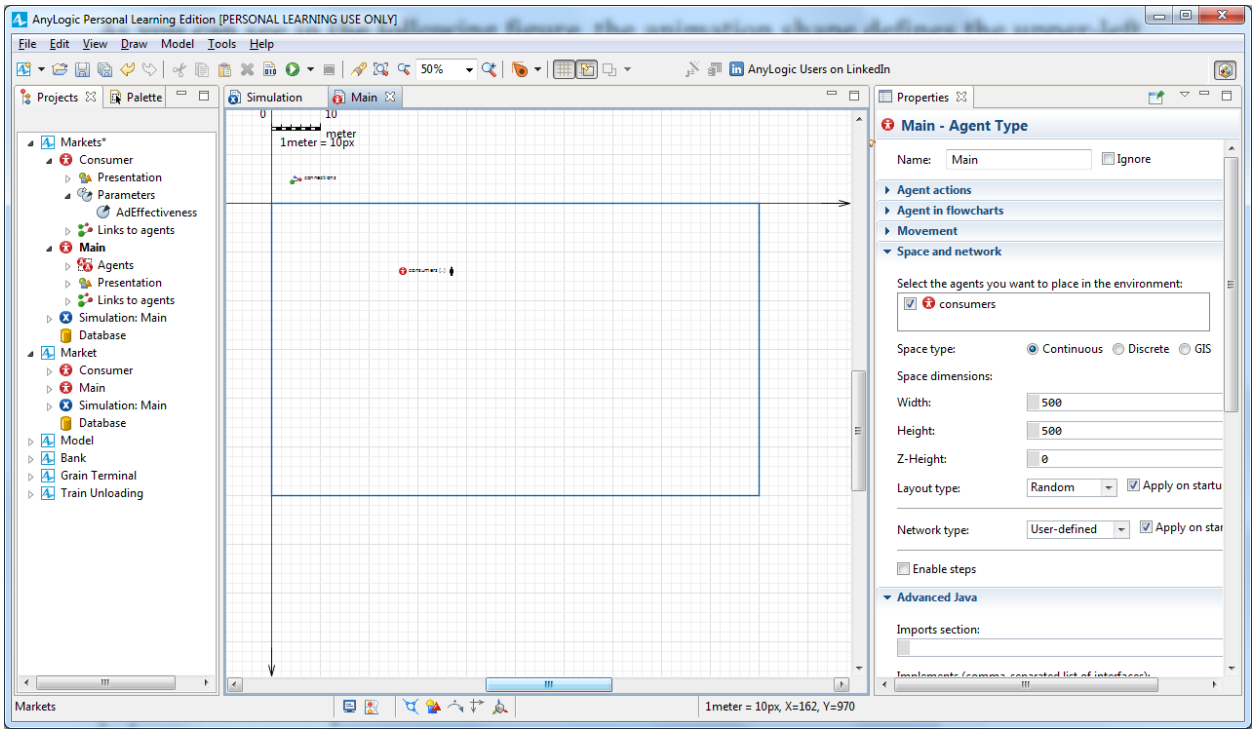



In the **Space and network** section of Main properties, you can adjust the environment settings for the consumers agent population.



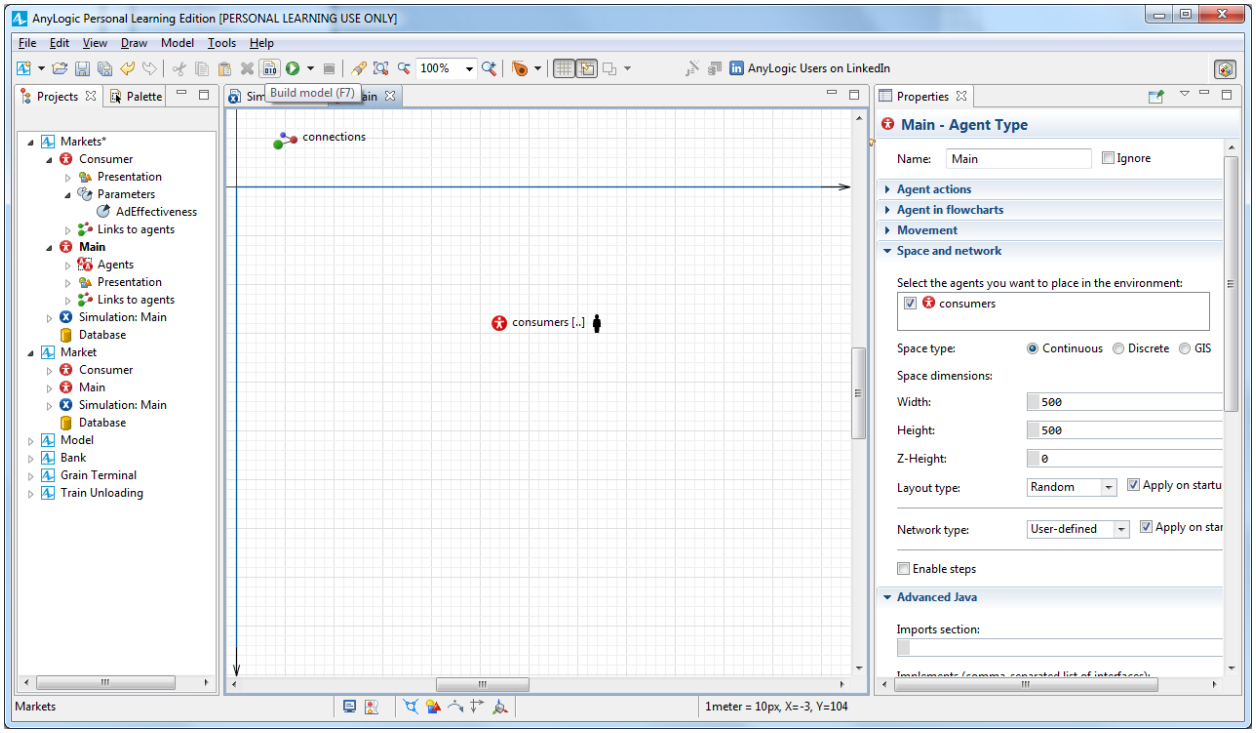

On Main diagram, select the agent population's non-editable embedded animation shape , open the **Advanced** properties section, and select the **Draw agent with offset to this position** option.



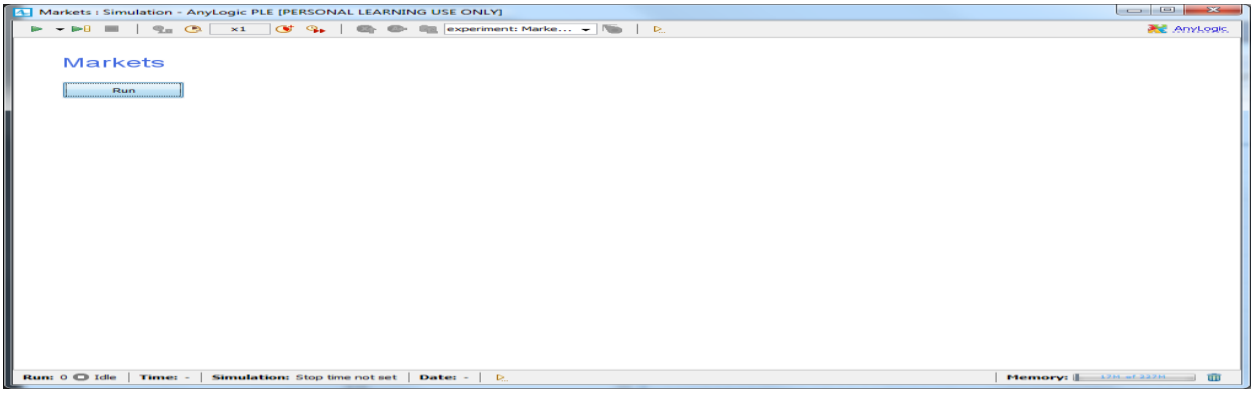



We've finished building this very simple model, and you can now run it and observe its behavior.

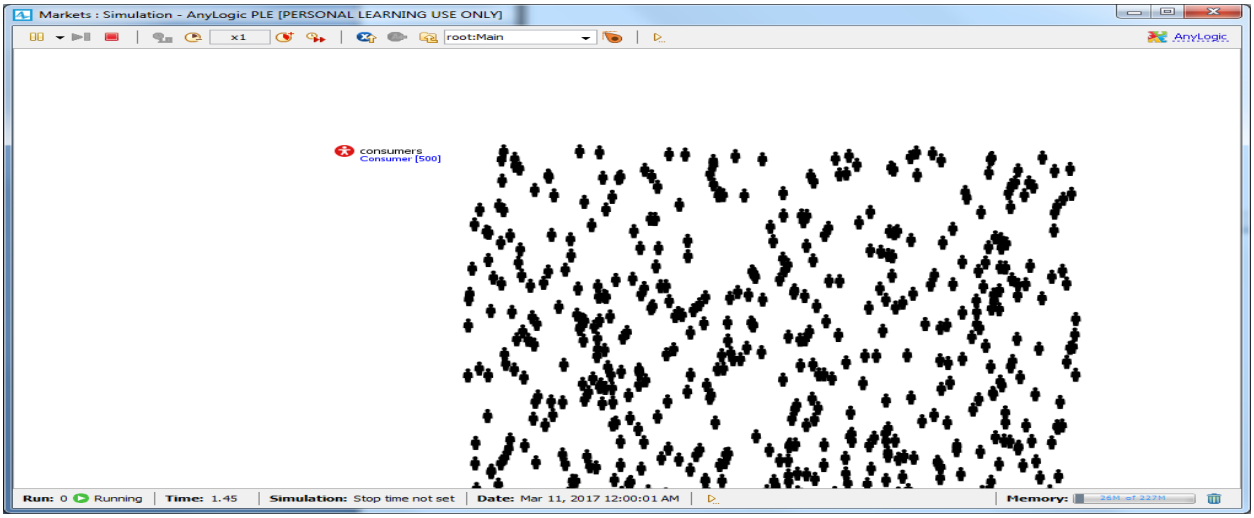
On the toolbar, click the **Build** button to build the model and check it for errors.



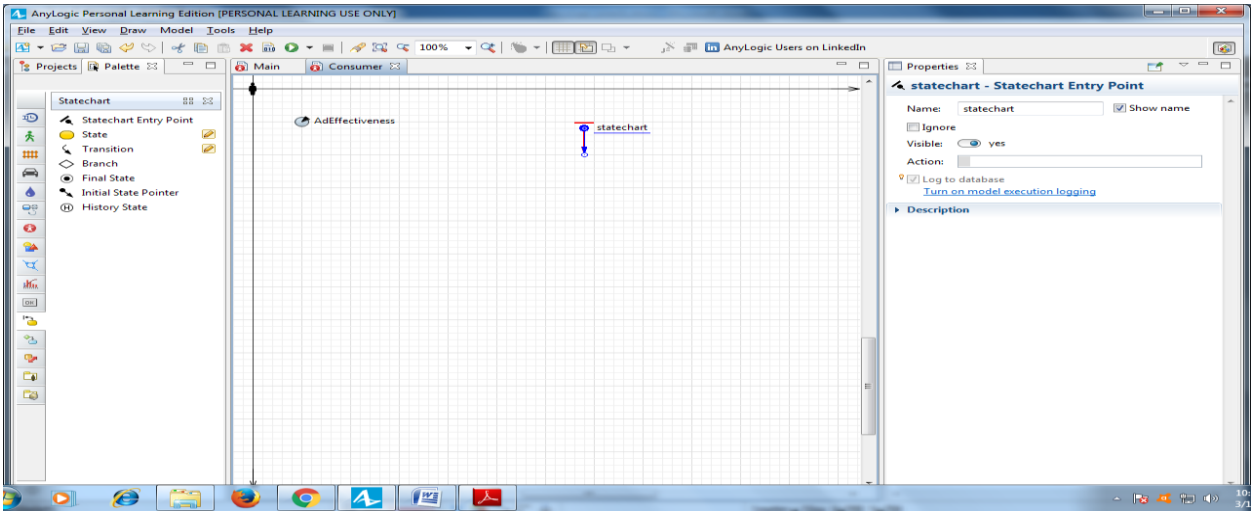
Locate the **Run** button, and click the small triangle to the right. Select the experiment you want to run. Choose **Markets / Simulation** from the list.



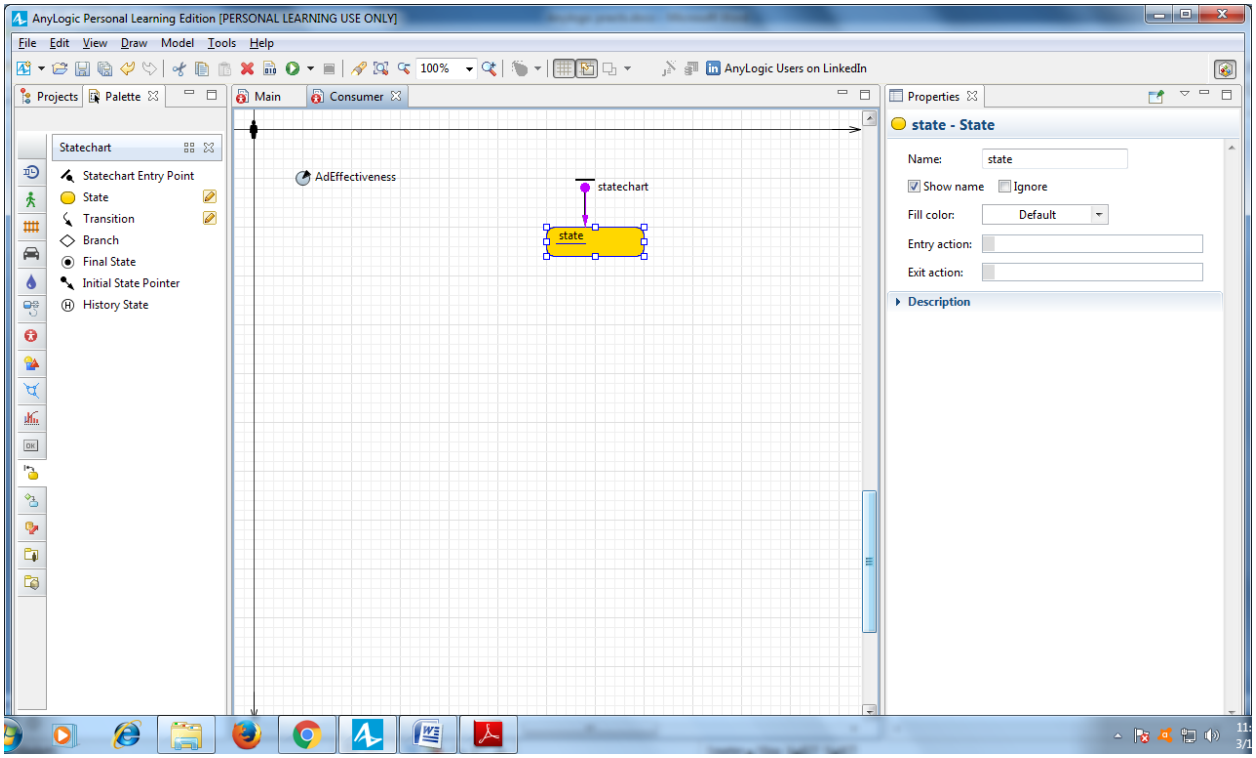

Click the **Run** button to run the model.



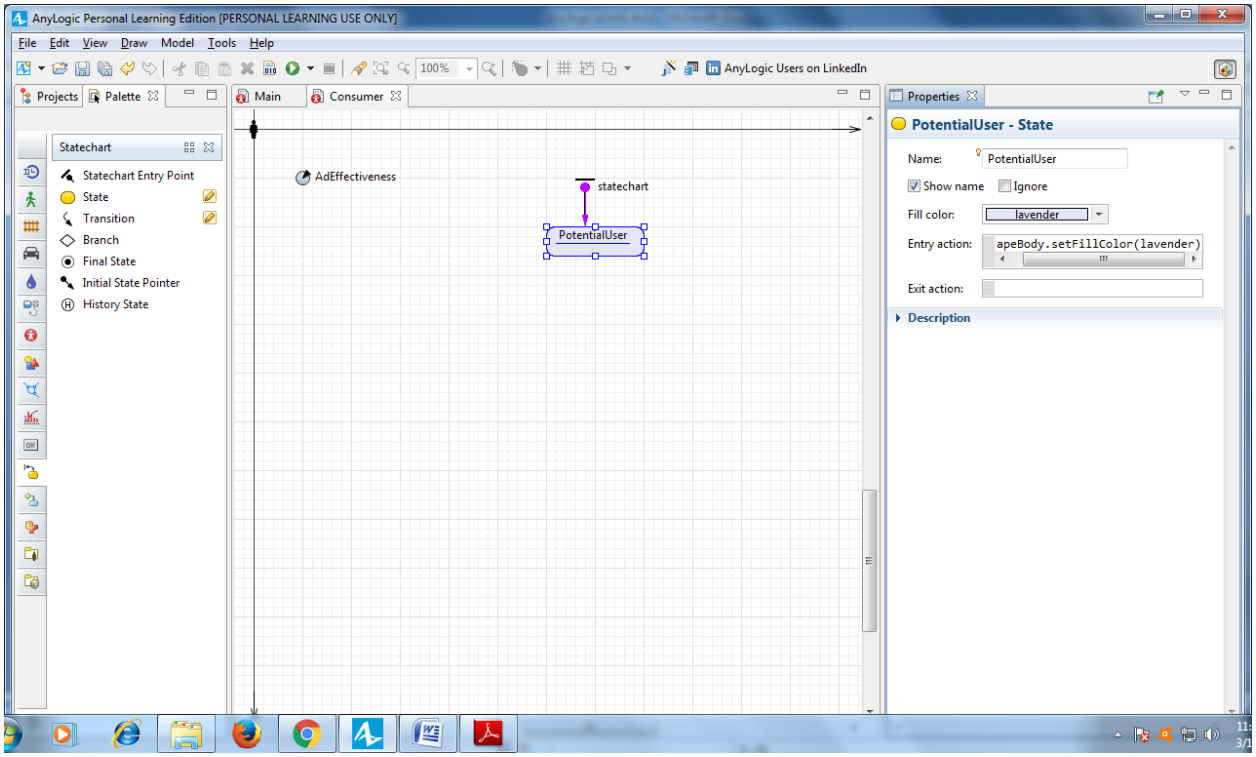

# Defining a consumer behavior



Drag the **State** from the **Statechart** palette on to the graphical diagram, and connect it to the statechart entry point.



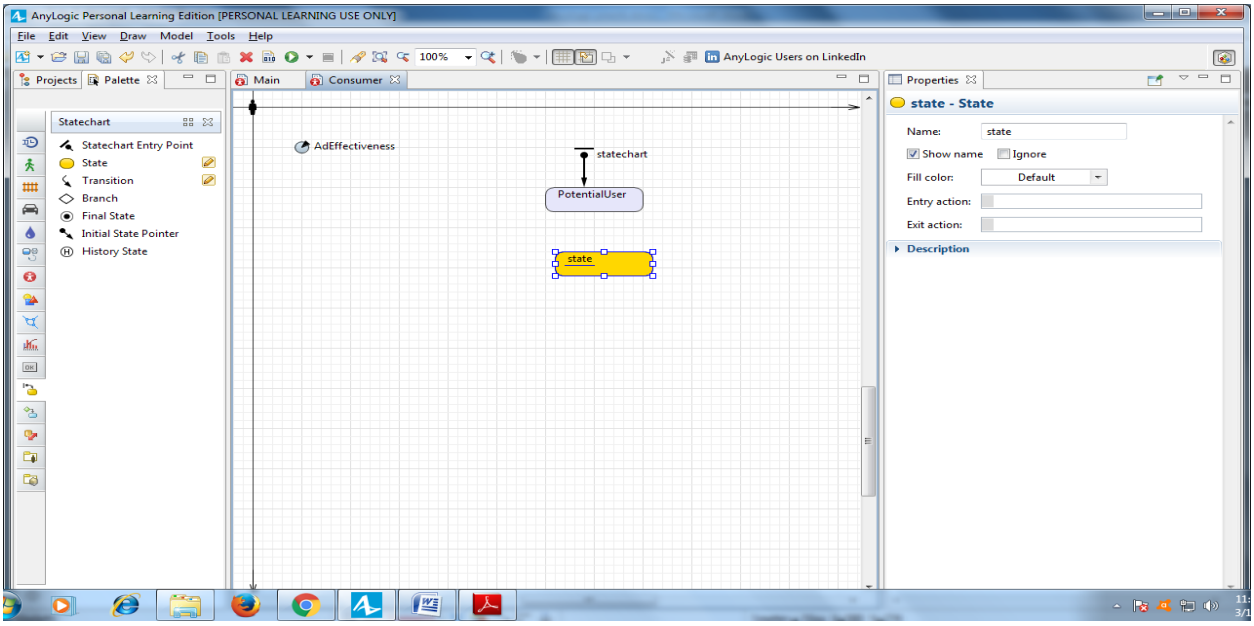

Select the state in the graphical editor, and modify its properties. Name the state PotentialUser.



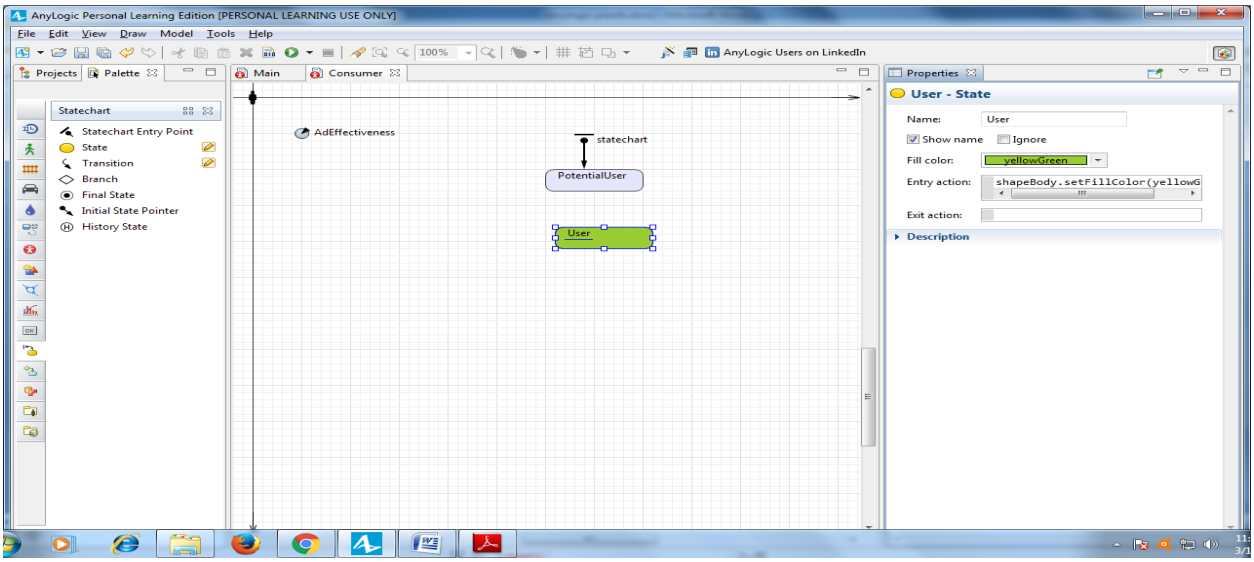
Use the **Fill color** control to change the state's color to lavender.

Type the following Java code in the state's **Entry action** field:
shapeBody.setFillColor(lavender)

Add another state in the consumer's statechart:

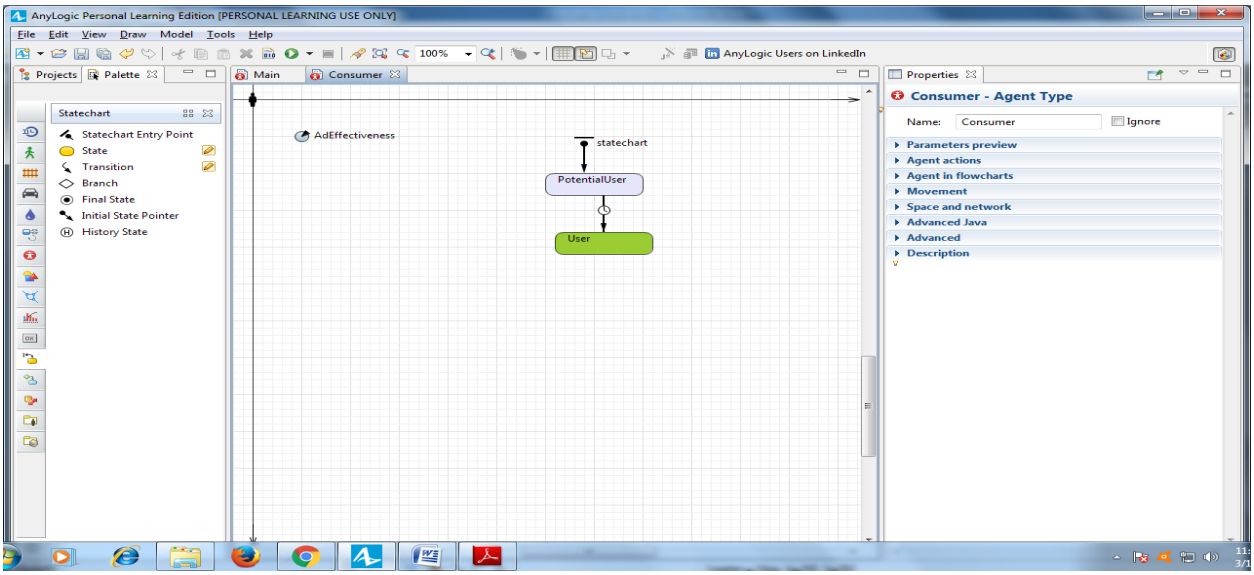



Modify the state's properties like you did earlier:
**Name**: User
**Fill color**: yellowGreen
**Entry action:** shapeBody.setFillColor(yellowGreen);




Draw a transition from PotentialUser to User state to model how persons purchase the product and become product users. To do so, double-click the **Statechart** palette's **Transition** element (the element's palette icon should

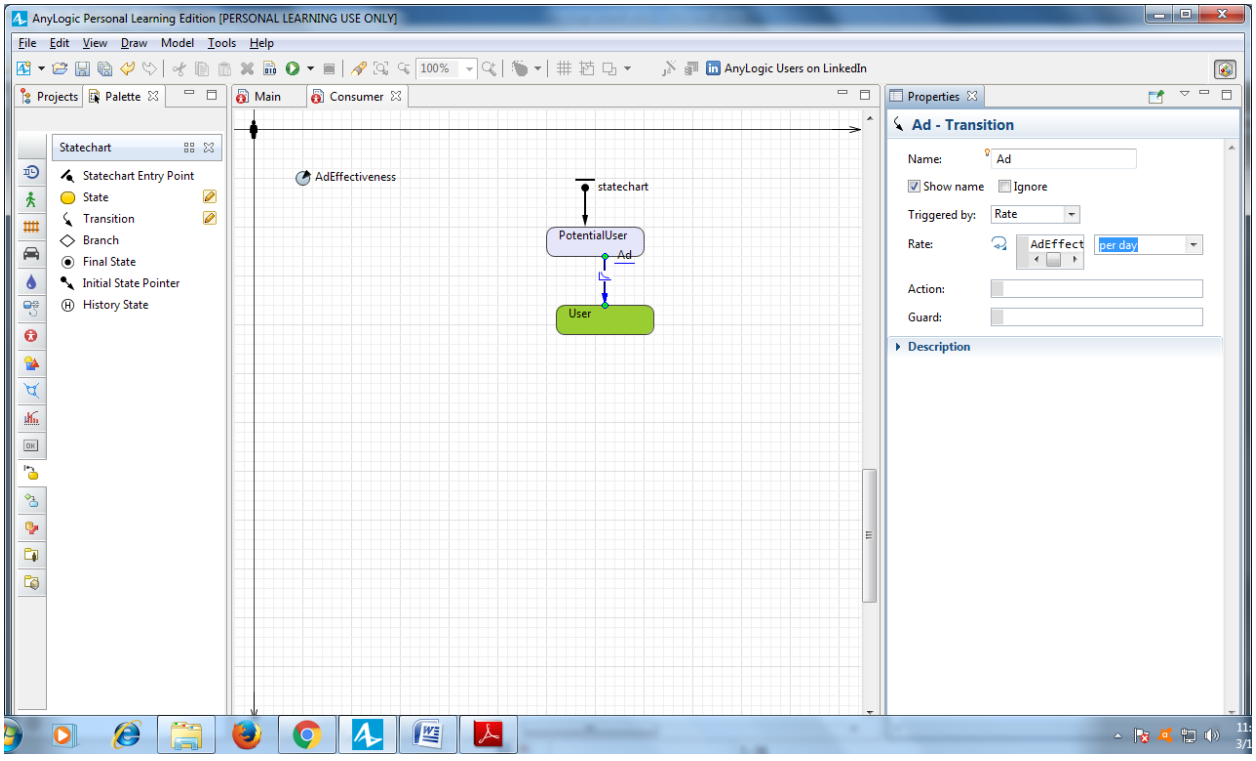
change to ), click PotentialUser state, and click User.




Name the transition Ad to represent “advertising”.

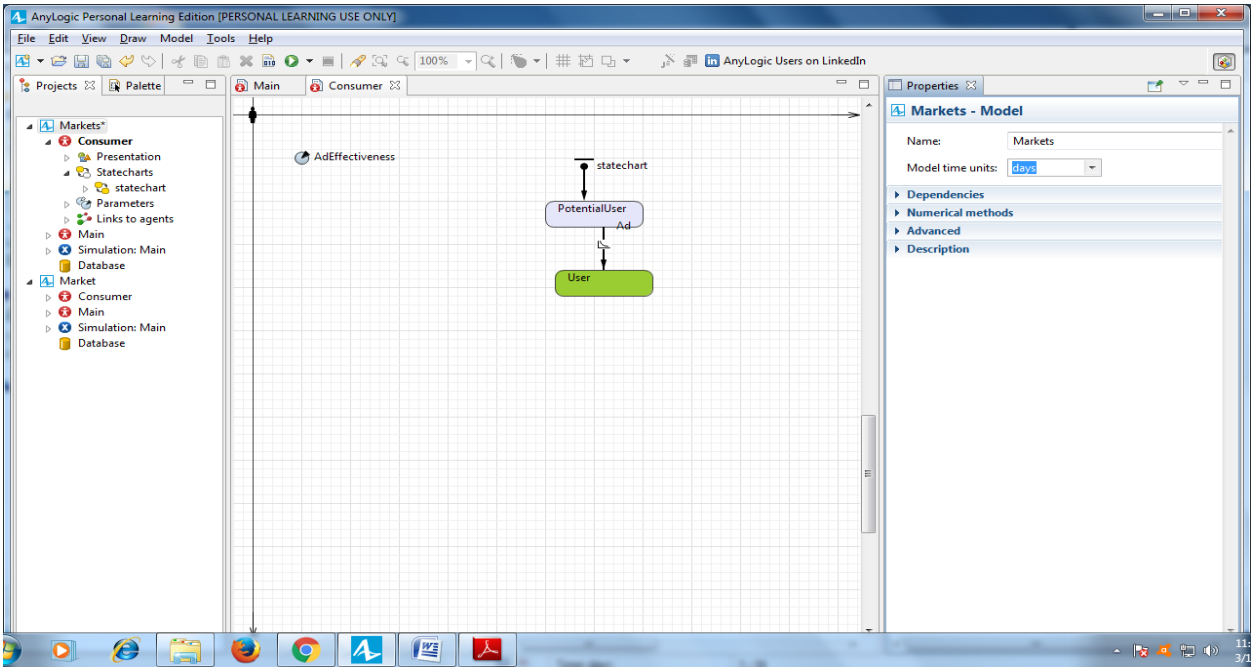
**12.** Select the **Show name** checkbox to display the transition’s name on the graphical diagram.

**13.** The transition from PotentialUser to User state will model how advertising leads the person to buy the product. In the **Triggered by** list, click **Rate**. In the **Rate** field, type AdEffectiveness, and then click **per day**.

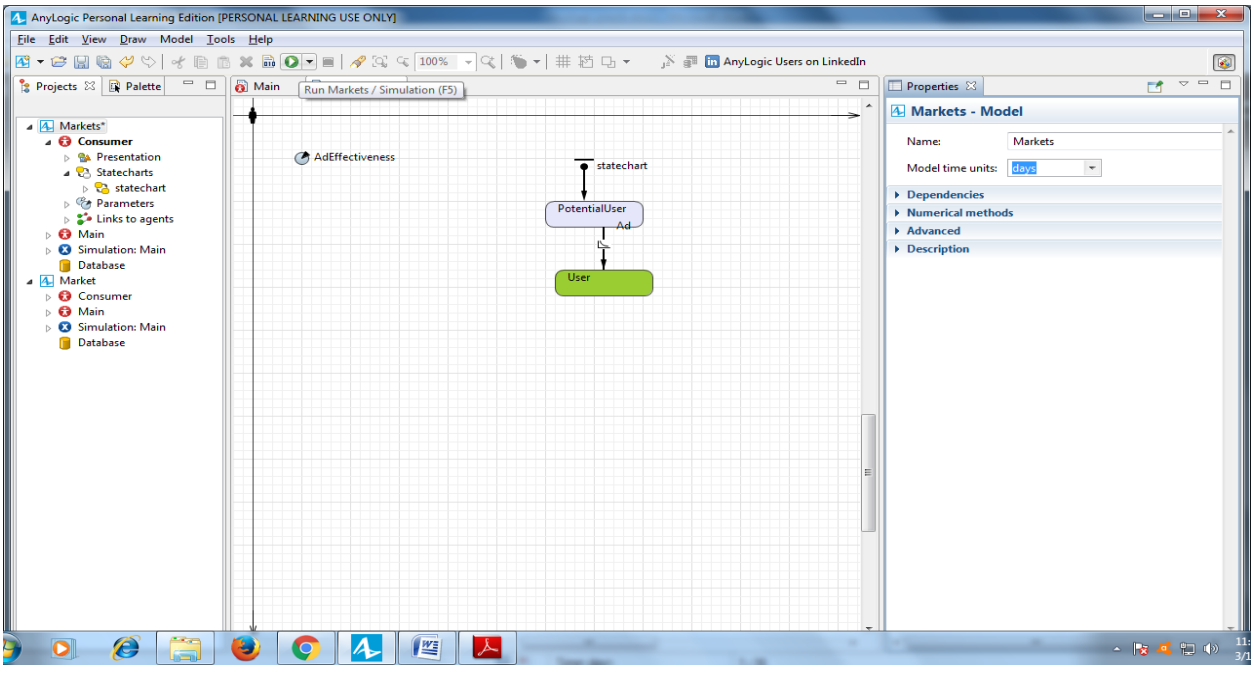


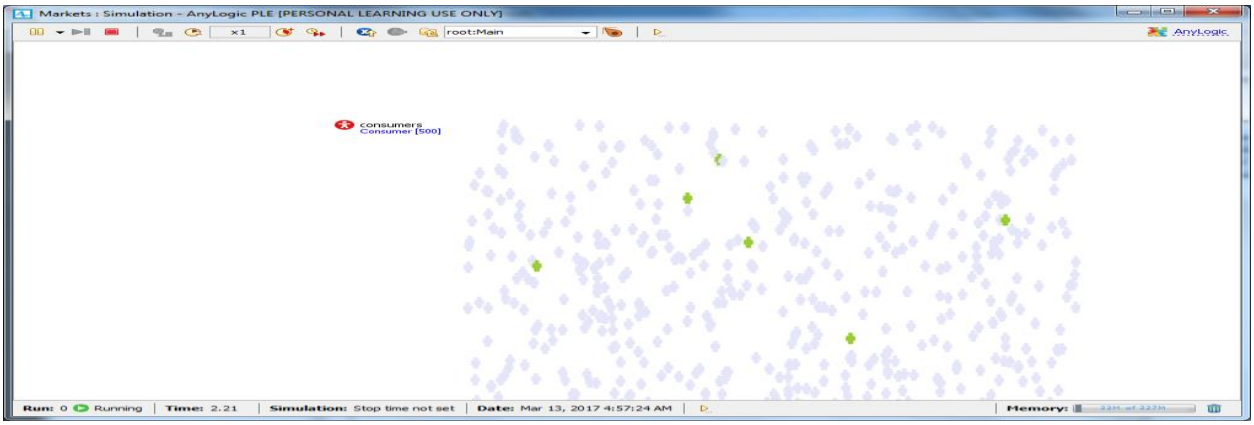

Now, let’s set up the model’s time units. To tune the model setting, switch from **Palette** to **Projects**, and then click the model item in the tree (the tree's

top object, Market ). In the **Properties** view, choose **days** as the **Model time units**.

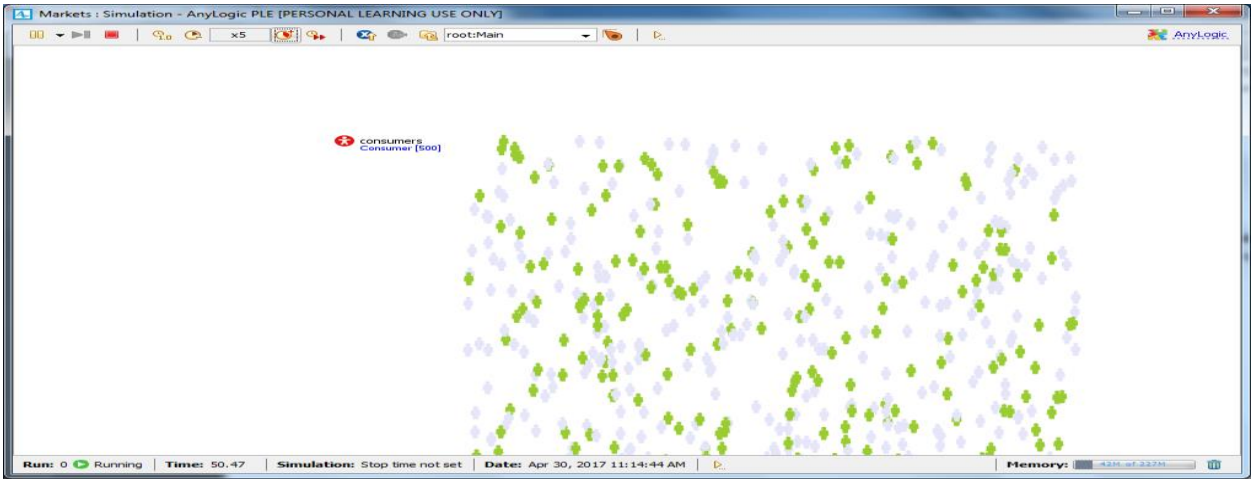


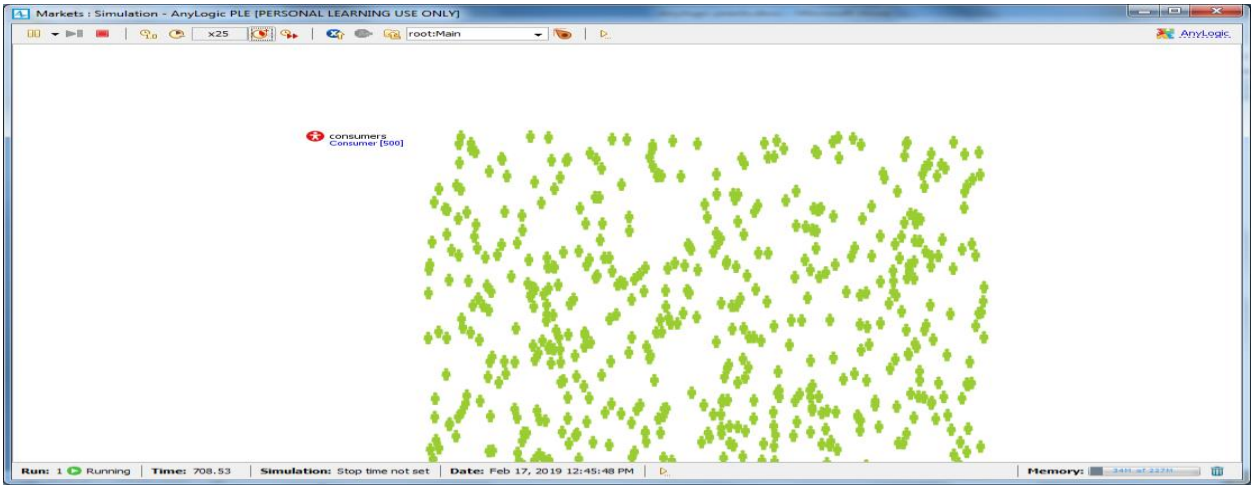

Run the model. The population should gradually turn green – a change that represents the effect of advertising - until every consumer buys the product.





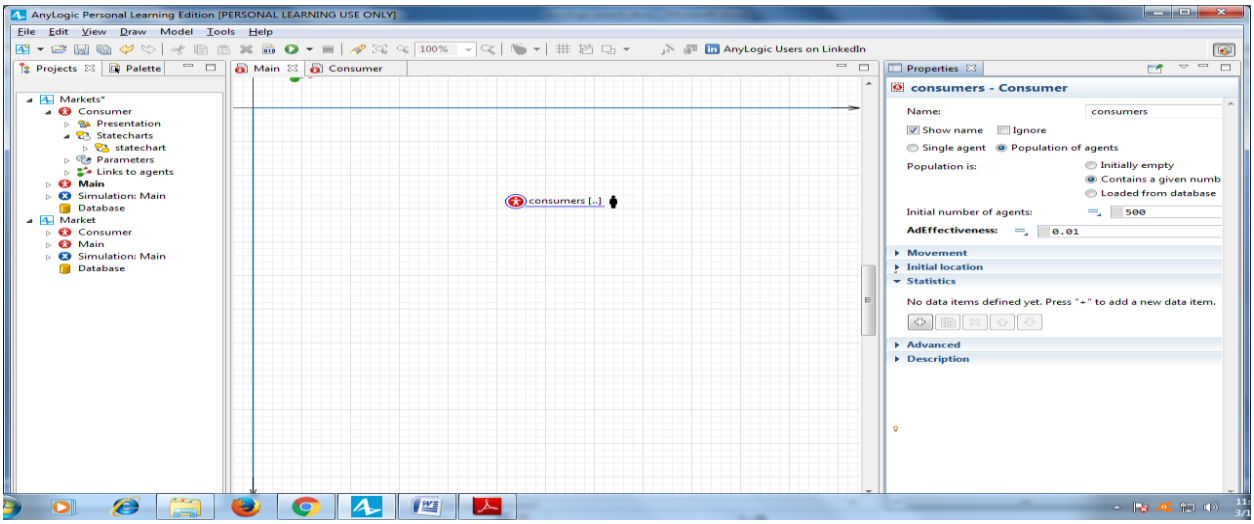

To adjust the model's execution speed, click the toolbar's **Slow down** or **Speed up** buttons. If you increase the speed to 10x – you'll see the speed at which the population turns green also increase



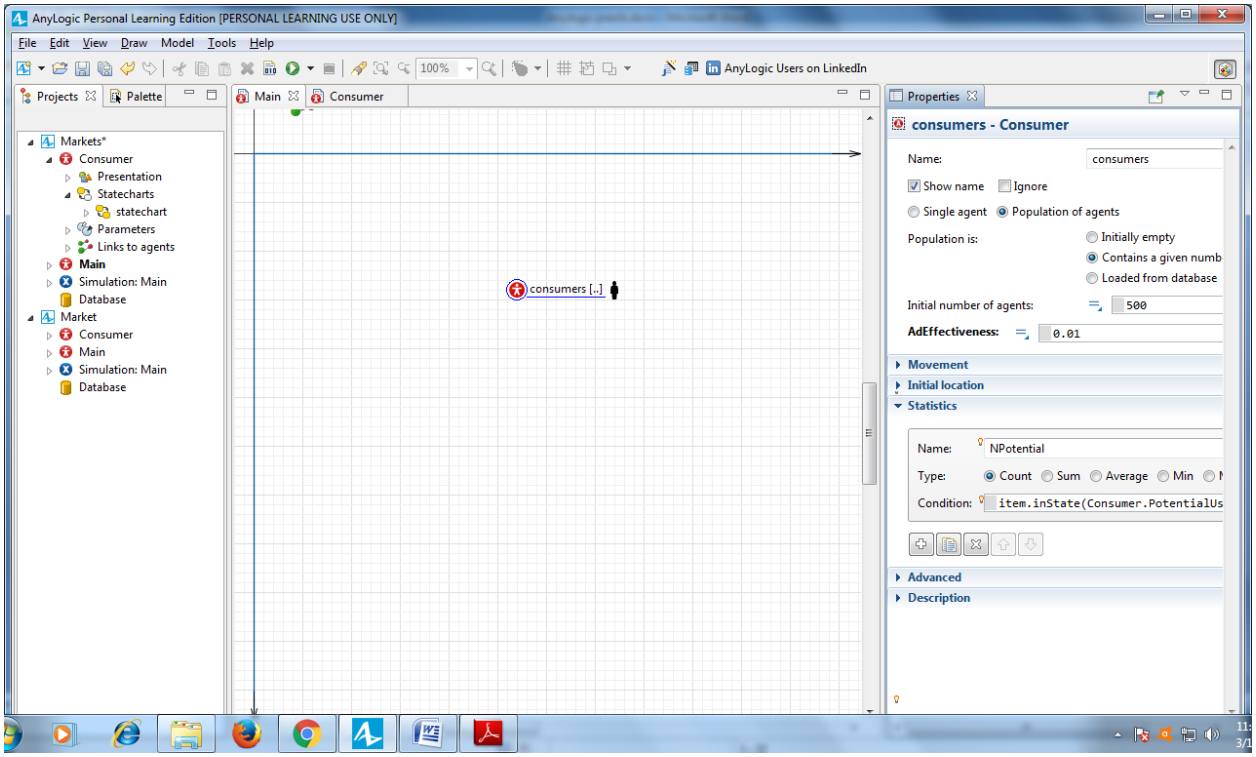

## Adding a chart to visualize the model Output

First, define a function to count potential users. To add a new function that collects statistics for agents, open the diagram of the agent type Main, select the agent population consumers, and go to the **Statistics** properties section.

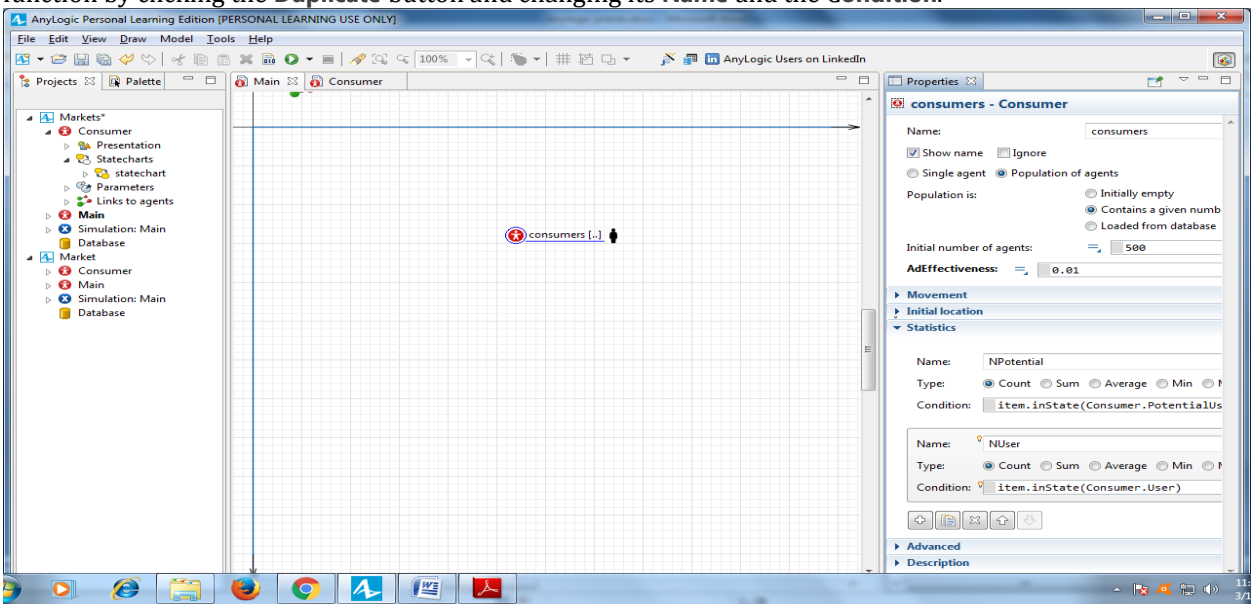



Define the function of type **Count** with the **Name** NPotential. The statistics of type **count** iterates through a given population – in our case, the number of agents – to count those that meet the selected condition.

Enter item.inState(Consumer.PotentialUser) as the function **Condition**.

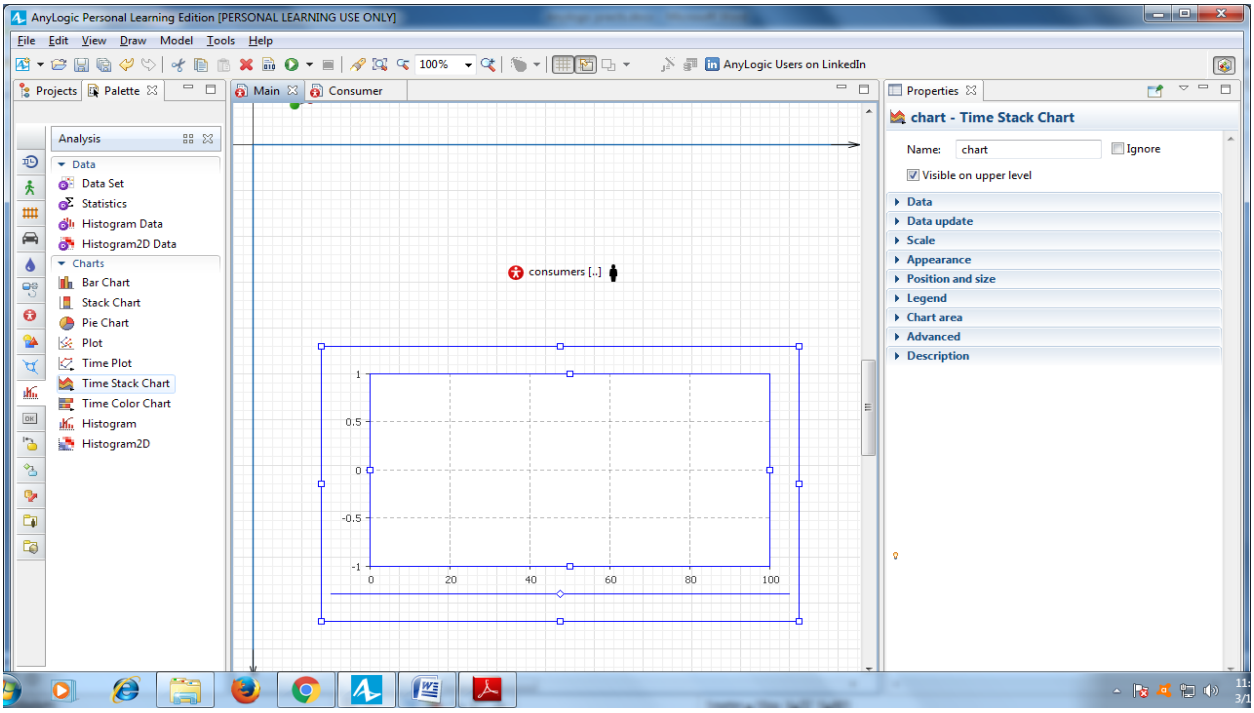


Define a second statistics function to calculate the number of product users. Name it NUser and let it count the number of agents, conforming the **Condition** item.inState(Consumer.User). You can duplicate the other statistics function by clicking the **Duplicate** button and changing its **Name** and the **Condition**.



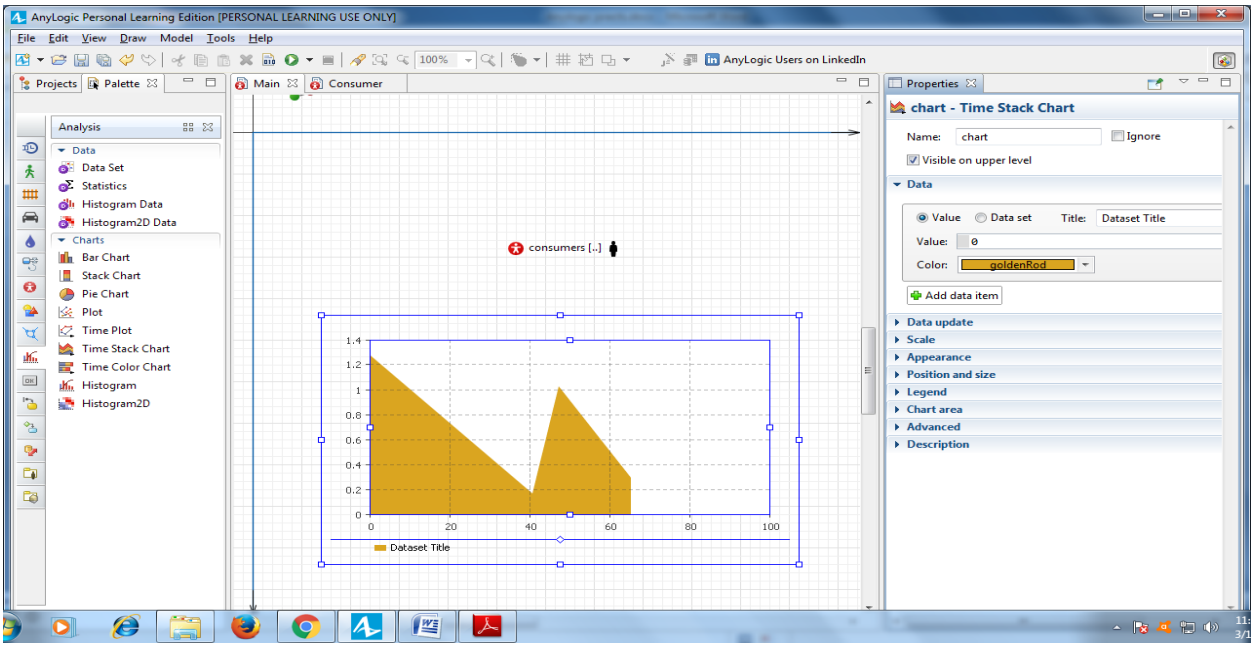

Now, let's add a chart to show the statistics these functions collect and display the adoption process dynamics.

Open the **Analysis** palette, and drag the **Time Stack Chart** from the **Analysis** palette on to the Main diagram to create a chart that will display the dynamics of users and potential users. Increase the time stack chart as shown in the figure below:



Add two data items for the chart to display. Here we'll call our statistics functions NUser and NPotential we have defined for consumers population on the previous step.

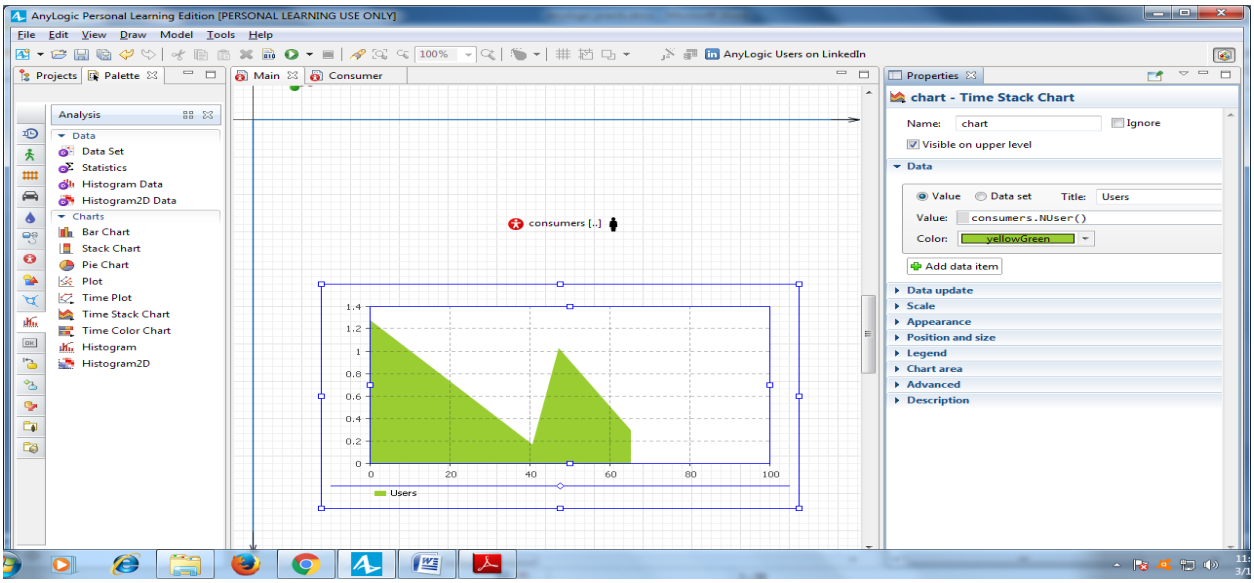
Click **Add data item** to add the statistics you want to draw on the chart.




Modify the data item's properties:

- **Title**: Users – the data item's title.
- **Color**: yellowGreen
- **Value**: consumers.NUser()

Our agent population name is consumers, and NUser() is the statistics function that we defined for this population.

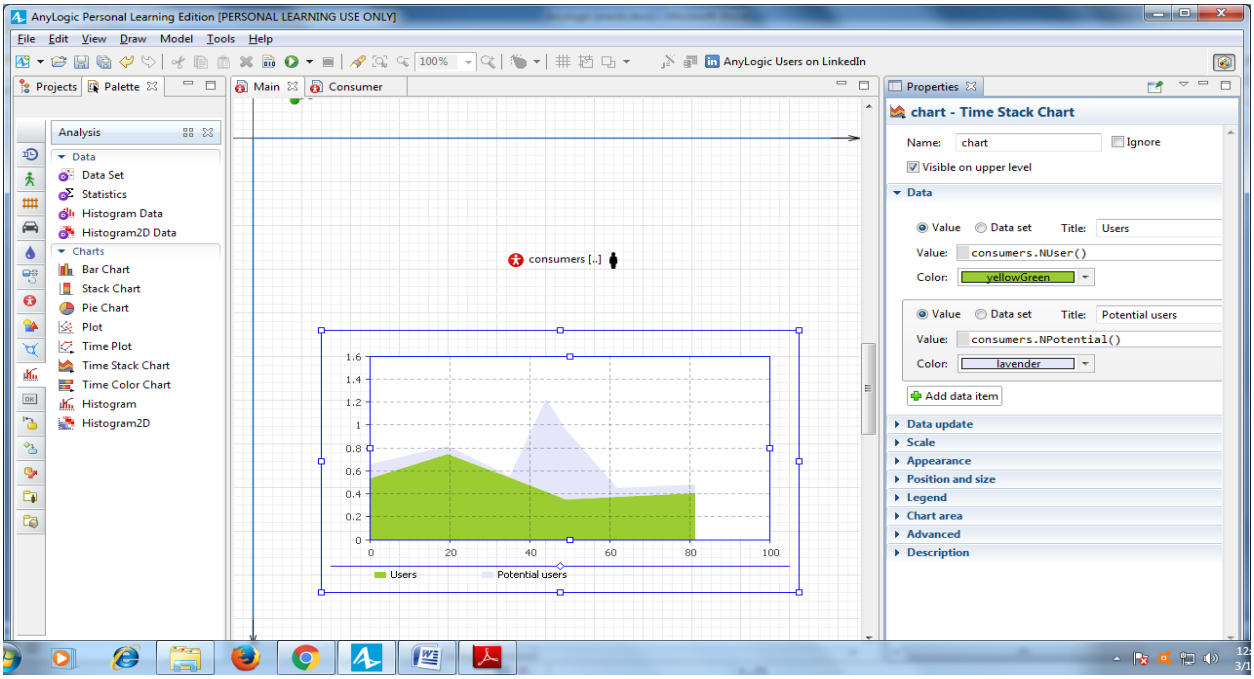


Add one more data item:

- **Title**: Potential users

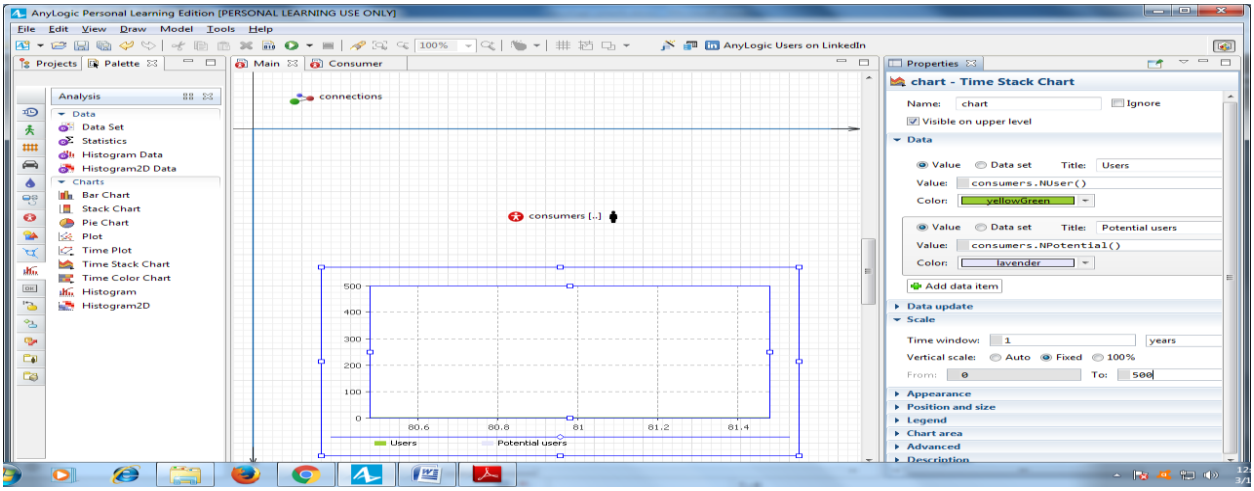


- **Color**: lavender
- **Value**: consumers.NPotential()



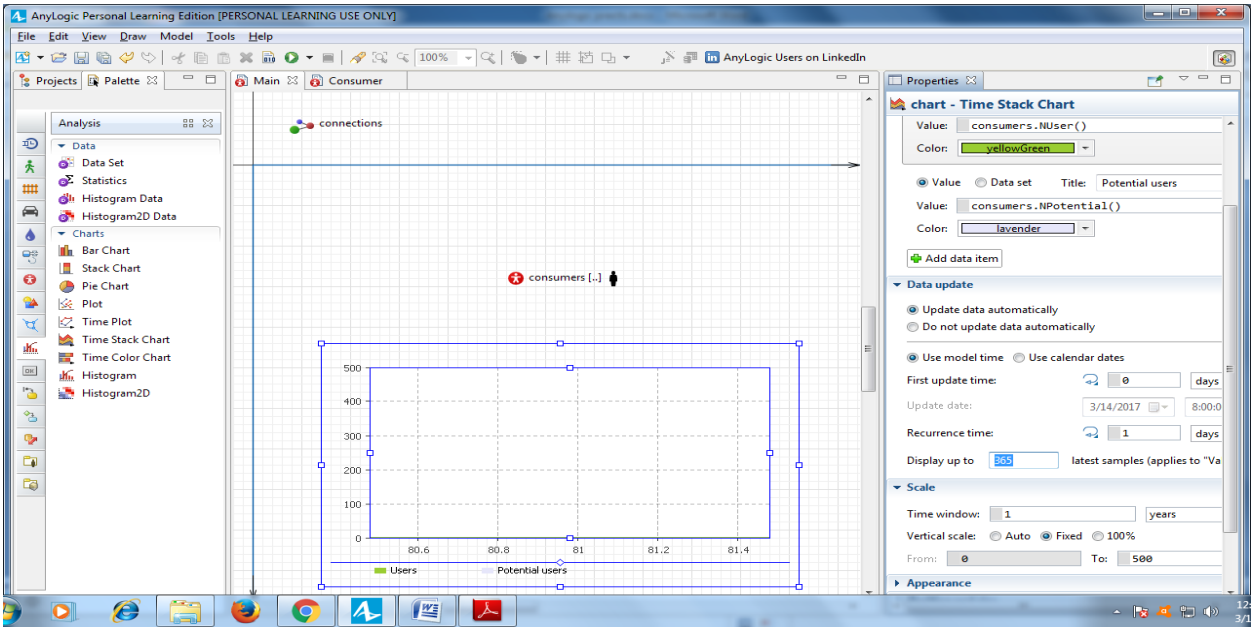

Go to the **Scale** section and set **Time window** equal to 1 year.

Since our chart will show statistics for consumers population and our model has 5,000 consumers, set the chart's **Vertical scale** to **Fixed**, and enter 5000 in the **To:** box.

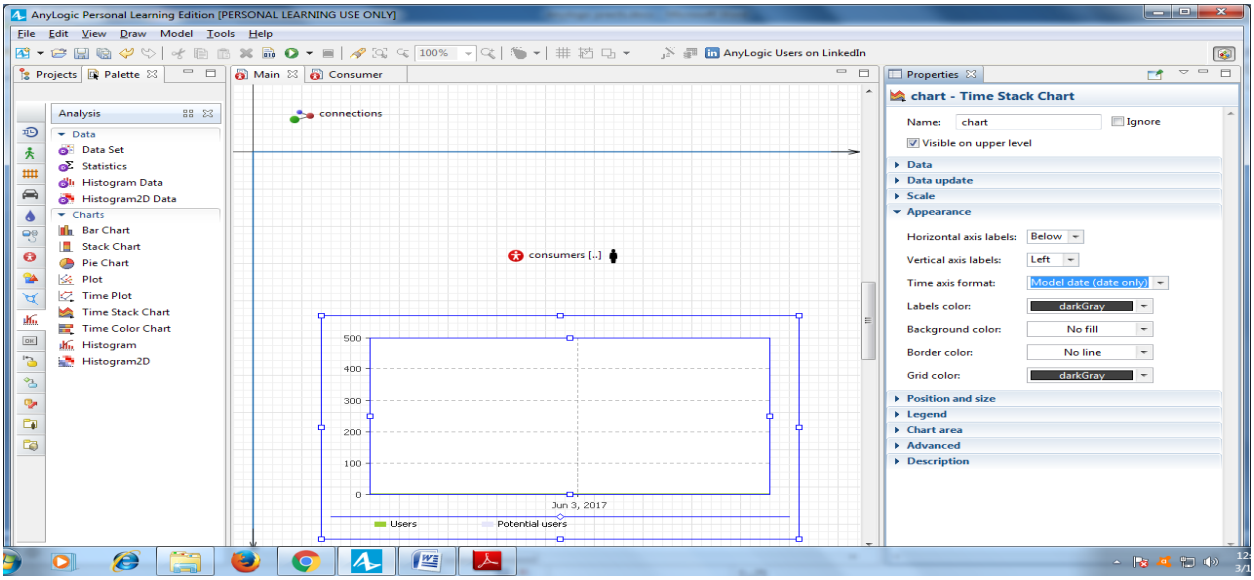



Now that we've set the time window, change the maximum number of data

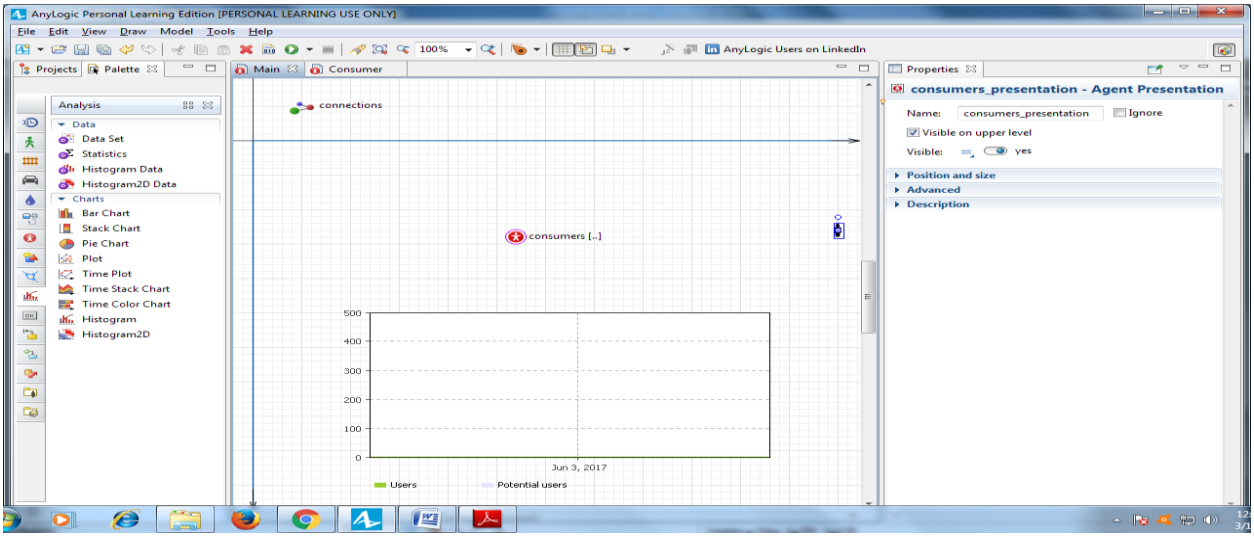
samples that the chart displays by navigating to the section **Data update** and setting **Display up to 365 latest samples**. Since we'll add one data sample each day, 365 data samples is an ideal amount for a one year range.



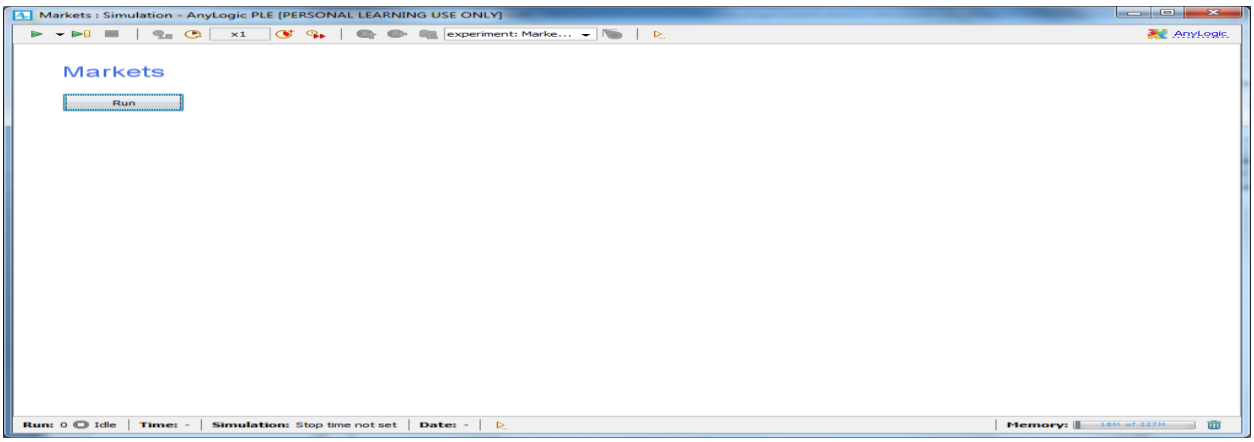

Go to the time stack chart's **Appearance** properties and set it to display **Model date (date only)** near the time axis.

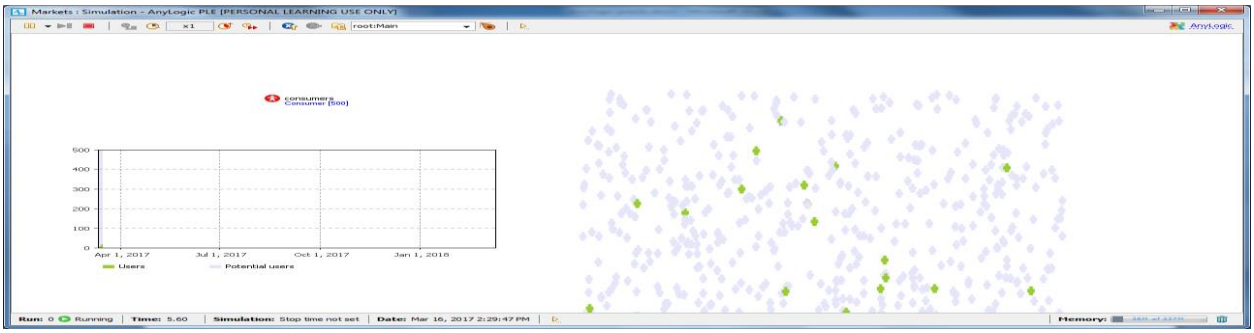


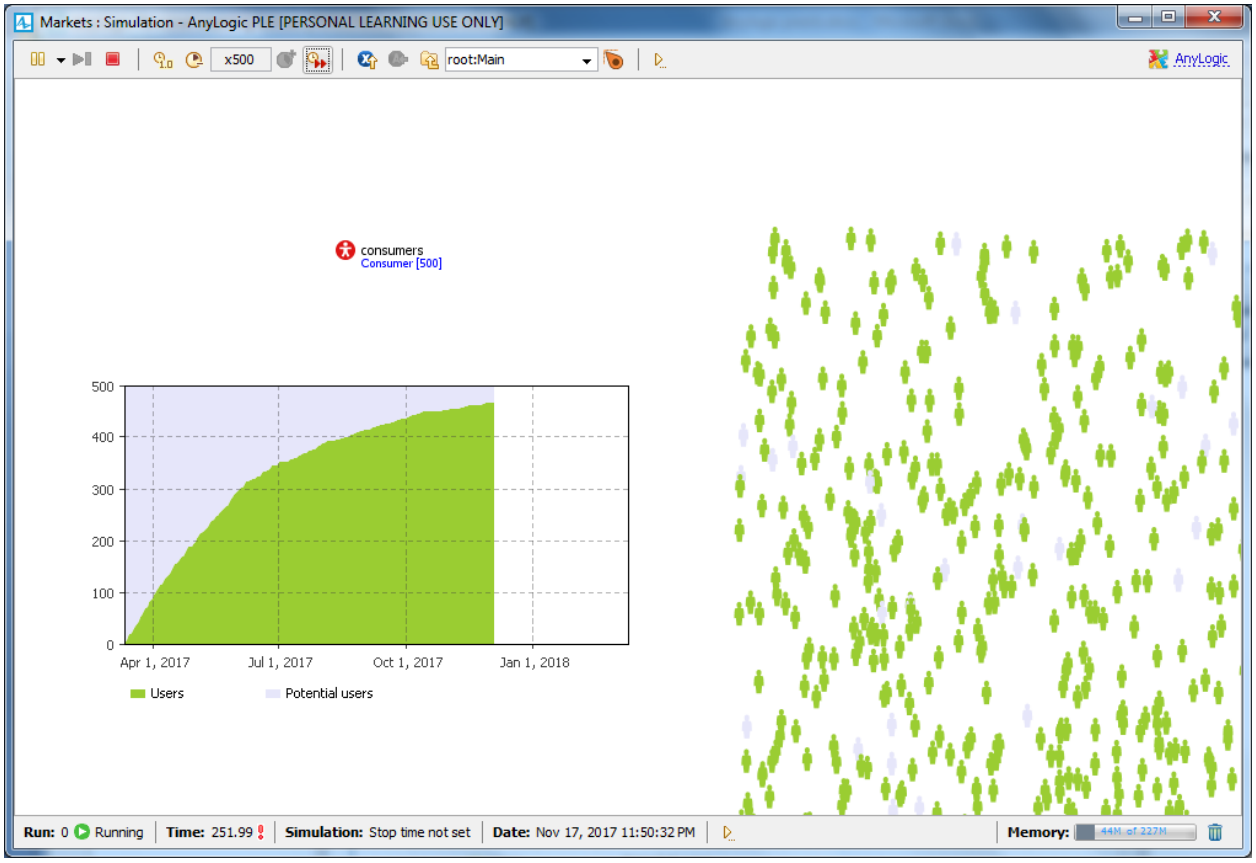

On the Main diagram, move the presentation of the consumers agent population to the right.

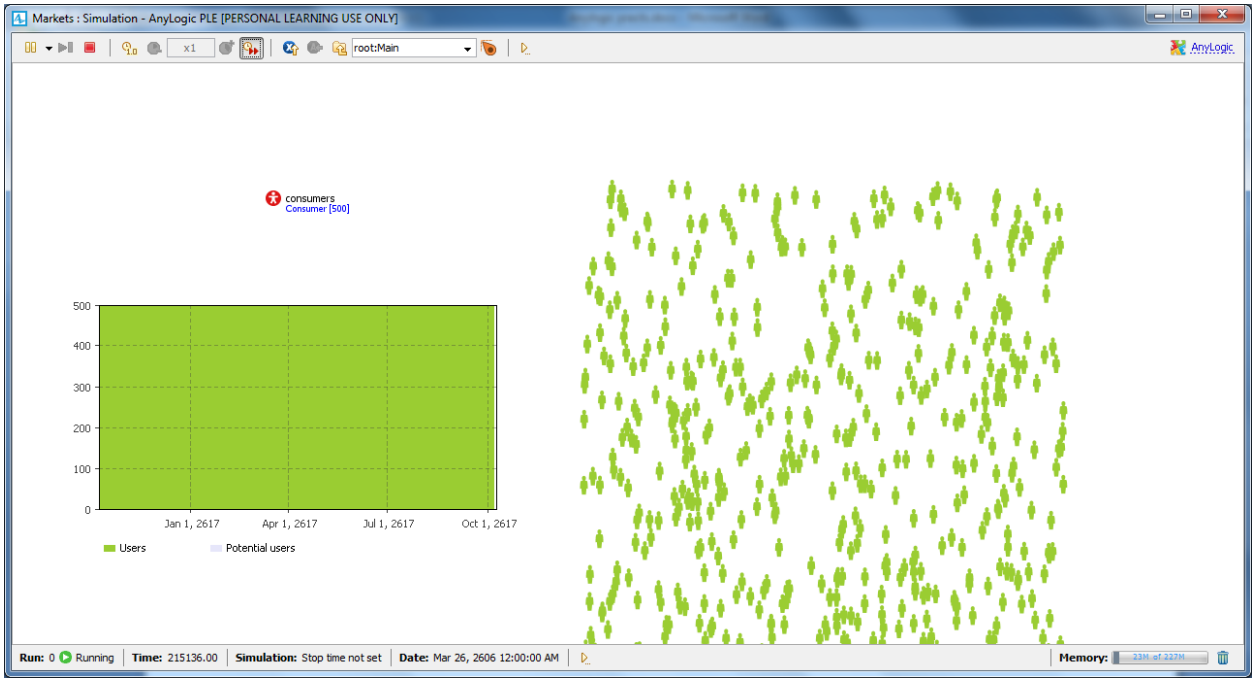



**15.** Run the model and use the time stack chart to review the process.




Markets : Simulation - AnyLogic PLE [PERSONAL LEARNING USE ONLY]
x500
root:Main
AnyLogic
consumers
Consumer [500]
500
400
300
200
100
0
Apr 1, 2017
Jul 1, 2017
Oct 1, 2017
Jan 1, 2018
Users
Potential users
Run: 0 Running
Time: 251.99
Simulation: Stop time not set
Date: Nov 17, 2017 11:50:32 PM
Memory:


Markets : Simulation - AnyLogic PLE [PERSONAL LEARNING USE ONLY]
x1
root:Main
AnyLogic
consumers
Consumer [500]
500
400
300
200
100
0
Jan 1, 2617
Apr 1, 2617
Jul 1, 2617
Oct 1, 2617
Users
Potential users
Run: 0 Running
Time: 215136.00
Simulation: Stop time not set
Date: Mar 26, 2606 12:00:00 AM
Memory:

Practical 2

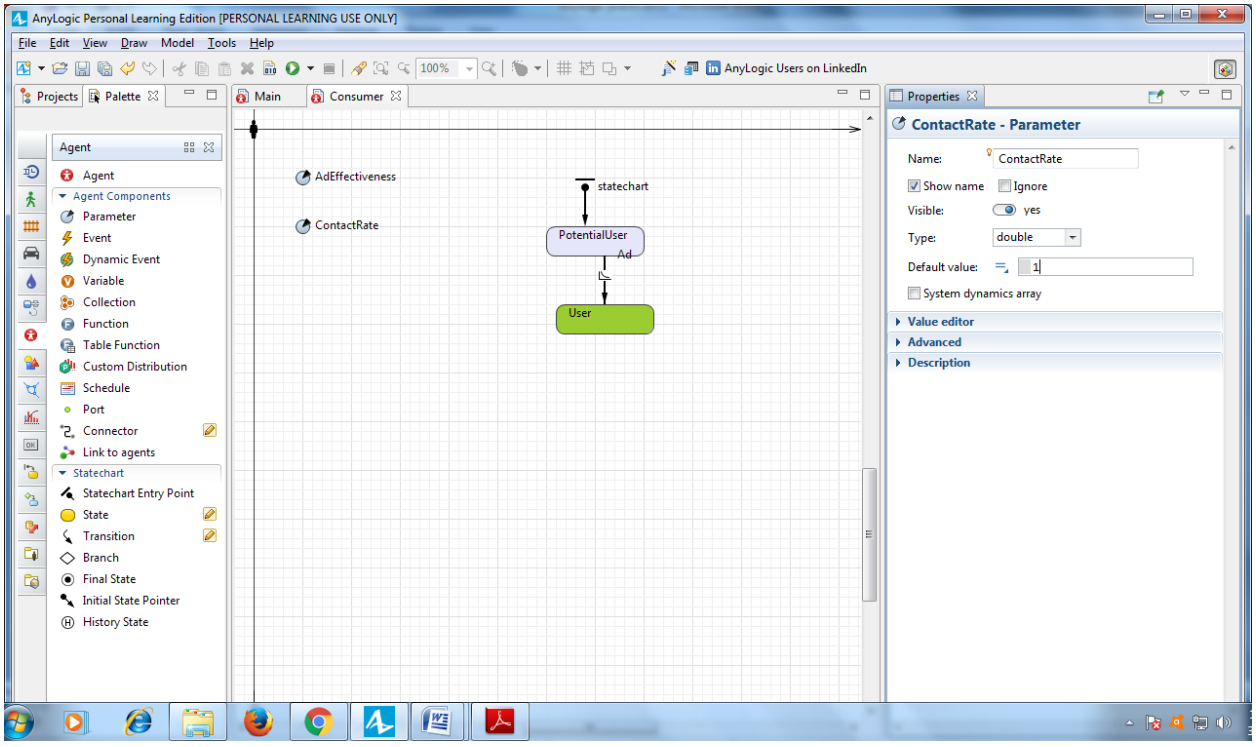
Design and develop agent based model by

- Creating the agent population
- Defining the agent behavior
- Adding a chart to visualize the model output
- Adding word of mouth effect
- Considering product discards

Considering delivery time

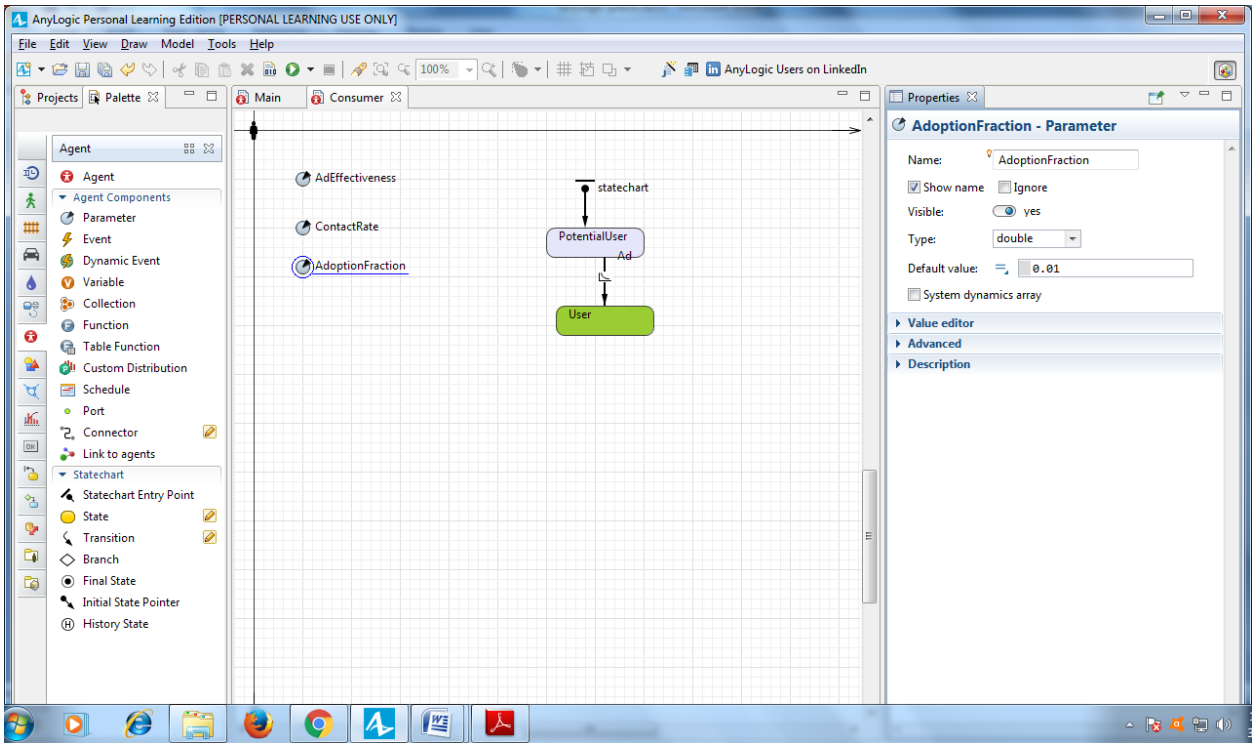
# Adding word of mouth effect

In the **Projects** tree, open Consumer diagram by double-clicking Consumer.
2. Add a parameter to define a consumer's average daily contacts. Drag the **Parameter** from the **Agent** palette on to the diagram.
3. Name the parameter ContactRate.
4. The rate is 1 contact per day, so type 1 as the parameter's **Default value**.

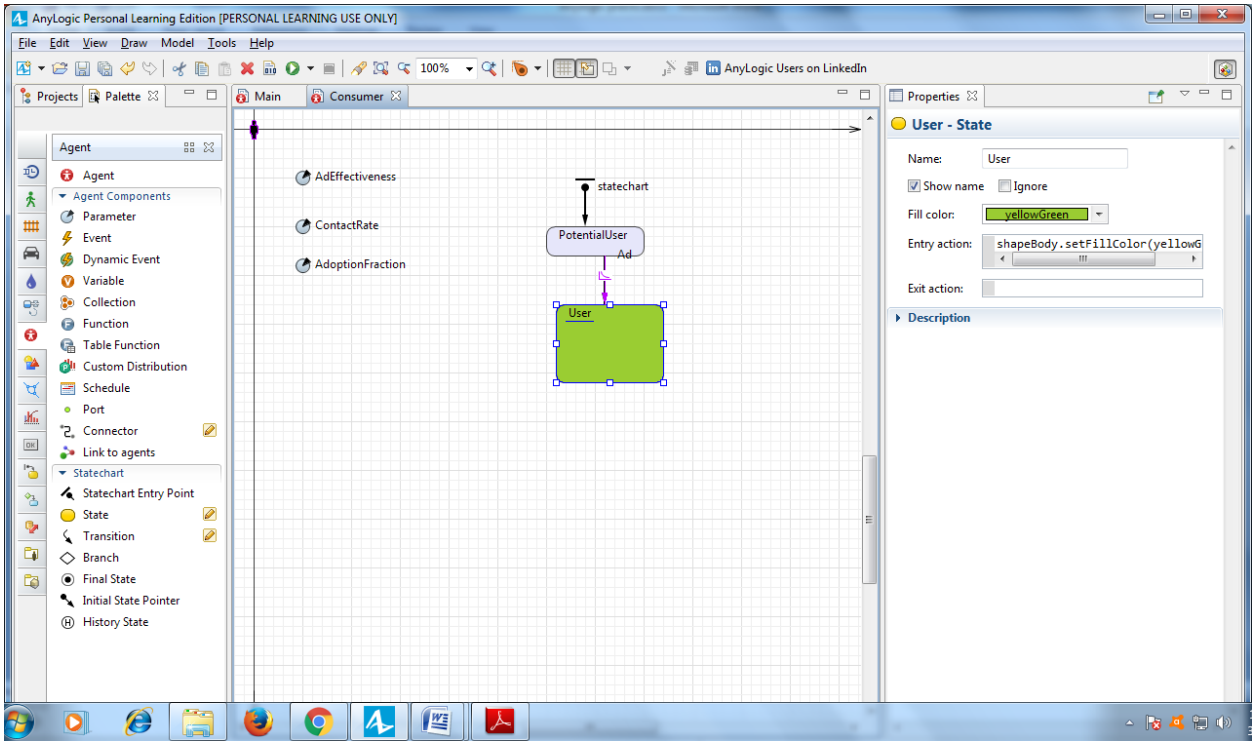



Add another parameter - AdoptionFraction - to define a person's influence on others, a number that we'll express as the percentage of people who will use the product after they come into contact with the consumer. Leave the default parameter's **Type**: double, and set the **Default value:** 0.01.

The Consumer diagram should look like this:



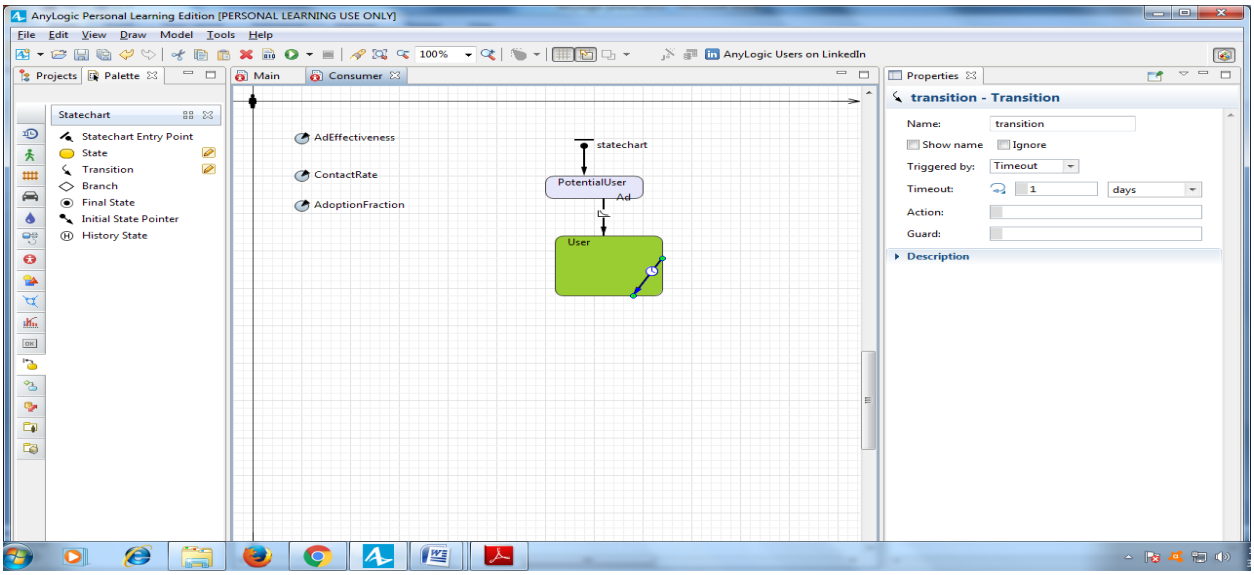

Open the Consumer diagram, and increase the User state to fit the internal transition we'll draw inside the state on the next step.




Draw an internal transition inside the User state. To draw a transition like the

one shown below, drag the **Transition** from the **Statechart** palette inside the state so the transition's start point lies on the state border.

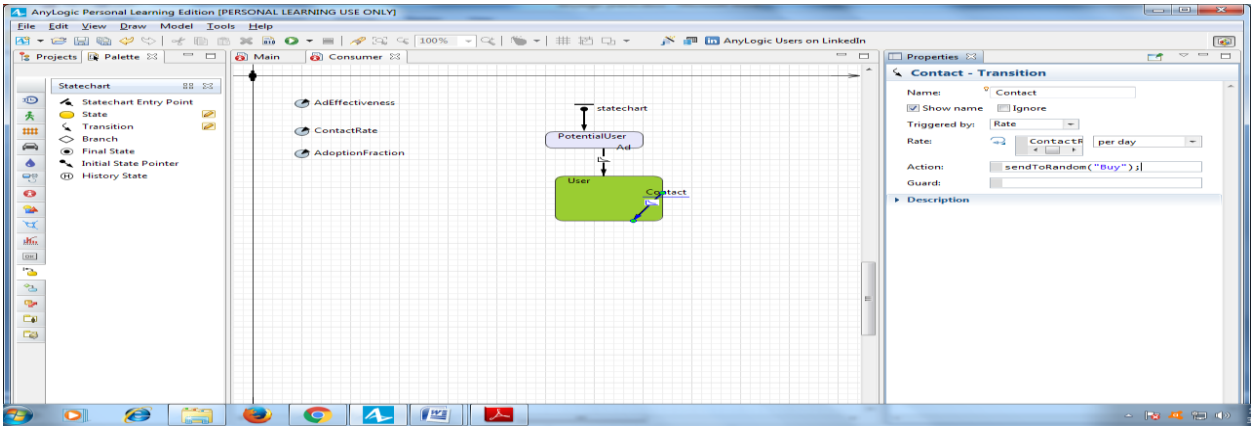
Afterward, you can move the transition end point to another point on the state border. To add a salient point, double-click the transition.




Modify the transition properties. This transition will occur with the specified **Rate** ContactRate (use code completion rather than typing the parameter's full name). Name the transition Contact and set it to show its name.

Specify the **Action** that will be executed on triggering this transition (use the code completion to write the code):

```
sendToRandom("Buy");
```




Draw another transition from PotentialUser to User state, and name it WOM (Word of Mouth). This transition will model purchases caused by word of mouth.

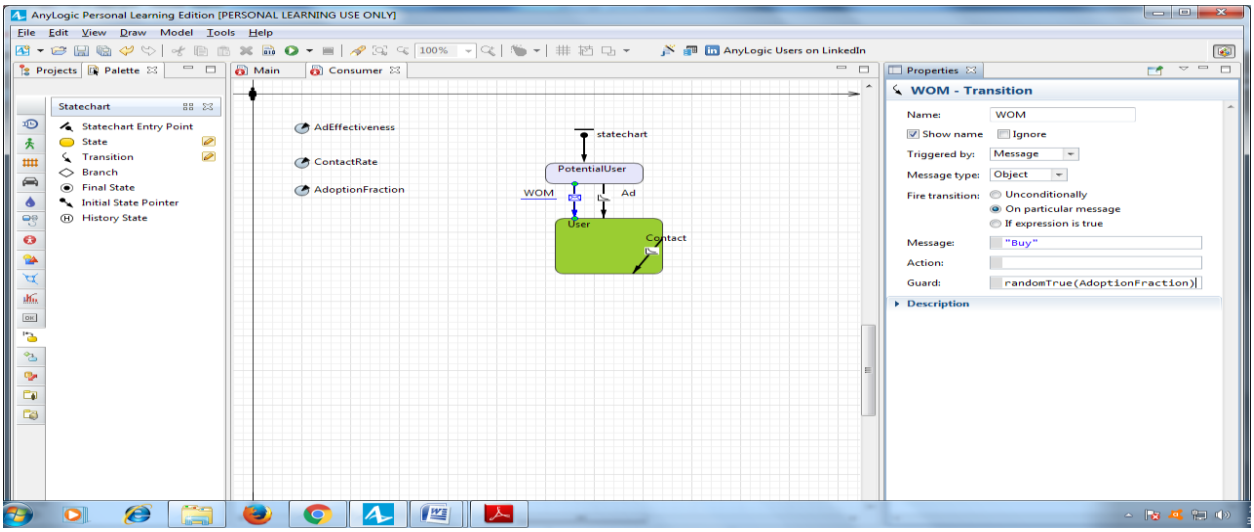
Modify the transition properties:

- In the **Triggered by** list, click **Message**.

• In the **Fire transition** area, select **On particular message**.

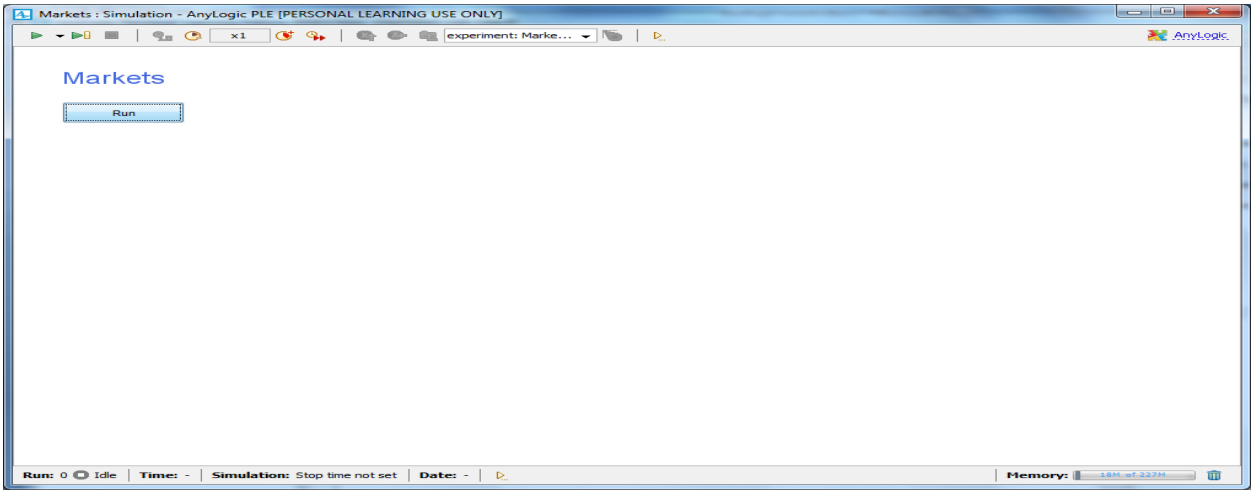
• In the **Message** field, type `"Buy"`

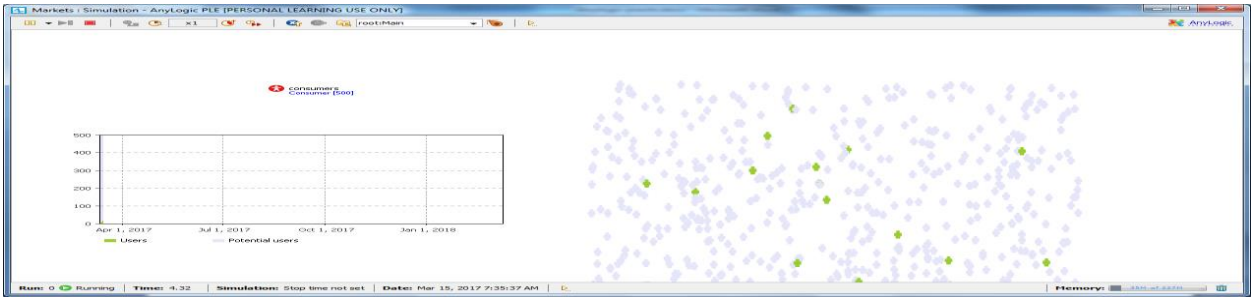
• Since we know not every contact is successful – in other words, a contact may not convince the potential user to buy our product – we'll use AdoptionFraction to make successful contacts less common. Specify the transition's **Guard:** `randomTrue(AdoptionFraction)`

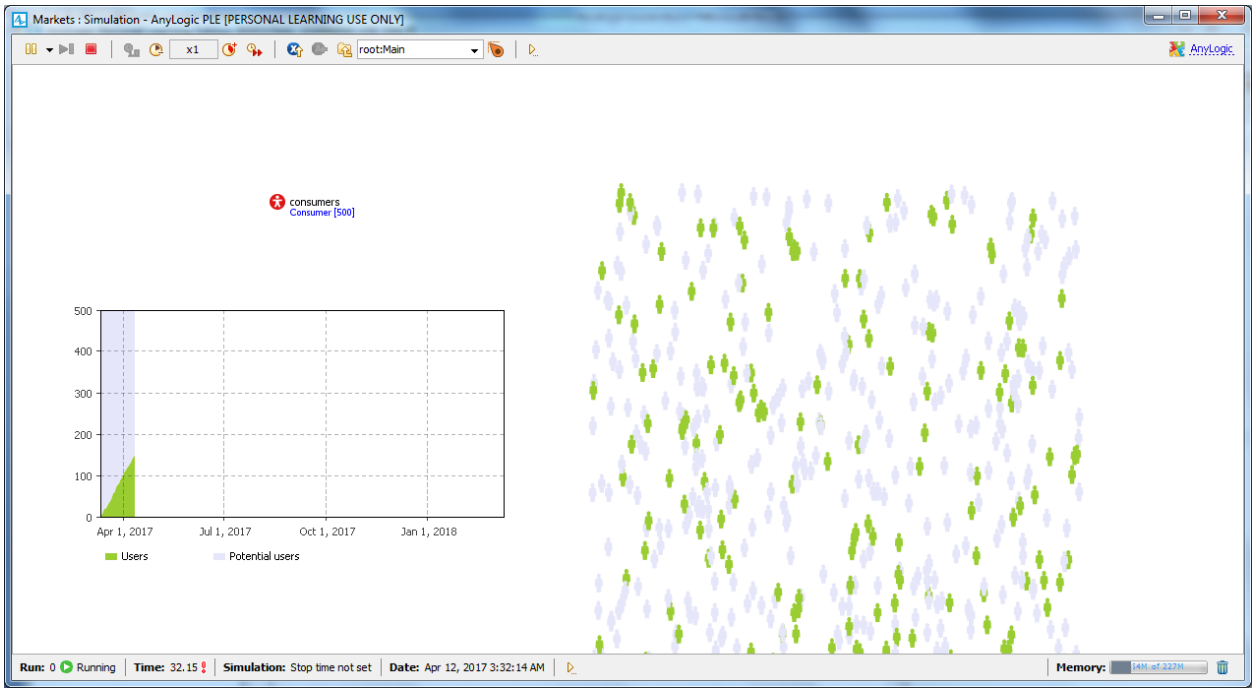


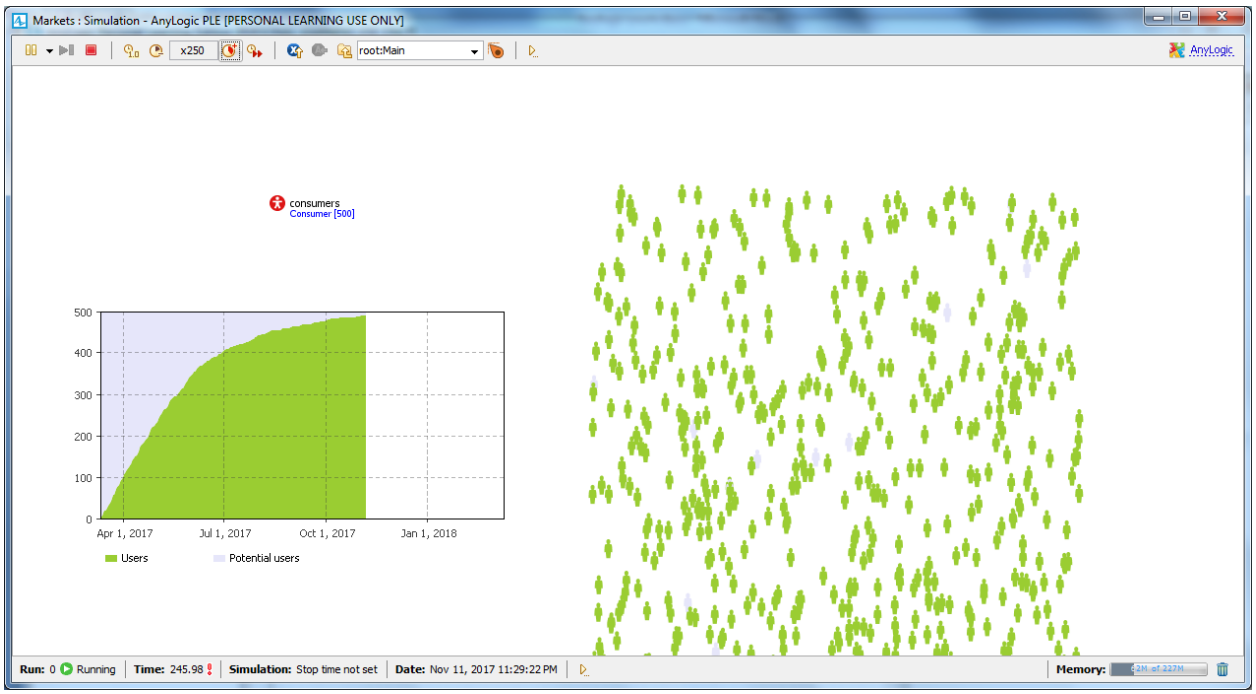

In the **Projects** view, you may see an asterisk near the model item that shows your model has unsaved changes. On the toolbar, click **Save** to save your model.

Run the model.



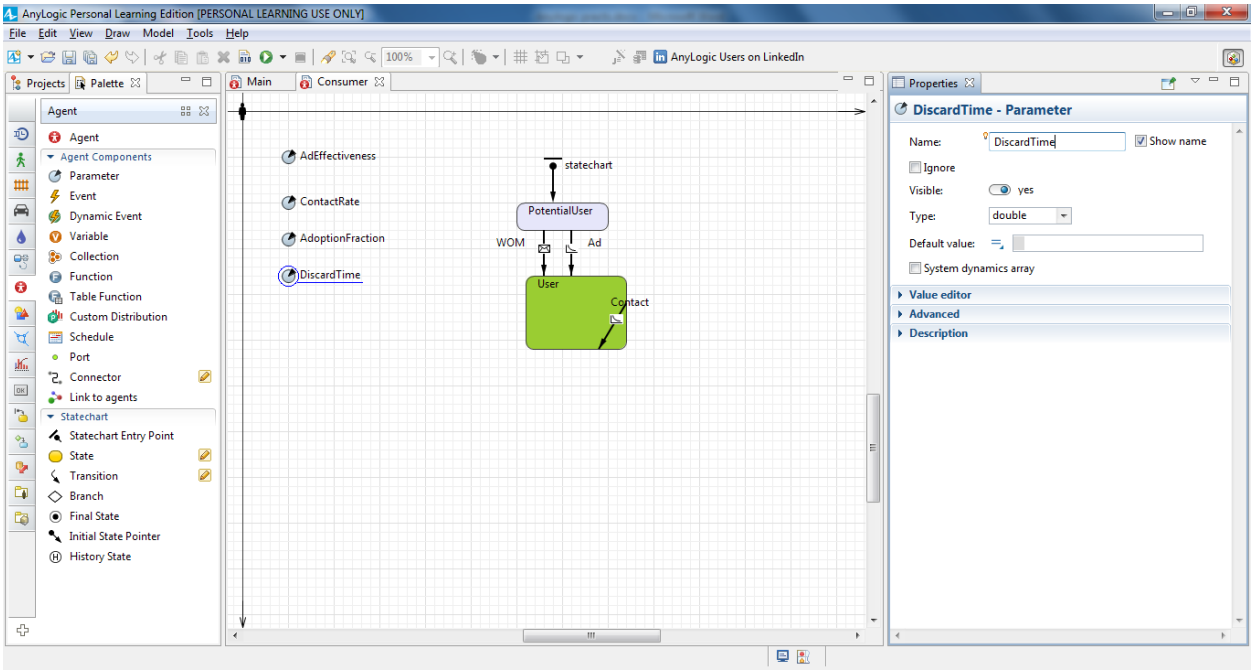

Markets : Simulation - AnyLogic PLE [PERSONAL LEARNING USE ONLY]
consumers
Consumer [500]
500
400
300
200
100
0
Apr 1, 2017
Jul 1, 2017
Oct 1, 2017
Jan 1, 2018
Users
Potential users
Run: 0 Running
Time: 4.32
Simulation: Stop time not set
Date: Mar 15, 2017 7:35:37 AM
Memory:


Markets : Simulation - AnyLogic PLE [PERSONAL LEARNING USE ONLY]
x1
root:Main
AnyLogic
consumers
Consumer [500]
500
400
300
200
100
0
Apr 1, 2017
Jul 1, 2017
Oct 1, 2017
Jan 1, 2018
Users
Potential users
Run: 0 Running
Time: 32.15
Simulation: Stop time not set
Date: Apr 12, 2017 3:32:14 AM
Memory:

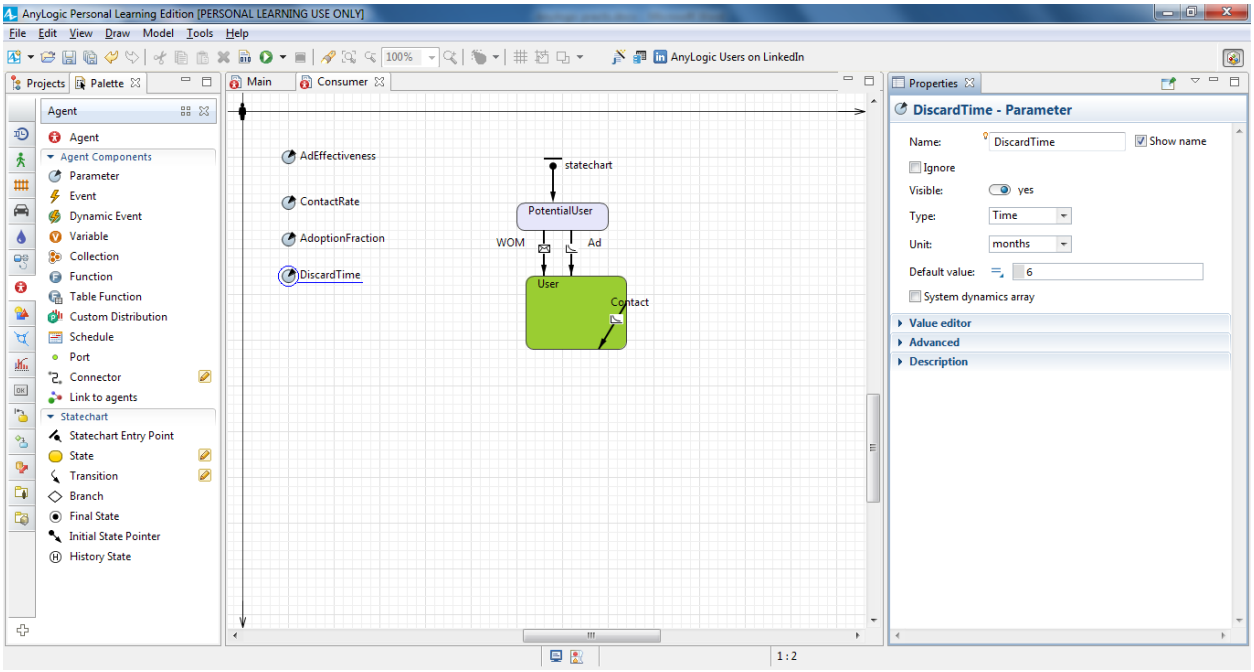

Markets : Simulation - AnyLogic PLE [PERSONAL LEARNING USE ONLY]
x250
root:Main
AnyLogic
consumers
Consumer [500]
500
400
300
200
100
0
Apr 1, 2017
Jul 1, 2017
Oct 1, 2017
Jan 1, 2018
Users
Potential users
Run: 0 Running
Time: 245.98
Simulation: Stop time not set
Date: Nov 11, 2017 11:29:22 PM
Memory:

# Considering product discards

Open the Consumer diagram and add a DiscardTime parameter

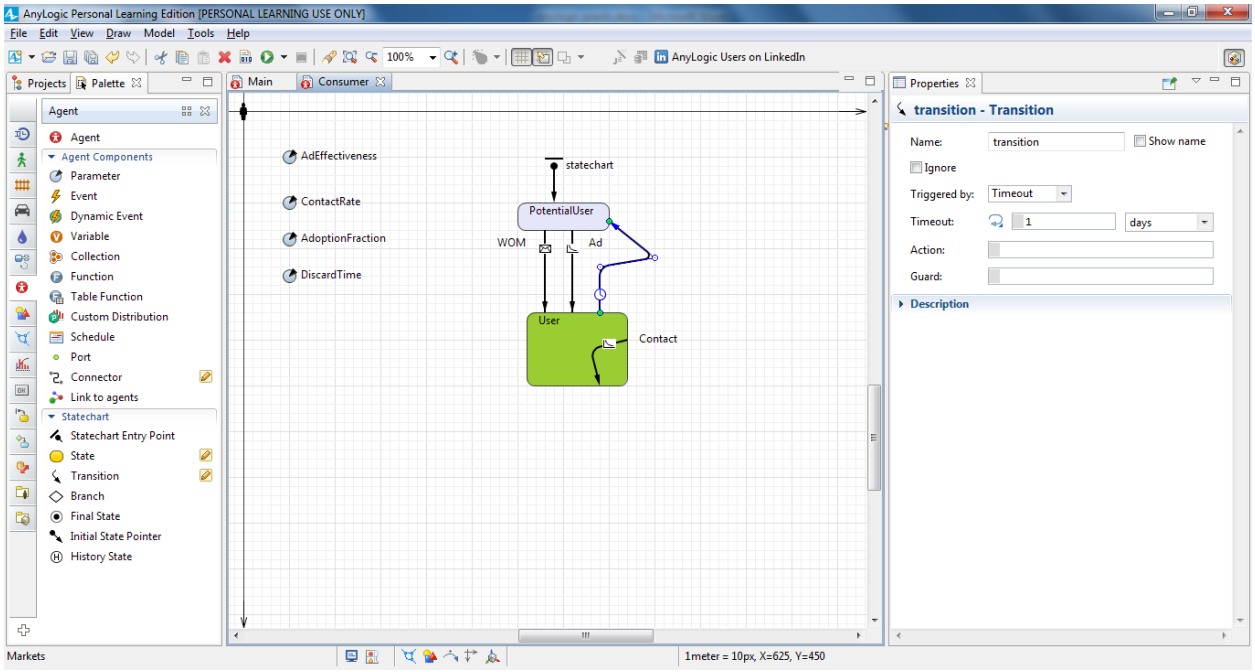



This parameter will define our product's lifespan. Choose **Time** as the parameter's **Type**, click **months** in the **Unit** list, and type 6 as the **Default value**.

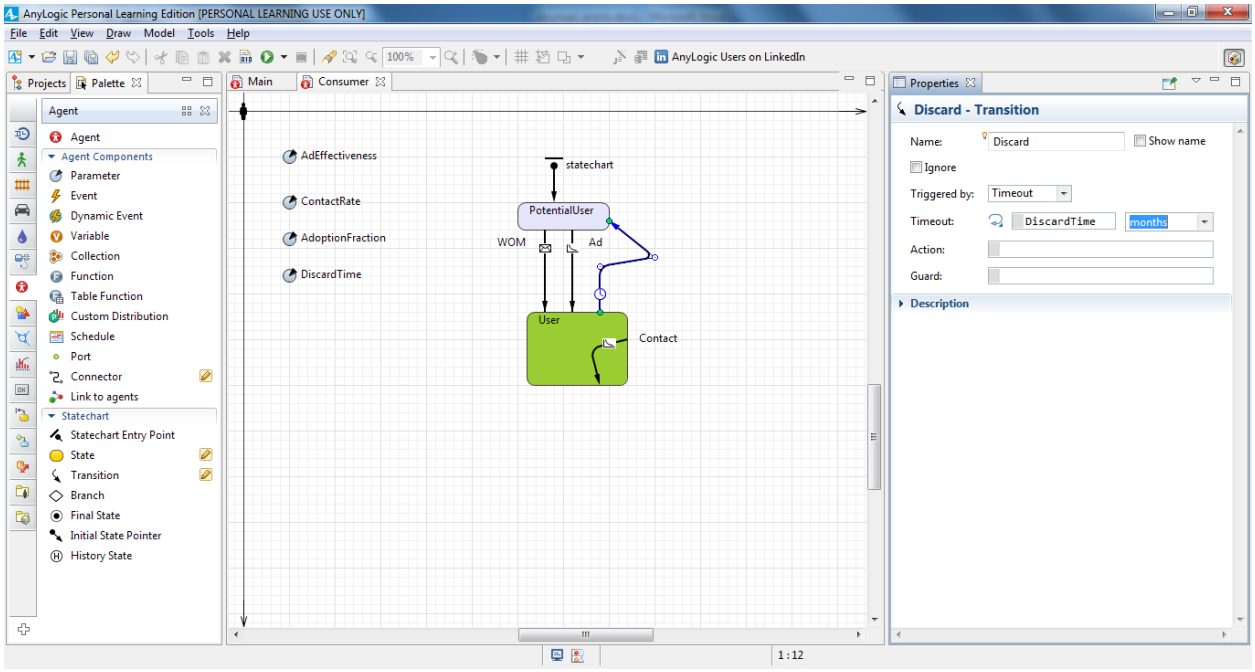


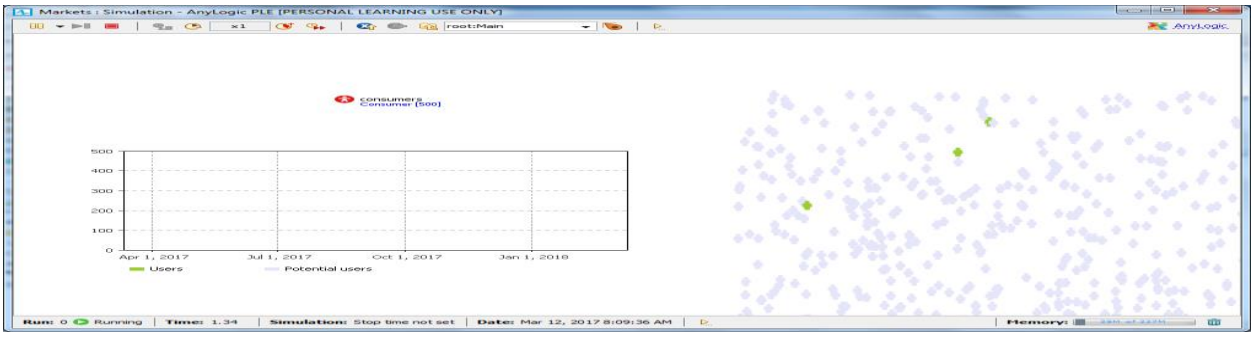

Draw a transition from User to PotentialUser state to model product discards.
To draw a transition with salient points like those shown in the figure,

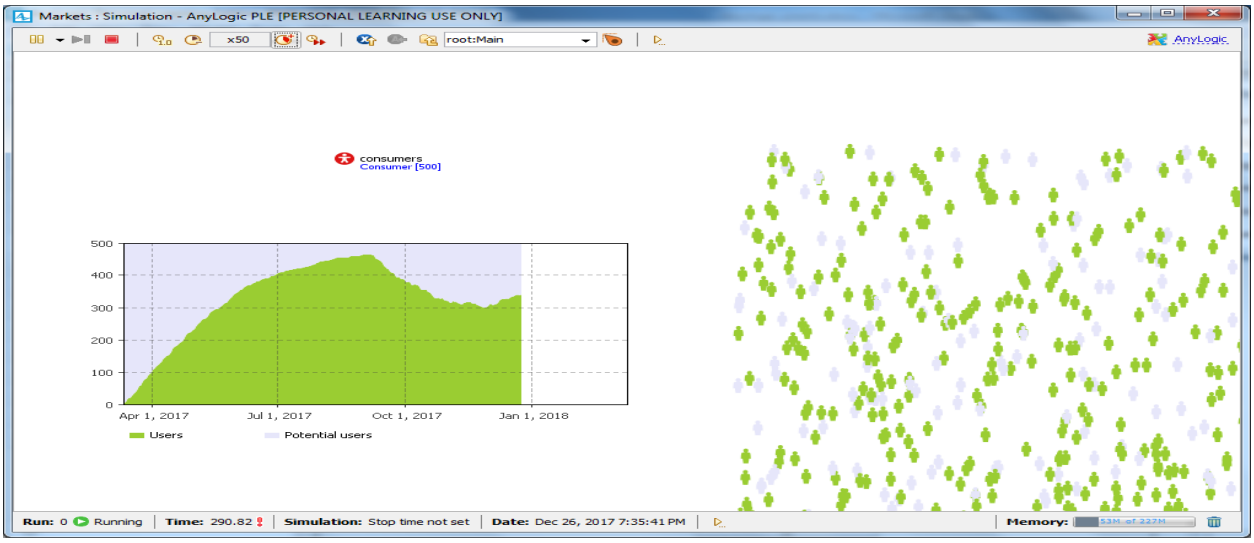
double-click the **Transition** element in the **Statechart** palette (this should change the element's icon in the palette to ), click the transition's source state User, click at the salient point places, and click the target state PotentialUser.




Name the transition Discard and set it to be triggered by a constant timeout DiscardTime. In the list to the right, click **months**.

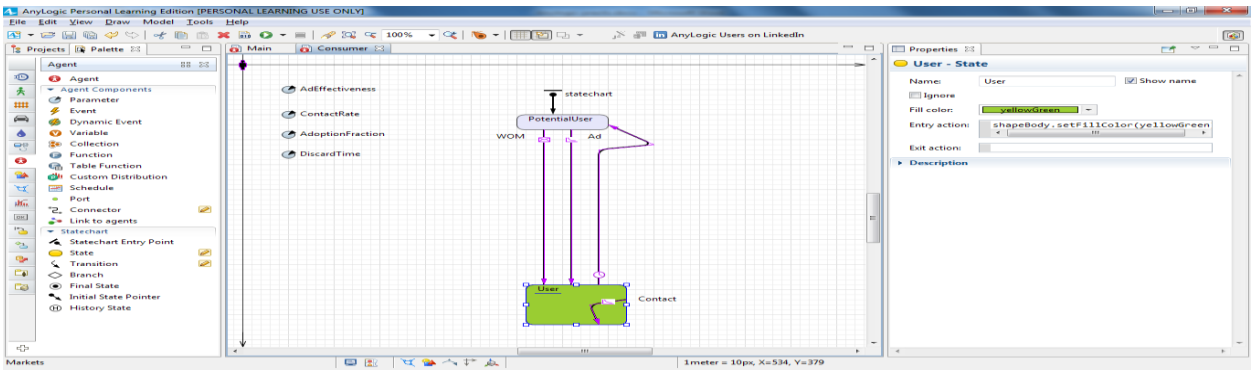




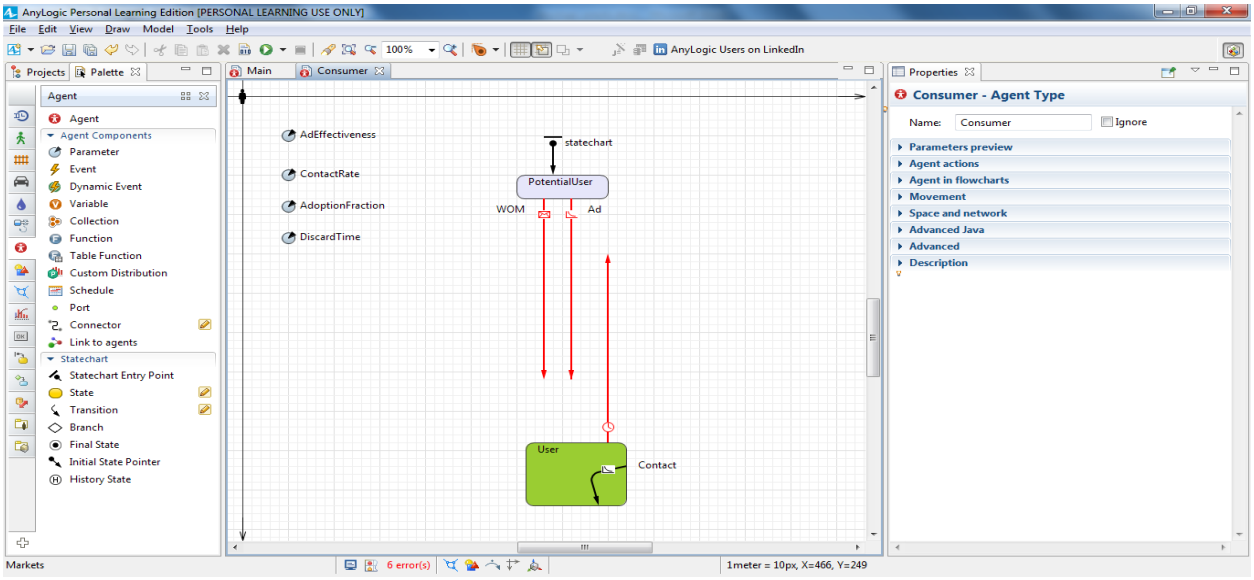




# Considering delivery time

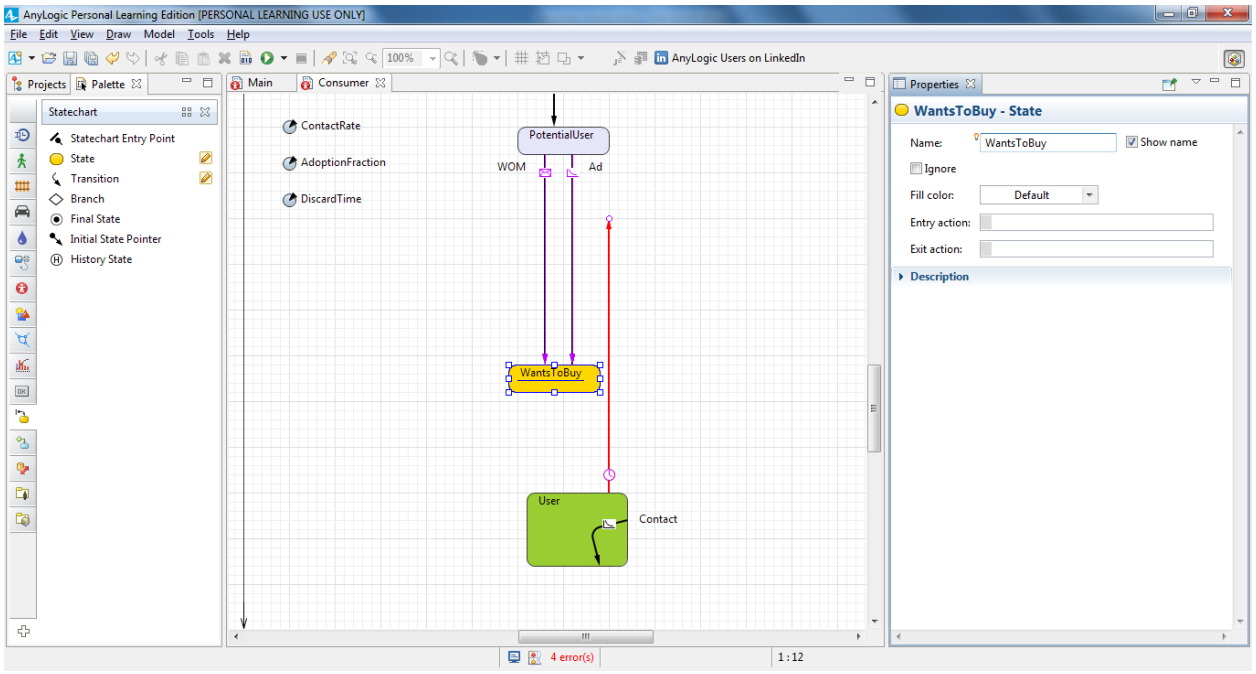
Prepare a place for another state between PotentialUser and User by moving the User state toward the bottom of the screen.



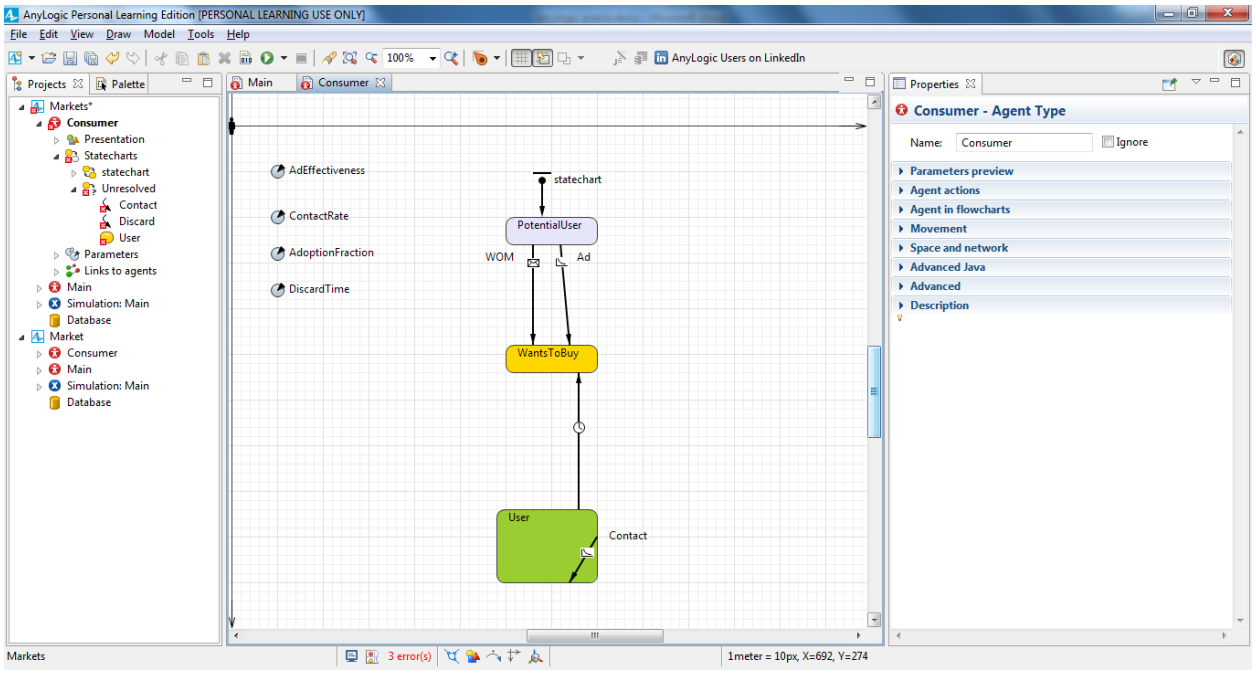

Disconnect the User state from the transitions.
Select the WOM and Ad transitions, move their end points toward the top of the screen, and disconnect the Discard transition from PotentialUser.
Afterward, you'll notice the disconnected transitions are drawn in red.




Add another **State** from the **Statechart** palette to the middle of the consumer's statechart and name it WantsToBuy. Consumers in this state have decided to purchase the product, but they have not done so.



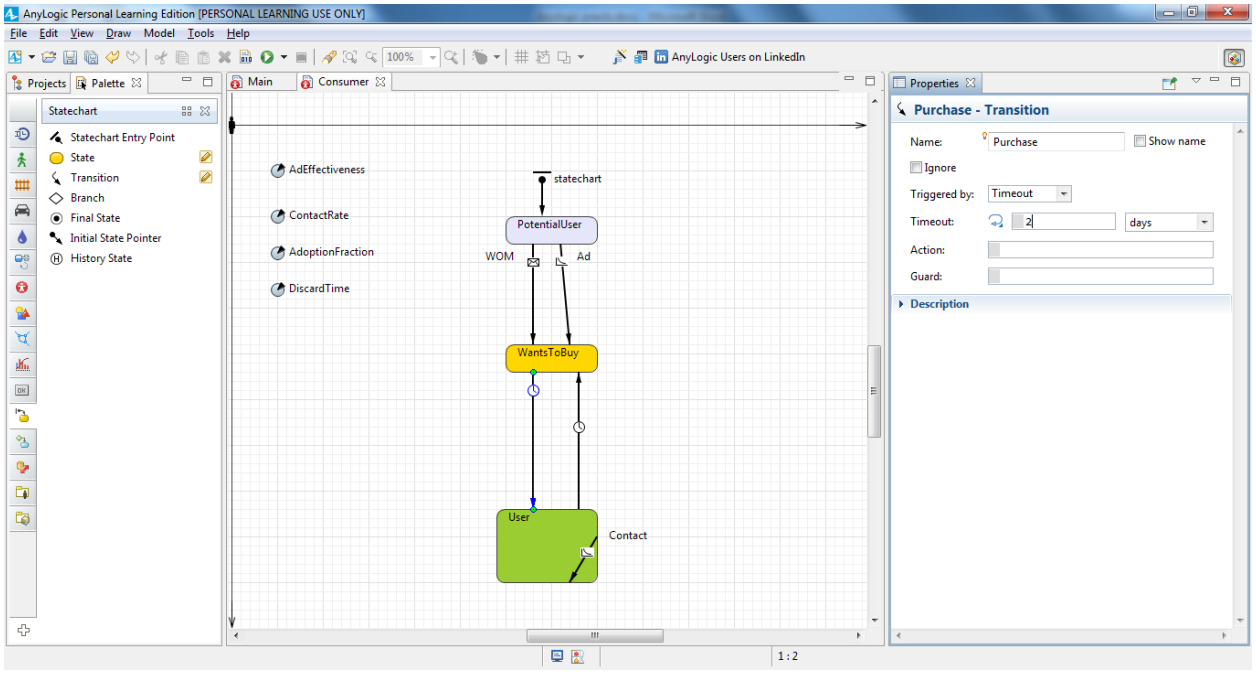

Reconnect transitions to the middle state: the WOM, Ad, and Discard transitions should now end in the WantsToBuy state.



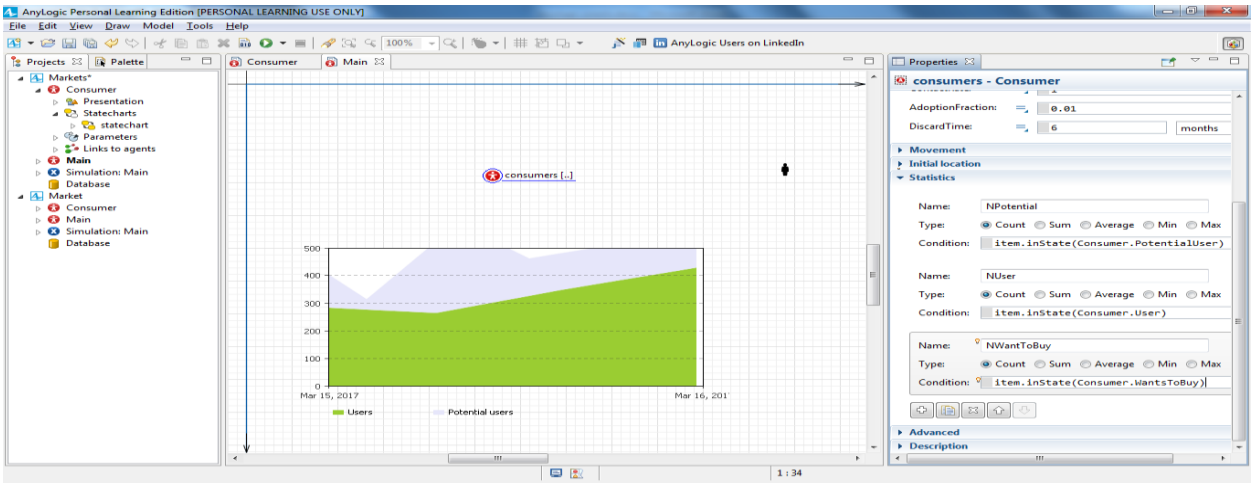

Modify WantsToBuy similar to other states:
**Fill color**: gold
**Entry action**: shapeBody.setFillColor(gold)

**6.** Add a transition from WantsToBuy to User state to model the product shipment and name it Purchase.

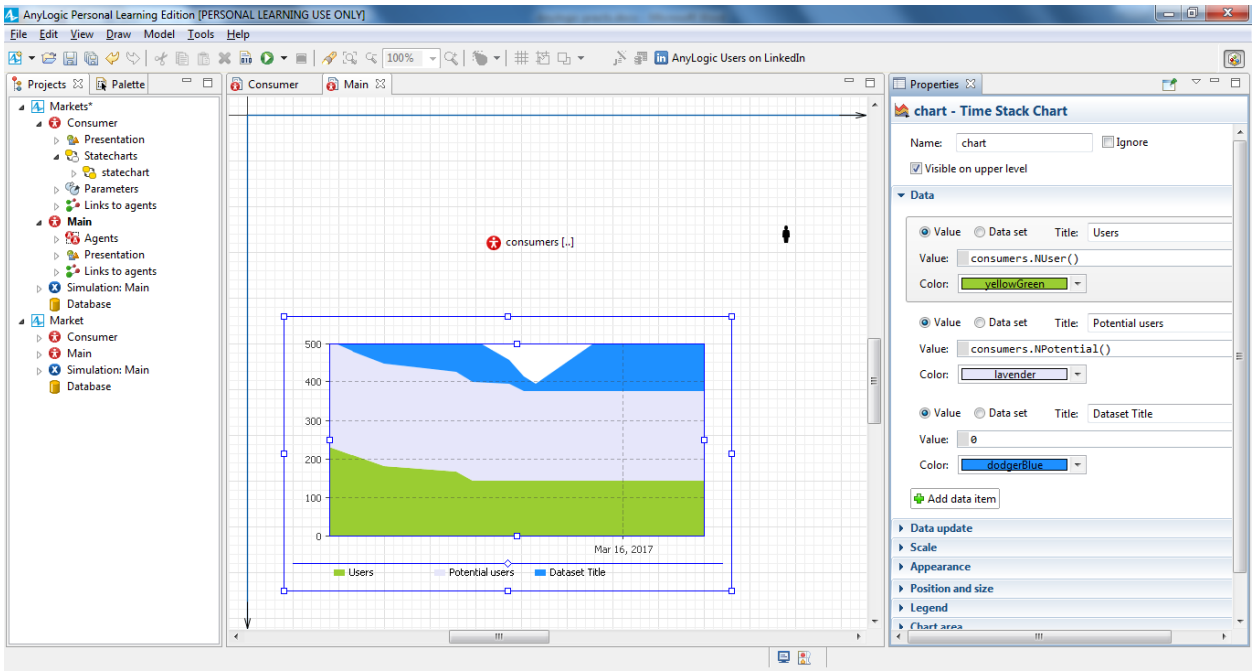
Let's assume it typically takes a user two days to get the product. This means once the consumer's statechart enters the state WantsToBuy, it will proceed to the state User with a two-day delay. With this in mind, set 2 days timeout for the Purchase transition:

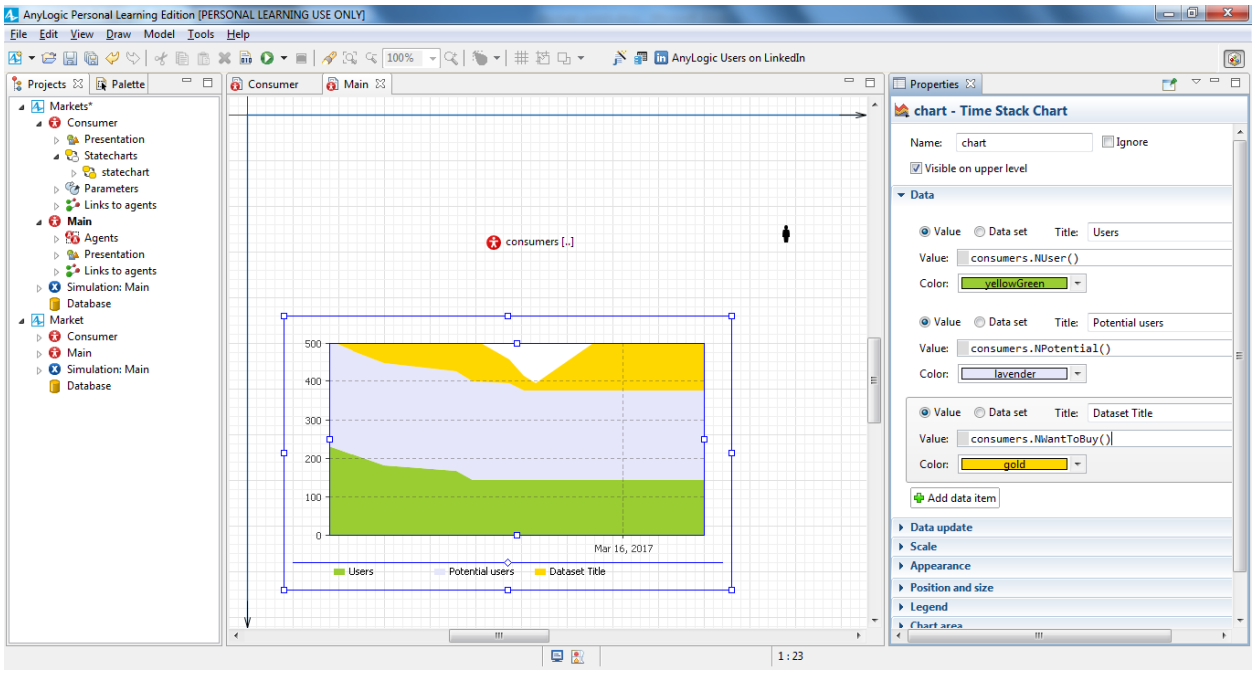


Define one more statistics function to count the product's market-driven demand. In the editor of Main, click the consumers, go to the **Statistics** properties section, and add a statistics item: NWantToBuy with condition `item.inState(Consumer.WantsToBuy)`

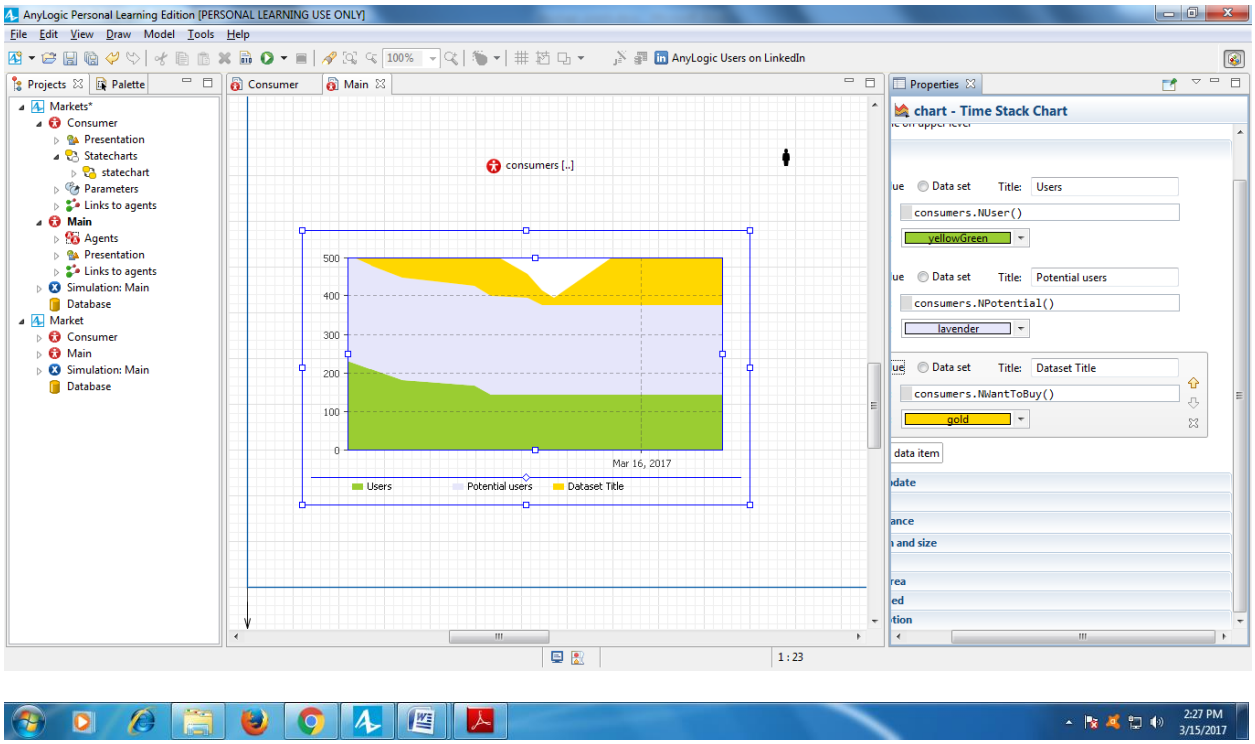



On Main, select the time stack chart, and add another data item to be displayed with the chart: consumers.NWantToBuy() with the title Want to buy and color gold.

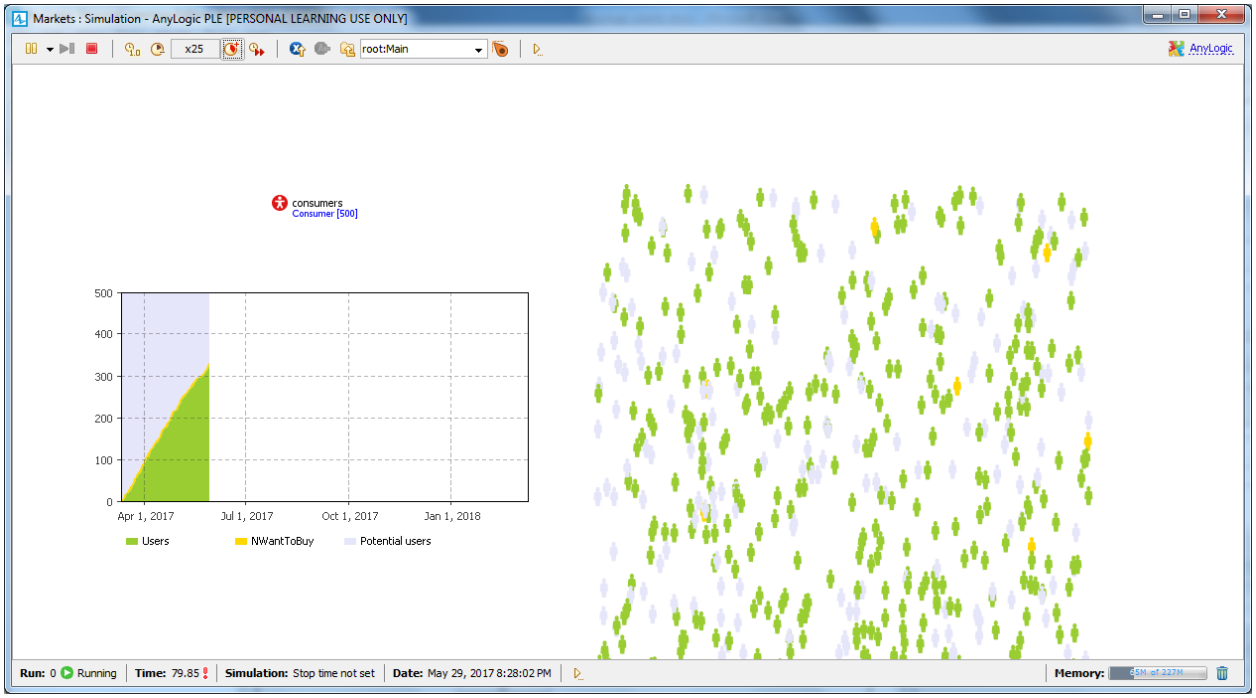


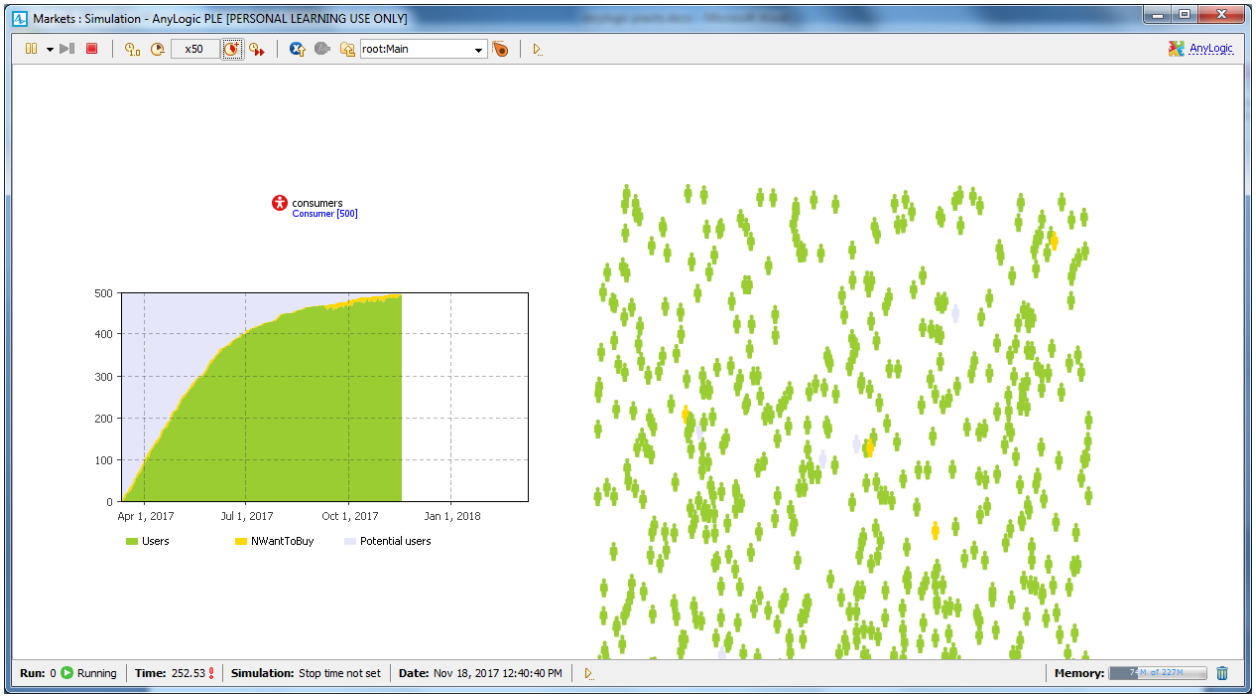



Make the newly-defined data item second in the list by selecting the item's section and clicking the "up" button .




Run the model, and you'll notice AnyLogic displays the number of consumers who are waiting for the product in yellow.


Markets : Simulation - AnyLogic PLE [PERSONAL LEARNING USE ONLY]
x25
root:Main
AnyLogic
consumers
Consumer [500]
500
400
300
200
100
0
Apr 1, 2017
Jul 1, 2017
Oct 1, 2017
Jan 1, 2018
Users
NWantToBuy
Potential users
Run: 0 Running
Time: 79.85
Simulation: Stop time not set
Date: May 29, 2017 8:28:02 PM
Memory:

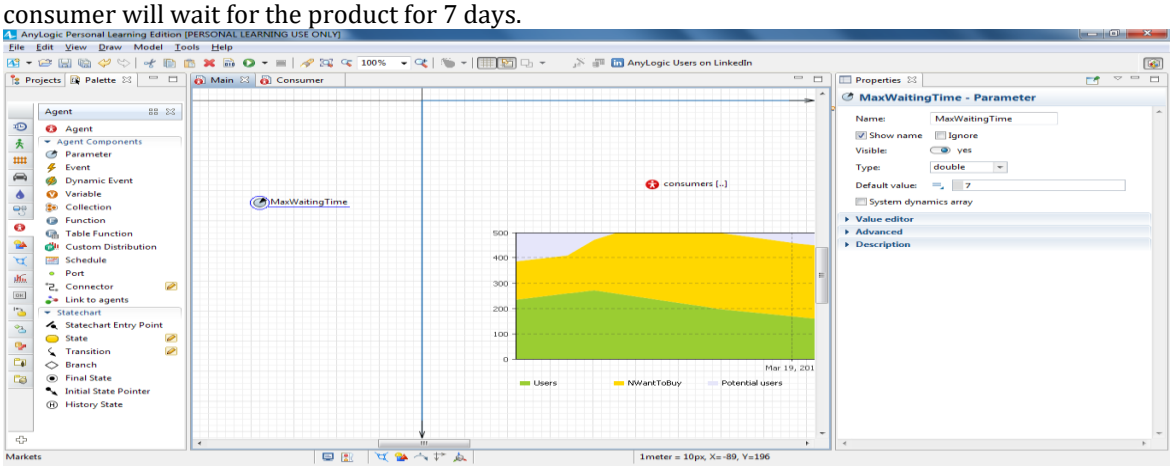

Markets : Simulation - AnyLogic PLE [PERSONAL LEARNING USE ONLY]
x50
root:Main
AnyLogic
consumers
Consumer [500]
500
400
300
200
100
0
Apr 1, 2017
Jul 1, 2017
Oct 1, 2017
Jan 1, 2018
Users
NWantToBuy
Potential users
Run: 0 Running
Time: 252.53
Simulation: Stop time not set
Date: Nov 18, 2017 12:40:40 PM
Memory:

Practical No. 3

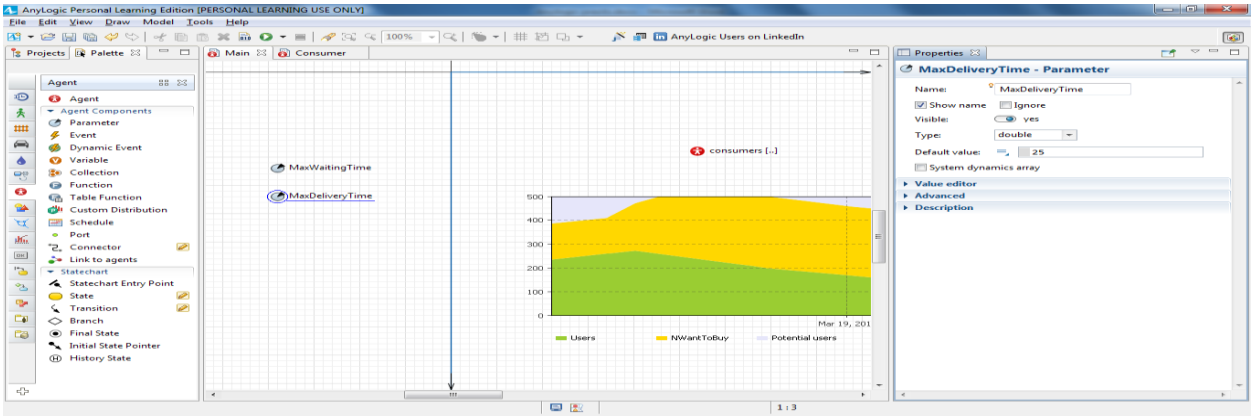
Design and develop agent based model by

- Creating the agent population
- Defining the agent behavior
- Adding a chart to visualize the model output
- Adding word of mouth effect
- Considering product discards
- Consider delivery time
- **Simulating agent/Consumer impatience**
- **Comparing model runs with different parameter values**

Configure the parameters. MaxWaitingTime defines the maximum time a consumer will wait for the product for 7 days.



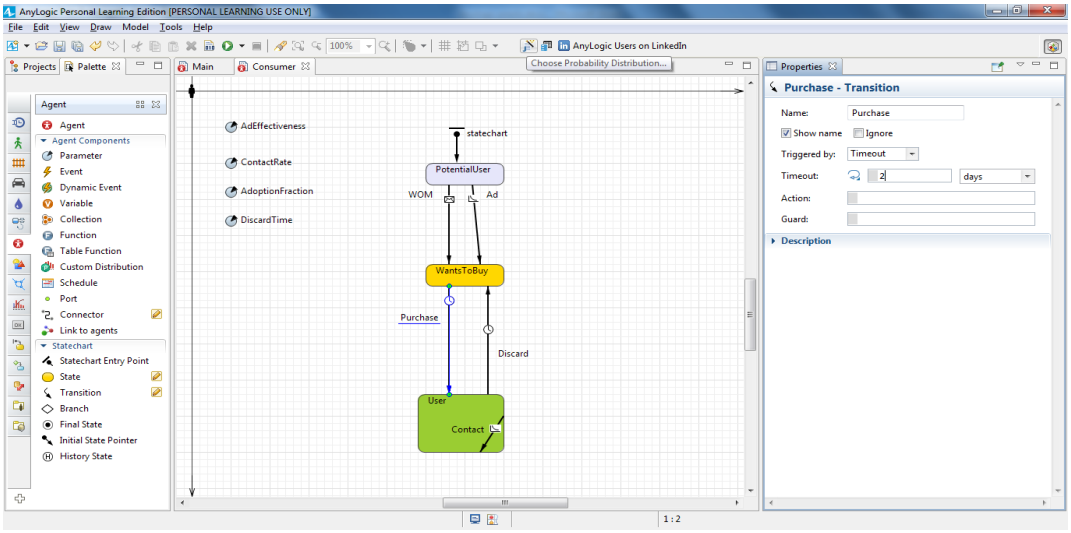

Set the other parameter, MaxDeliveryTime to 25 days to reflect our assumption it may take up to 25 days to deliver a product.



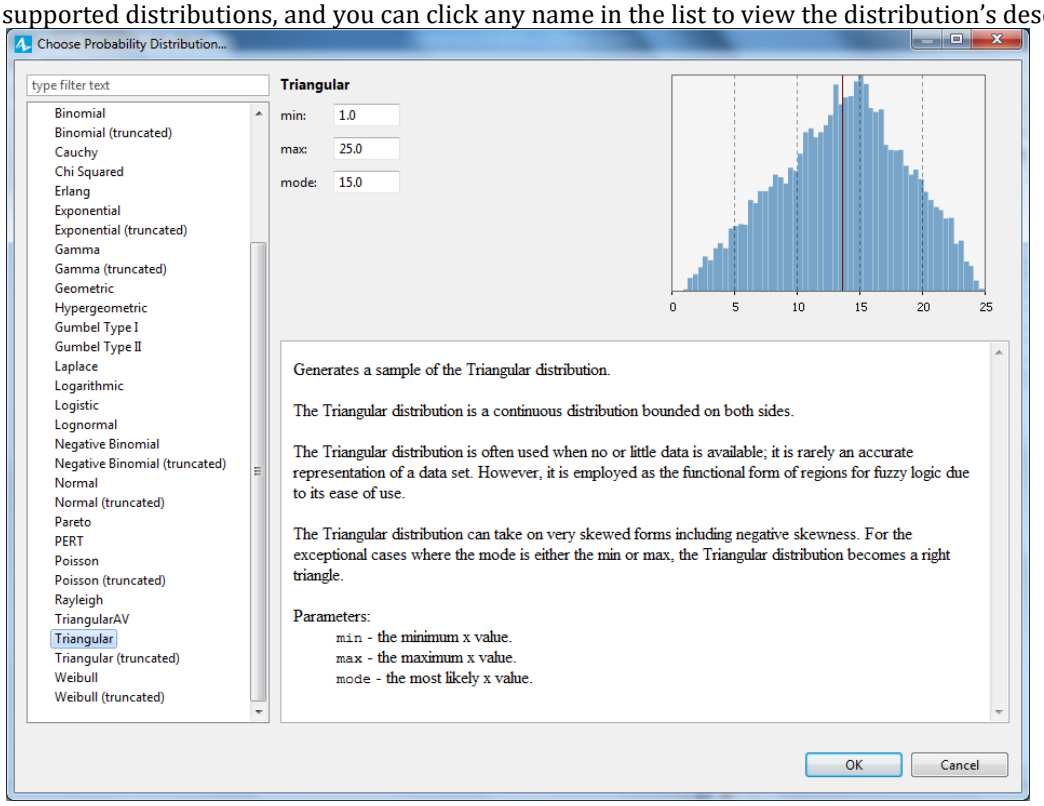

We assume it takes between one and 25 days – with an average of two days – to deliver the product. With that in mind, let's change the delivery time from a fixed

two day delivery period to the stochastic expression that describes this pattern.

Open the Consumer diagram and select the Purchase transition. We want to change the transition's timeout expression, and we'll do that by using a wizard to choose the distribution function and insert the function's name in the property. To substitute the existing value, use your mouse to select the existing **Timeout** expression.
Click the **Choose Probability Distribution...** toolbar button.

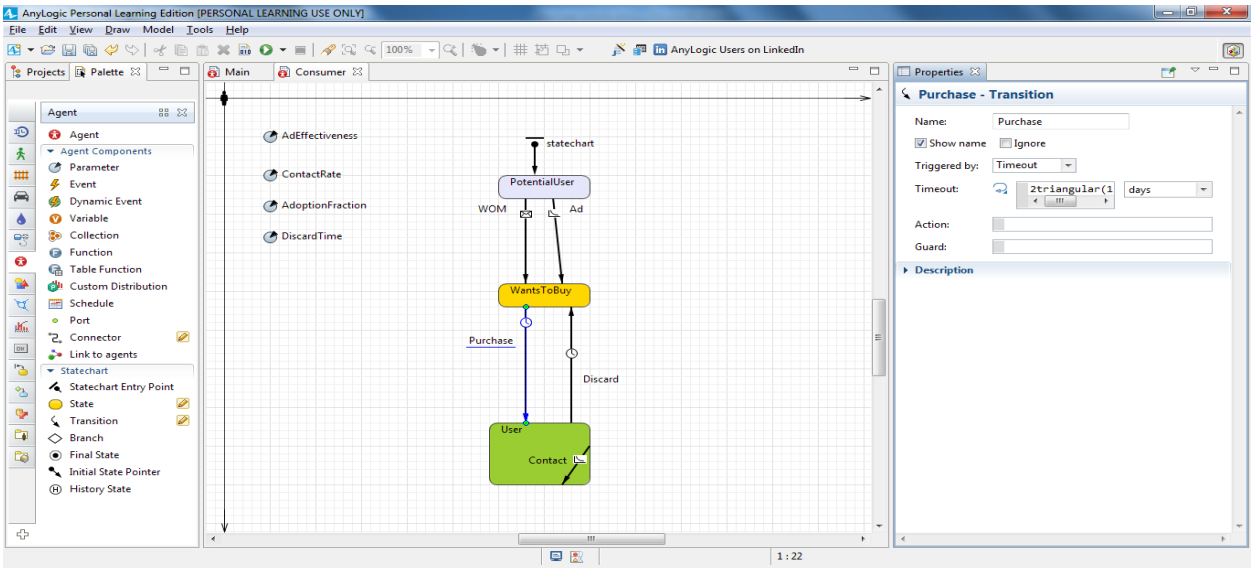



You'll see the **Choose Probability Distribution...** dialog box.
The **Choose Probability Description** screen allows you to view the list of supported distributions, and you can click any name in the list to view the distribution's description.

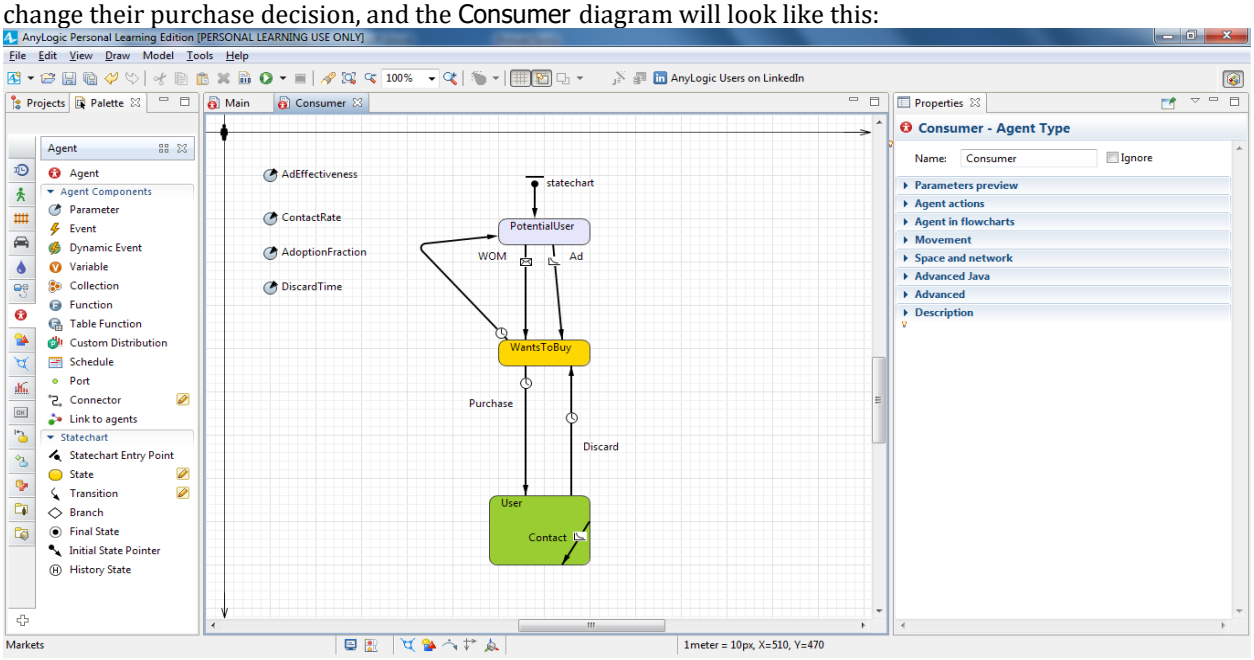


Choose triangular in the list. Set **min**, **max** and **mode** parameters equal to 1, 25, 2 respectively. In the upper right, you'll see PDF instantly built for the distribution with the specified parameters. Click **OK** when finished.

You'll see the expression triangular(1, 25, 2) automatically inserted as the timeout value. Let's modify the line to triangular(1, main.MaxDeliveryTime, 2) Here main is how we access the Main agent from the consumer agent.

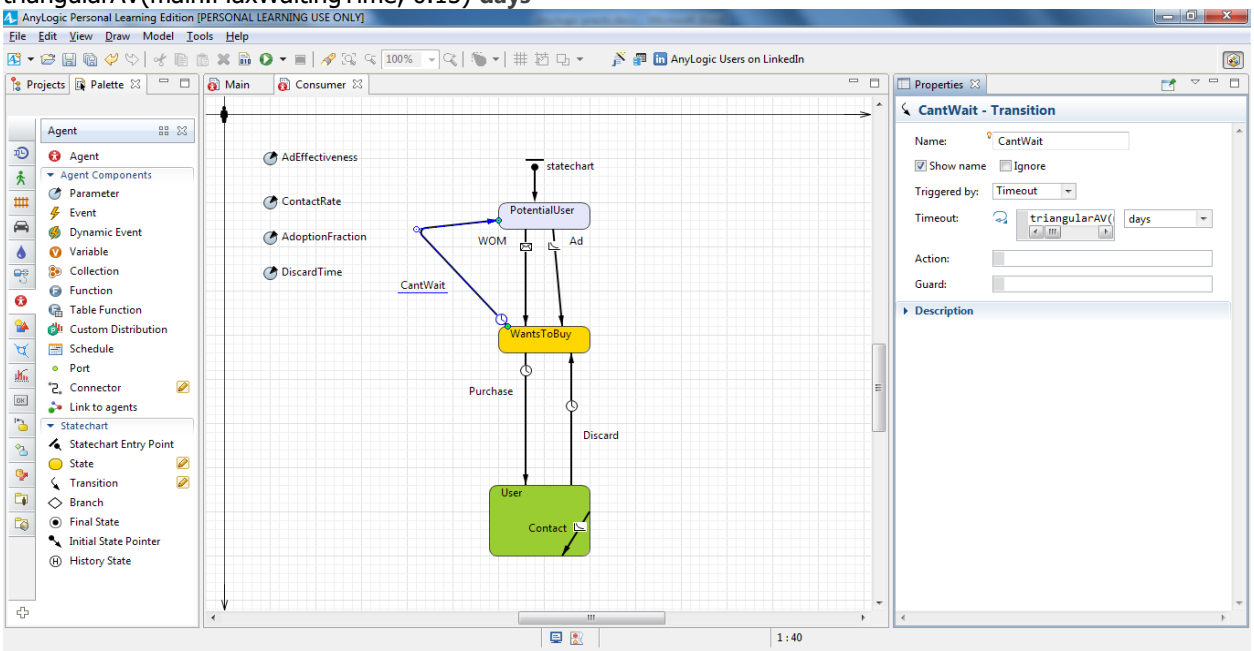



Draw the last transition CantWait that goes from WantsToBuy to PotentialUser state. This transition will model how a consumer's impatience causes them to change their purchase decision, and the Consumer diagram will look like this:



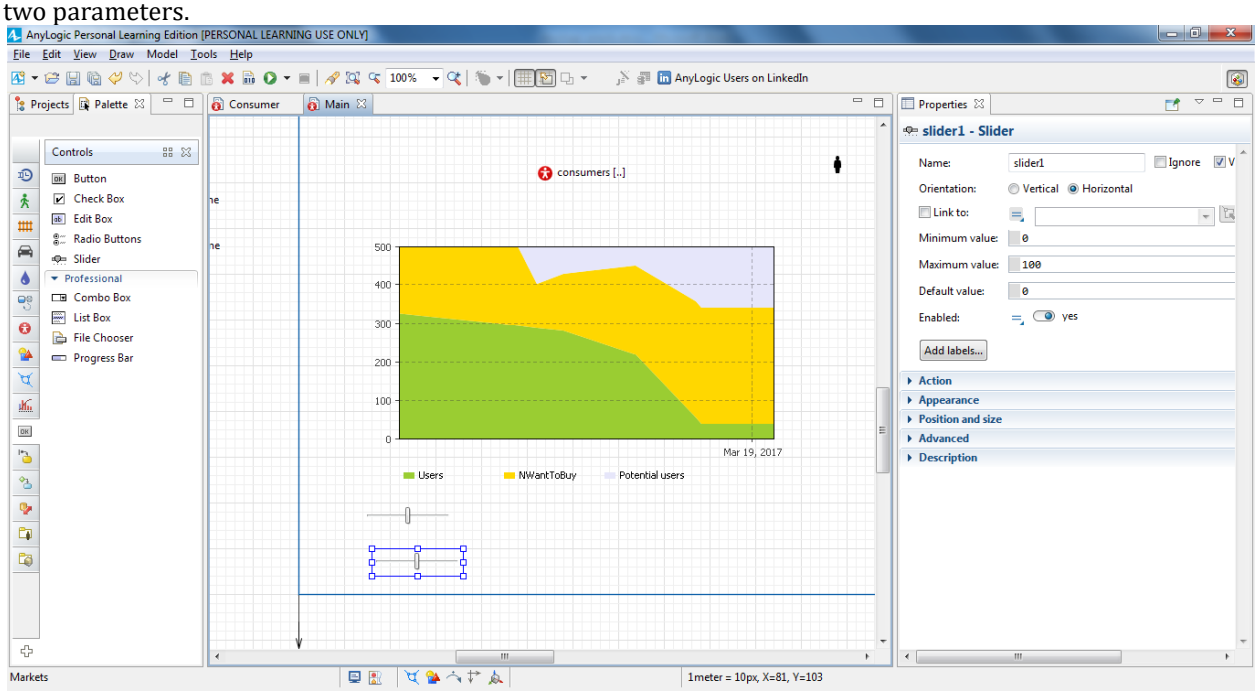

Modify the transition properties so it is triggered by **Timeout** which equals

triangularAV(main.MaxWaitingTime, 0.15) **days**

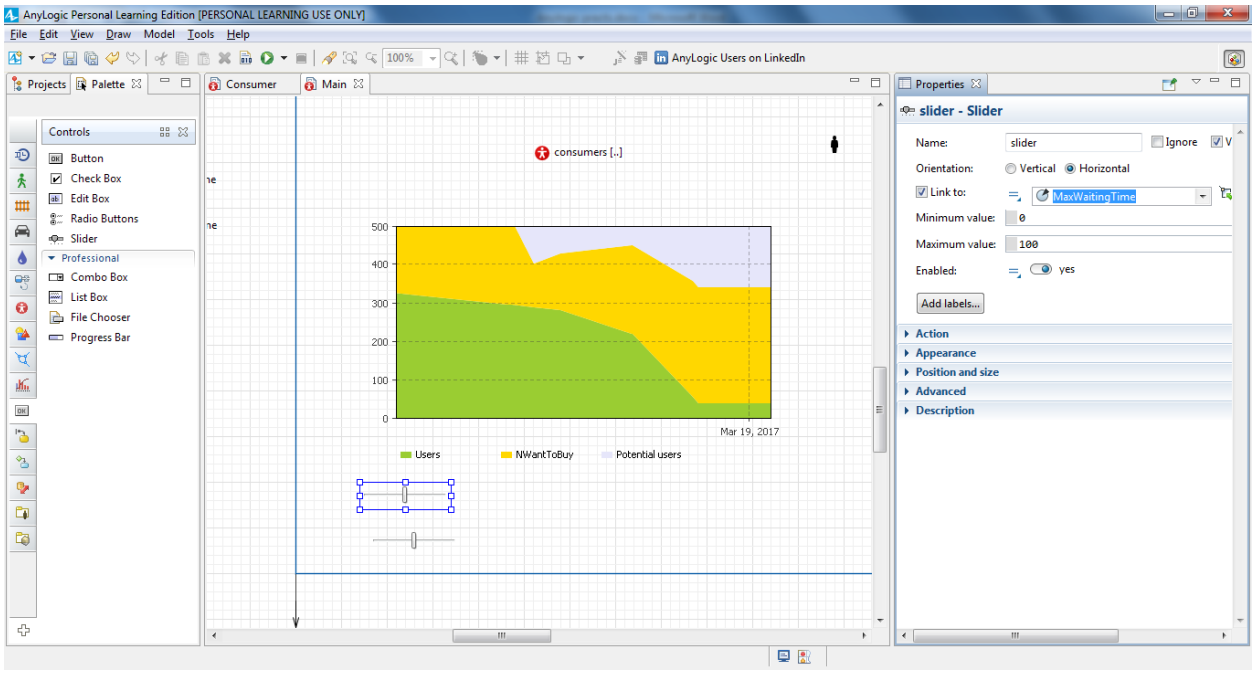



Rather than setting the maximum waiting time equal to constant MaxWaitingTime, we assume it follows a triangular distribution with an average of one week and a possible variation to up to 15 percent.

Go back to Main diagram. Open the **Controls** palette and drag two **Sliders** on to the diagram below the chart. We'll eventually link the sliders to our two parameters.

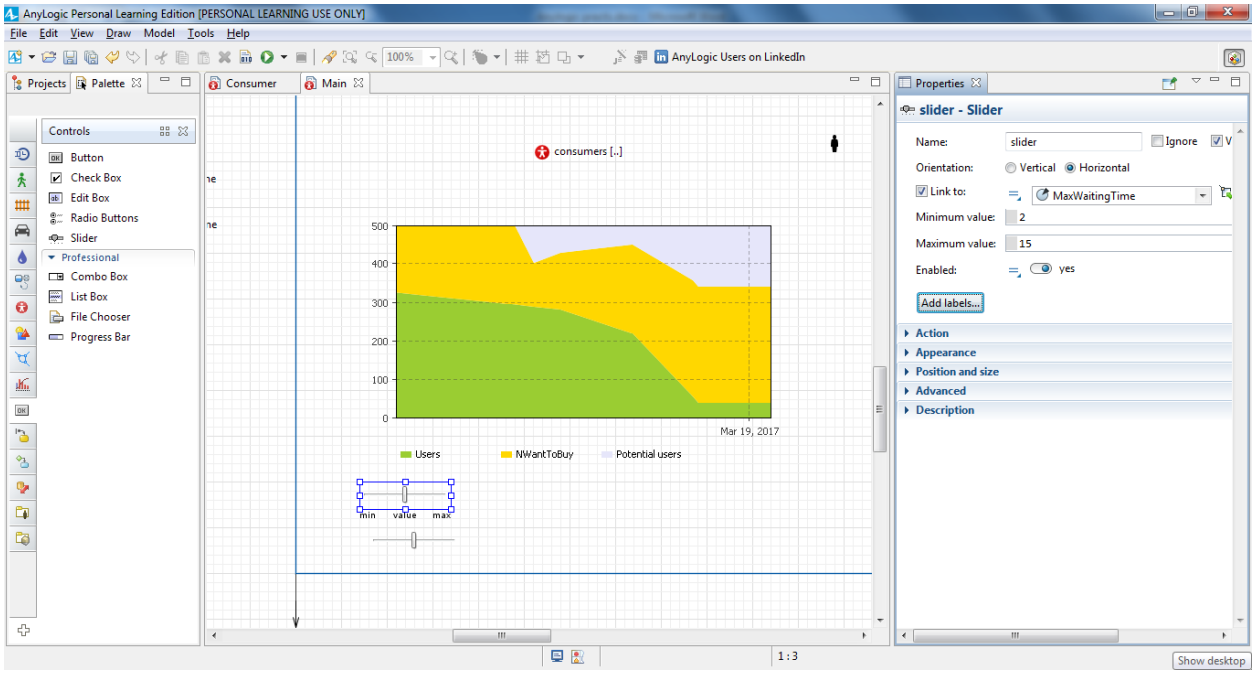



Select the checkbox **Link to** and select the parameter MaxWaitingTime to the right.

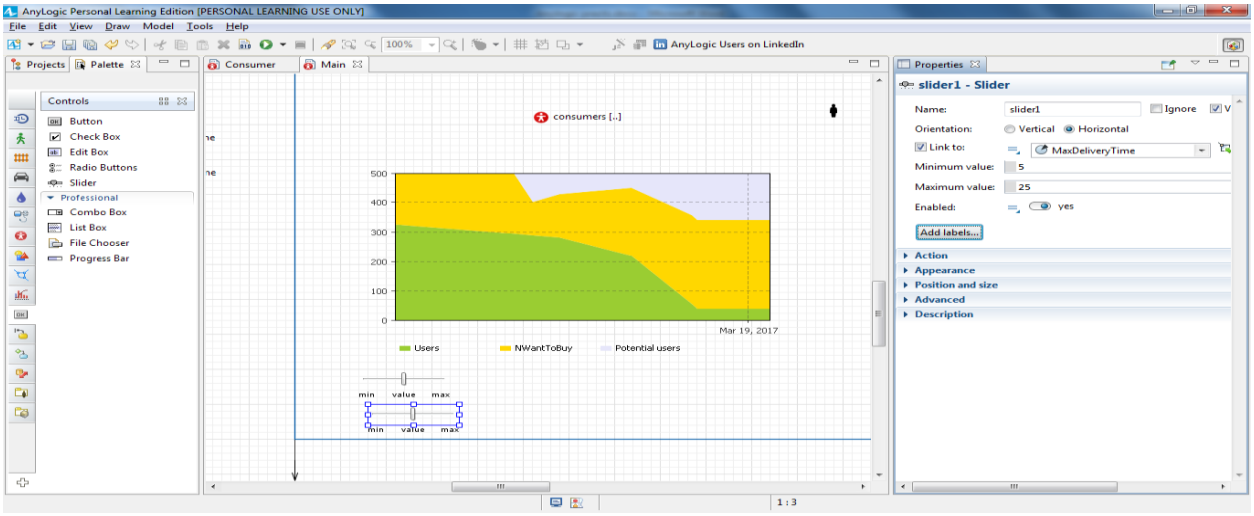



Set the slider's **Minimum** and **Maximum values**. The parameter value can vary within the range you define here, and we'll set 2 as **Minimum** and 15 as **Maximum value**.

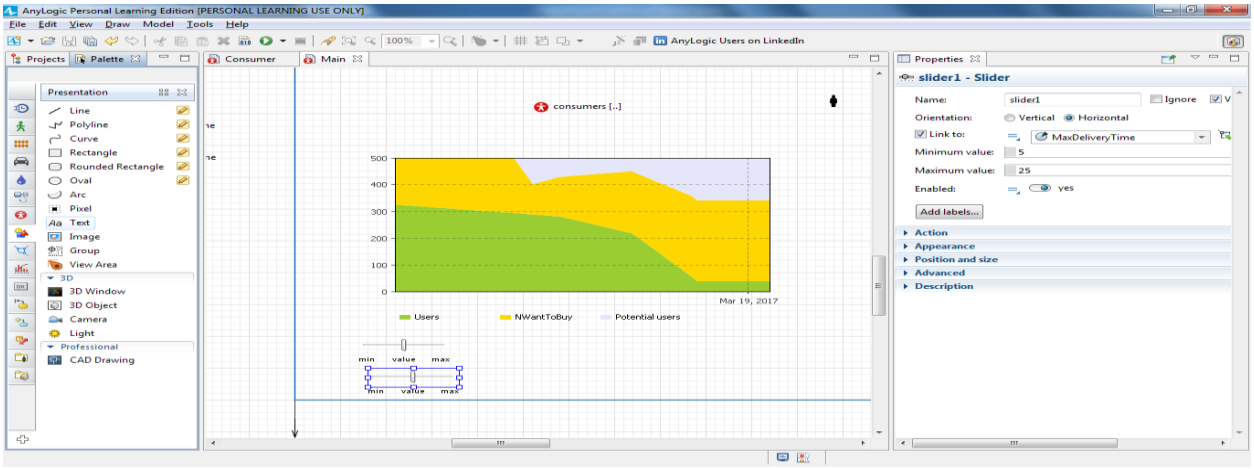
- Finally, click the **Add labels...** button to display the slider's minimum, maximum, and current values at runtime (the min, value, and max text shapes will appear beneath the slider).



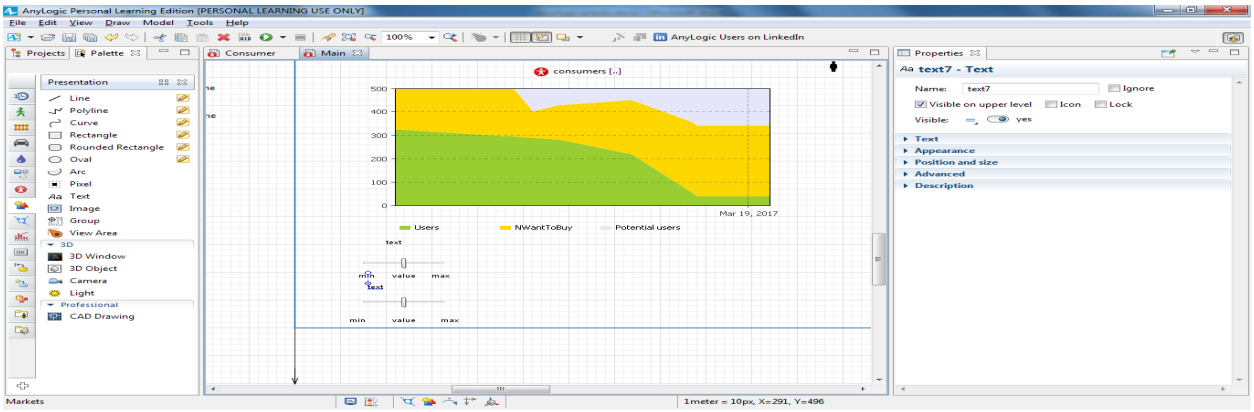

Select the checkbox **Link to** and select the parameter MaxDeliveryTime to the right.



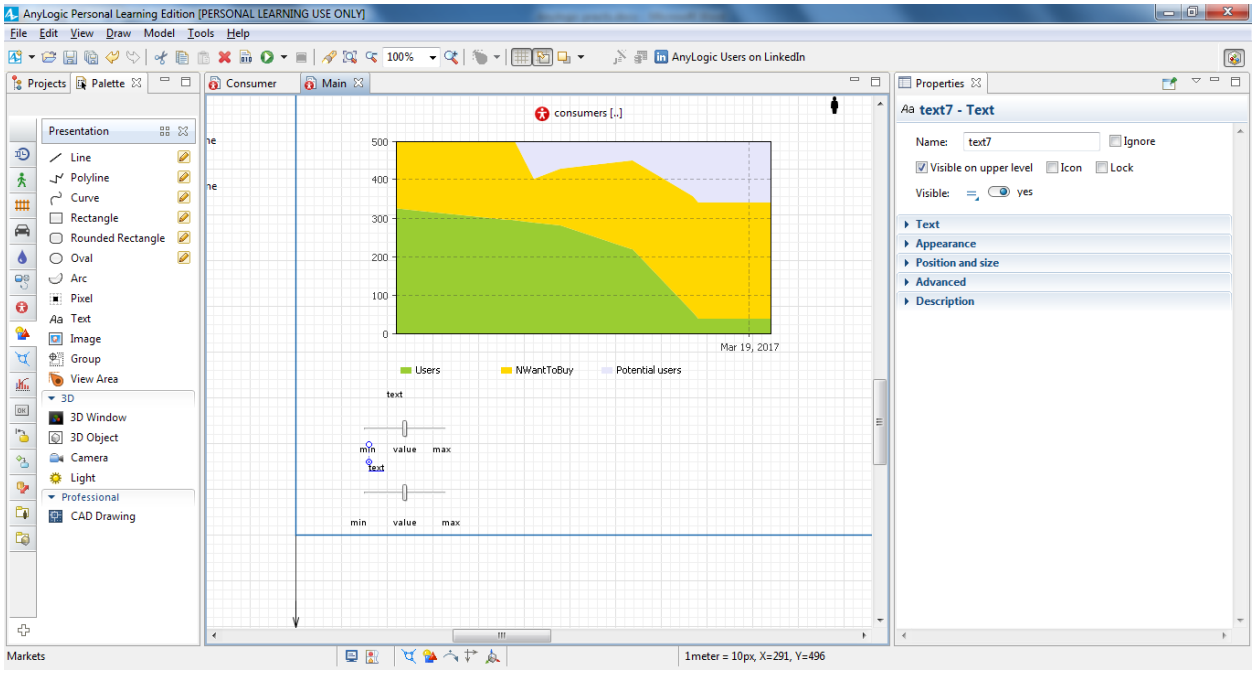

Open the **Presentation** palette, drag two **Text** shapes on to the diagram, and place them above the sliders. Let's configure the titles of these controls.

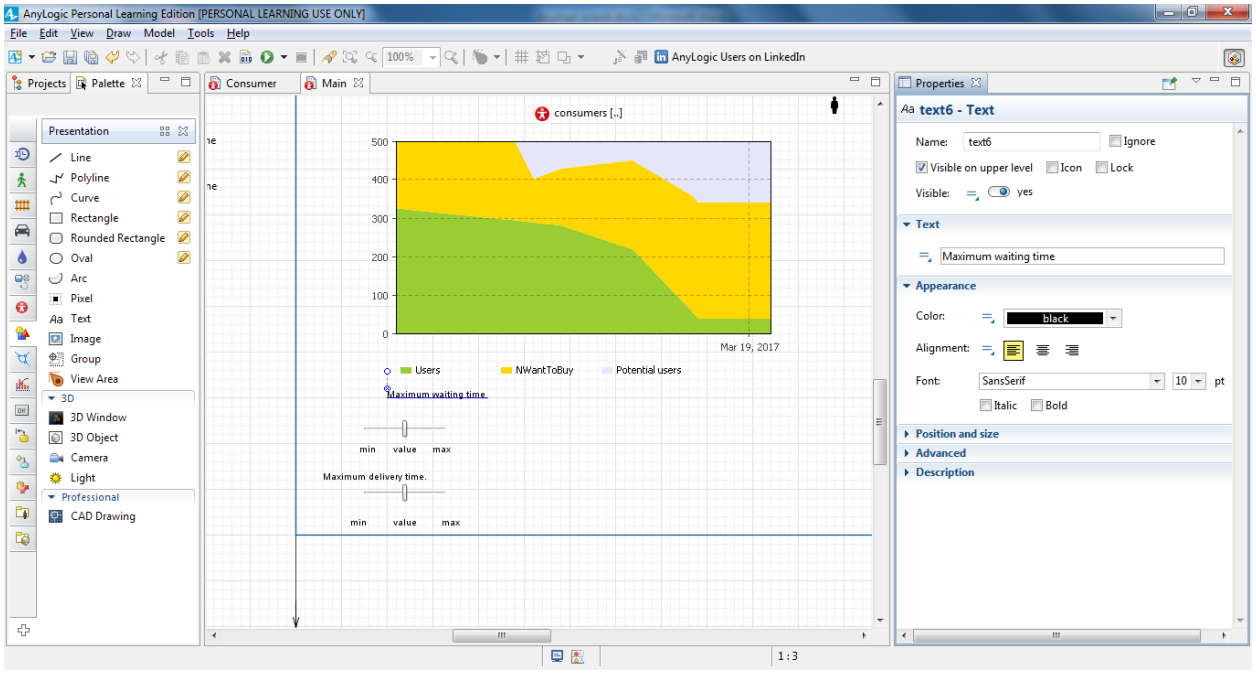



In the properties view, in the **Text** section, enter the text that the model will display. Using text shapes, name one slider Maximum waiting time and the other slider Maximum delivery time.

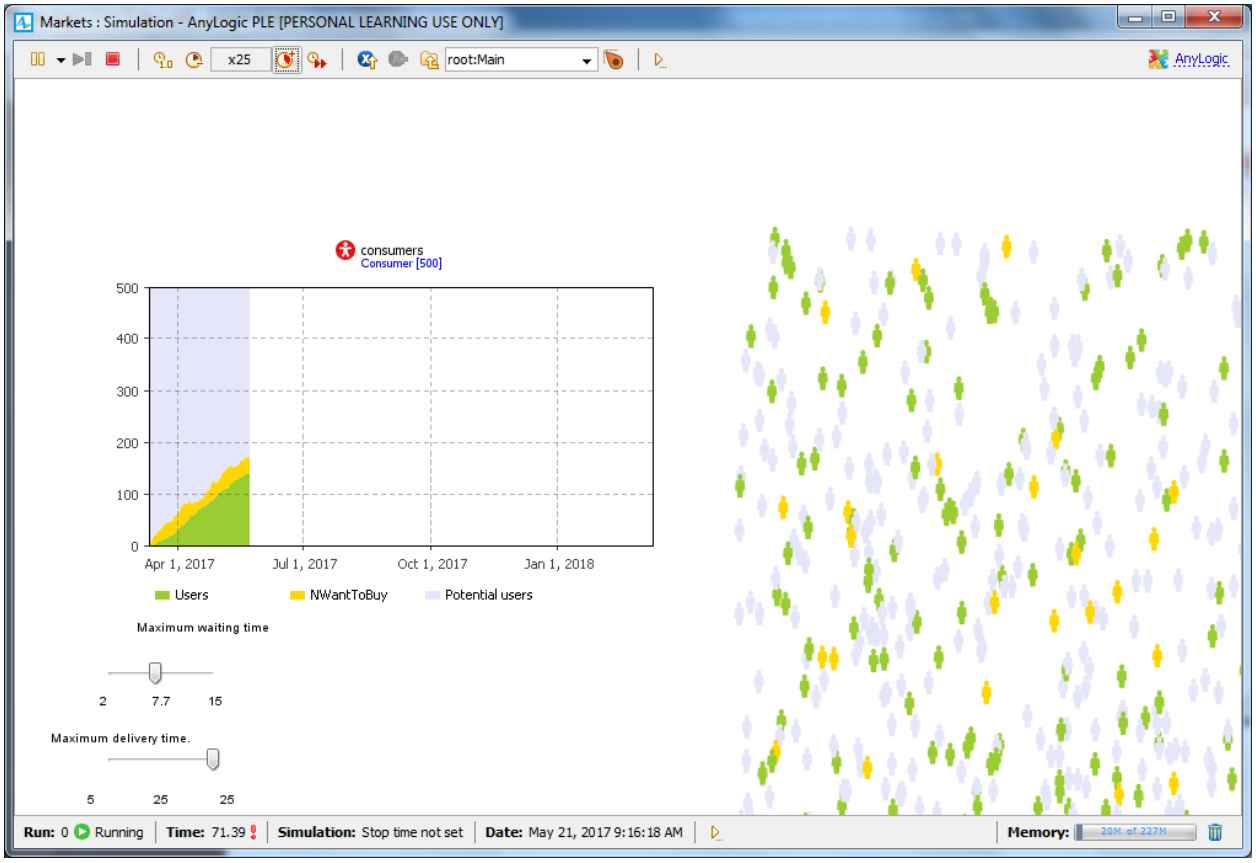


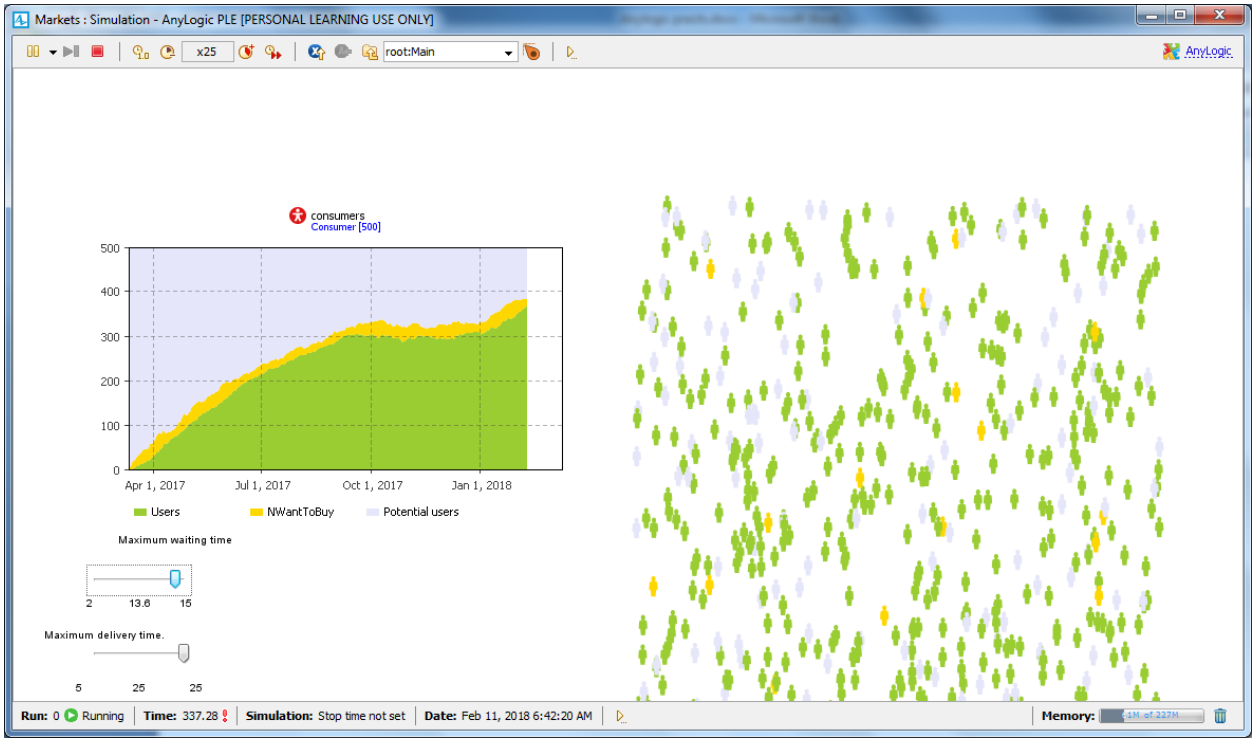
In the properties view, in the **Text** section, enter the text that the model will display. Using text shapes, name one slider Maximum waiting time and the other slider Maximum delivery time.



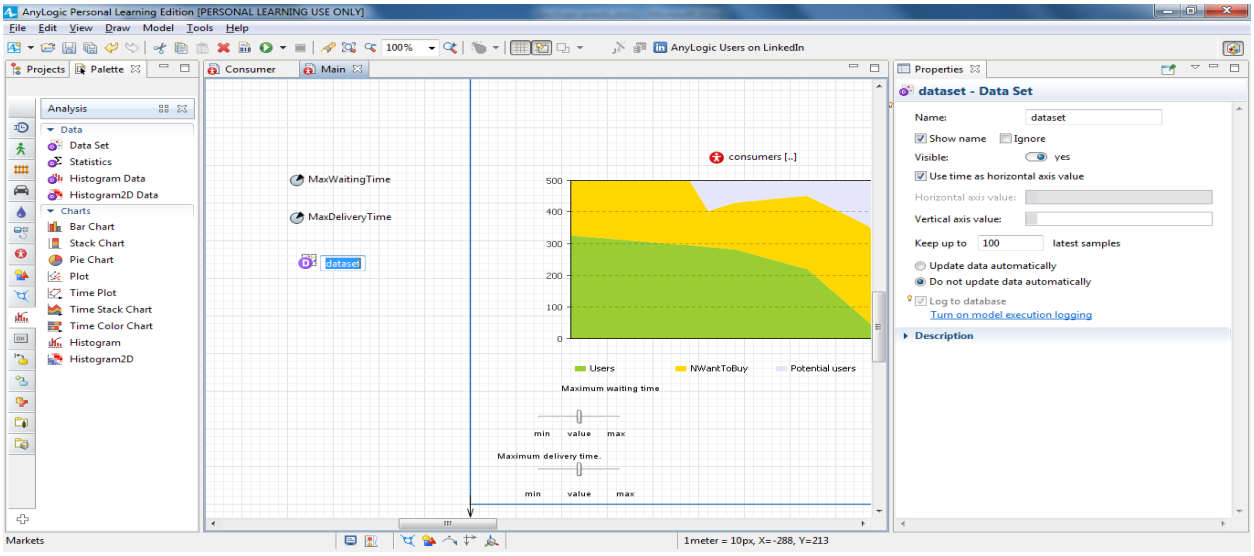

Run the model and observe the behavior. As you use the sliders to change the maximum waiting time or delivery time, you'll see your changes reflected in consumer behaviors and whole adoption dynamics.


Markets : Simulation - AnyLogic PLE [PERSONAL LEARNING USE ONLY]
x25
root:Main
AnyLogic
consumers
Consumer [500]
500
400
300
200
100
0
Apr 1, 2017
Jul 1, 2017
Oct 1, 2017
Jan 1, 2018
Users
NWantToBuy
Potential users
Maximum waiting time
2
7.7
15
Maximum delivery time.
5
25
25
Run: 0 Running
Time: 71.39
Simulation: Stop time not set
Date: May 21, 2017 9:16:18 AM
Memory:


Markets : Simulation - AnyLogic PLE [PERSONAL LEARNING USE ONLY]
x25
root:Main
AnyLogic
consumers
Consumer [500]
500
400
300
200
100
0
Apr 1, 2017
Jul 1, 2017
Oct 1, 2017
Jan 1, 2018
Users
NWantToBuy
Potential users
Maximum waiting time
2
13.6
15
Maximum delivery time.
5
25
25
Run: 0 Running
Time: 337.28
Simulation: Stop time not set
Date: Feb 11, 2018 6:42:20 AM
Memory:

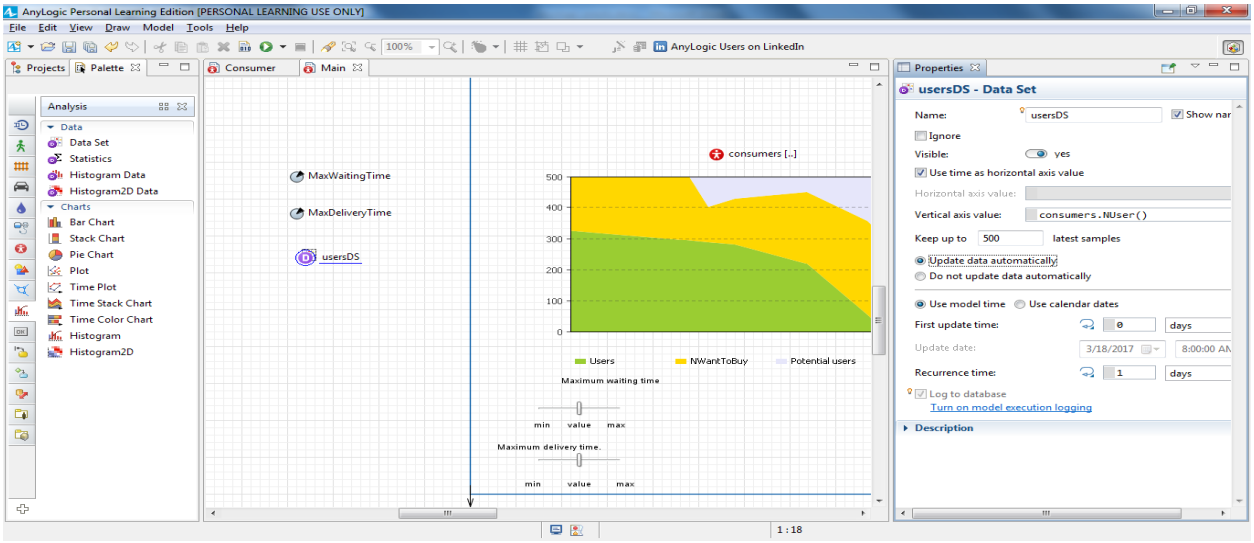
# Comparing model runs with different parameter values

Open the Main diagram and add a **Data Set** from the **Analysis** palette.
Name it usersDS

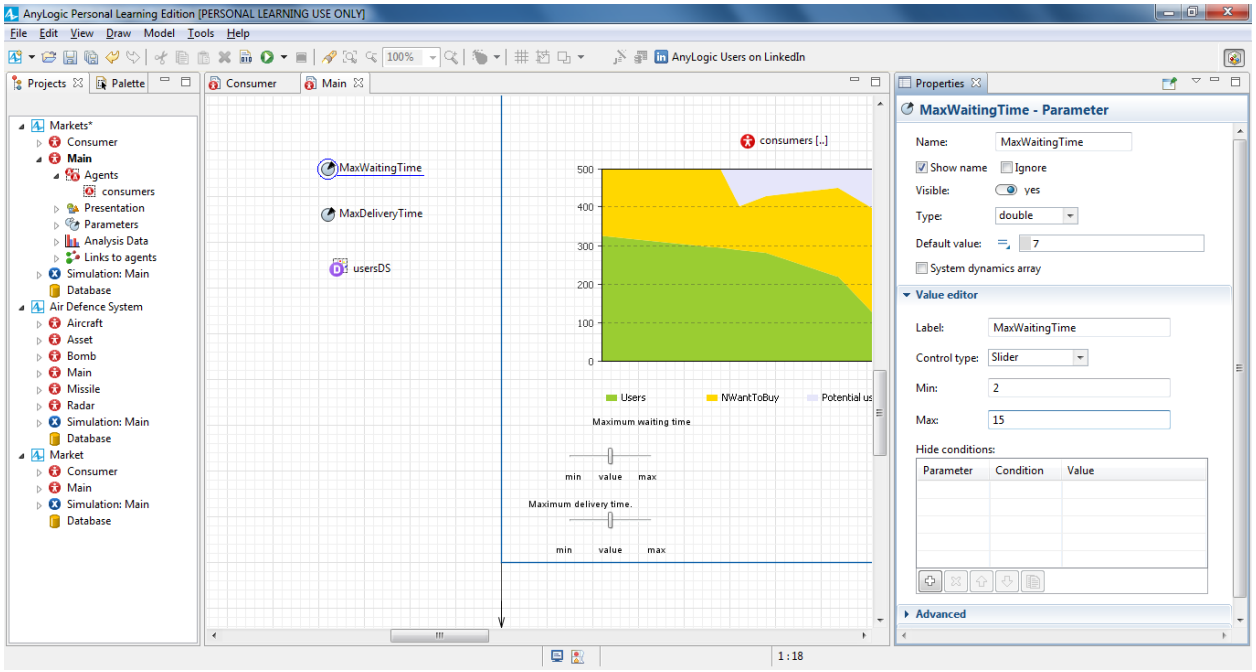



**Data Set** is capable of storing 2D (X,Y) data of type double. We want this dataset to store the history of product sales dynamics. We'll store data samples, each with a timestamp and the current number of the product users.

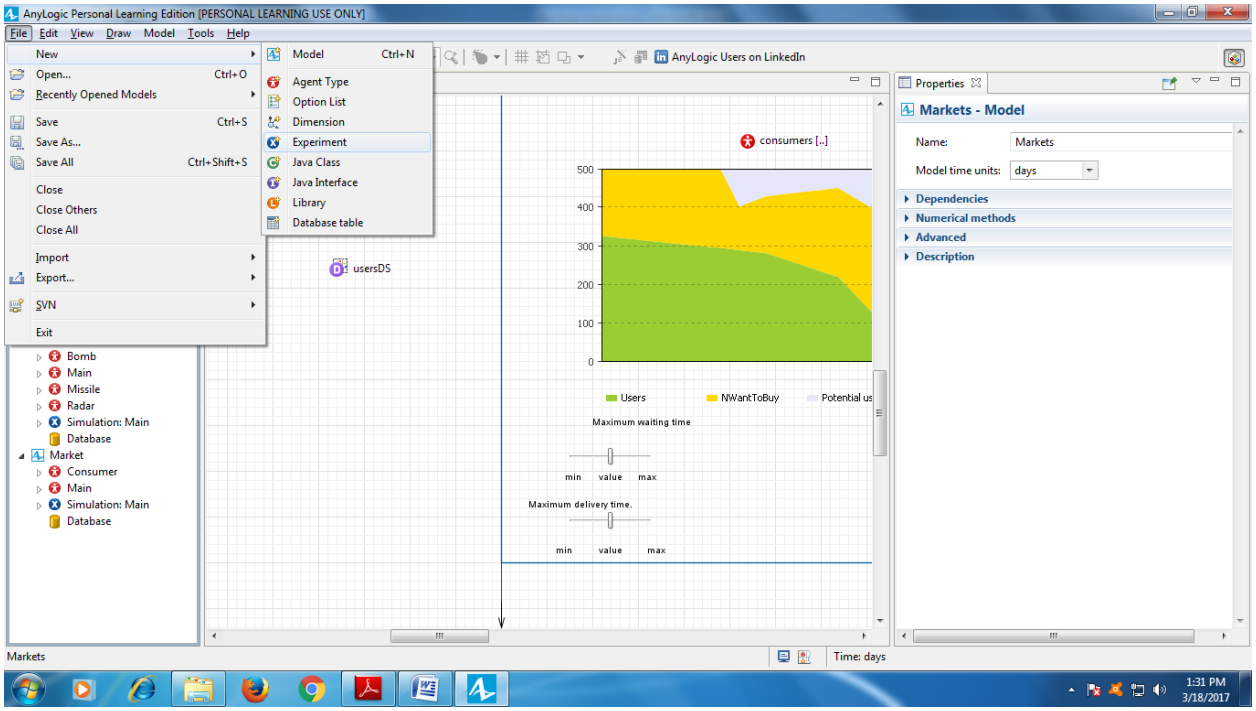
Set the value that the dataset will store. In the **Vertical axis value** property, type: consumers.NUser().
The dataset keeps a limited number of recent latest data items, and we'll limit our sample size to 500. Set the dataset to **Keep up to 500 latest samples**. Set it to **Update data automatically** with the default **Recurrence time**: 1. We'll add one data sample for one day of the model's lifetime.

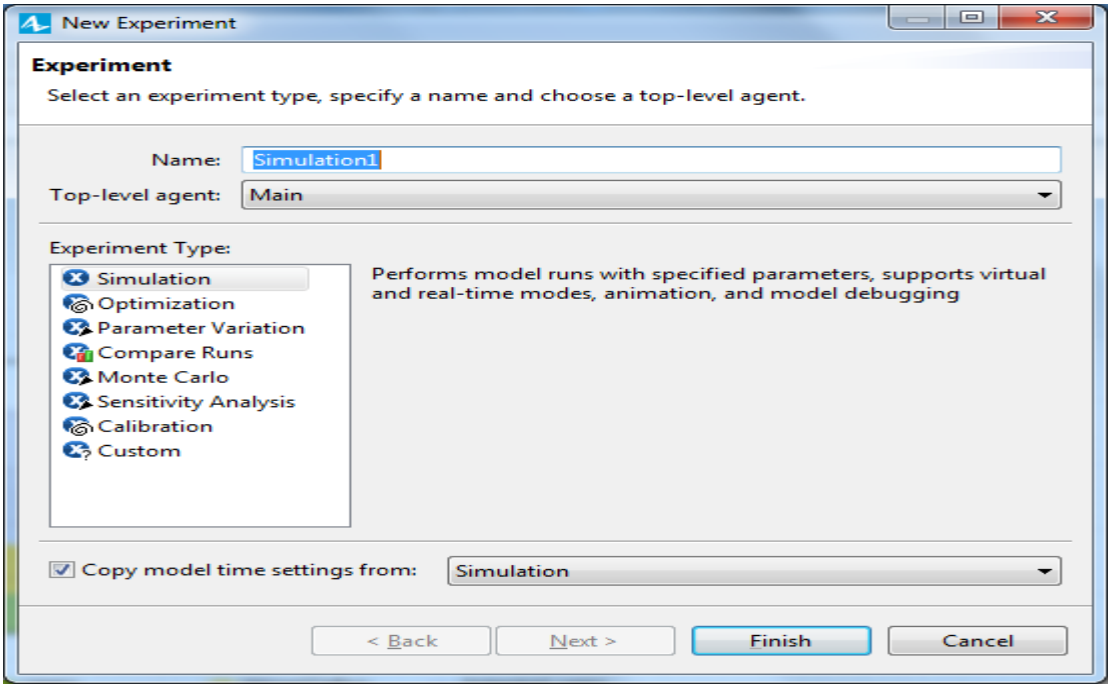


Next, make changes in the **Value editor** section for both parameters on Main diagram (MaxWaitingTime and MaxDeliveryTime). Choose Slider as **Control type**, set **Min** and **Max** values the same as we have in the sliders on Main, and if you want, change the default label (say, Maximum waiting time).

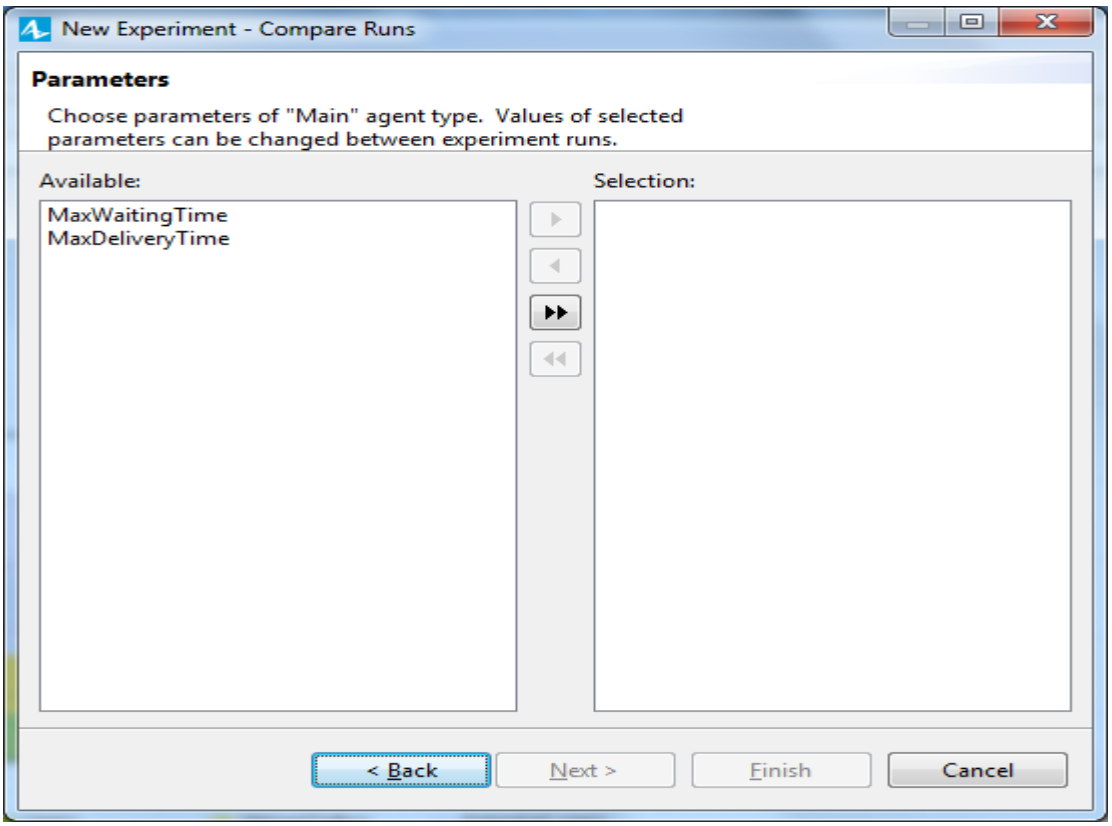



Open the **Projects** view, right-click the model item, and select **New > Experiment** from the context menu. The **New experiment** wizard will pop up.

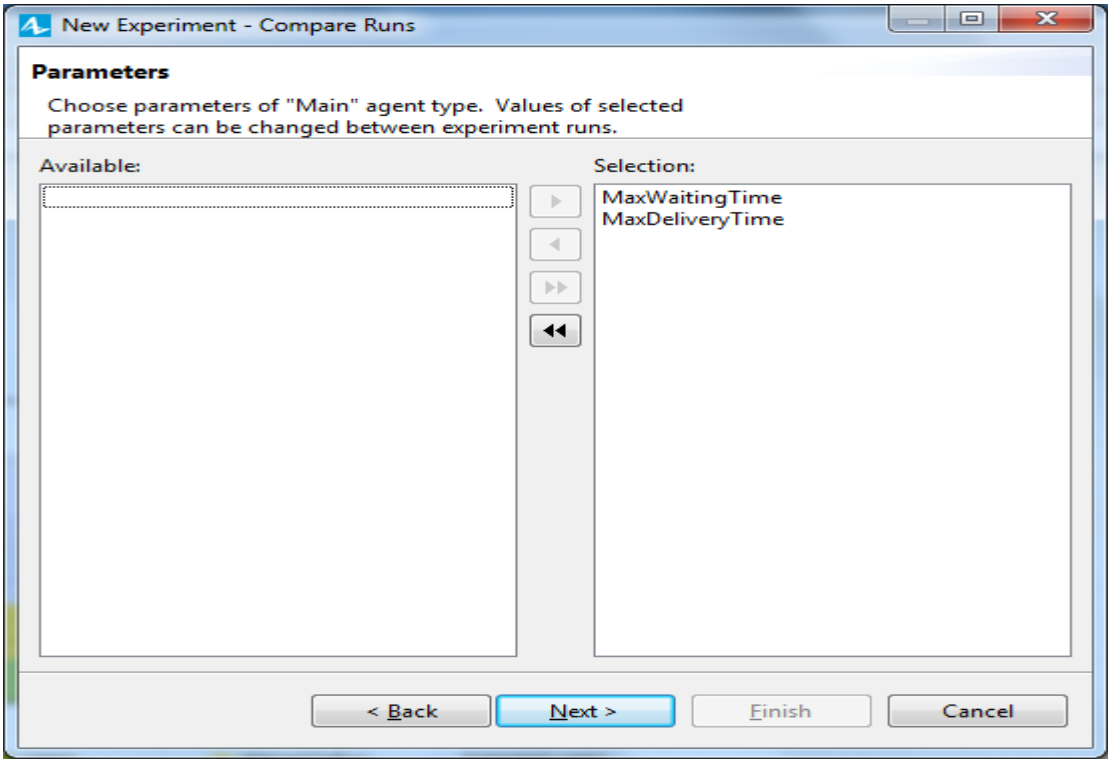


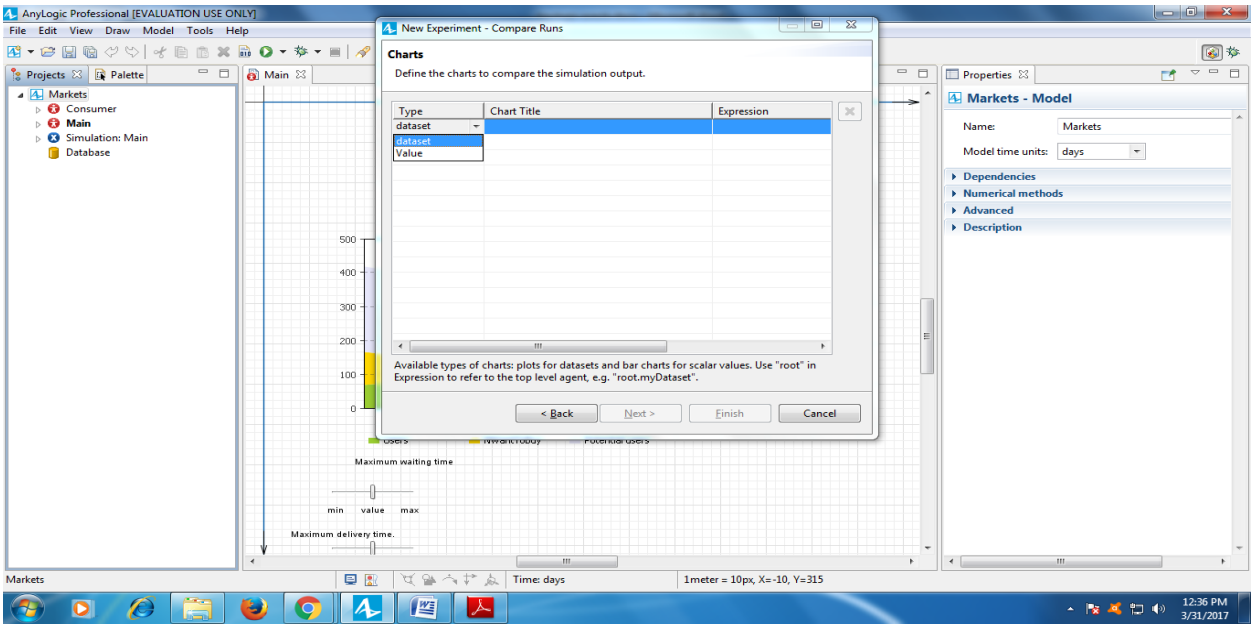



Select **Compare Runs** experiment from the list of experiment types andclick **Next**.

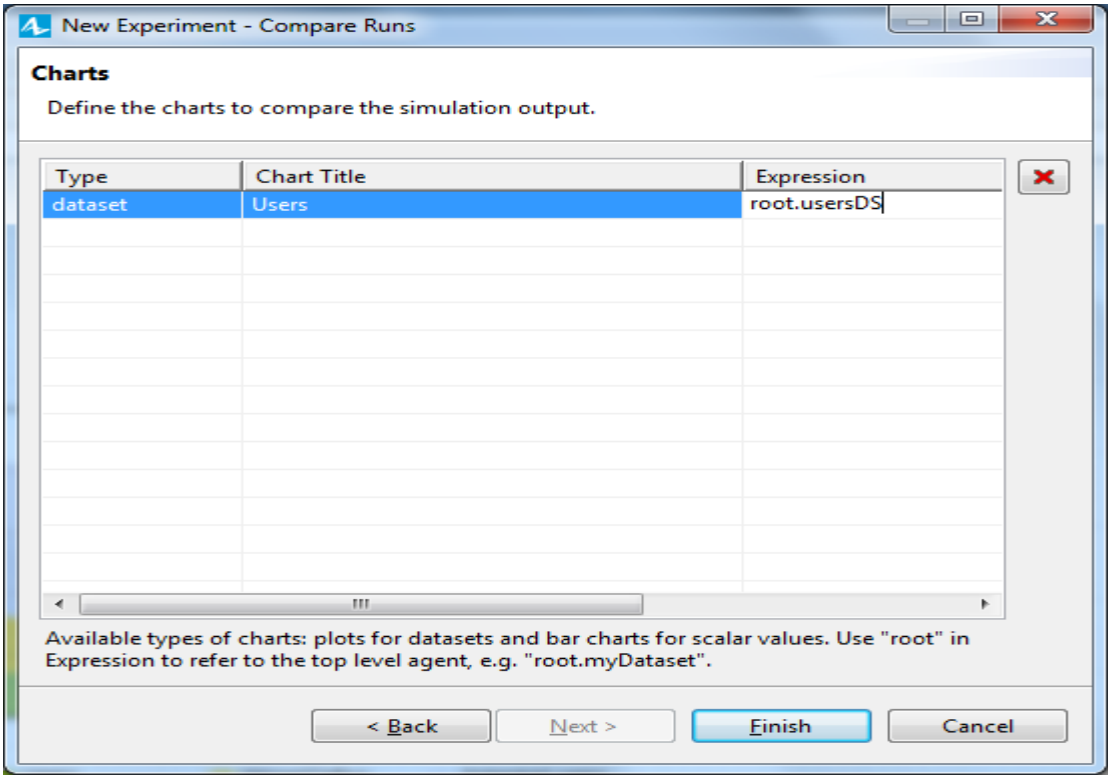



On the **Parameters** page, add both parameters to the **Selection** column. To add a parameter, select it in the **Available** list on the left and click the arrow.
You can also click the button to add all the parameters. Click **Next** after both parameters are in **Selection**.

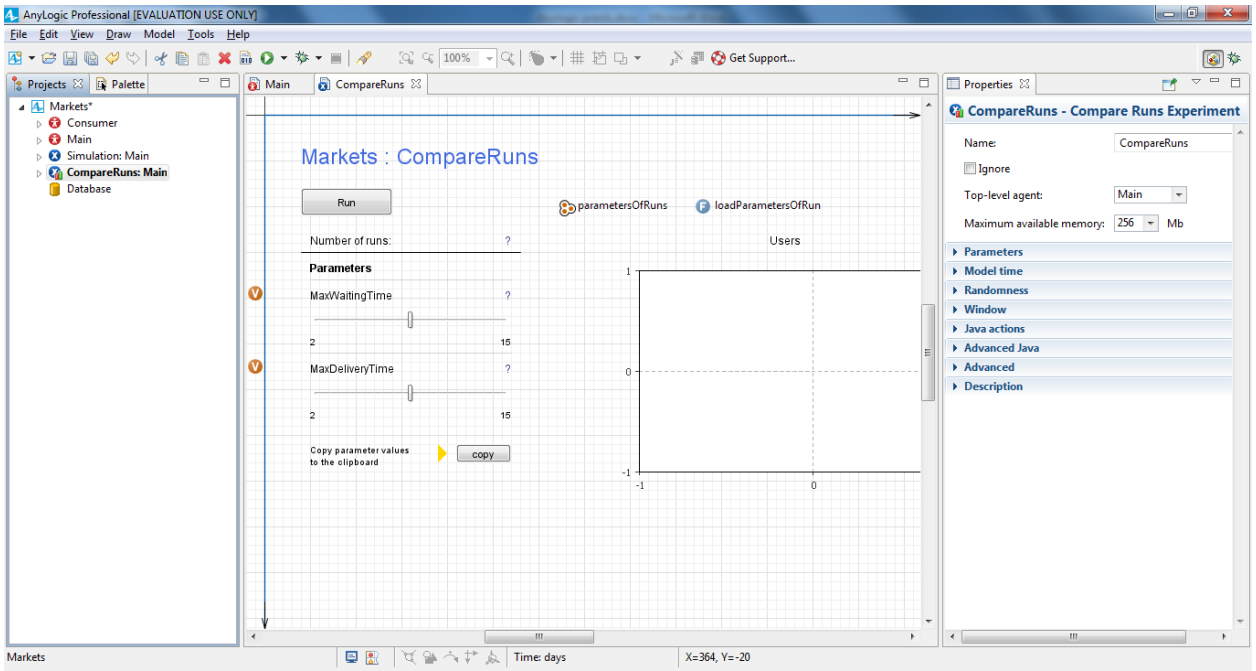


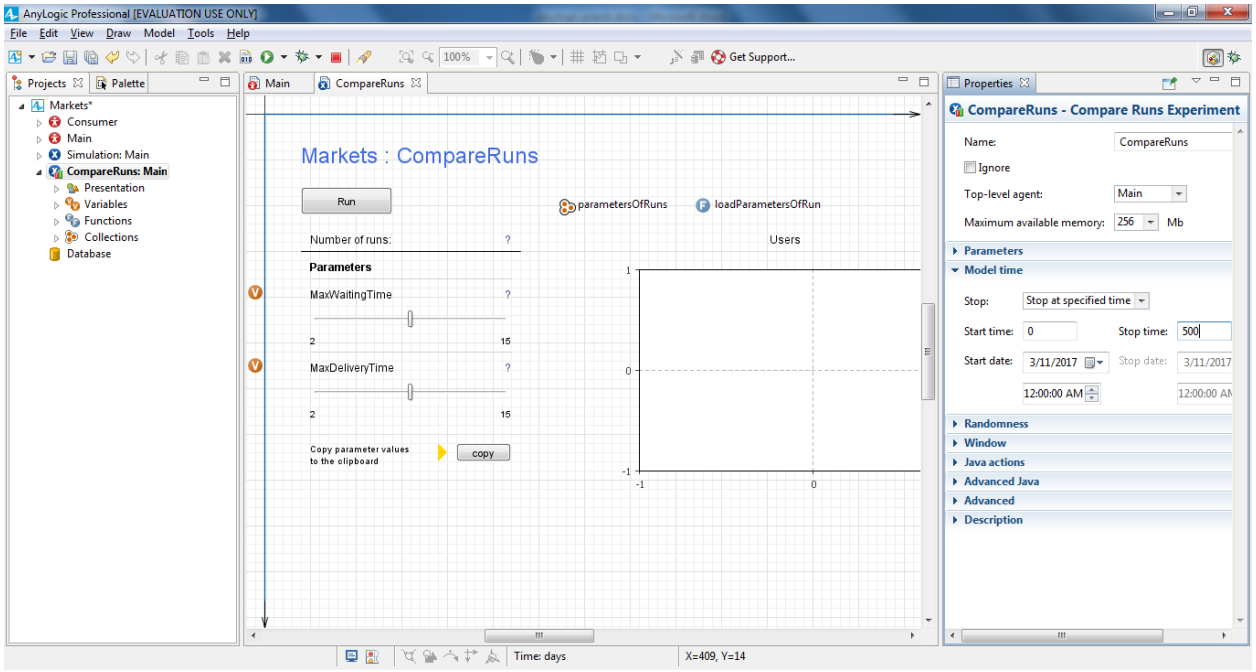

In the **Type** column, select **dataset**.
b. In the **Chart Title** column, type Users.
c. In the **Expression** column, refer to the dataset you defined in Main as root.usersDS where root is the model's top-level agent (Main)



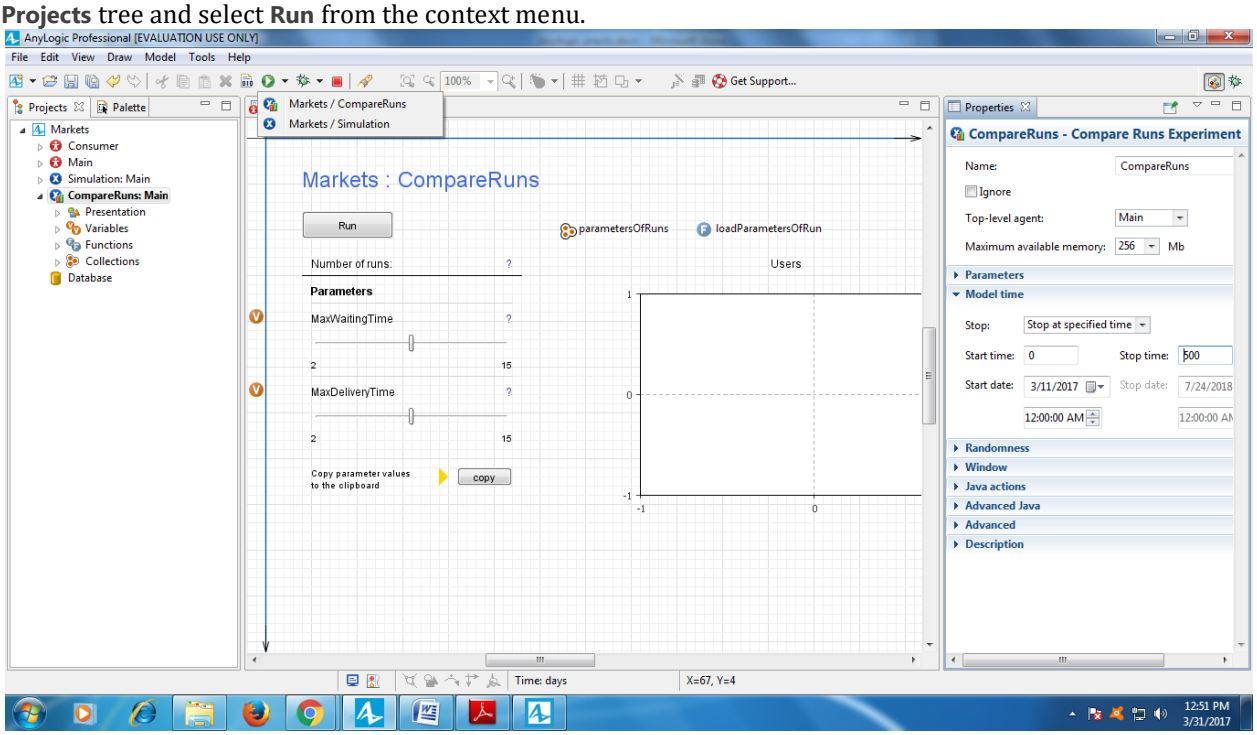

The chart will display the data collected by the dataset usersDS. Click **Finish**.
The CompareRuns experiment diagram should open automatically, and you'll see the default user interface we created with the wizard.



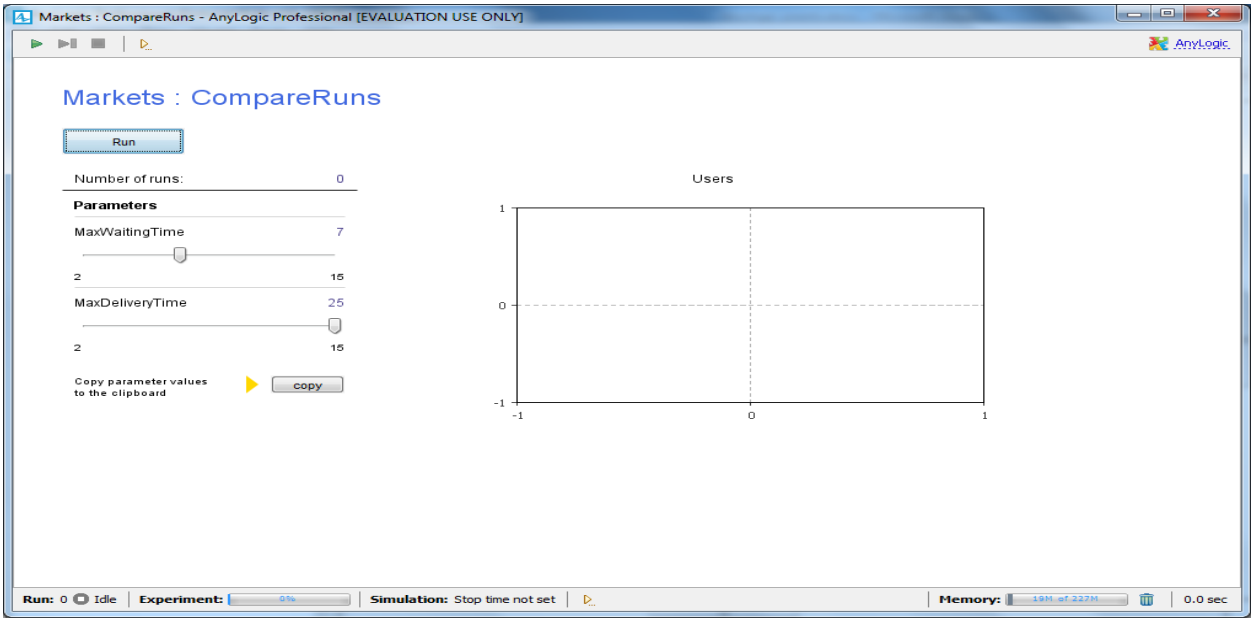

We want our experiment to simulate the model for just 500 days. To do this, select CompareRuns experiment in the **Projects** tree. In the experiment properties, open the **Model time** properties section, and type 500 in the **Stop time** field.

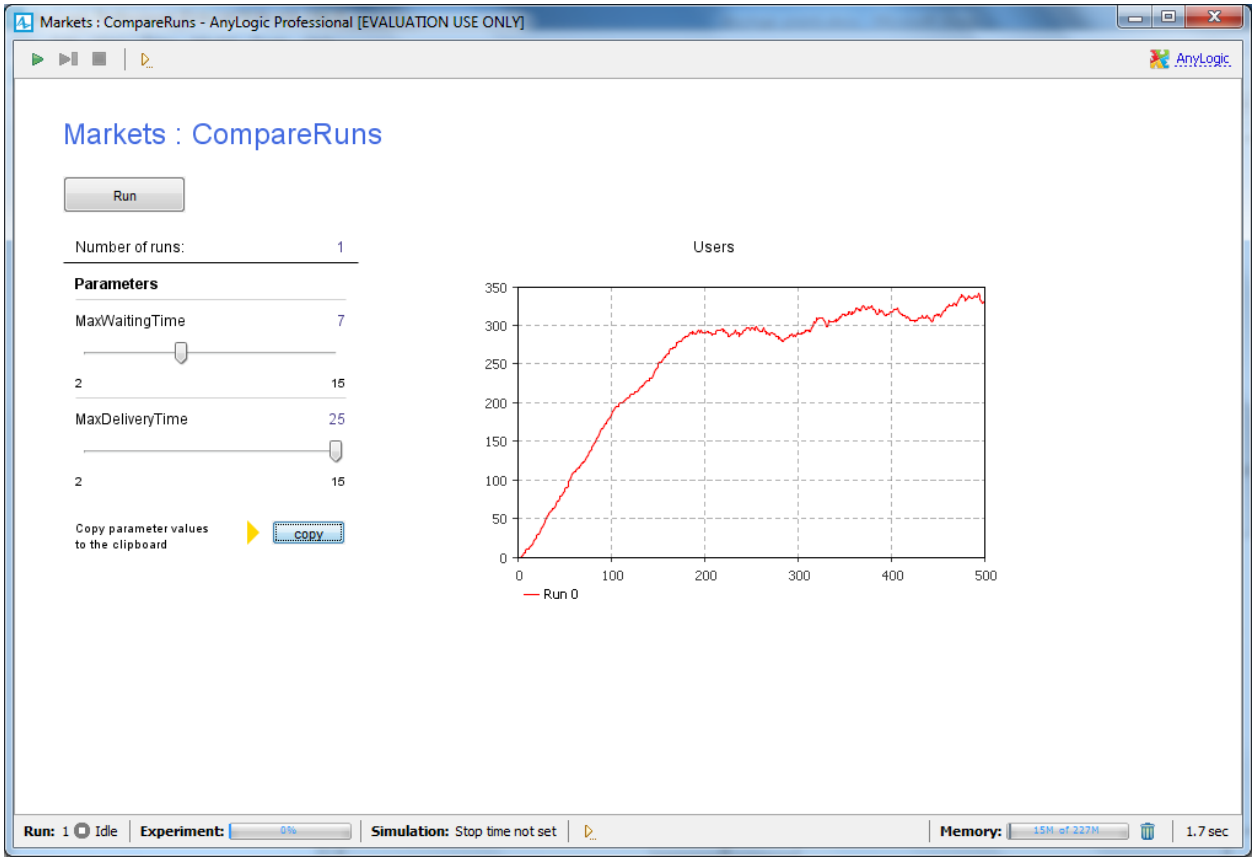


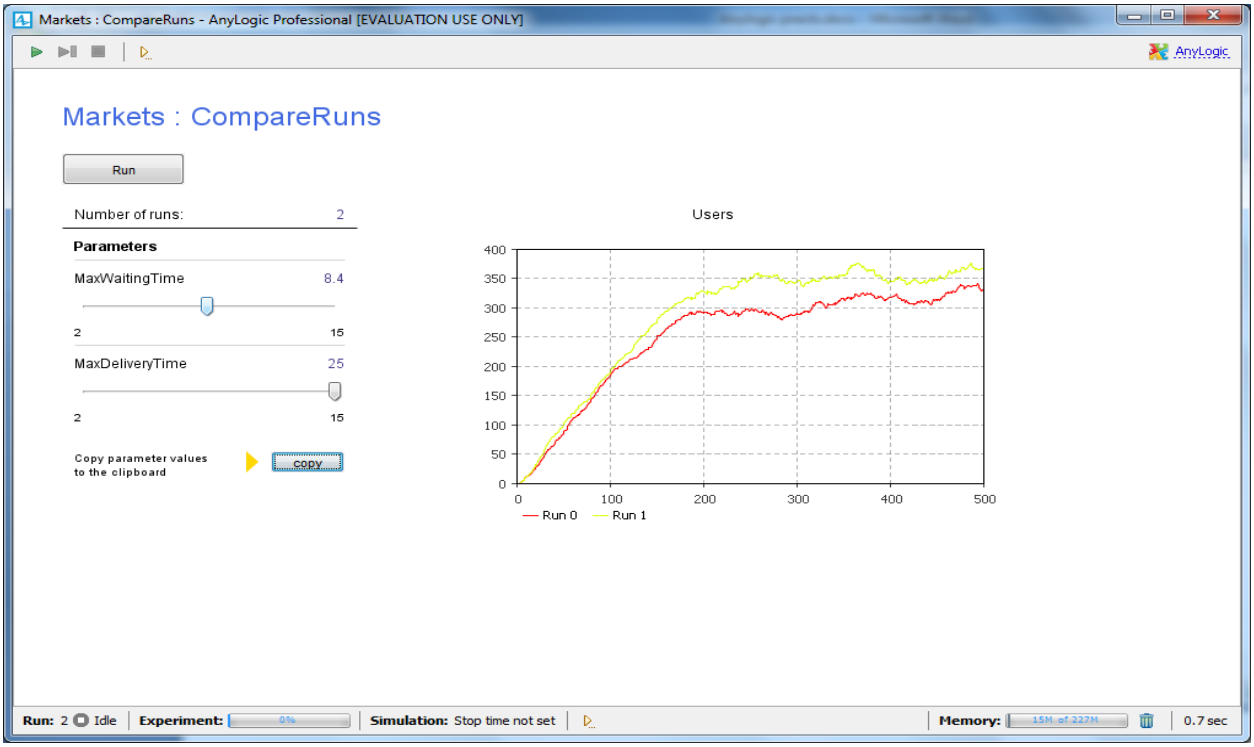

Run the experiment. Select the newly-created experiment from the **Run** list: **Market / CompareRuns**, or right-click the CompareRuns experiment in the **Projects** tree and select **Run** from the context menu.

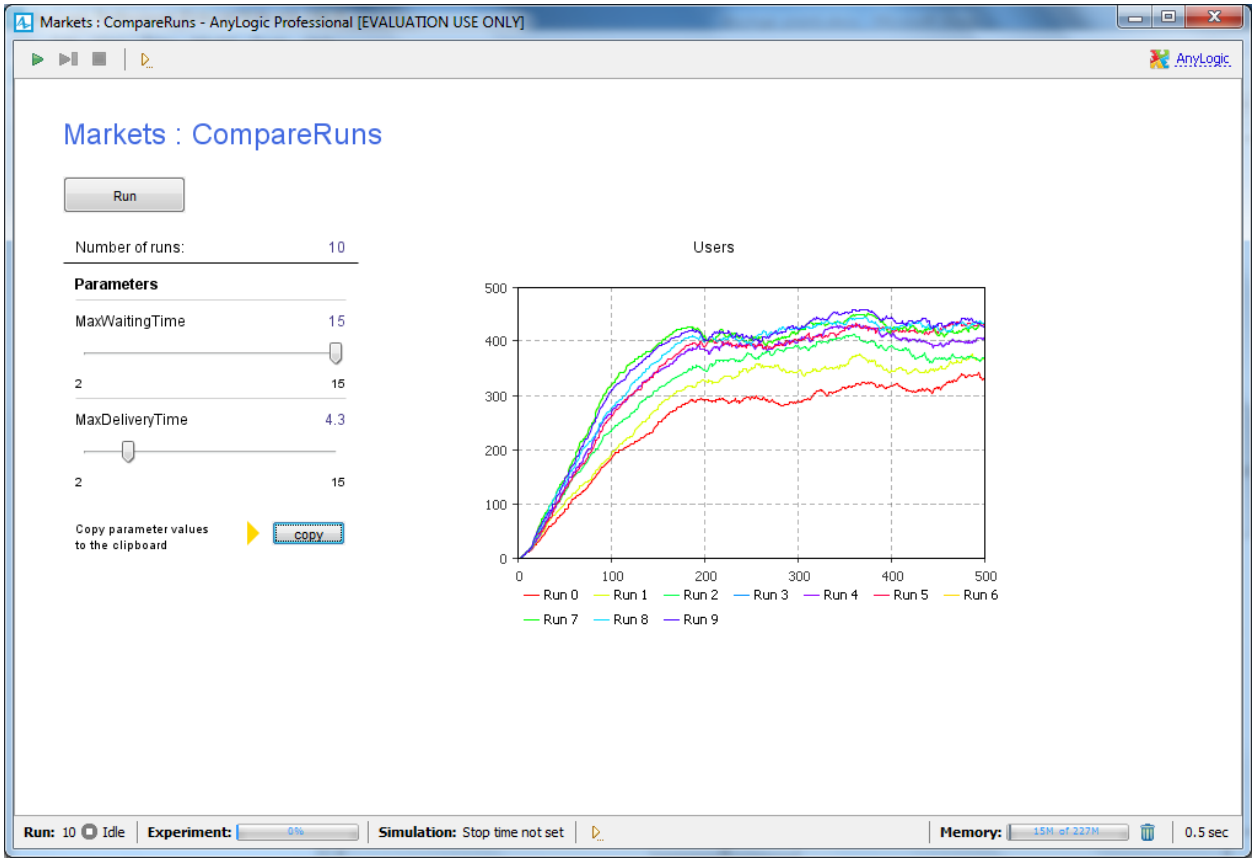



In the model window, click the **Run** button to see the result associated with the default parameter values. Afterward, change the parameter values and click **Run** again to observe the system behavior for the new settings. The chart displays all the results for your review.


Markets : CompareRuns - AnyLogic Professional [EVALUATION USE ONLY]
AnyLogic
Markets : CompareRuns
Run
Number of runs: 0
Parameters
MaxWaitingTime 7
2 15
MaxDeliveryTime 25
2 15
Copy parameter values to the clipboard
copy
Users
1
0
-1
-1
0
1
Run: 0 Idle
Experiment:
Simulation: Stop time not set
Memory:
0.0 sec


Markets : CompareRuns - AnyLogic Professional [EVALUATION USE ONLY]
AnyLogic
Markets : CompareRuns
Run
Number of runs: 1
Parameters
MaxWaitingTime 7
2 15
MaxDeliveryTime 25
2 15
Copy parameter values to the clipboard
copy
Users
350
300
250
200
150
100
50
0
0
100
200
300
400
500
Run 0
Run: 1 Idle
Experiment:
Simulation: Stop time not set
Memory:
1.7 sec

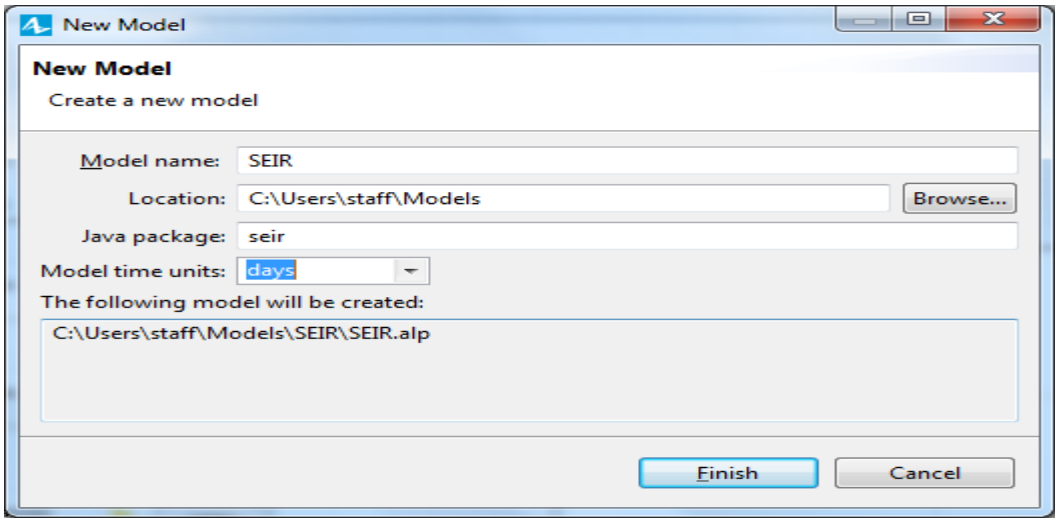

Markets : CompareRuns - AnyLogic Professional [EVALUATION USE ONLY]
AnyLogic
Markets : CompareRuns
Run
Number of runs: 2
Parameters
MaxWaitingTime 8.4
2 15
MaxDeliveryTime 25
2 15
Copy parameter values to the clipboard
copy
Users
400
350
300
250
200
150
100
50
0
0 100 200 300 400 500
Run 0 Run 1
Run: 2 Idle
Experiment: 0%
Simulation: Stop time not set
Memory: 15M of 227M
0.7 sec


Markets : CompareRuns - AnyLogic Professional [EVALUATION USE ONLY]
AnyLogic
Markets : CompareRuns
Run
Number of runs: 10
Parameters
MaxWaitingTime 15
2 15
MaxDeliveryTime 4.3
2 15
Copy parameter values to the clipboard
copy
Users
500
400
300
200
100
0
0 100 200 300 400 500
Run 0 Run 1 Run 2 Run 3 Run 4 Run 5 Run 6
Run 7 Run 8 Run 9
Run: 10 Idle
Experiment: 0%
Simulation: Stop time not set
Memory: 15M of 227M
0.5 sec

Practical No.4

Design and develop System Dynamic model by

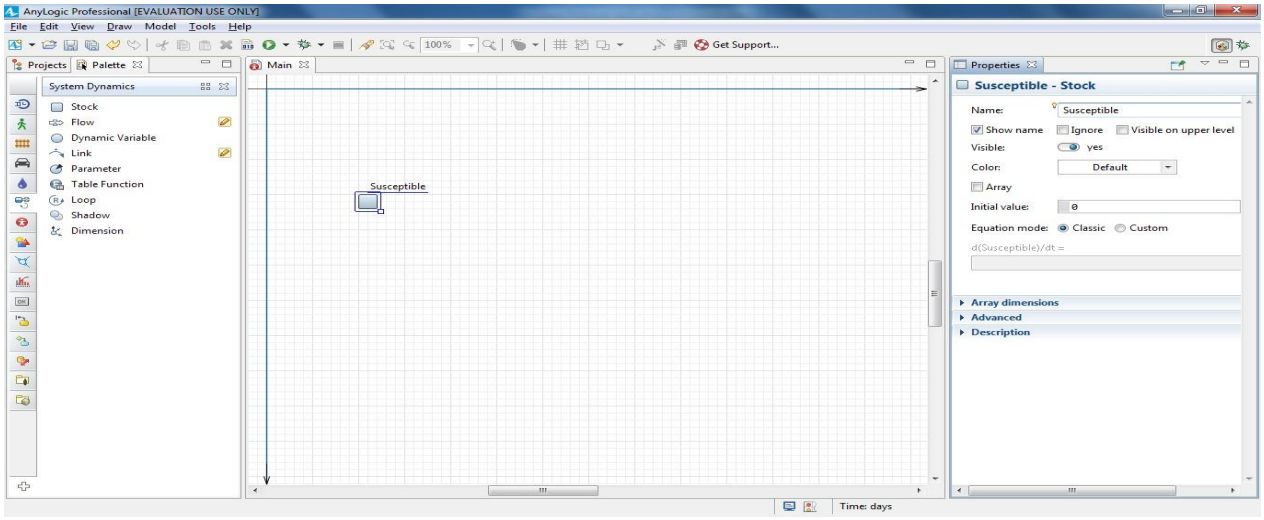
Creating a stock and flow diagram Adding a plot to visualize dynamics Parameter Variation Calibration
[ Use a case scenario like spread of contagious disease for the purpose]
.
SEIR model
We're about to build a model that displays the spread of a contagious disease among a large population. Our sample model will have a population of 10,000 people – a value we call TotalPopulation – of which one person is infectious.
During the infectious phase, a person comes into contact with an average of ContactRateInfectious = 1.25 people each day. If an infectious person comes into contact with a susceptible person, the susceptible person's probability of infection is Infectivity = 0.6.
After a susceptible person is infected, the infection latent phase lasts for AverageIncubationTime = 10 days. We'll use the word exposed to describe people who are in the latent phase.
After the latent phase, infectious phase starts. This phase lasts for AverageIllnessDuration = 15 days.
Persons who have recovered from the disease are immune to a second infection.
Phase 1. Creating a stock and flow diagram
Create a new model by selecting File > New > Model from the menu, and then name it SEIR.
Select days as the Model time units.



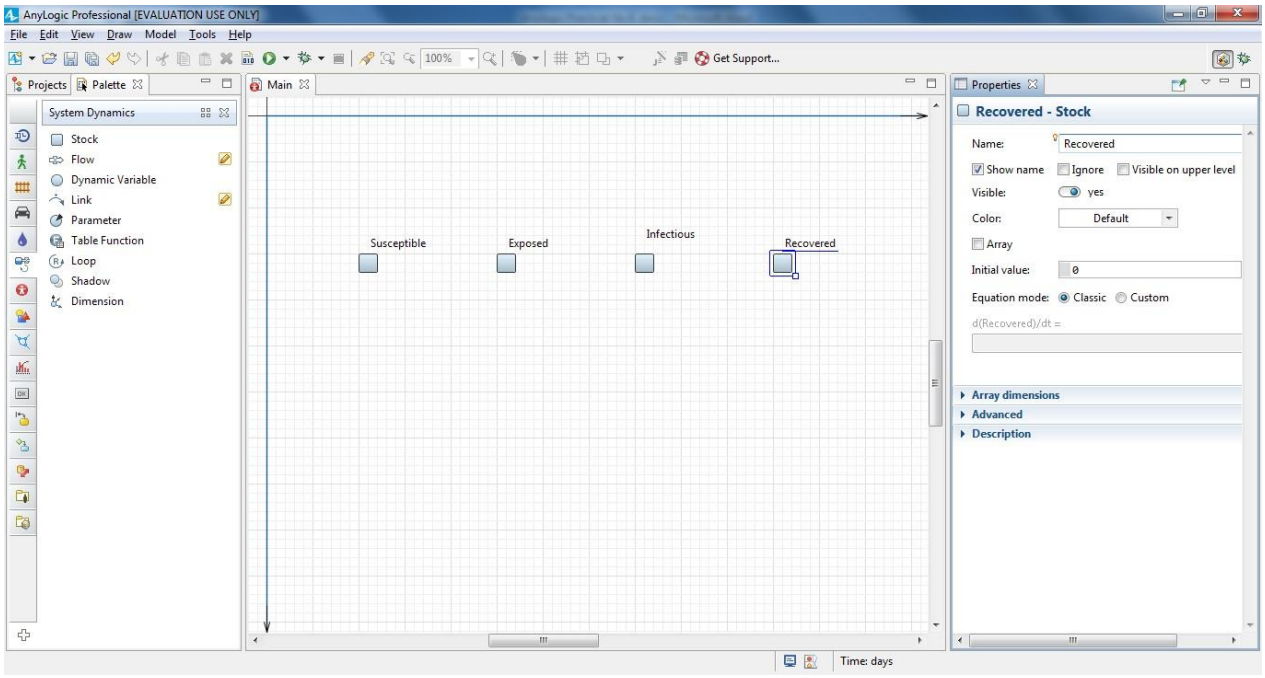

In this example, we'll
consider four important characteristics:
Susceptible - people who are not infected by the virus
Exposed - people who are infected but who can't infect others
Infectious - people who are infected and who can infect others
Recovered – people who have recovered from the virus
SEIR is an acronym that represents the four stages: Susceptible-Exposed- Infectious-Recovered.
The terminology and the overall structure of the problem is taken from the ("Compartmental models in epidemiology". n.d.) -- namely, from the SEIR (Susceptible Exposed Infectious Recovered) model.

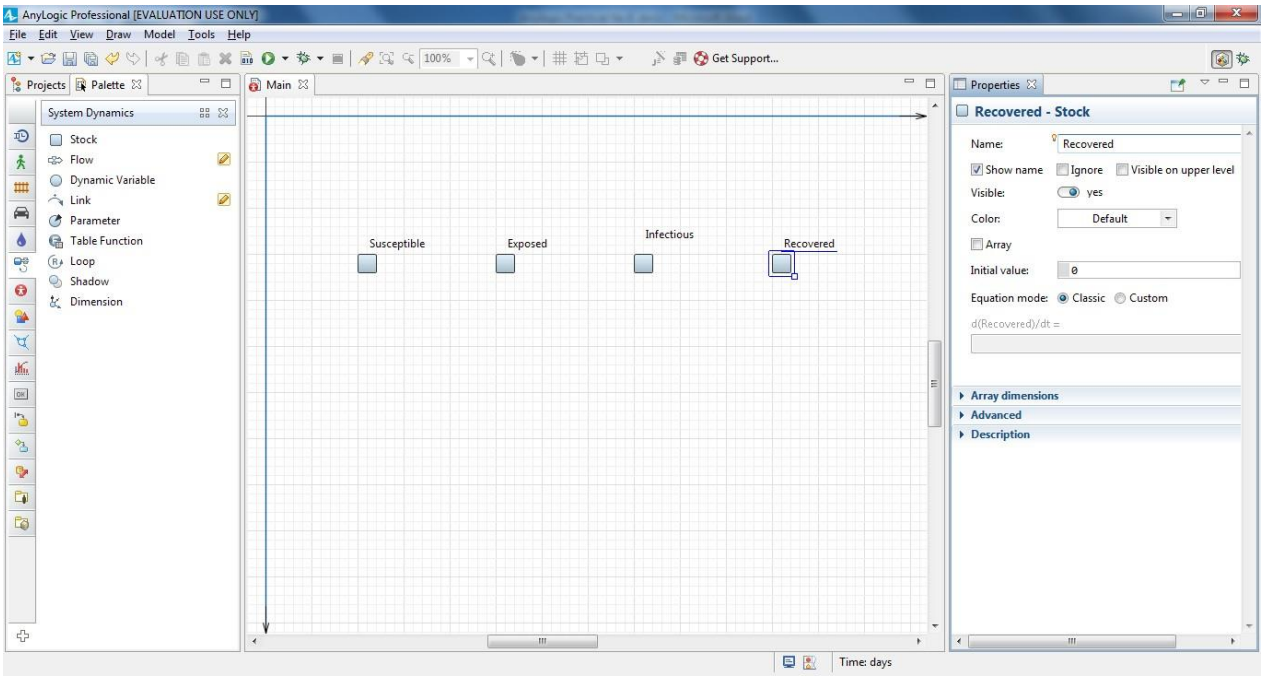
There are four stocks in our model - one for each stage.12.222

Open the System Dynamics palette. Drag the Stock from the System Dynamics palette on to the diagram. Name it Susceptible.

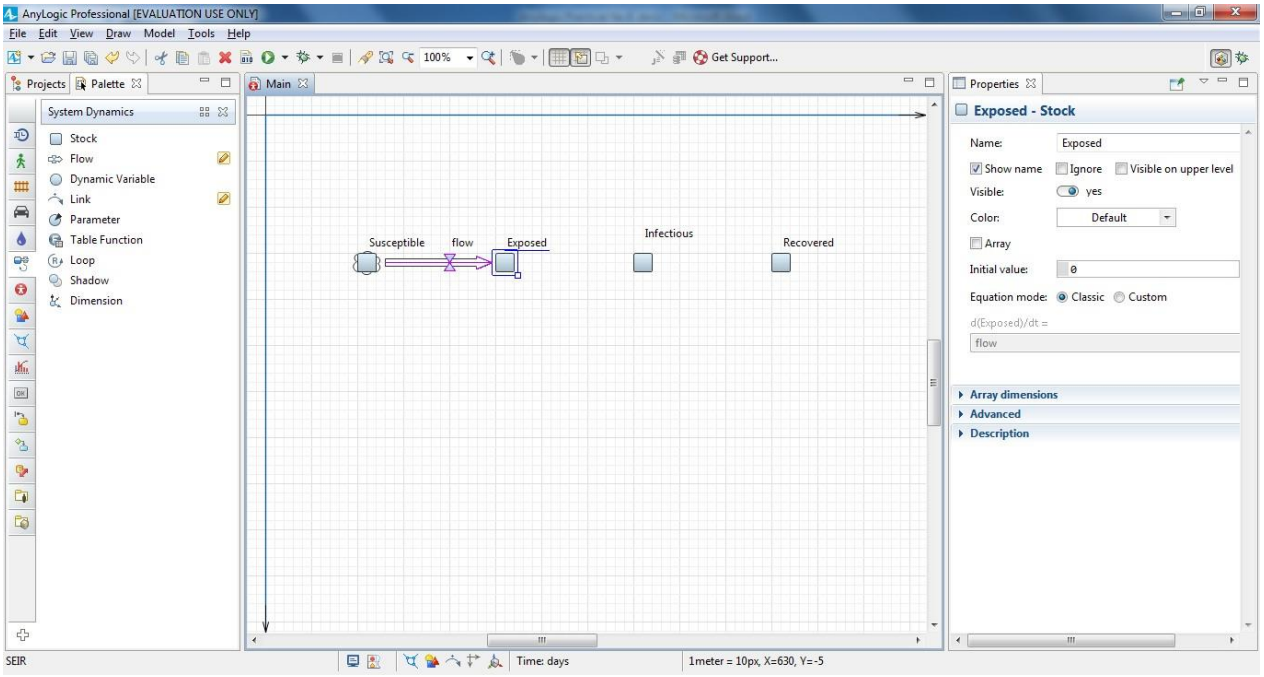



Add three more stocks. Place them as shown in the figure and name them Exposed, Infectious, and Recovered.





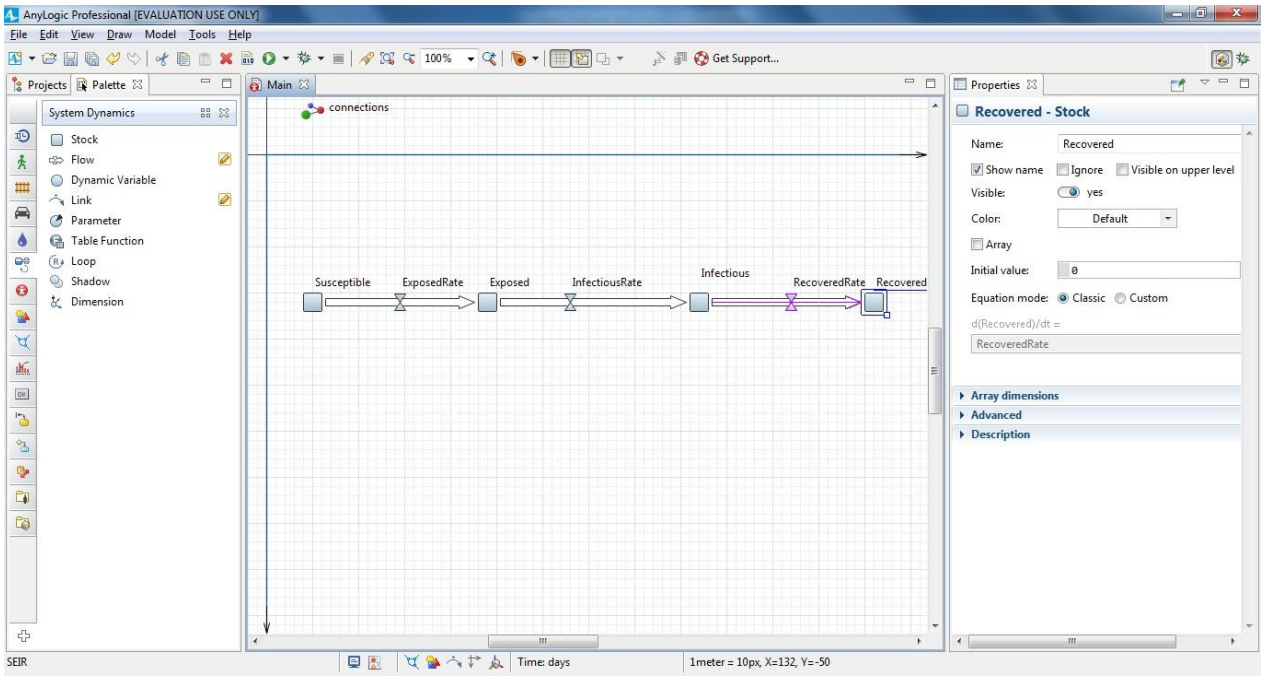

The flow's arrow shows its direction.
In our model, susceptible people are exposed to the virus, become infectious, and then recover. It's a progression that requires our model to use three flows to drive people from one stock to the next. Add the first flow that flows from the stock Susceptible to Exposed. Doubleclick the stock where the flow flows out (Susceptible), and then click the stock
where it flows in (Exposed).



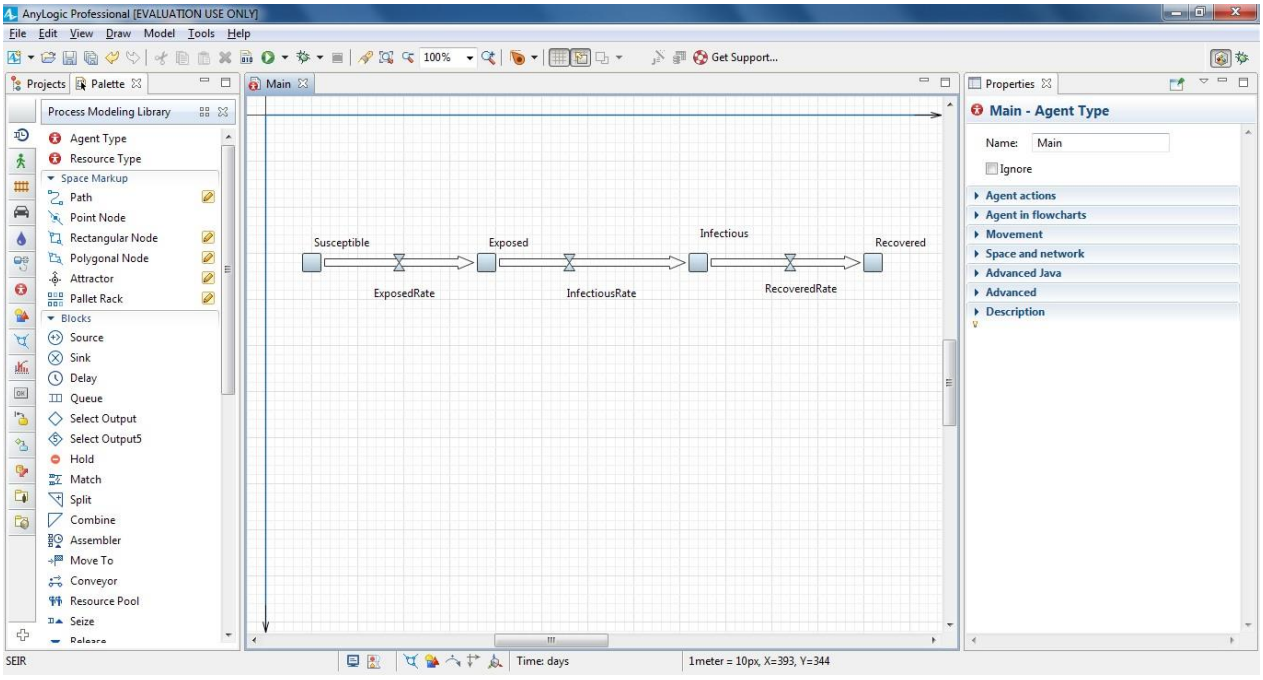
Name the flow ExposedRate.

Add a flow from Exposed to Infectious, and then name it InfectiousRate

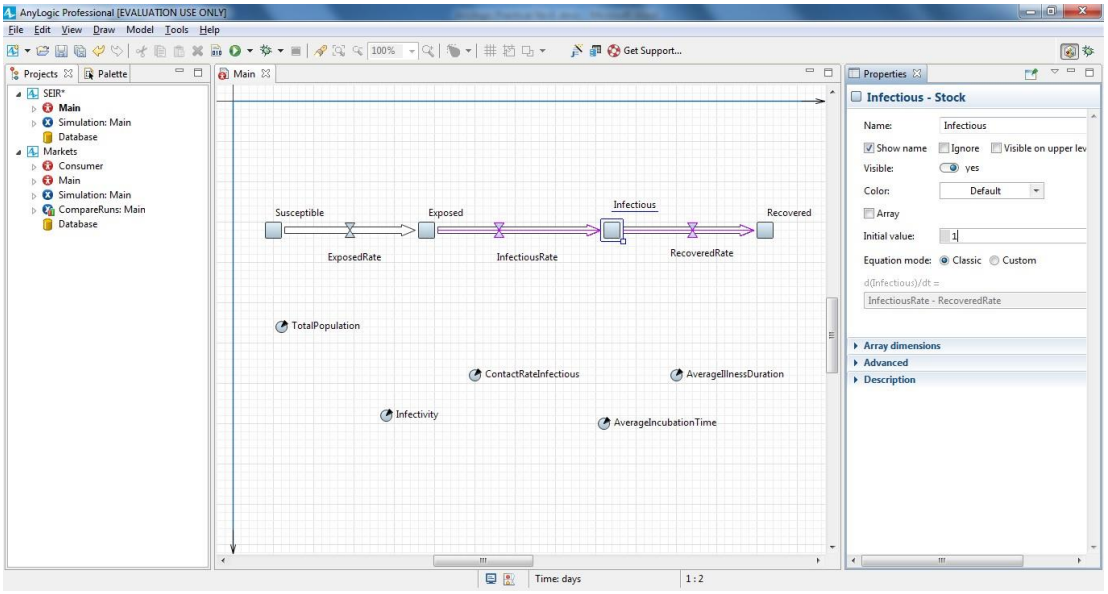
Add a flow from Infectious to Recovered, and then name it RecoveredRate



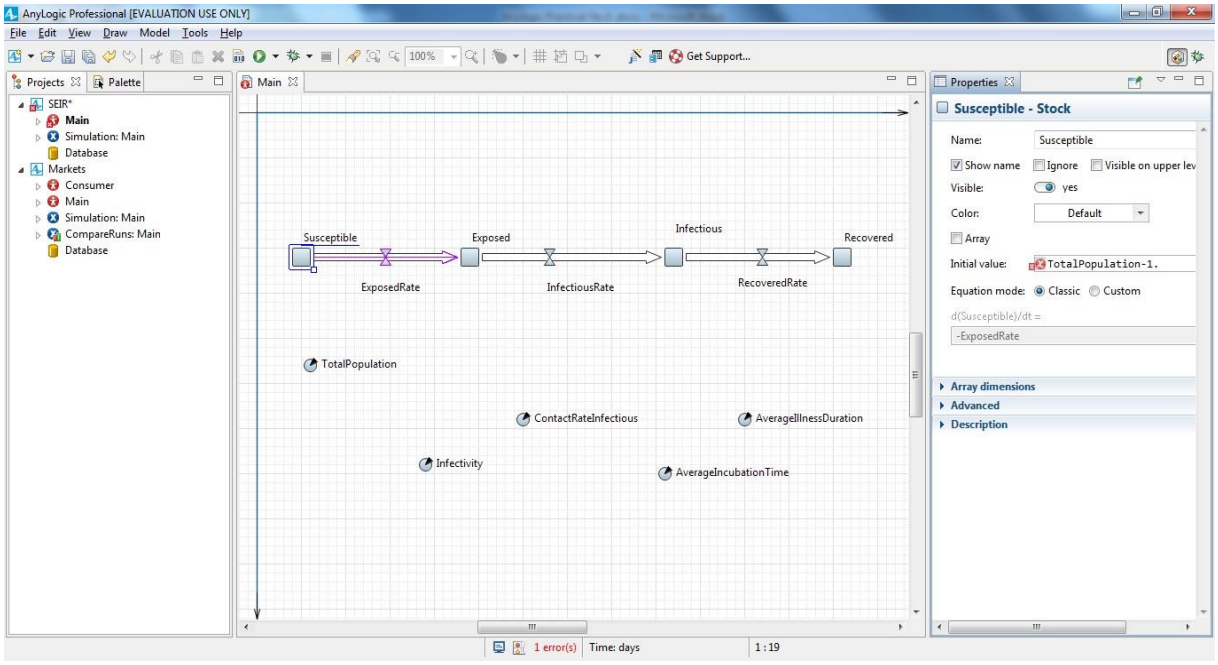

Rearrange the flow names as shown in the figure below. To do this, select a flow and then drag its name.



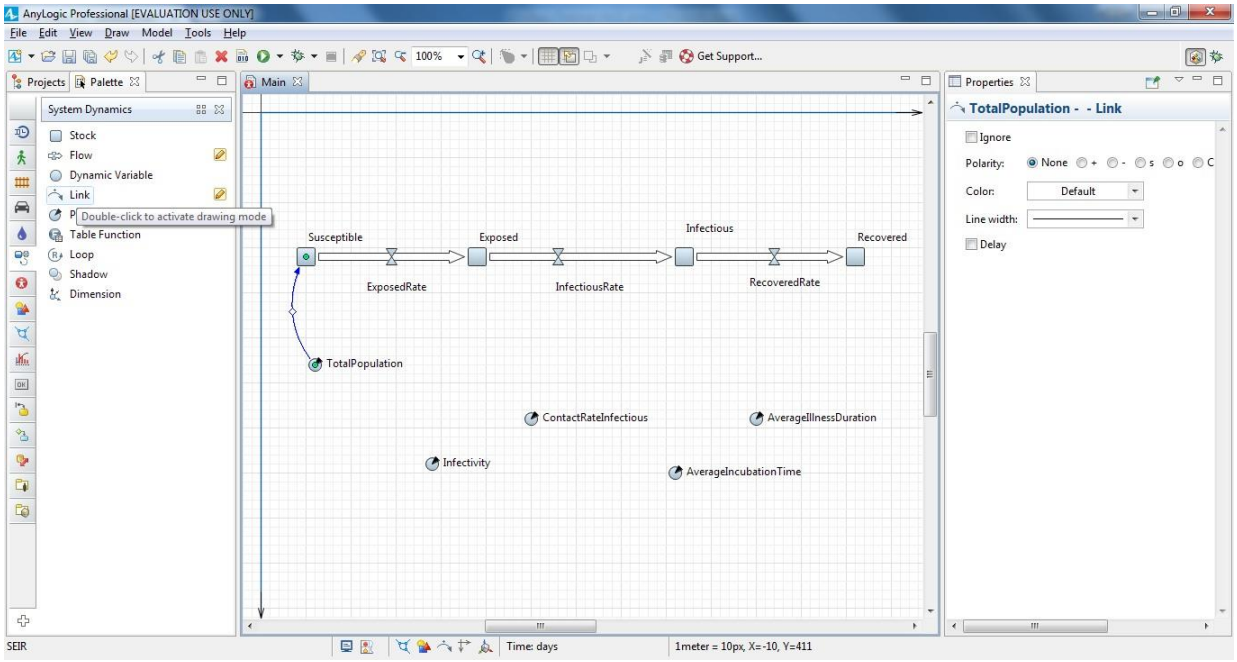
Add five Parameters ,
rename them, and define their default values according to the information below:
TotalPopulation = 10 000
Infectivity = 0.6
ContactRateInfectious = 1.25
AverageIncubationTime = 10
AverageIllnessDuration = 15



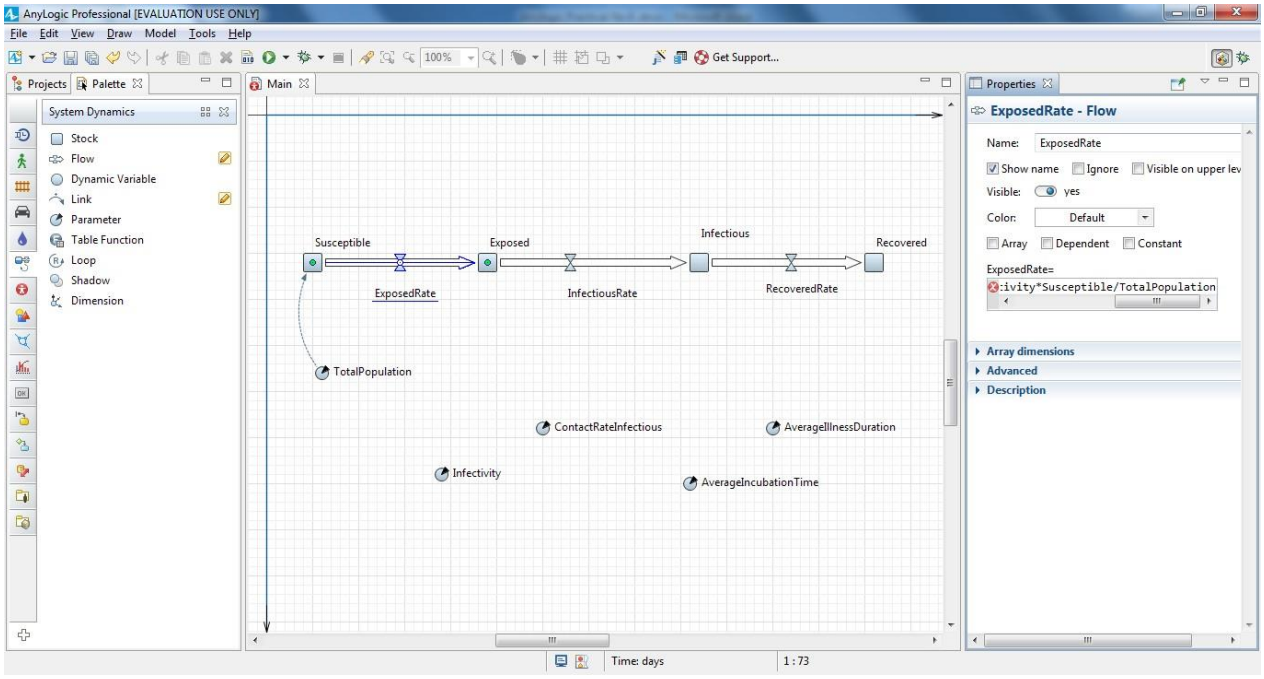

Define the number of infected people by specifying 1 as the Initial Value of the stock Infectious.Define the
Initial Value for the stock Susceptible: TotalPopulation-1.



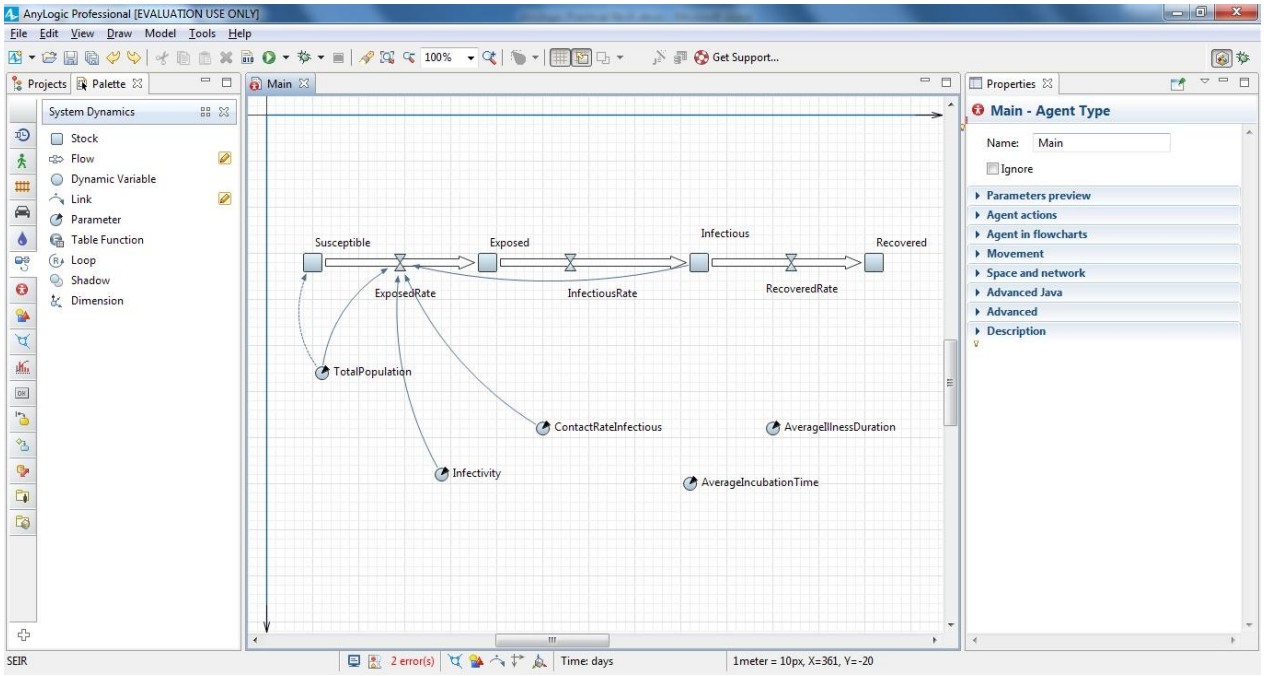
Draw a dependency link from TotalPopulation to Susceptible:
In the System Dynamics palette, double-click the Link element, click TotalPopulation, and then click the stock Susceptible. You should see the link with small circles drawn on its end points



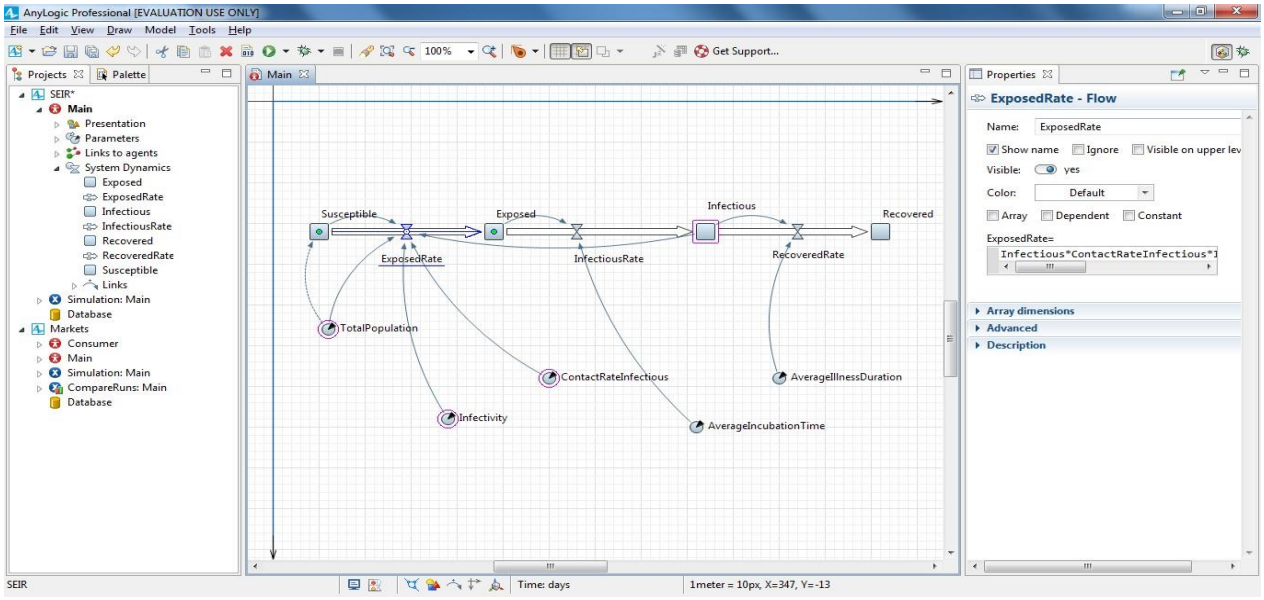

formula for the flow ExposedRate.
Click the flow and define the following formula using the Code Completion assistant:
Infectious*ContactRateInfectious*Infectivity*Susceptible/TotalPopulation.



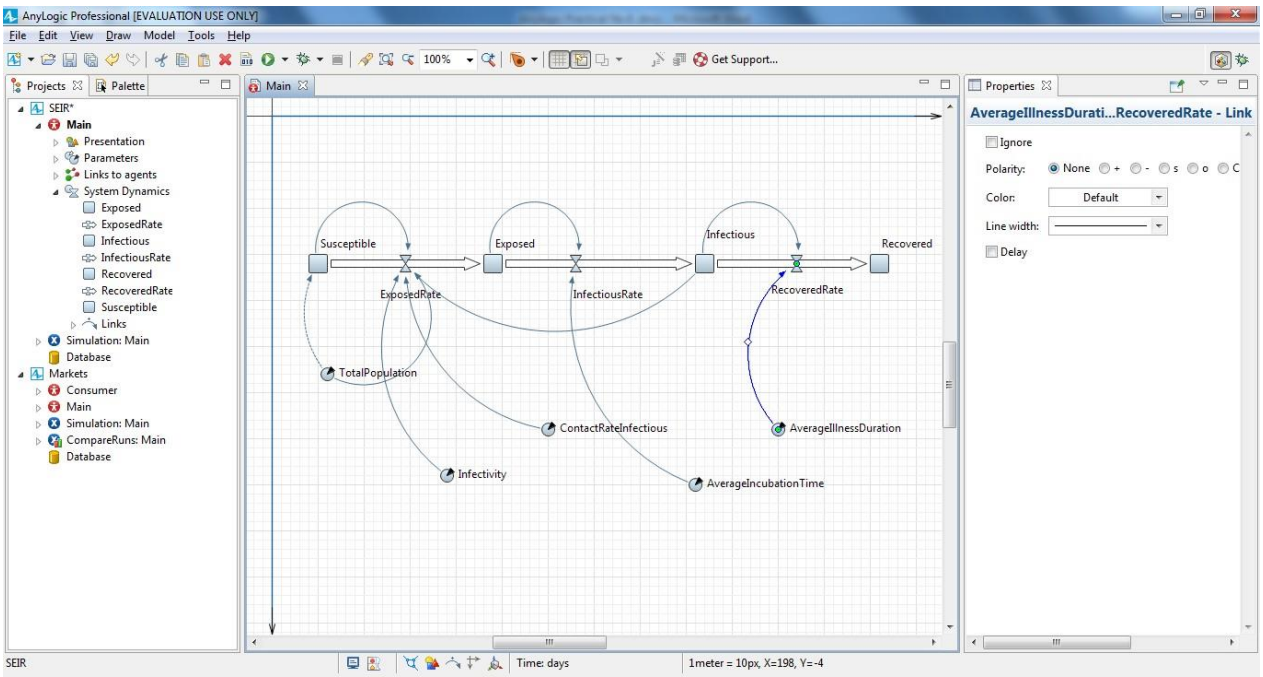
Right-click ExposedRate flow in the graphical diagram, and choose Fix Dependancy Links > Create Missing Links from the context menu. Afterward, you should see the links in the stock and flow diagram:



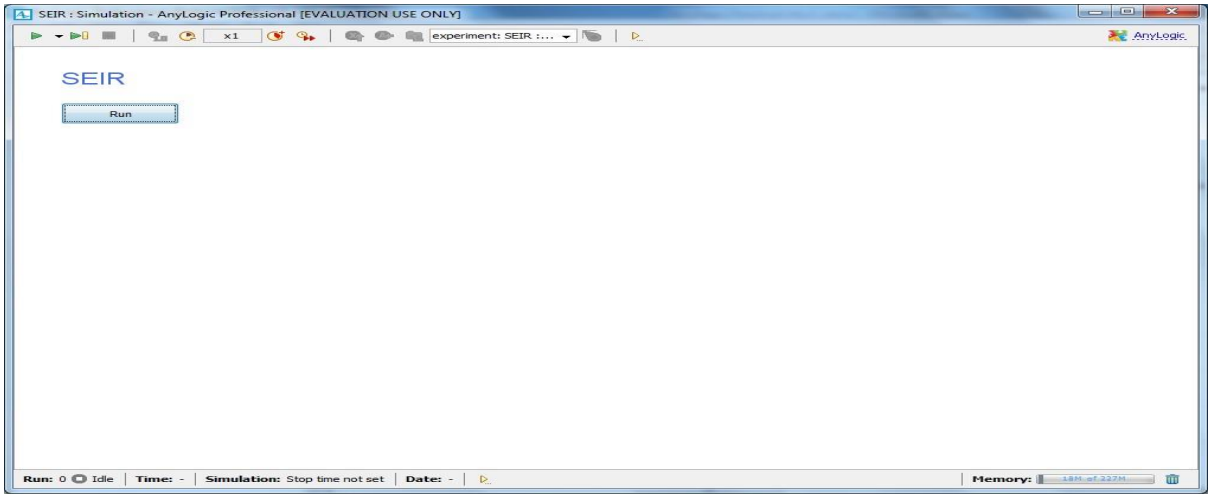

Define the following formula for InfectiousRate: Exposed/AverageIncubationTime
Define the following formula for RecoveredRate: Infectious/AverageIllnessDuration
Draw the missing dependency links, and your stock and flow diagram should resemble the following image:



Adjust the appearance of dependency links. Modify the links' bend angles to make the diagram match the figure below. To adjust the link's bend angle, select it, and then drag the handle in the middle of the link.




Run the model and inspect the dynamics using the variables' inspect windows. To open a variable's inspect window, click the variable to select it. To resize the window, drag its lower right corner.



Practical 5

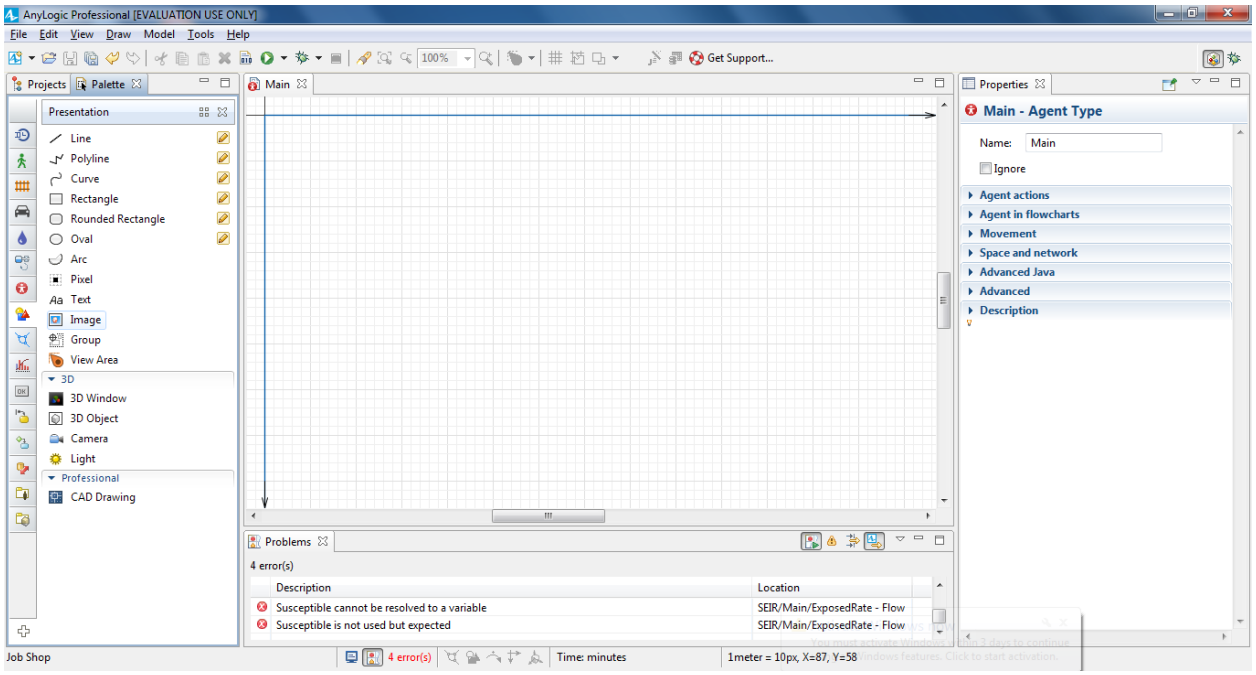
Design and develop a discrete-event model that will simulate process by:

- Creating a simple model
- Adding resources
- Creating 3D animation
- Modeling delivery

[Use a case situation like a company's manufacturing and shipping].

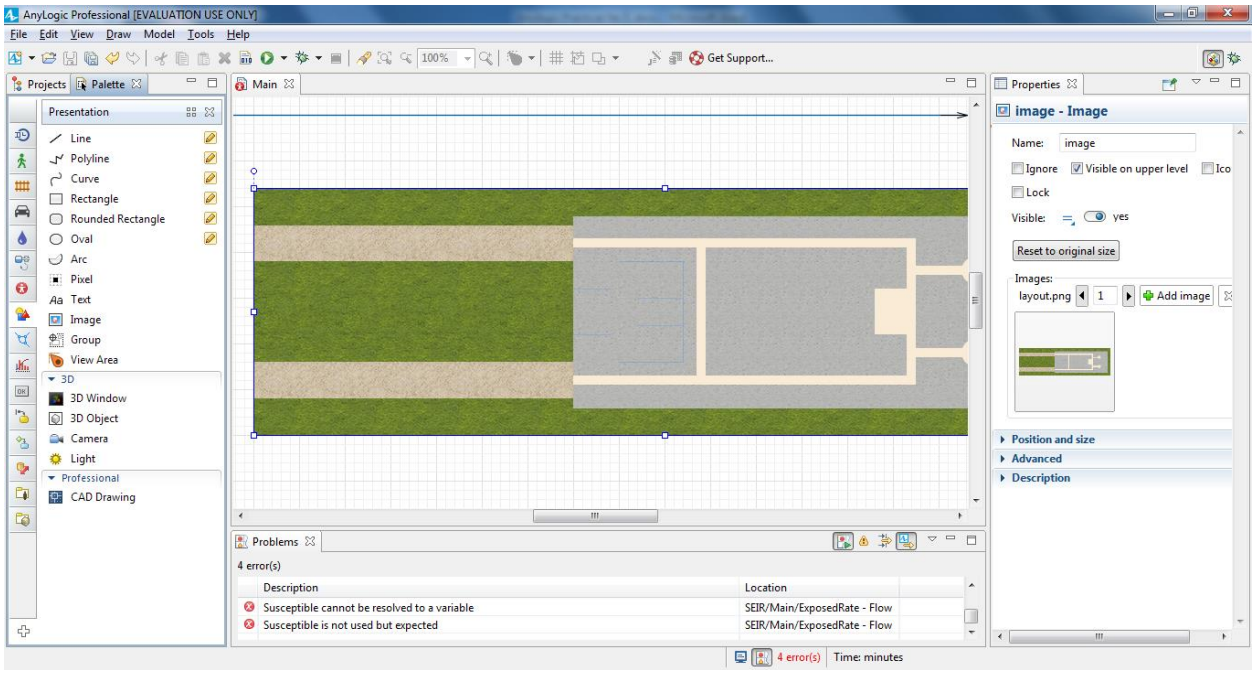
Create a new model. In the **New Model** wizard, set the **Model name**: Job Shop, and **Model time units**: **minutes**.

**2.** Open the **Presentation** palette. The palette has several shapes that you can use to draw model animation, including a rectangle, a line, an oval, a polyline and a curve.



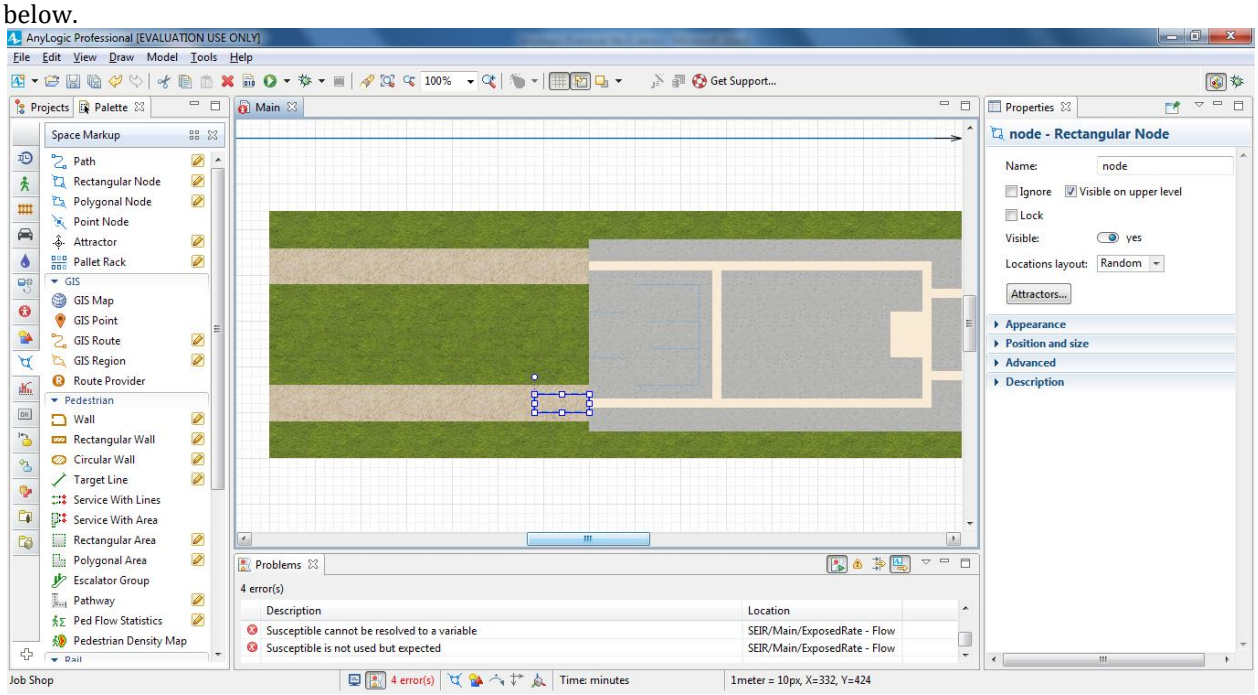

On the **Presentation** palette, select the **Image** shape and then drag it on to the Main diagram. You can use the **Image** shape to add images in several graphic formats -- including PNG, JPEG, GIF, and BMP – to your presentation. You'll see the dialog box that prompts you to choose the image file the shape will display.

**5.** Browse to the following location and then select the layout.png image: AnyLogic folder/resources/AnyLogic in 3 days/Job Shop

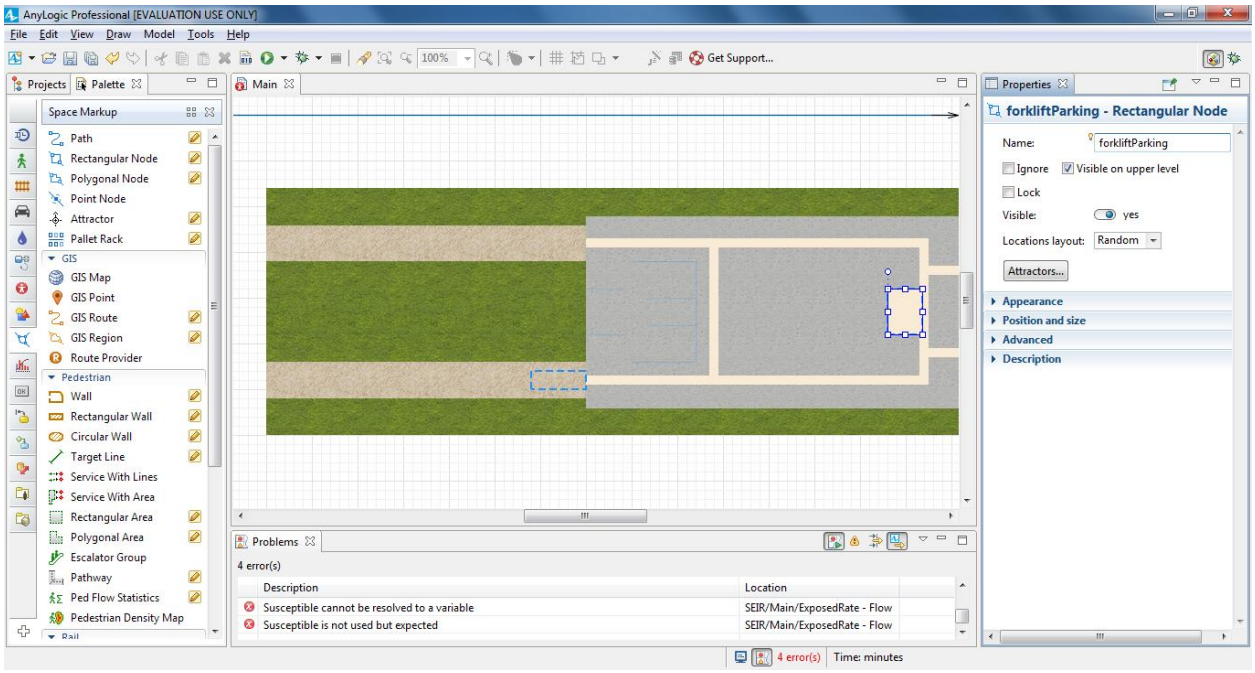



Select the image in the graphical editor. In the **Properties** view, select the **Lock** checkbox to lock the image.

Open the **Space Markup** palette, and drag the **Rectangular Node** element on to the Main diagram. Resize the node. The node should look as in the figure below.




Name the created node receivingDock.

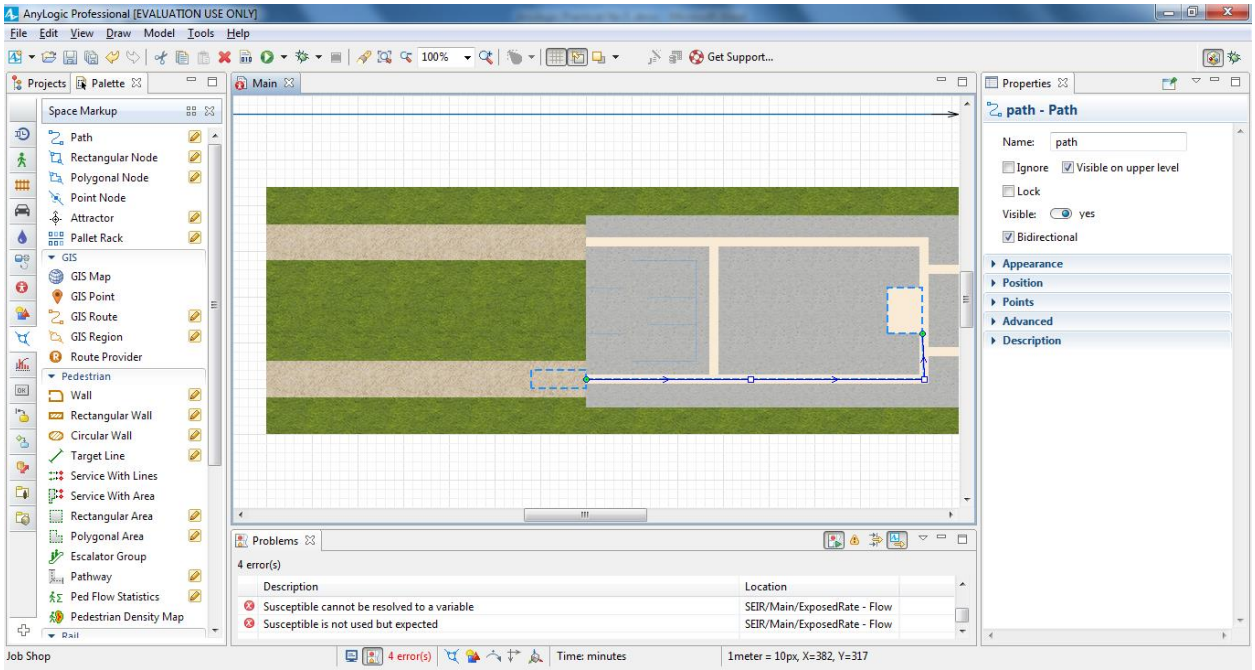



Use another **Rectangular Node** to draw the parking area as shown in the figure below and then name this node forkliftParking.

Do the following to draw a movement path that will guide our model's forklift trucks:

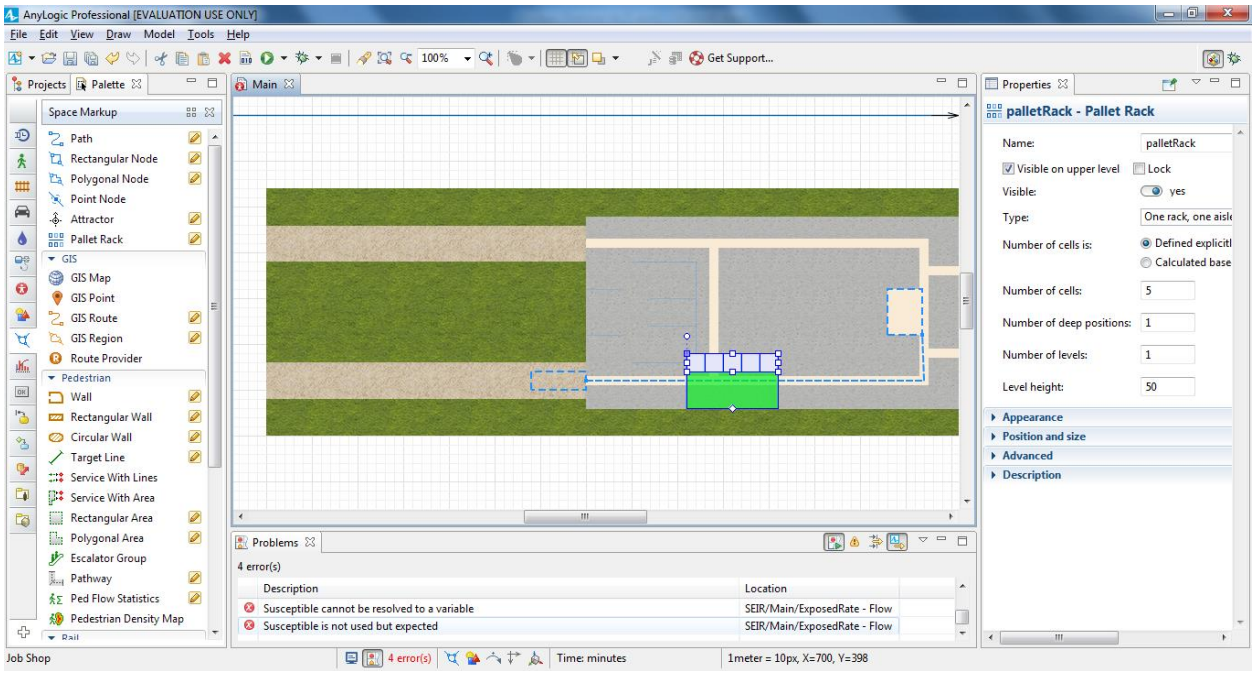
In the **Space Markup** palette, double-click the **Path** element to activate its drawing mode.

b. Draw the path as shown in the figure below by clicking the receivingDock border, clicking in the diagram to add the path's turning point, and then clicking the forkliftParking node's border.



If you've successfully connected the nodes, the path's connection points will display cyan highlights each time you select the path.

Define your model's warehouse storage by dragging the **Pallet Rack** element from the **Space Markup** palette on to the layout and placing its aisle on the path. A correctly-placed pallet rack will display a green highlight that shows it is connected to the network.

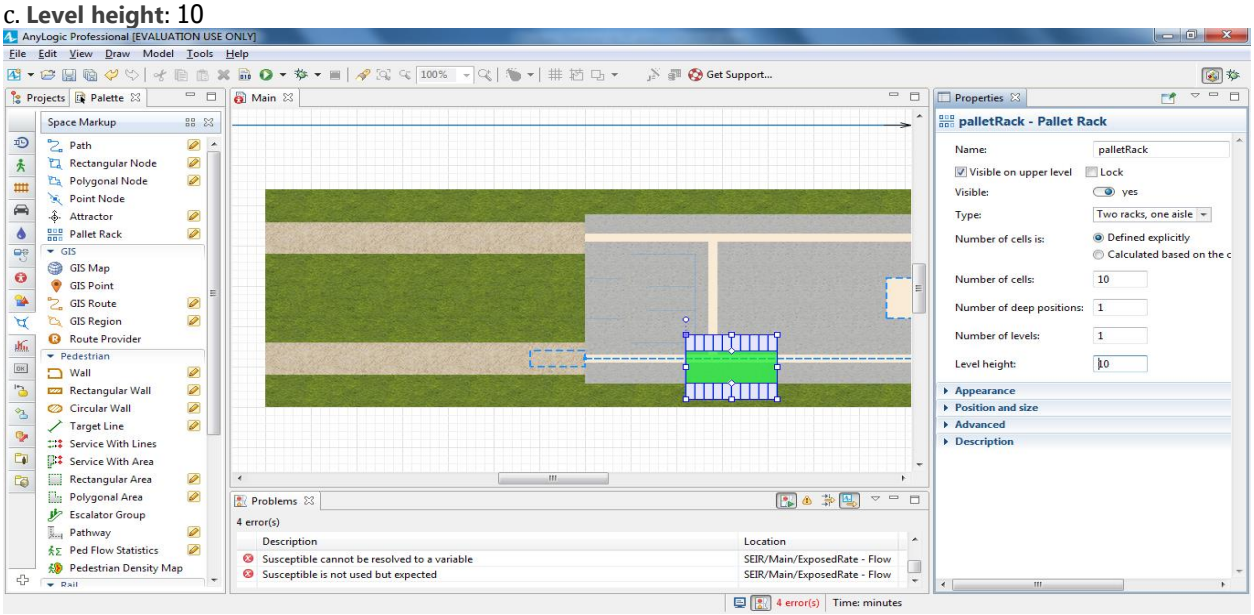



In the pallet rack's **Properties** area, do the following:

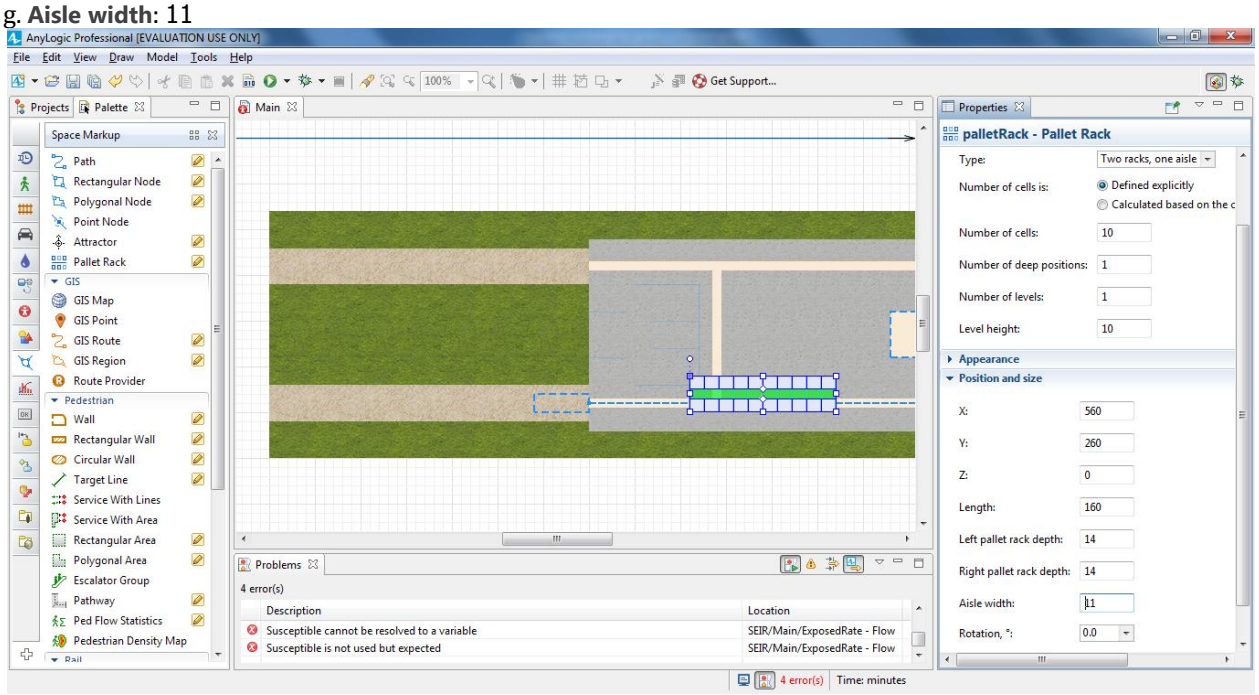
a. Set **Type** to: two racks, one aisle

b. **Number of cells**: 10

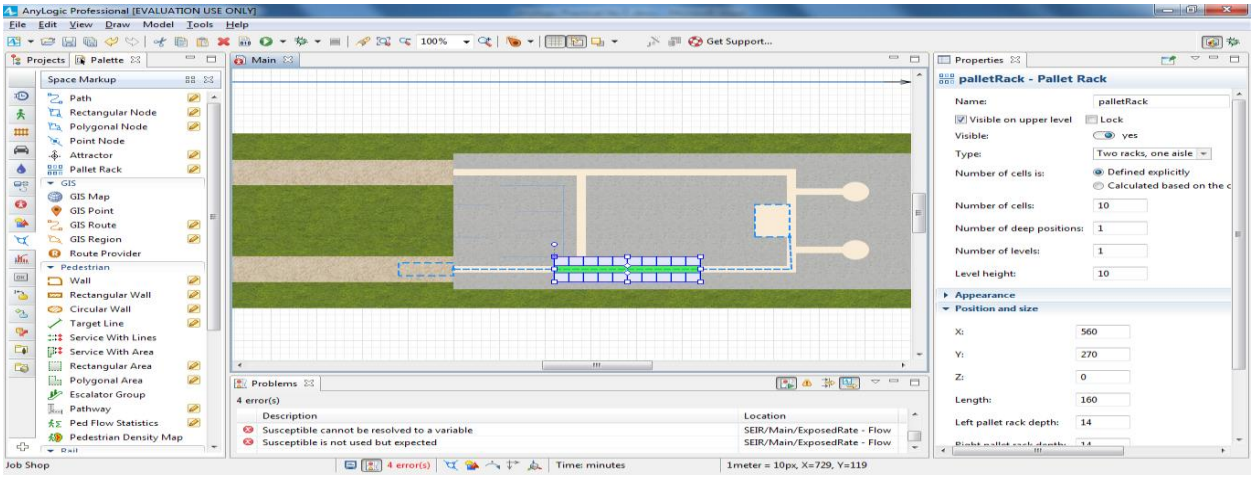
c. **Level height**: 10



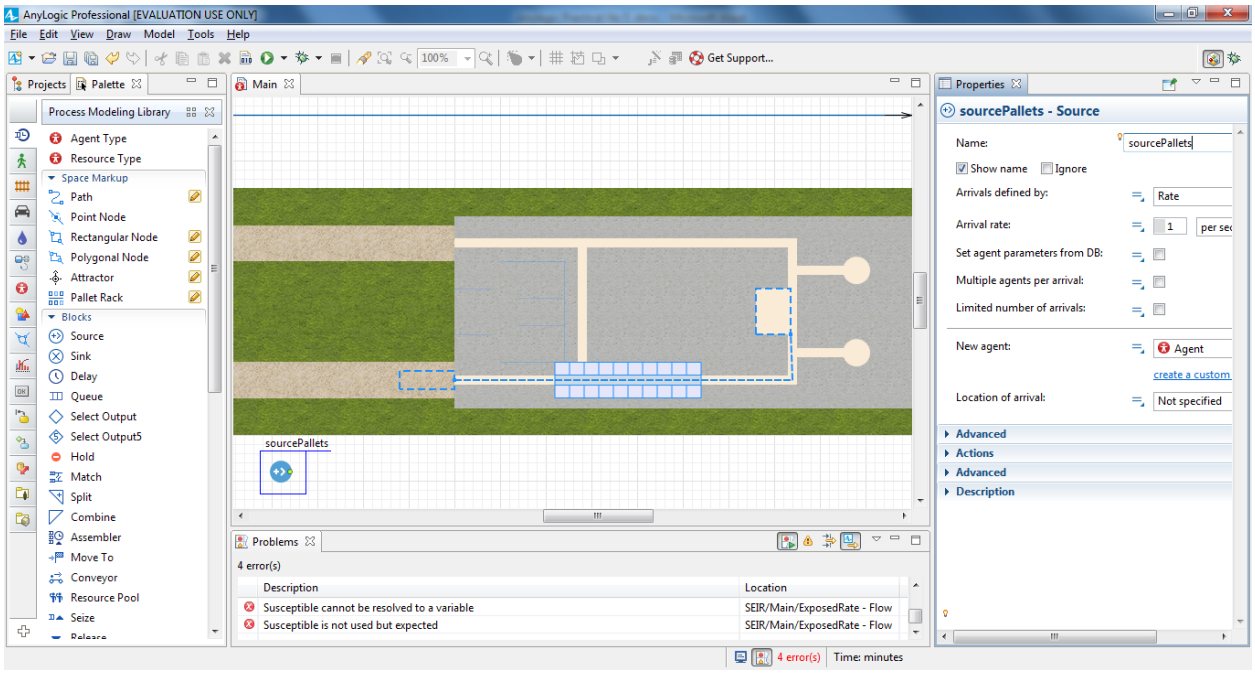
In the **Position and size** section:
d. **Length**: 160
e. **Left pallet rack depth**: 14
f. **Right pallet rack depth**: 14
g. **Aisle width**: 11




After you've completed these changes, the pallet rack should resemble the pallet rack shown in the figure below. If necessary, move the pallet rack so that its center aisle lies on the path. Make sure the pallet rack is connected to the network by clicking it twice to select it. Your first click will select the entire network, and the second will select the pallet rack. The pallet rack should display a green highlight that shows it is connected to the network.



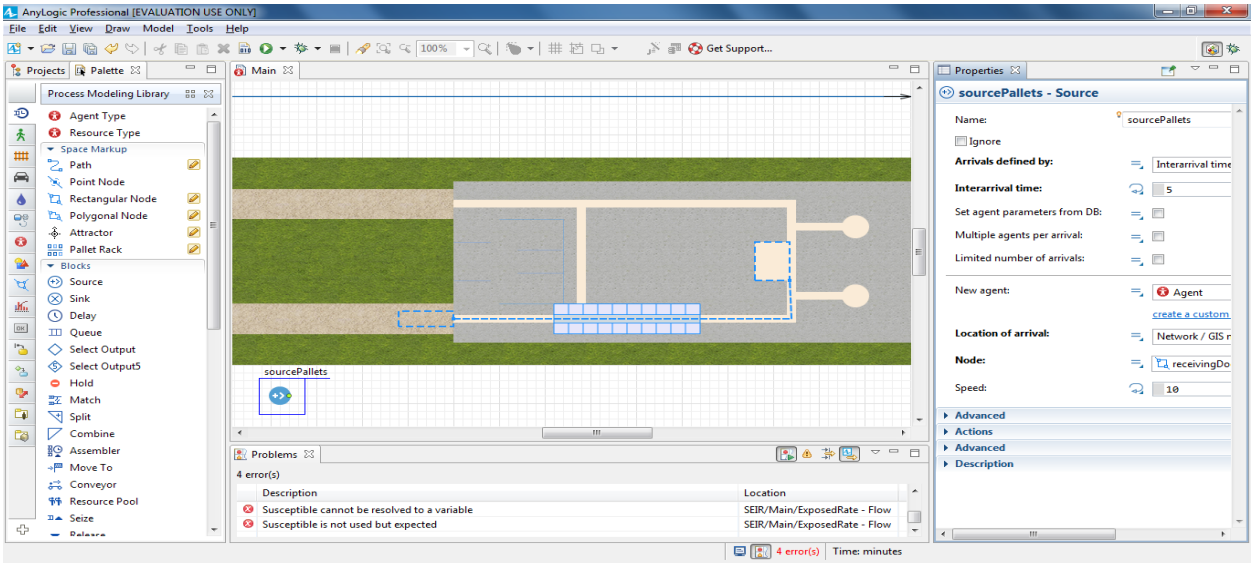

Drag the **Source** element from the **Process Modeling Library** palette on to the graphical diagram and name the block sourcePallets.



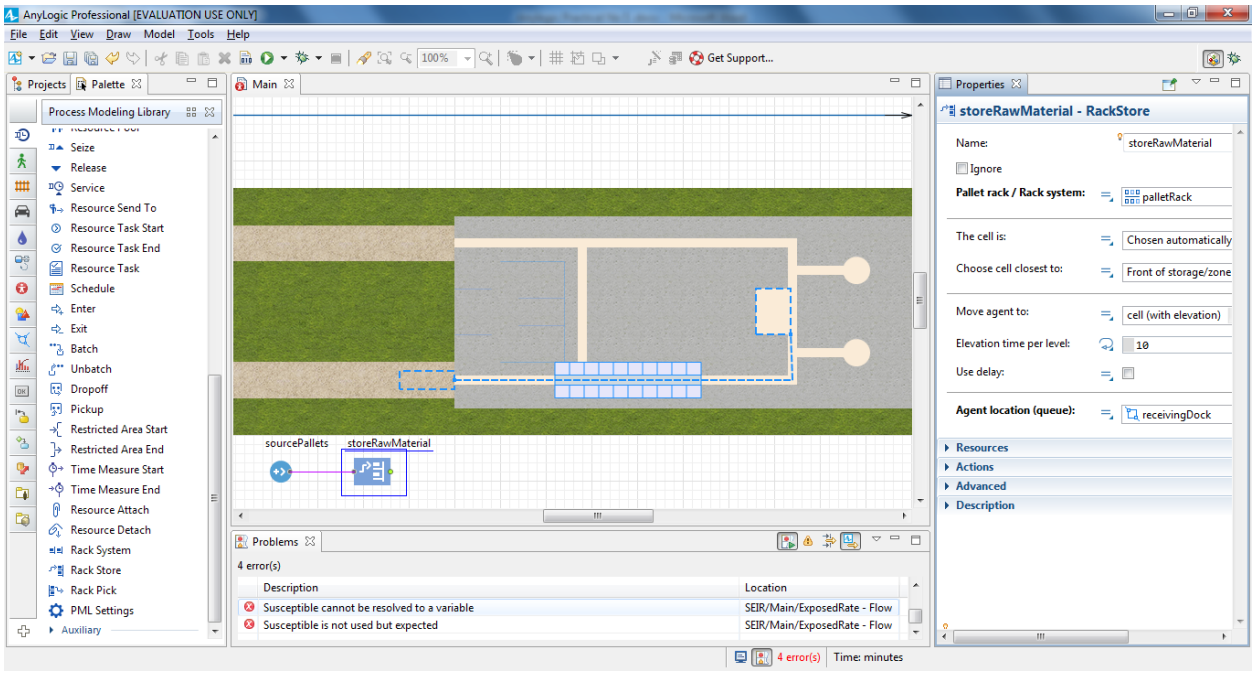

While the **Source** block usually acts as a process starting point, our model will use it to generate pallets.

In the sourcePallets block's **Properties** area, do the following to ensure the model's pallets arrive every five minutes and appear in the receivingDock node.

a. In the **Arrivals defined by** area, click **Interarrival time**.
b. In the **Interarrival time** box, type 5, and select **minutes** from the list on the right to have pallets arrive every five minutes.
c. In the **Location of arrival** area, click **Network / GIS node** in the list.
d. In the **Node** area, click receivingDock in the list.

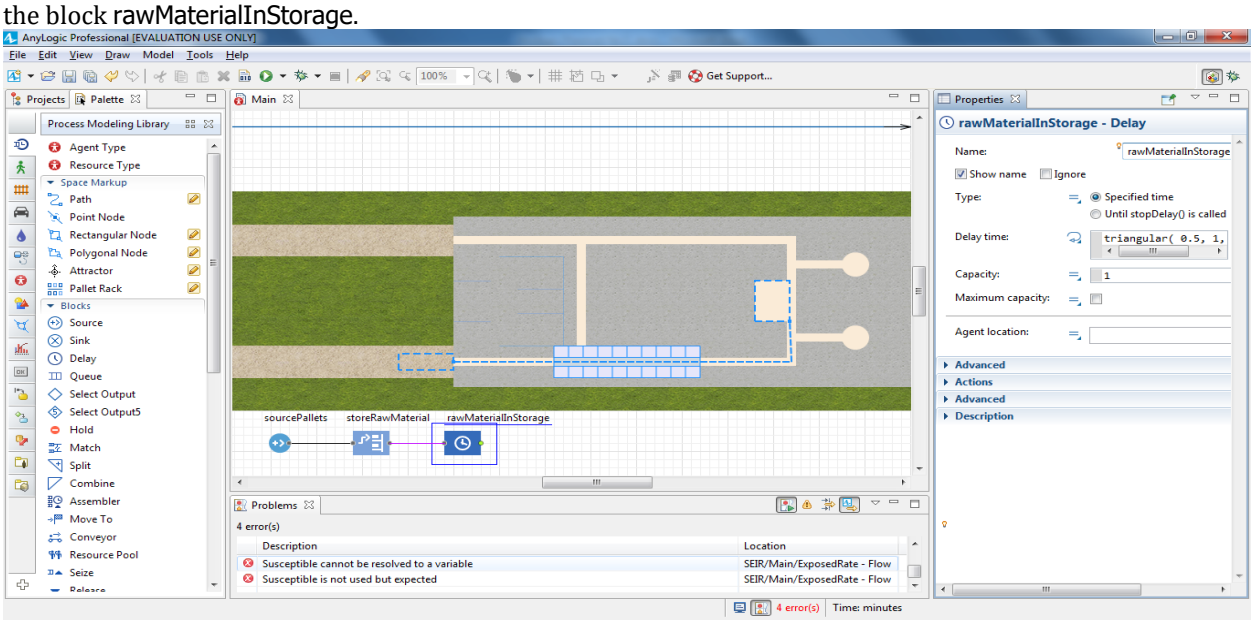


Drag the **RackStore** block from the **Process Modeling Library** palette on to the diagram and place it near the sourcePallets block so they are automatically connected as shown in the diagram below.
The **RackStore** block places pallets into a given pallet rack's cells.

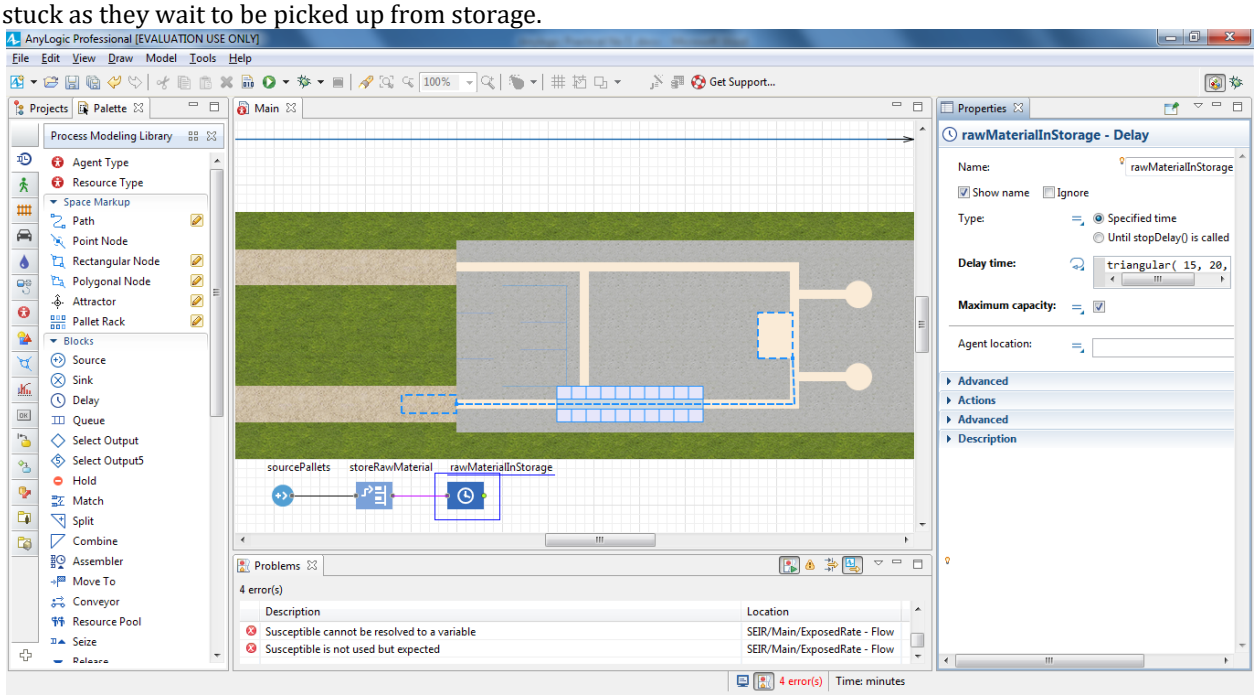



In the rackStore block's **Properties** area, do the following:
a. In the **Name** box, type storeRawMaterial.
b. In the **Pallet rack / Rack system** list, click palletRack.
c. In the **Agent location (queue)** list, click receivingDock to specify the location where agents wait to be stored.

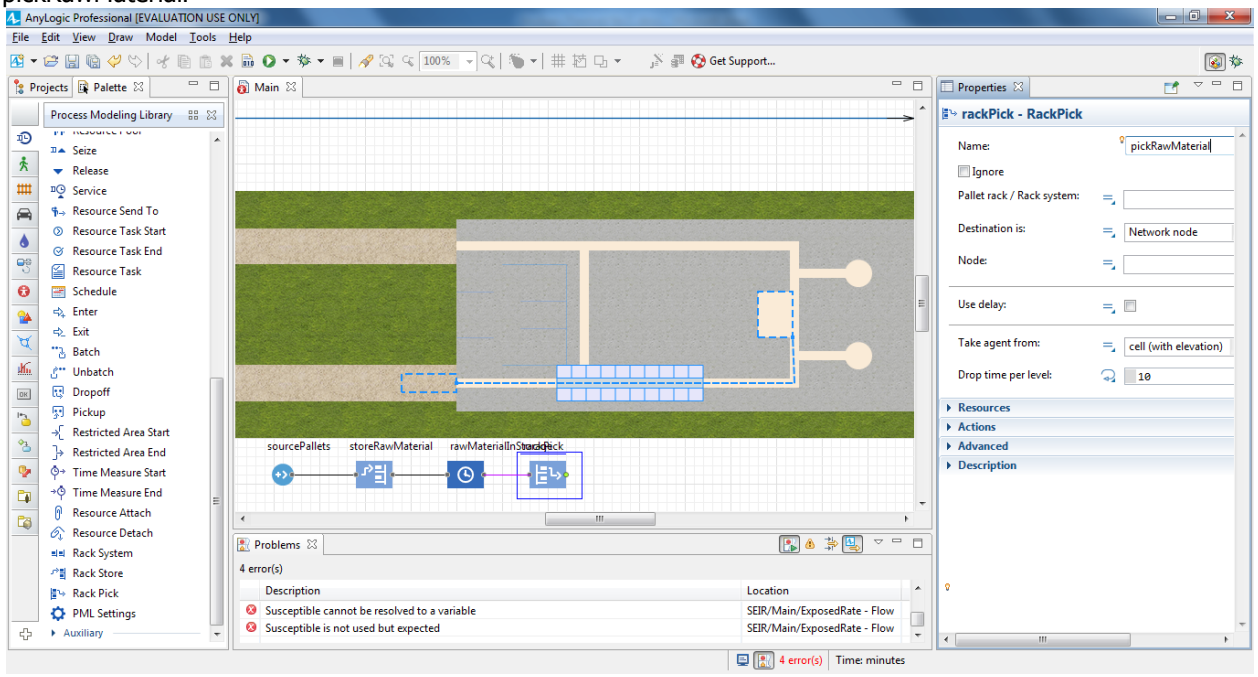
Add a **Delay** block to simulate how pallets wait in the rack and then name the block rawMaterialInStorage.



In the rawMaterialInStorage block's **Properties** area, do the following:
a. In the **Delay time** box, type triangular(15, 20, 30) and select **minutes** from the list.
b. Select the **Maximum capacity** checkbox to ensure agents will not get stuck as they wait to be picked up from storage.



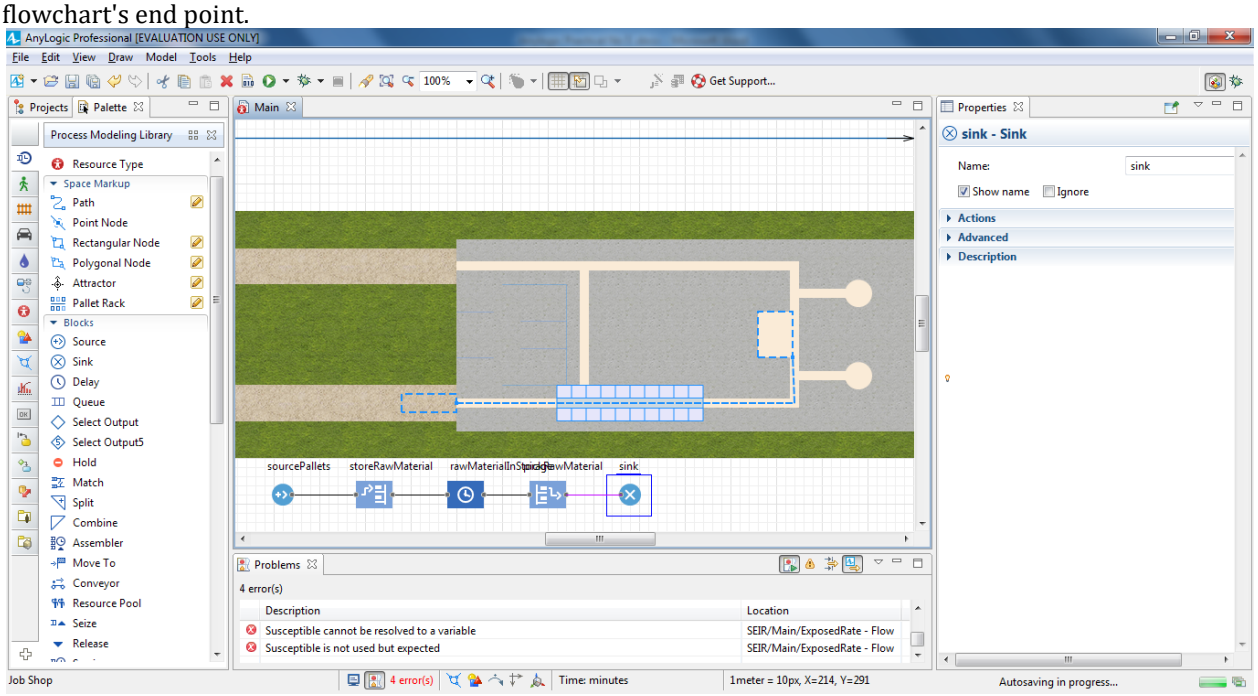

Add a **RackPick** block, connect it to the flowchart, and then name it pickRawMaterial.



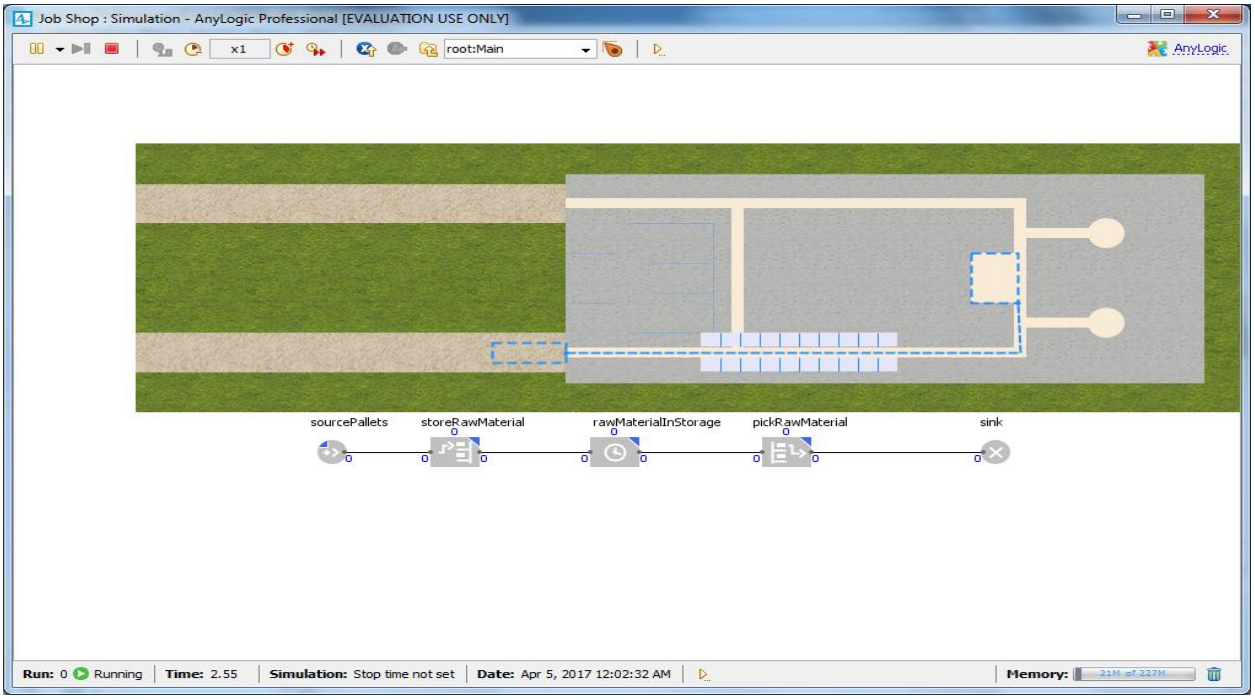

In the pickRawMaterial block's **Properties** area, do the following:
a. In the **Pallet rack / Rack system** list, click palletRack to select the

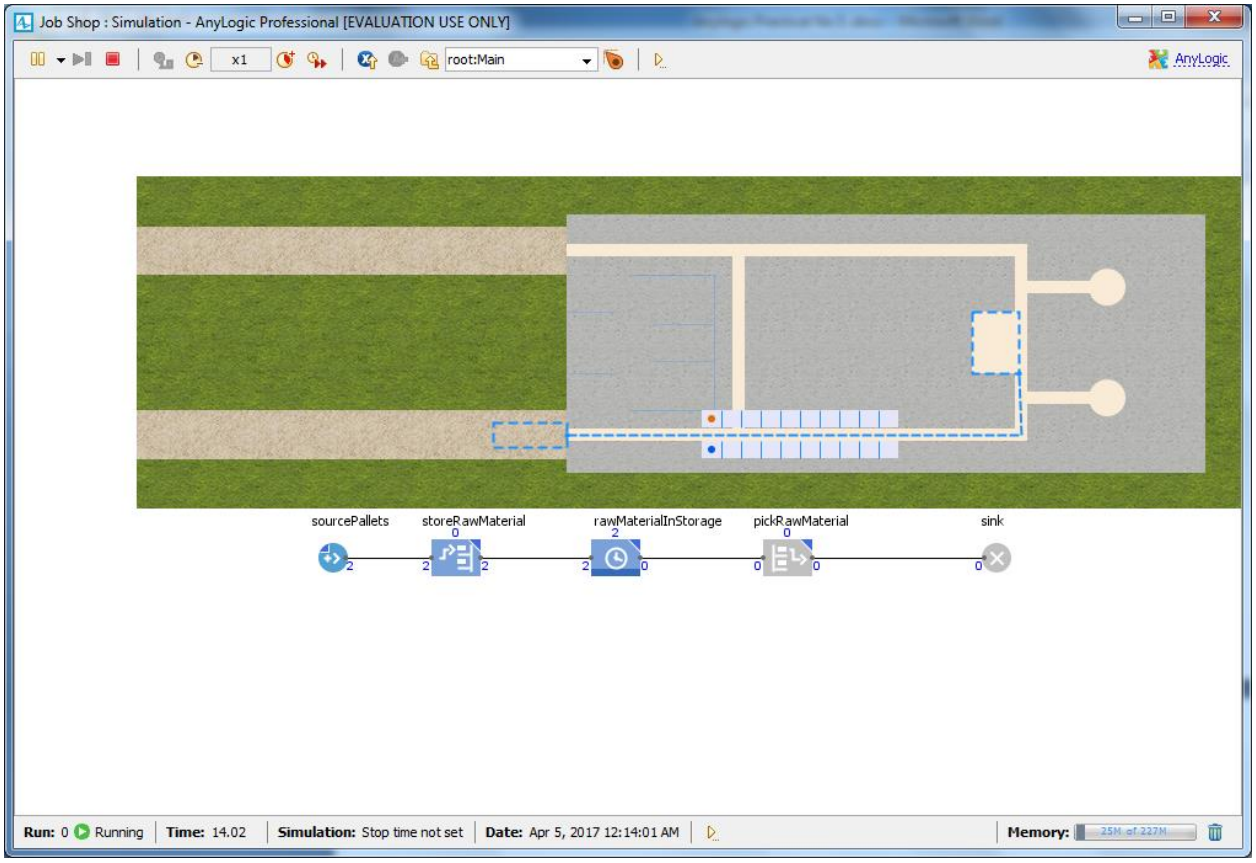
pallet rack that will provide pallets to agents.
b. In the **Node** list, click forkliftParking to specify where the agents should park forklift trucks.

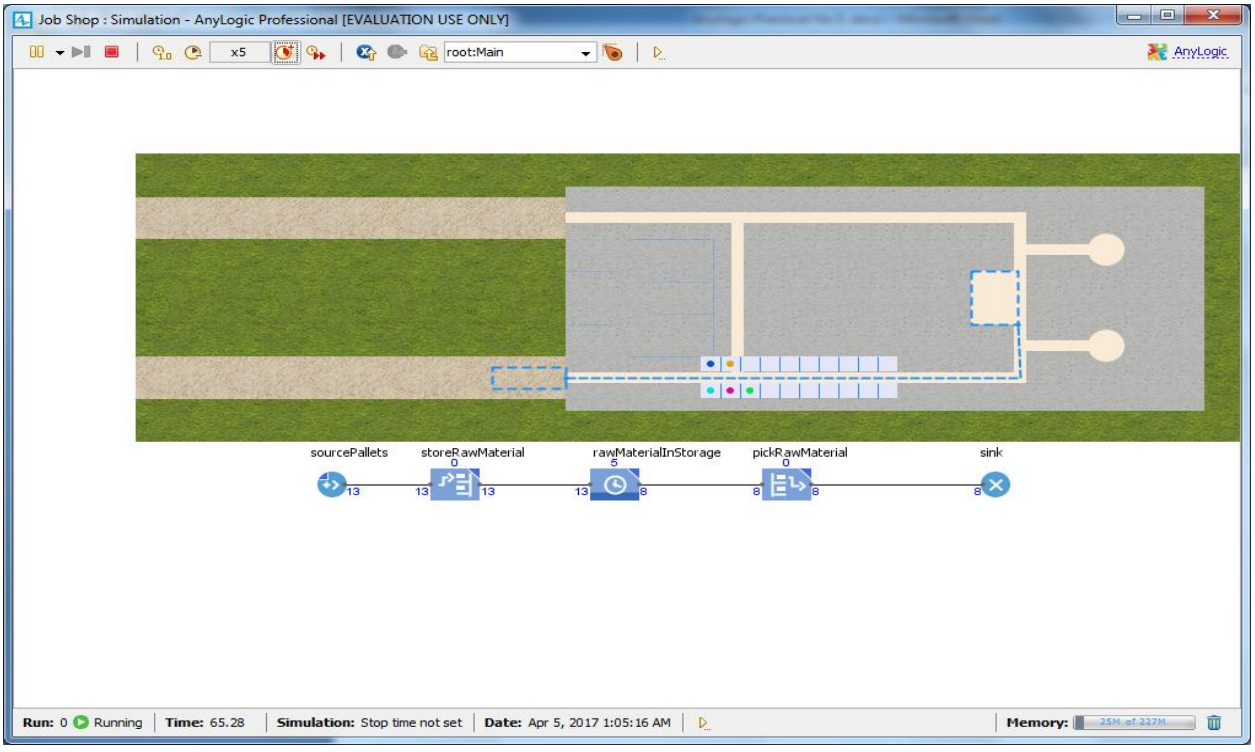
Add a **Sink** block. The **Sink** block disposes agents and is usually a flowchart's end point.




Run the model (Job Shop / Simulation experiment).




Job Shop : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x1
root:Main
AnyLogic
sourcePallets
storeRawMaterial
rawMaterialInStorage
pickRawMaterial
sink
Run: 0 Running
Time: 14.02
Simulation: Stop time not set
Date: Apr 5, 2017 12:14:01 AM
Memory:


Job Shop : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x5
root:Main
AnyLogic
sourcePallets
storeRawMaterial
rawMaterialInStorage
pickRawMaterial
sink
Run: 0 Running
Time: 65.28
Simulation: Stop time not set
Date: Apr 5, 2017 1:05:16 AM
Memory:

## Adding resources

Some examples of resources include:

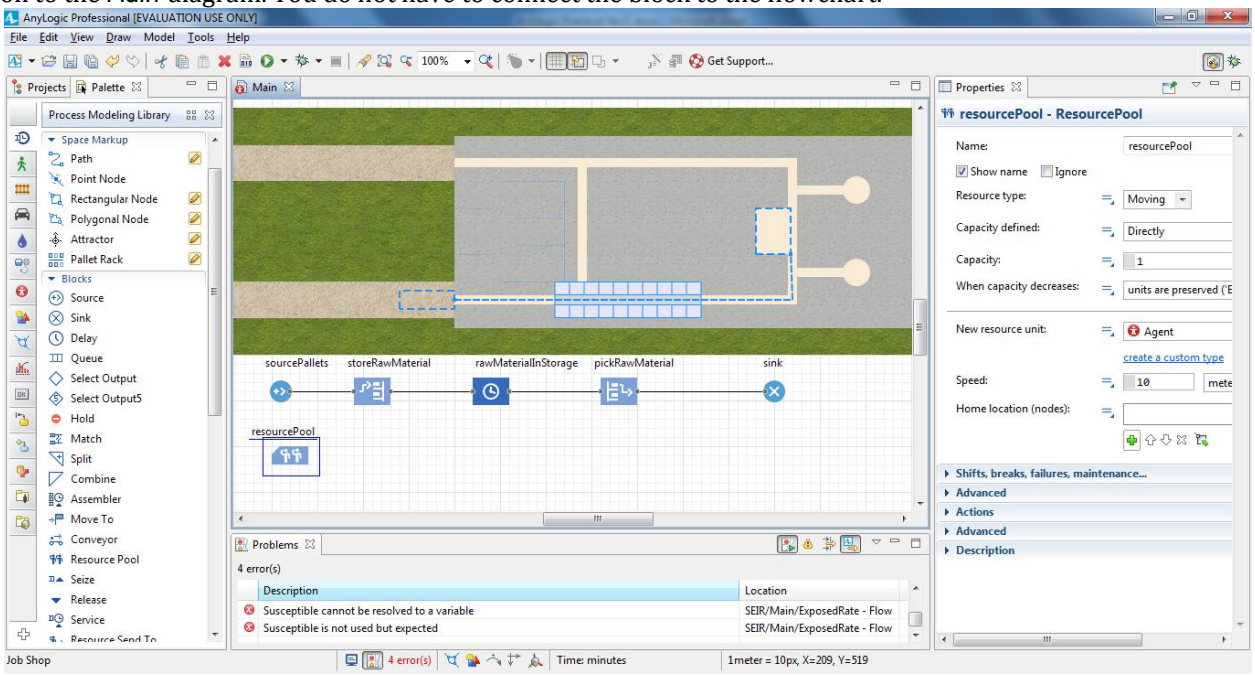
- A hospital model's doctors, nurses, equipment, and wheelchairs
- A supply chain model's vehicles and containers
- A warehouse model's forklift trucks and workers

There are three types of resources: *static*, *moving*, and *portable*.

- Static resources are bound to a specific location, and they cannot move or be moved.
- Moving resources can move independently.
- Portable resources can be moved by agents or by moving resources.

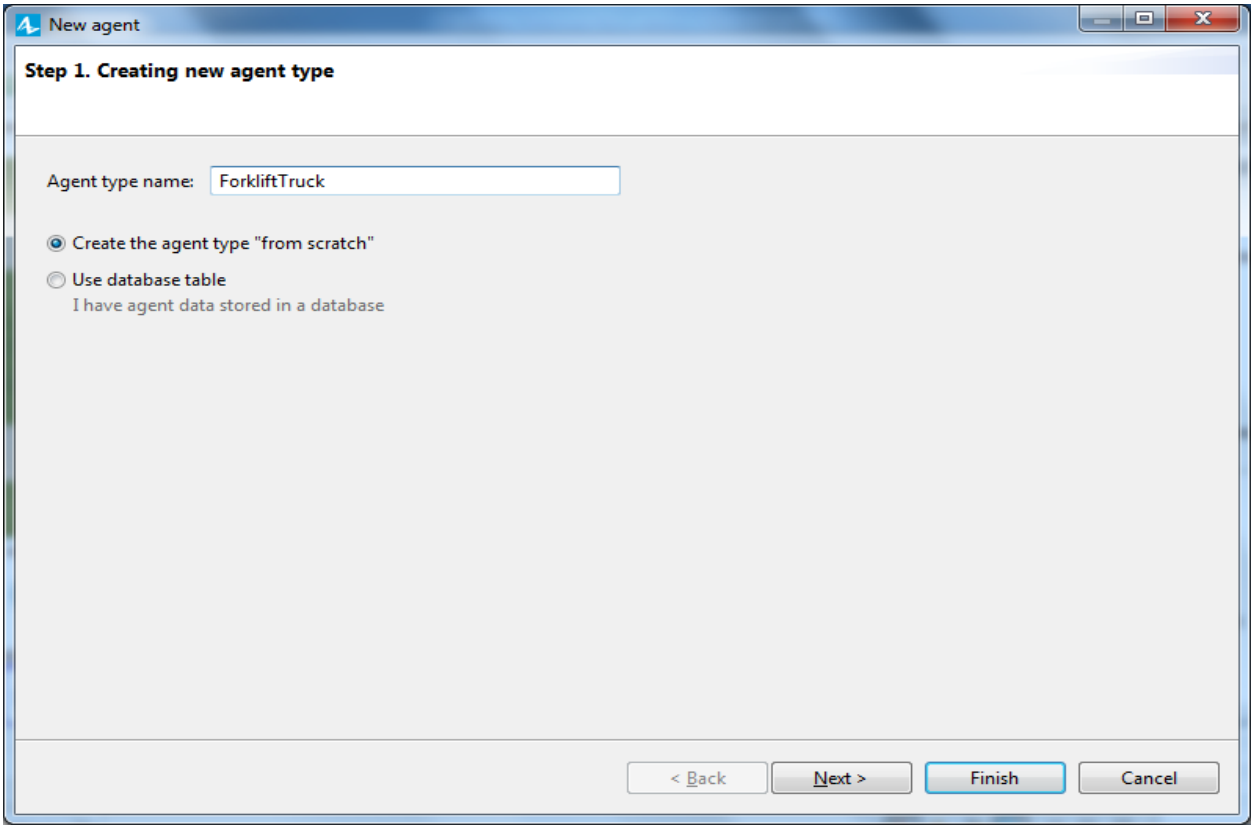
Our model's resources are the forklift trucks that move pallets from the unloading zone to a pallet rack and then deliver pallets from the rack to the production zone.

On the **Process Modeling Library** palette, drag the **ResourcePool** block on to the Main diagram. You do not have to connect the block to the flowchart.




Name the **ResourcePool** block forklifts

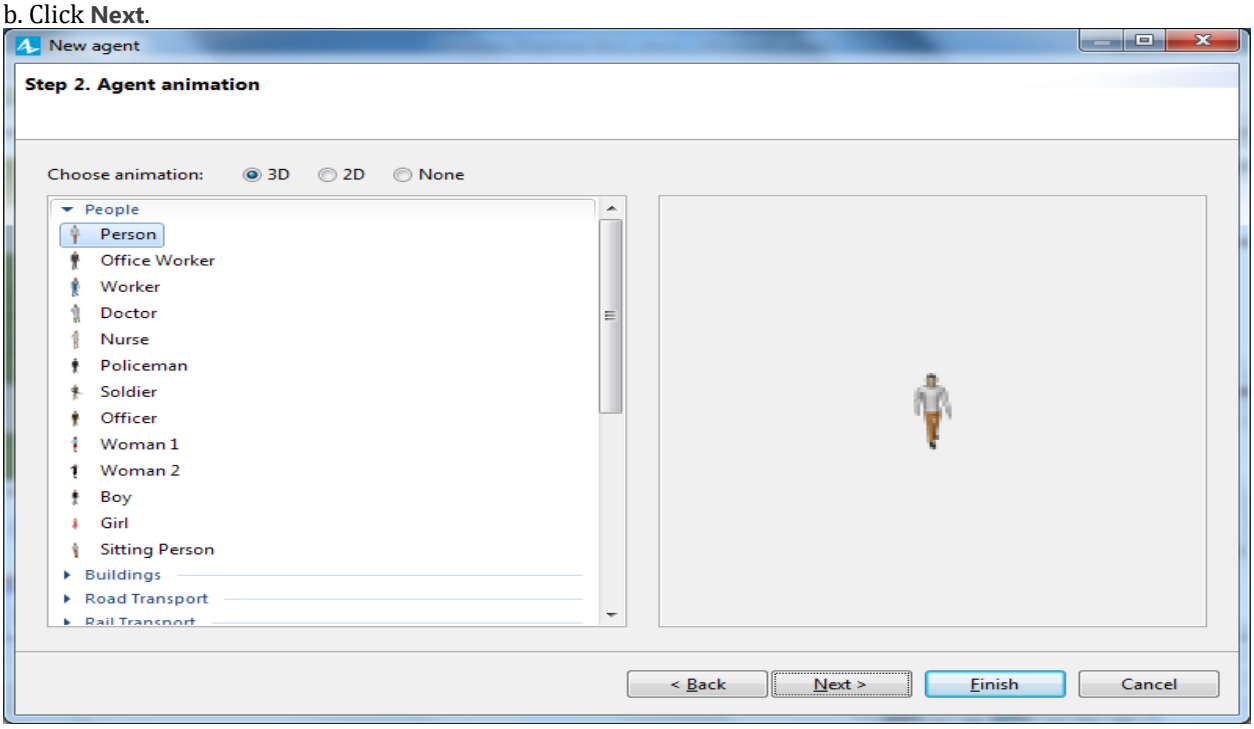
In the forklifts block's **Properties** area, in the new resource unit->click the button **create a custom type**.




In the **New agent** wizard:

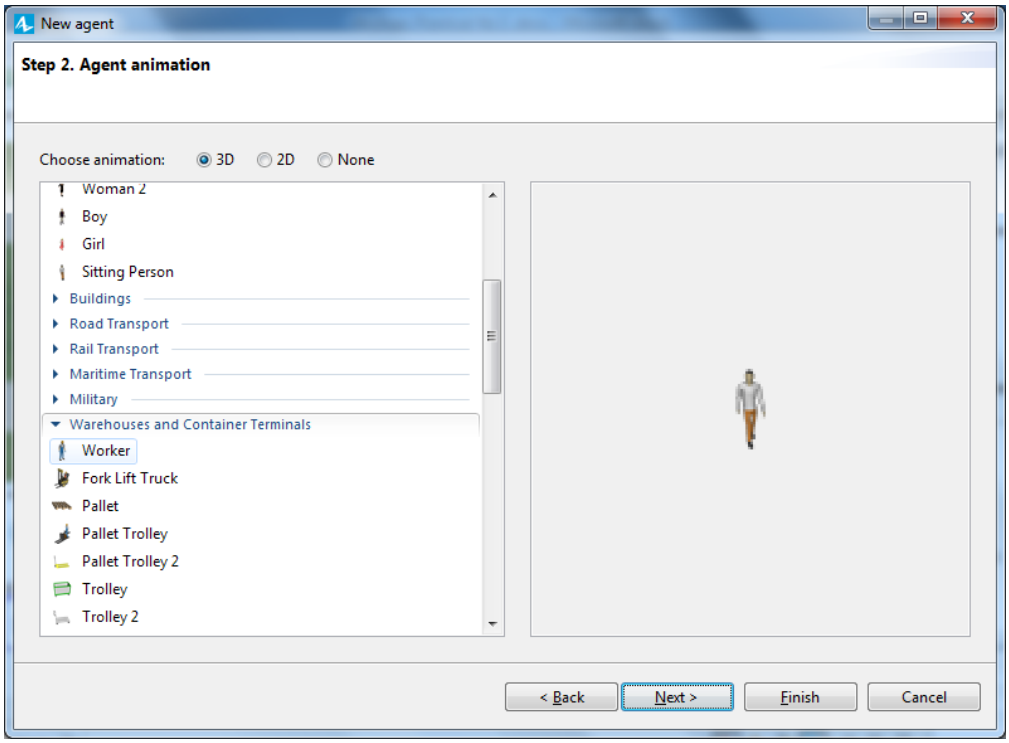
a. In **Agent type name** box, type ForkliftTruck.

b. Click **Next**.

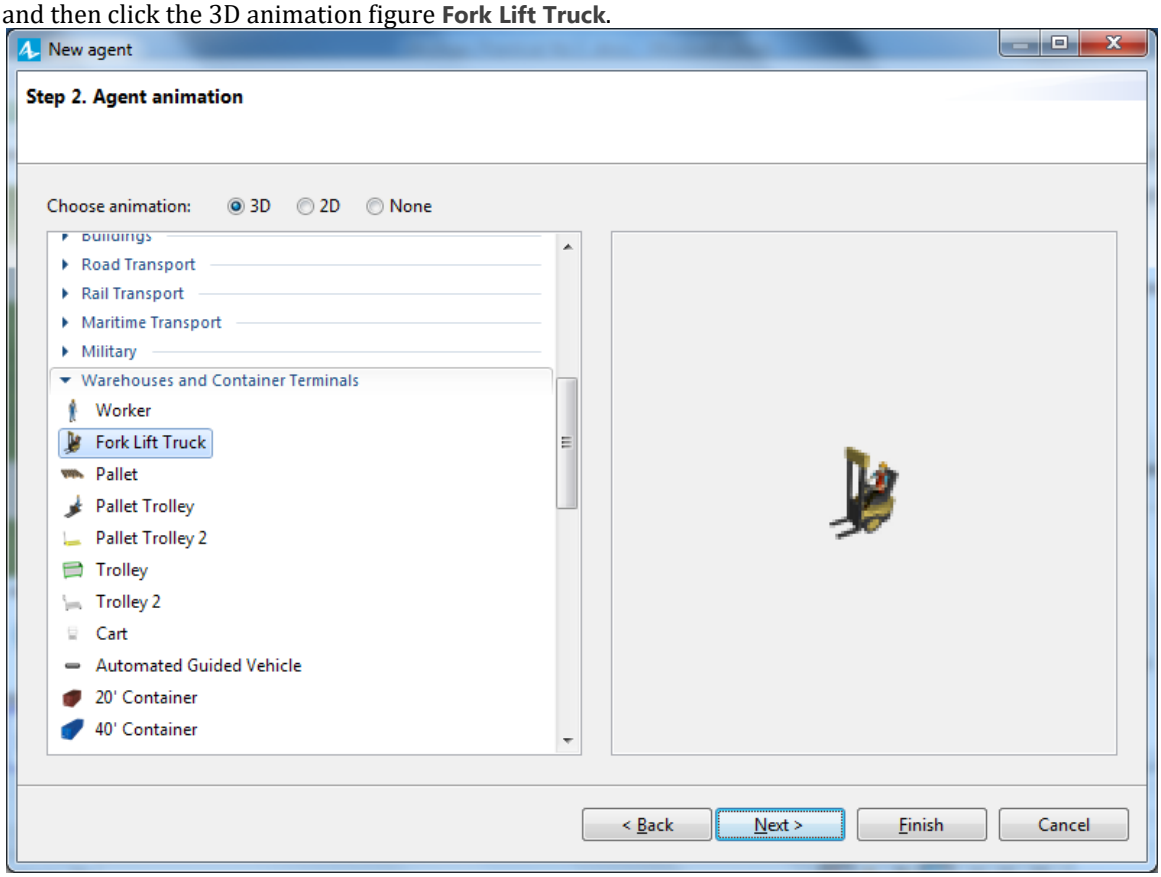



expand the

**Warehouses and Container Terminals** area,

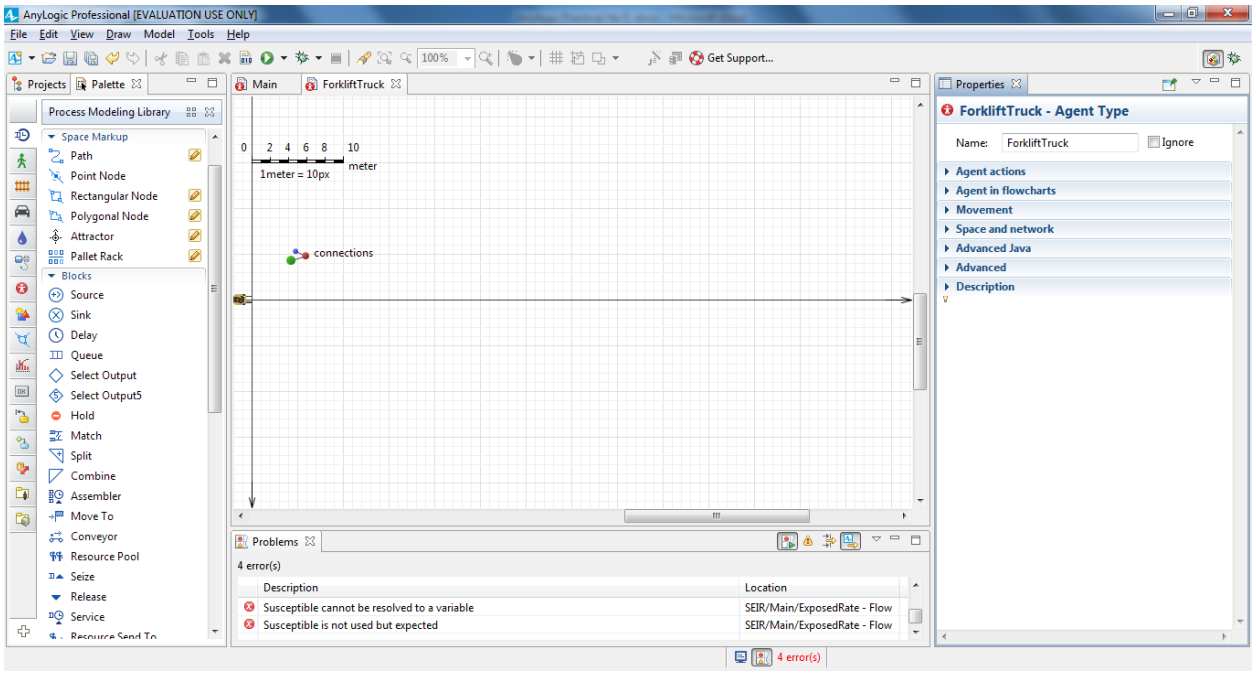



and then click the 3D animation figure **Fork Lift Truck**.

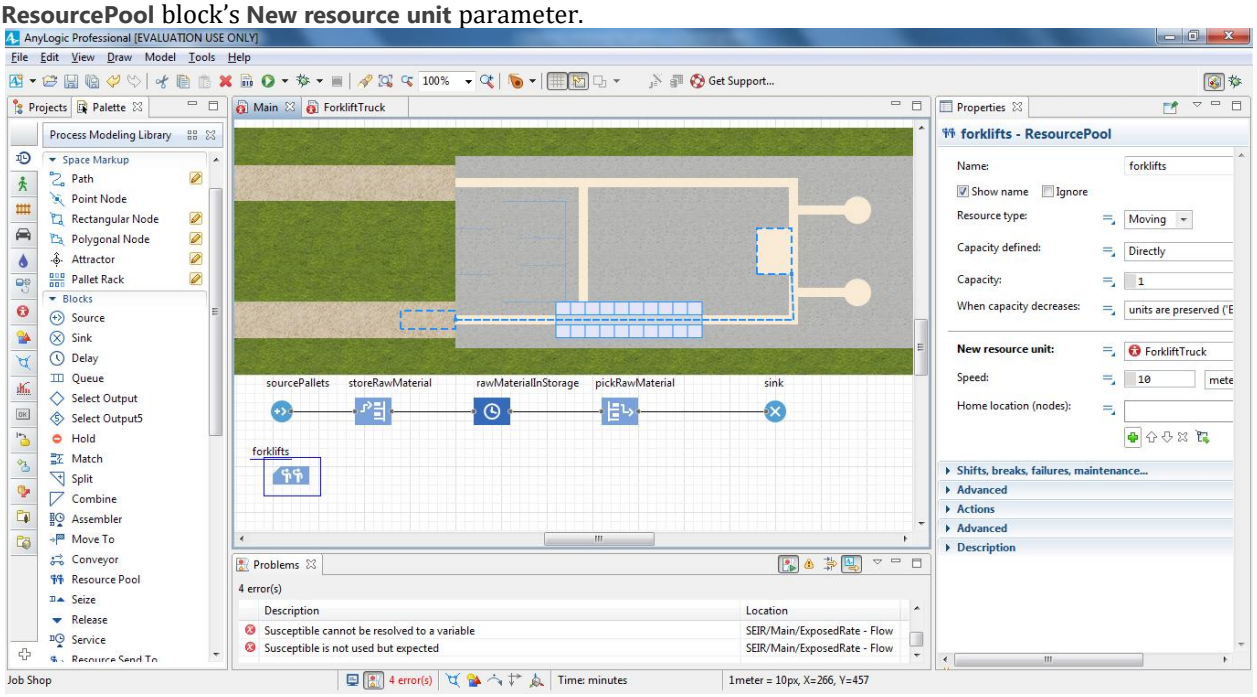



Click **Finish**.

The ForkliftTruck agent type diagram will open and display the animation shape you selected in the wizard.

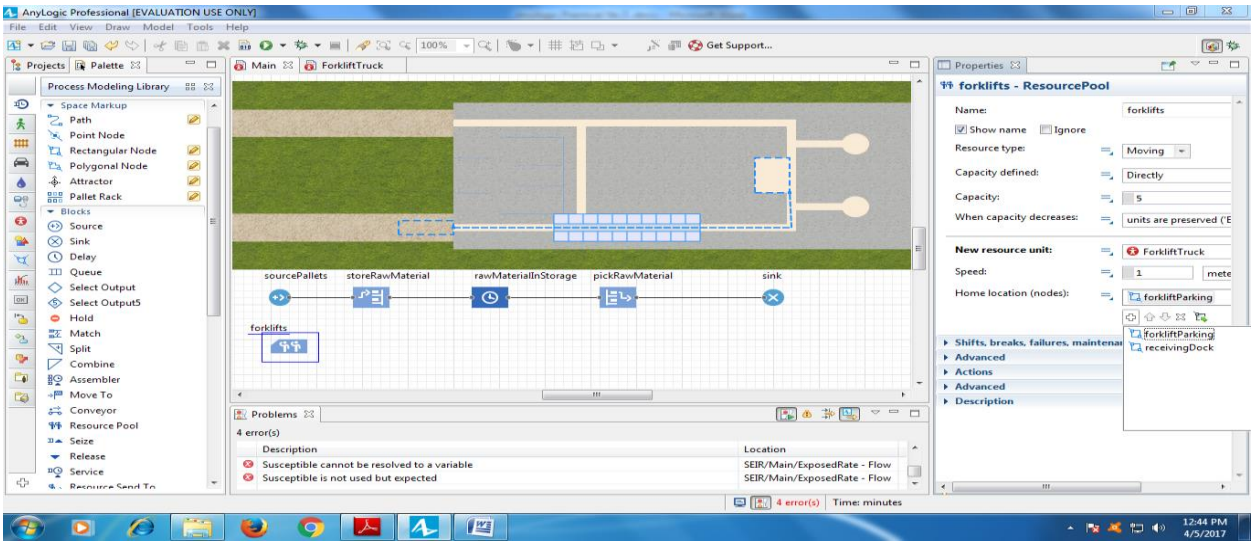



You'll see the ForkliftTruck resource type has been selected in the **ResourcePool** block's **New resource unit** parameter.

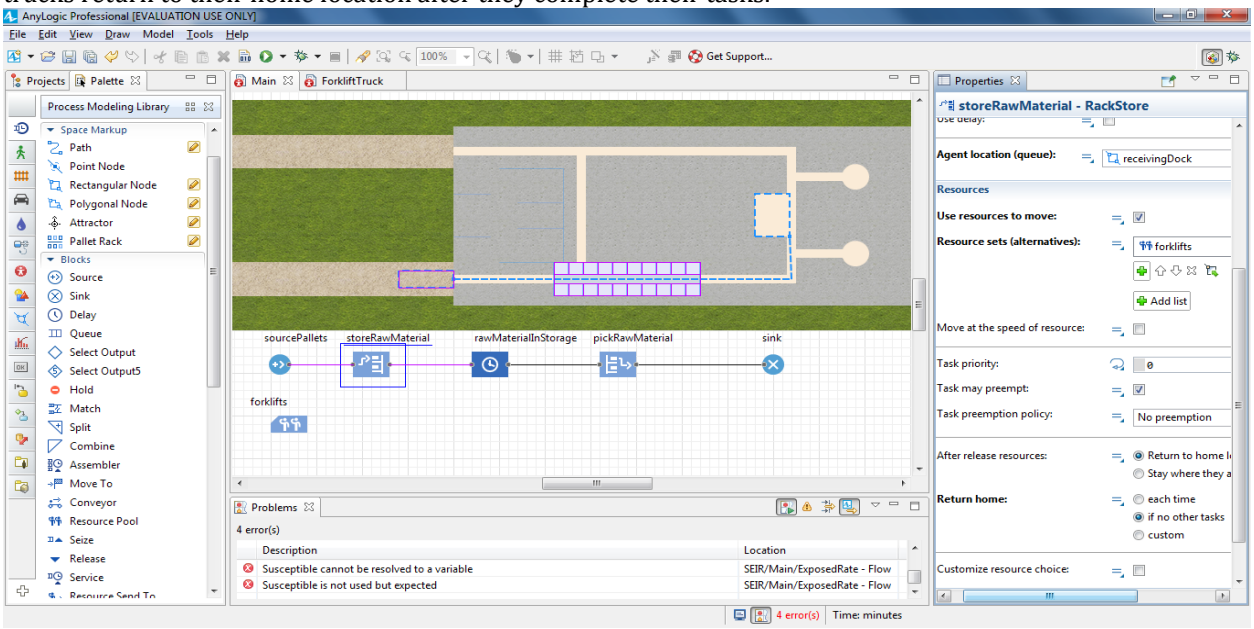



Modify the forklifts resource type's other parameters:
In the **Capacity** box, type **5** to set the number of forklift trucks in our

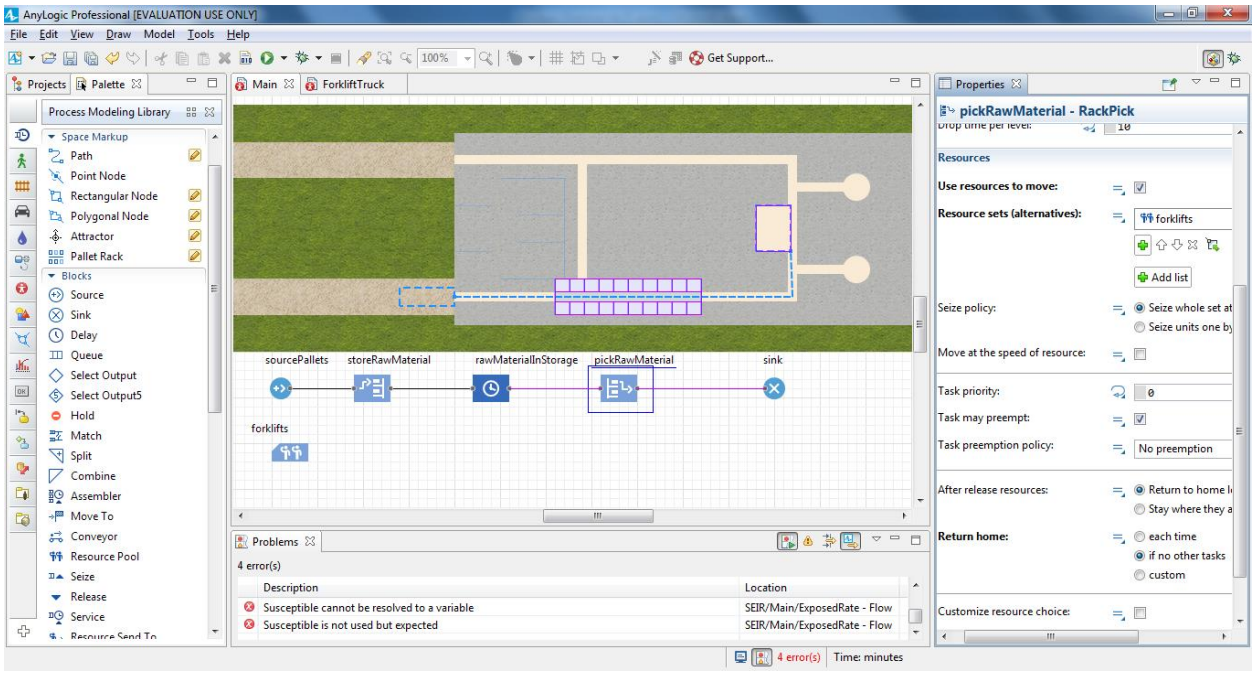
model.
b. In the **Speed** box, type **1** and choose **meters per second** from the list on the right.
c. In the **Home location (nodes)** area, select the forkliftParking node. Click the plus button and then click **forkliftParking** in the list of the model's nodes.



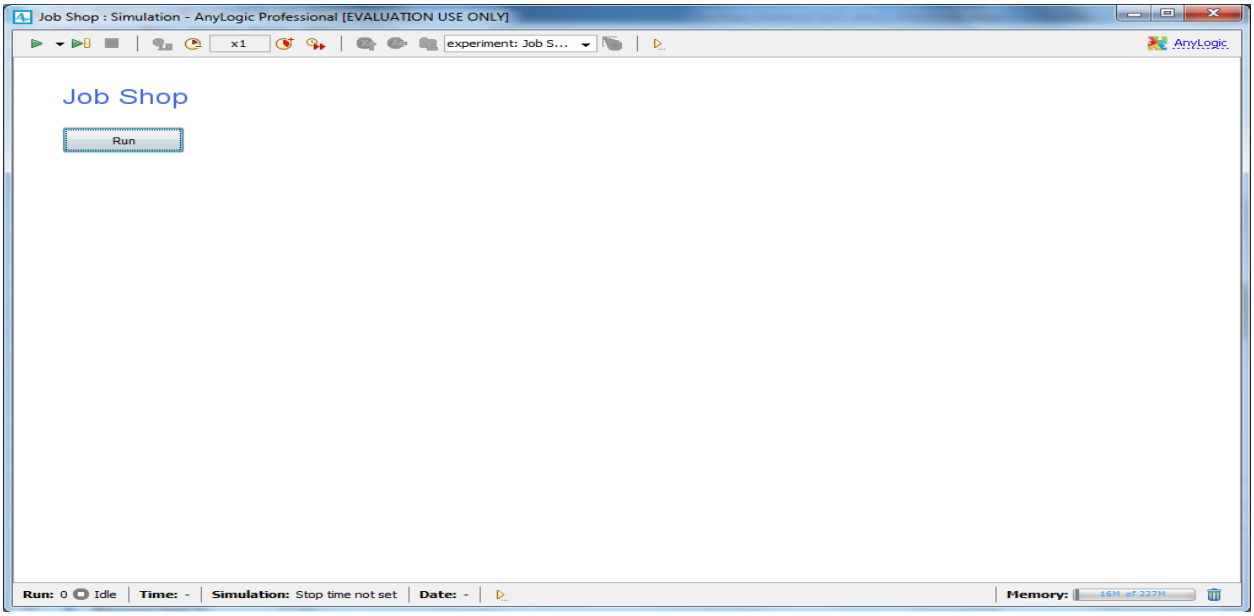

In the storeRawMaterial block's **Properties** area, do the following:
a. Click the arrow to expand the **Resources** area.
b. Select the **Use resources to move** check box.
c. In the **Resource sets (alternatives)** list, select **forklifts** to ensure the flowchart block uses the selected resources -- in our case, the forklift trucks -- to move the agents. Click the plus button and then click **forklifts** in the list of the model's resources.
d. In the **Return home** area, select **if no other tasks** to ensure the forklift trucks return to their home location after they complete their tasks.

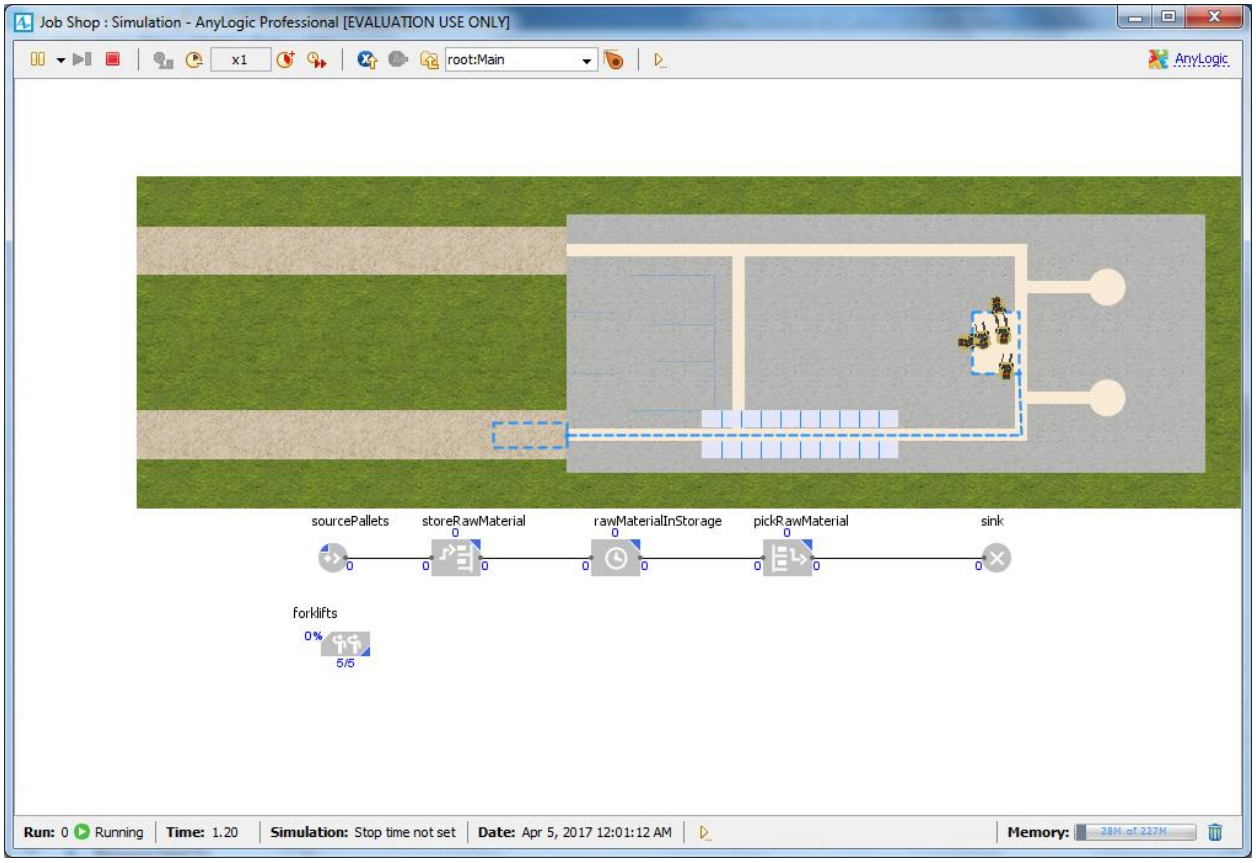


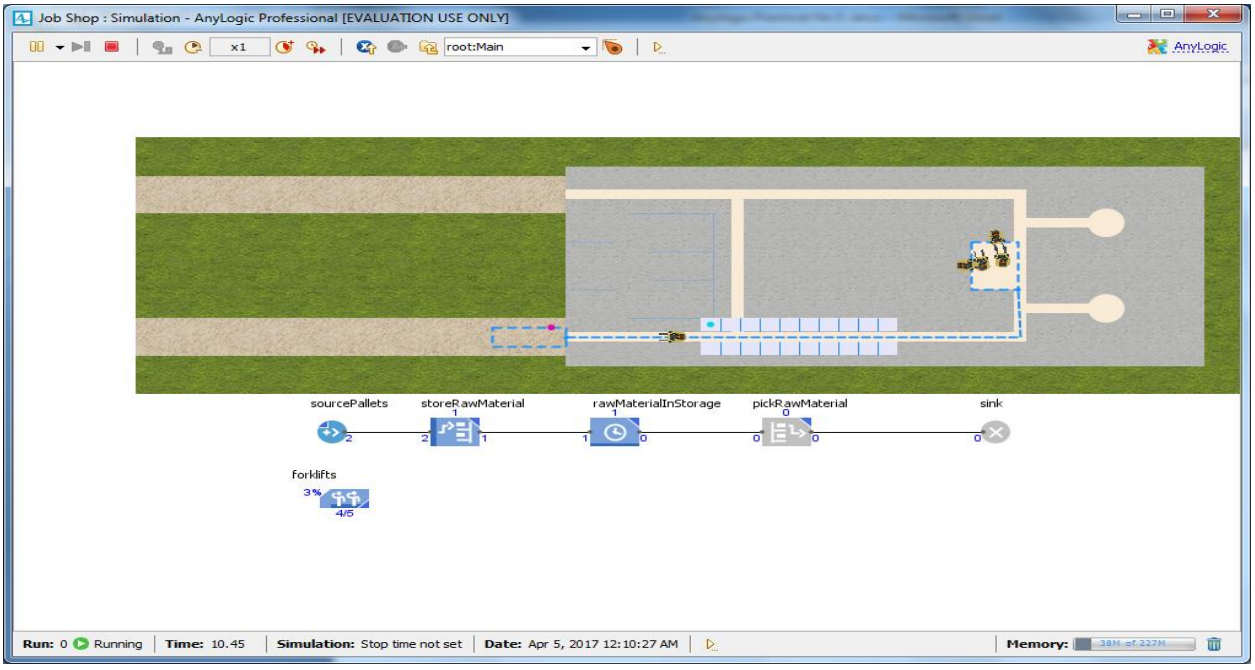
In the pickRawMaterial block's **Properties** area, do the following:
a. Click the arrow to expand the **Resources** area.
b. Select the **Use resources to move** check box.
c. In the **Resource sets (alternatives)** list, click **forklifts** to ensure the flowchart block uses the forklift trucks to move the agents.
d. In the **Return home** area, click **if no other tasks** to ensure the forklift trucks return to their home location after they complete their tasks.

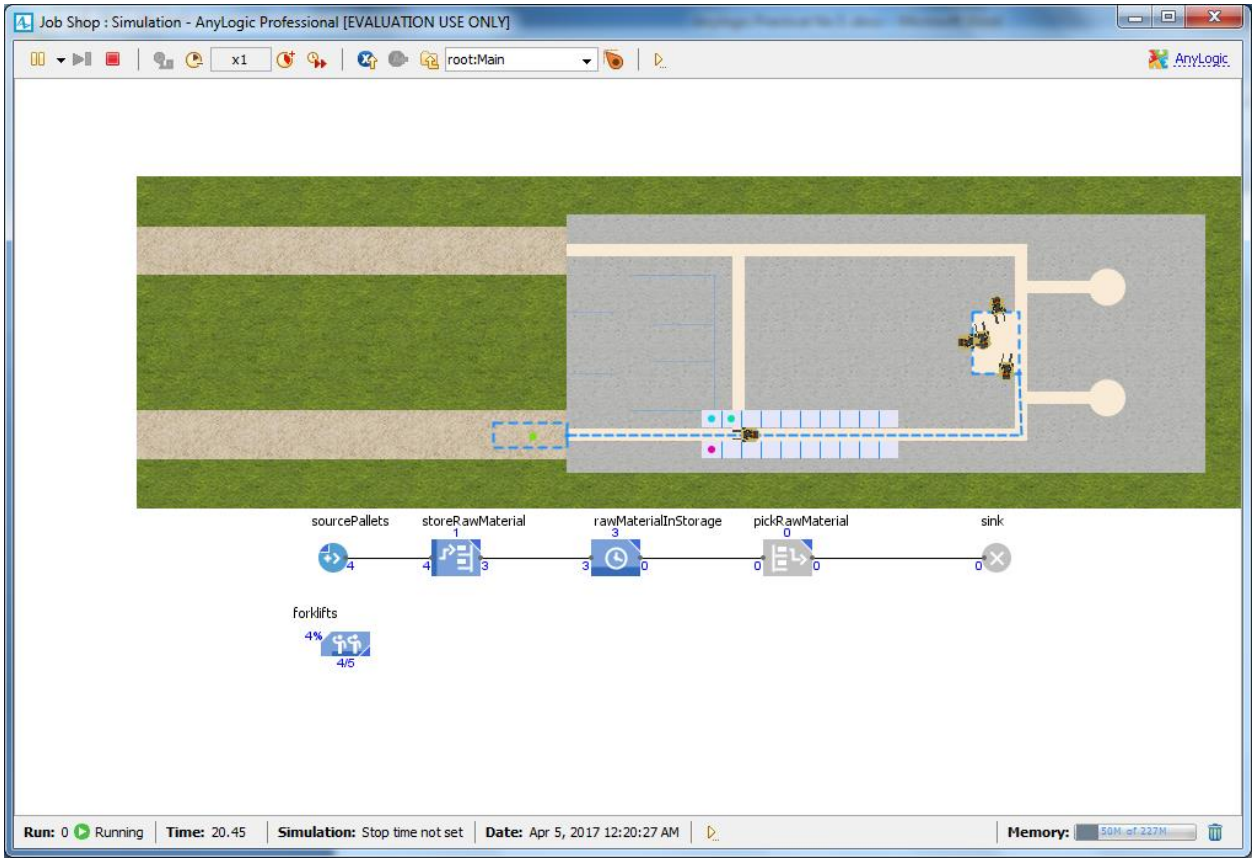


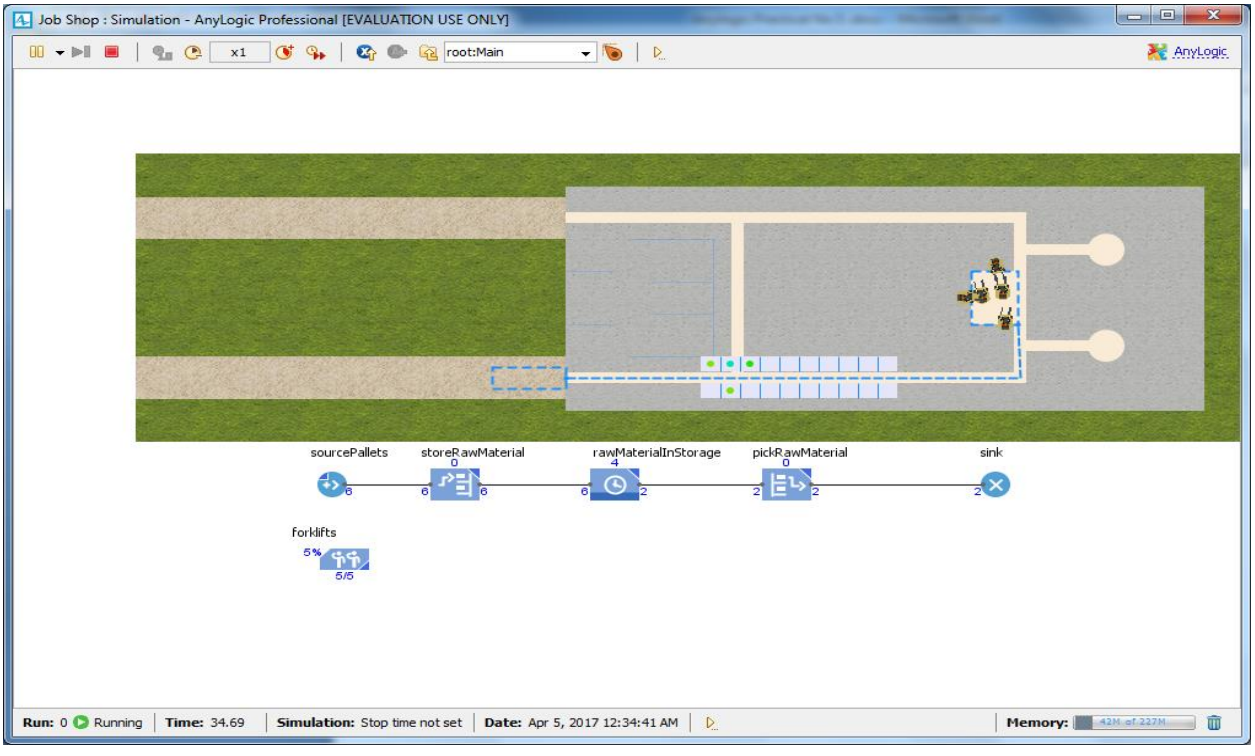

Run the model.

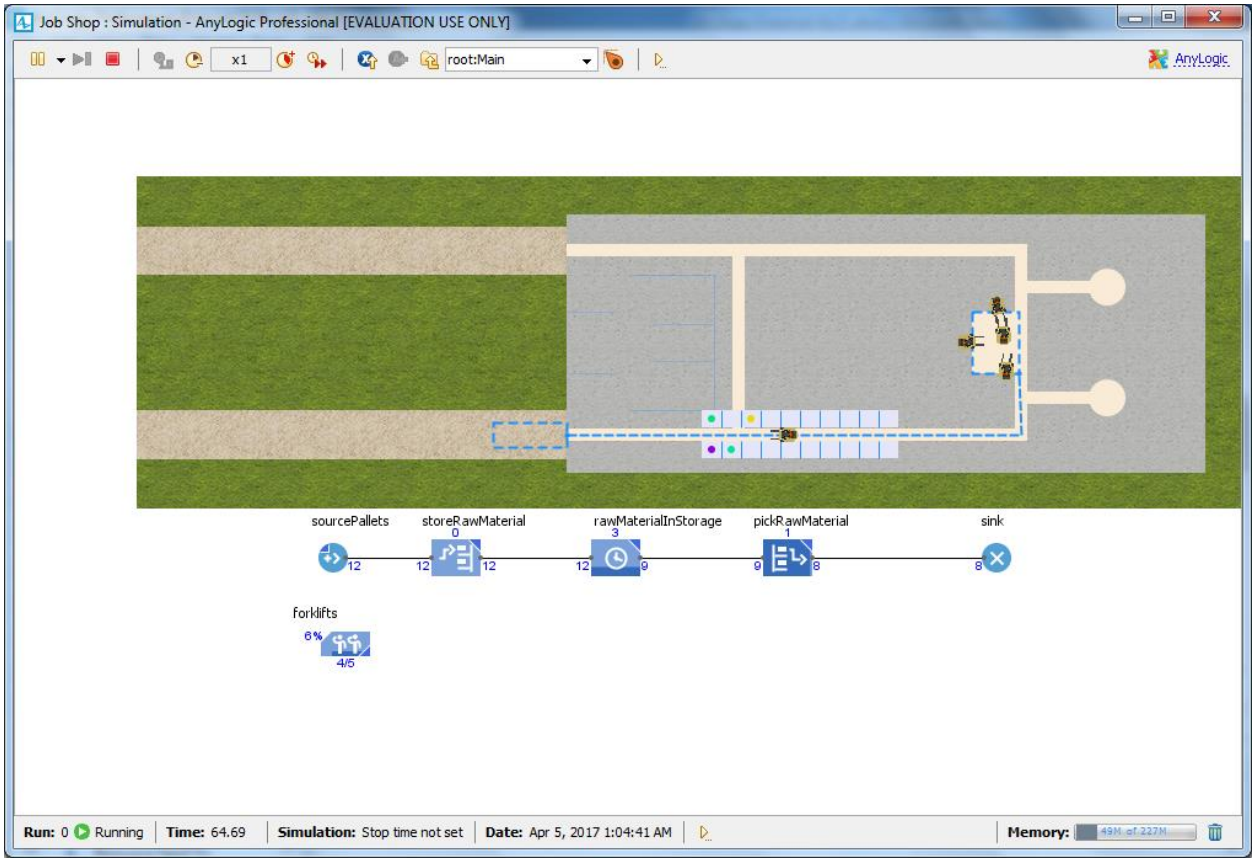


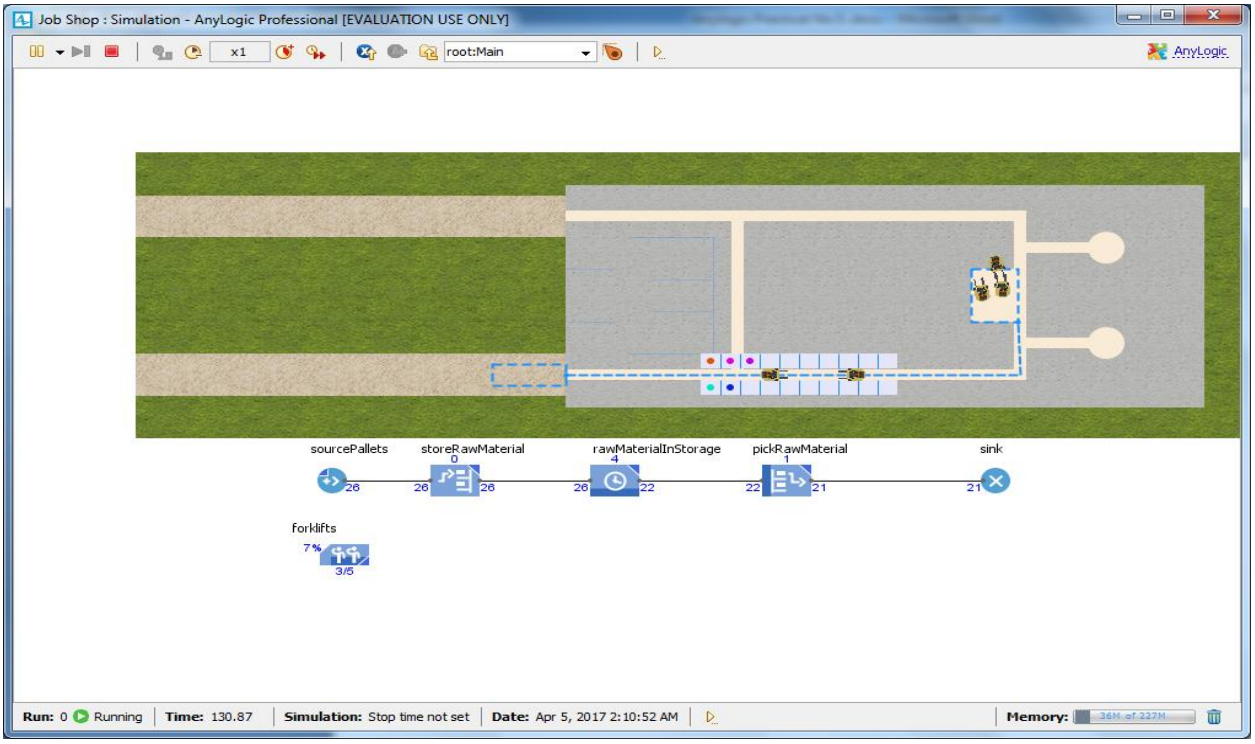

Job Shop : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x1
root:Main
AnyLogic
sourcePallets
storeRawMaterial
rawMaterialInStorage
pickRawMaterial
sink
forklifts
0%
5/5
Run: 0 Running
Time: 1.20
Simulation: Stop time not set
Date: Apr 5, 2017 12:01:12 AM
Memory:


Job Shop : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x1
root:Main
AnyLogic
sourcePallets
storeRawMaterial
rawMaterialInStorage
pickRawMaterial
sink
forklifts
3%
4/5
Run: 0 Running
Time: 10.45
Simulation: Stop time not set
Date: Apr 5, 2017 12:10:27 AM
Memory:


Job Shop : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x1
root:Main
AnyLogic
sourcePallets
storeRawMaterial
rawMaterialInStorage
pickRawMaterial
sink
forklifts
4%
4/5
Run: 0 Running
Time: 20.45
Simulation: Stop time not set
Date: Apr 5, 2017 12:20:27 AM
Memory:


Job Shop : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x1
root:Main
AnyLogic
sourcePallets
storeRawMaterial
rawMaterialInStorage
pickRawMaterial
sink
forklifts
5%
5/5
Run: 0 Running
Time: 34.69
Simulation: Stop time not set
Date: Apr 5, 2017 12:34:41 AM
Memory:

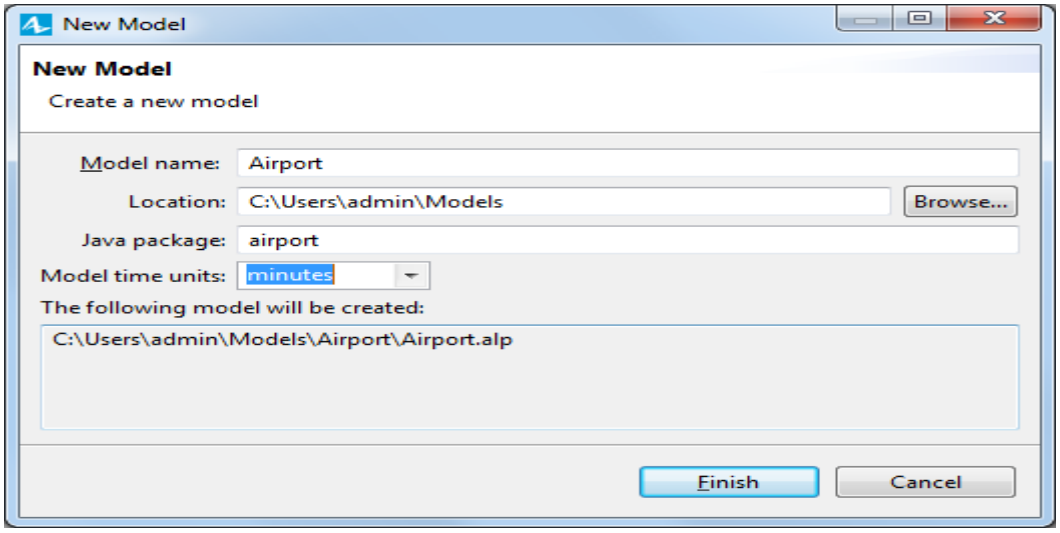

Job Shop : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x1
root:Main
AnyLogic
sourcePallets
storeRawMaterial
0
rawMaterialInStorage
3
pickRawMaterial
1
sink
12
12
12
12
9
9
8
8
forklifts
6%
4/5
Run: 0 Running
Time: 64.69
Simulation: Stop time not set
Date: Apr 5, 2017 1:04:41 AM
Memory:


Job Shop : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x1
root:Main
AnyLogic
sourcePallets
storeRawMaterial
0
rawMaterialInStorage
4
pickRawMaterial
1
sink
26
26
26
26
22
22
21
21
forklifts
7%
3/5
Run: 0 Running
Time: 130.87
Simulation: Stop time not set
Date: Apr 5, 2017 2:10:52 AM
Memory:

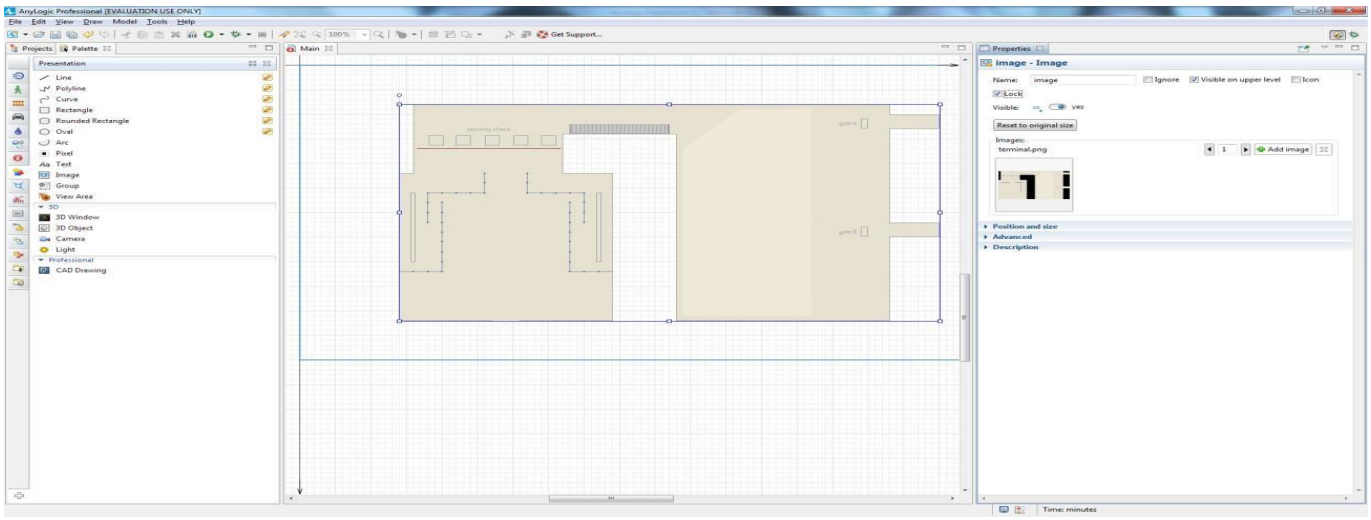
# Practical No.6

Design and develop time-slice simulation for a scenario like airport model to design how passengers move within a small airport that hosts two airlines, each with their own gate. Passengers arrive at the airport, check in, pass the security checkpoint and then go to the waiting area. After boarding starts, each airline's representatives check their passengers' tickets before they allow them to board.

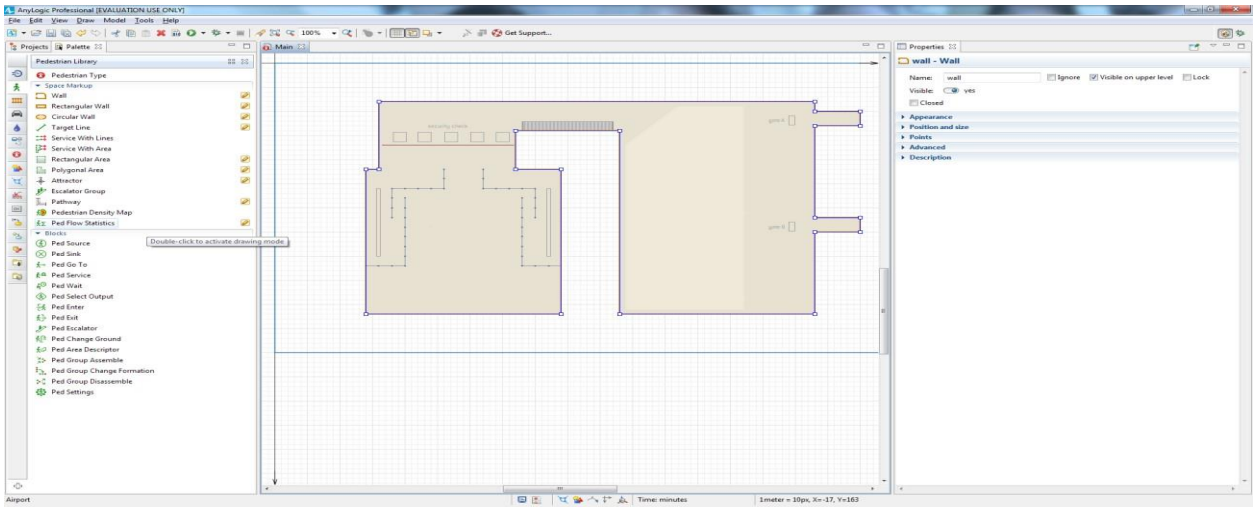
Create a new model and name it Airport. Select minutes as the model time units.



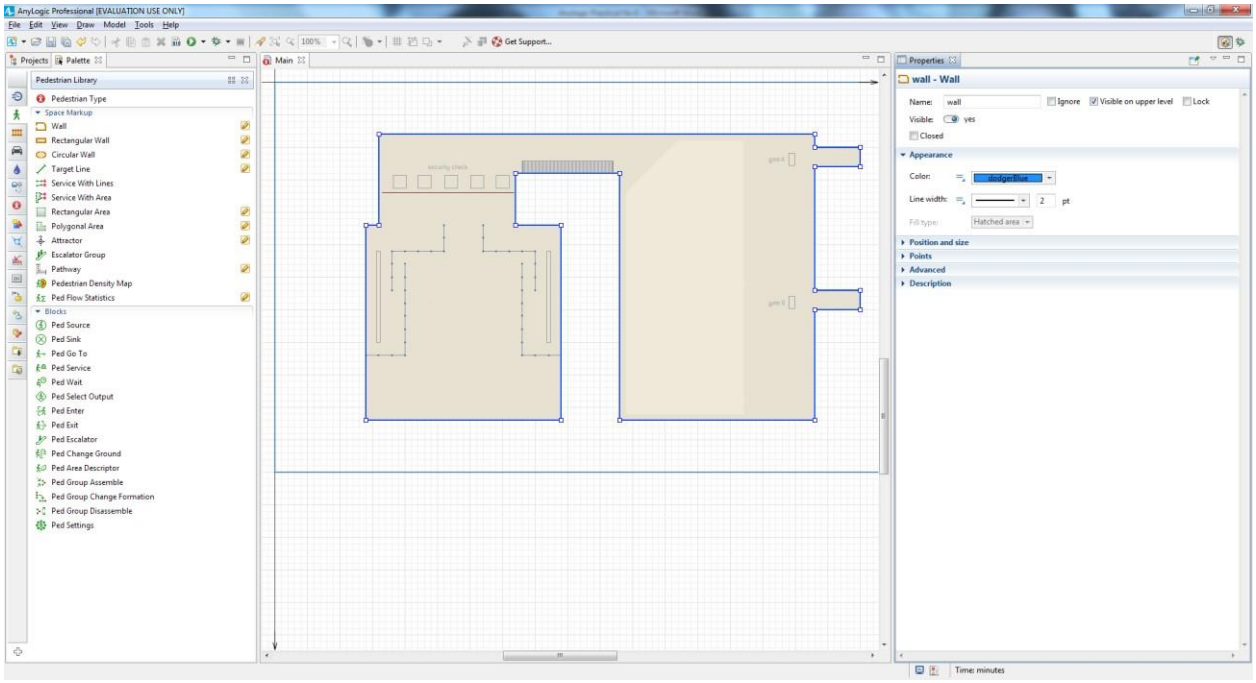

Drag an Image from the Presentation palette on to the Main diagram. Choose the image file you want to display. In this example, you'll select the terminal.png image file from AnyLogic folder/resources/AnyLogic in 3 days/Airport.On the Main diagram, place the image in the blue frame's lower left corner. If the image is distorted, click the Reset to original size button and then select the Lock checkbox to lock the image shape.



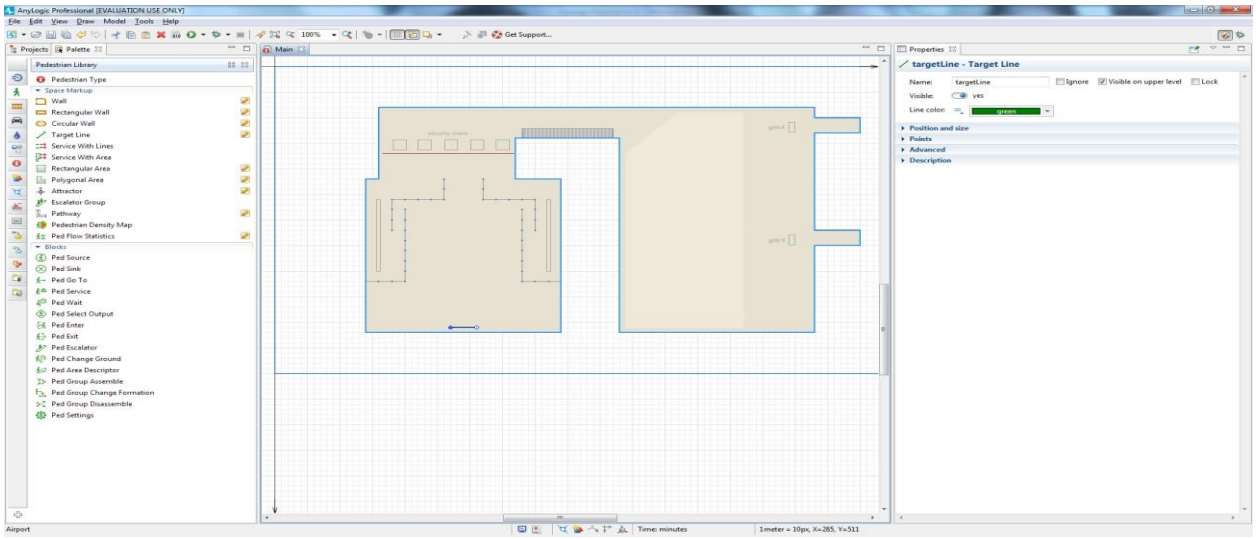
Use the Pedestrian Library palette to draw the airport's walls. Double-click the Wall element you'll find in the Pedestrian Library palette's Space markup section and then draw the wall around the airport building's border
by clicking your mouse each time you want to add a point. When you're ready to set the wall's final point, double-click your mouse.



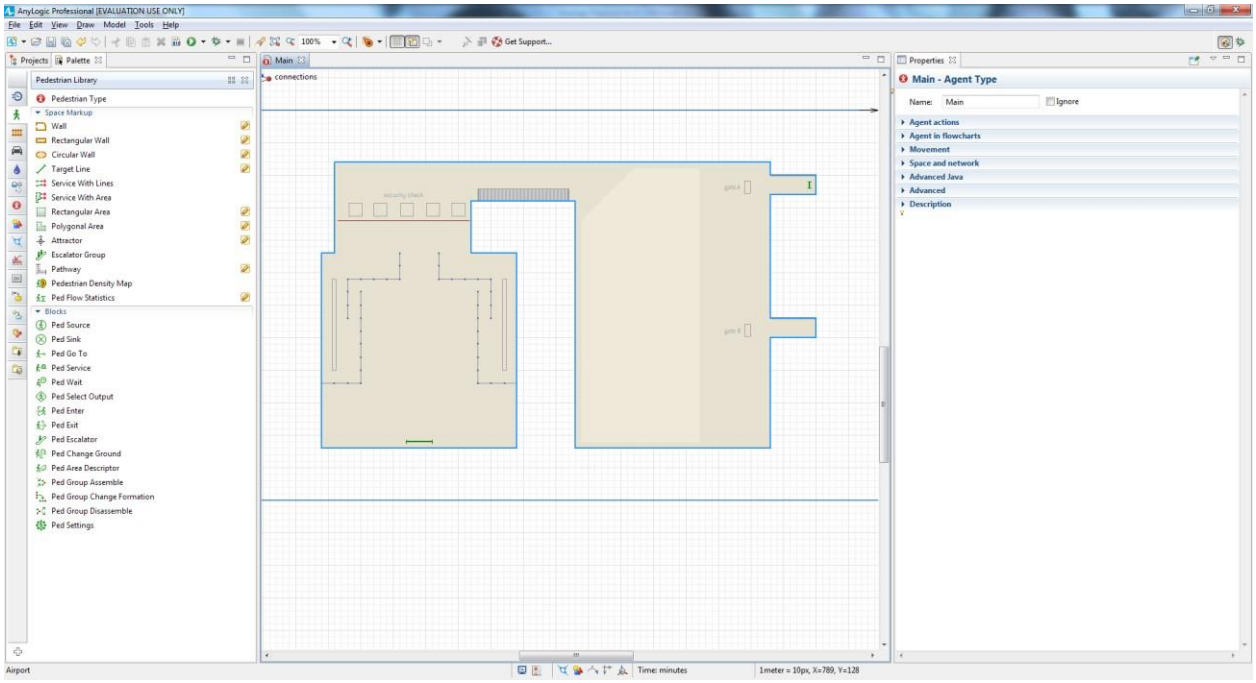
Navigate to the wall's properties and then select the Color: dodgerBlue in the Appearance section.



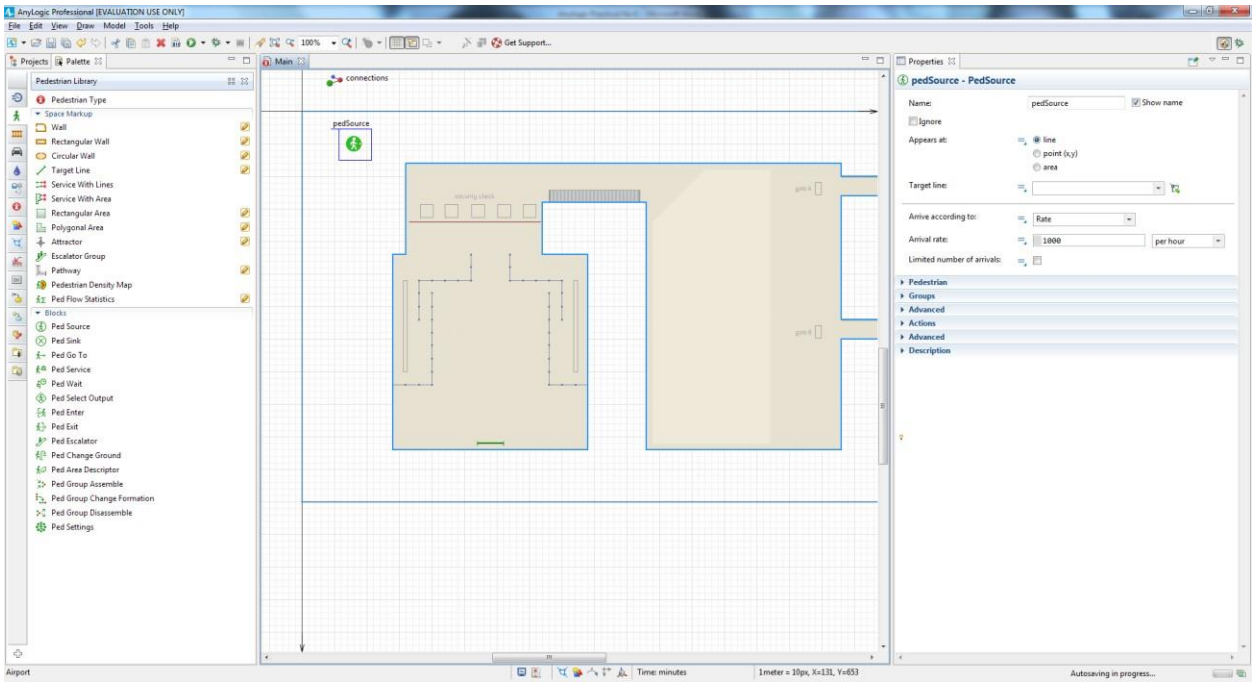
Define the location where your model's passengers appear by dragging the Target Line element from the Pedestrian Library palette on to the graphical diagram, as shown in the figure below.



Define a second target line that passengers will move toward after they enter the airport, place it in the gate area as shown in the figure below, and then name it gateLine1.



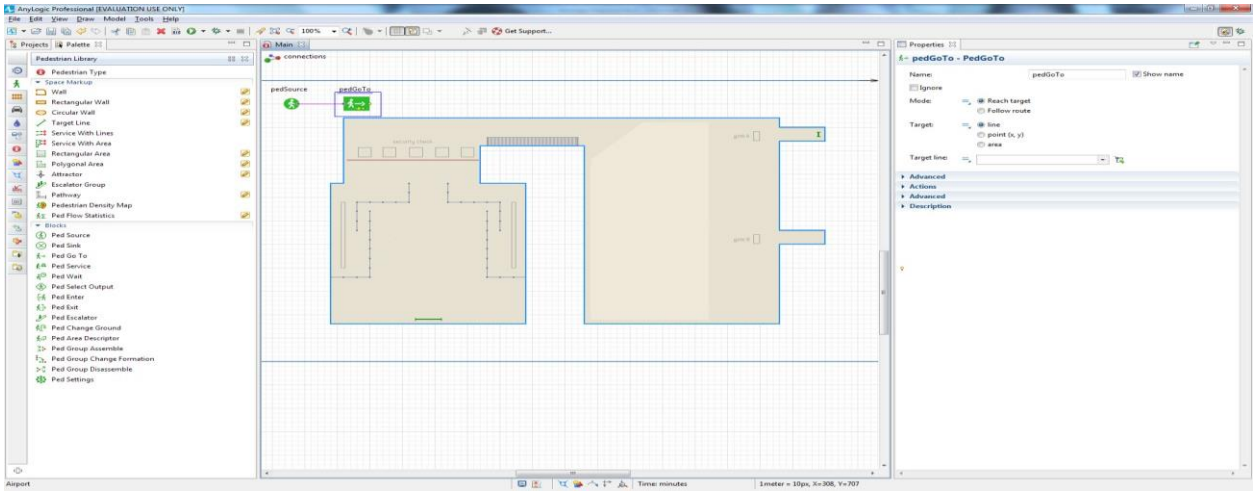
Start by dragging the PedSource block from the Pedestrian Library palette on to our Main diagram.



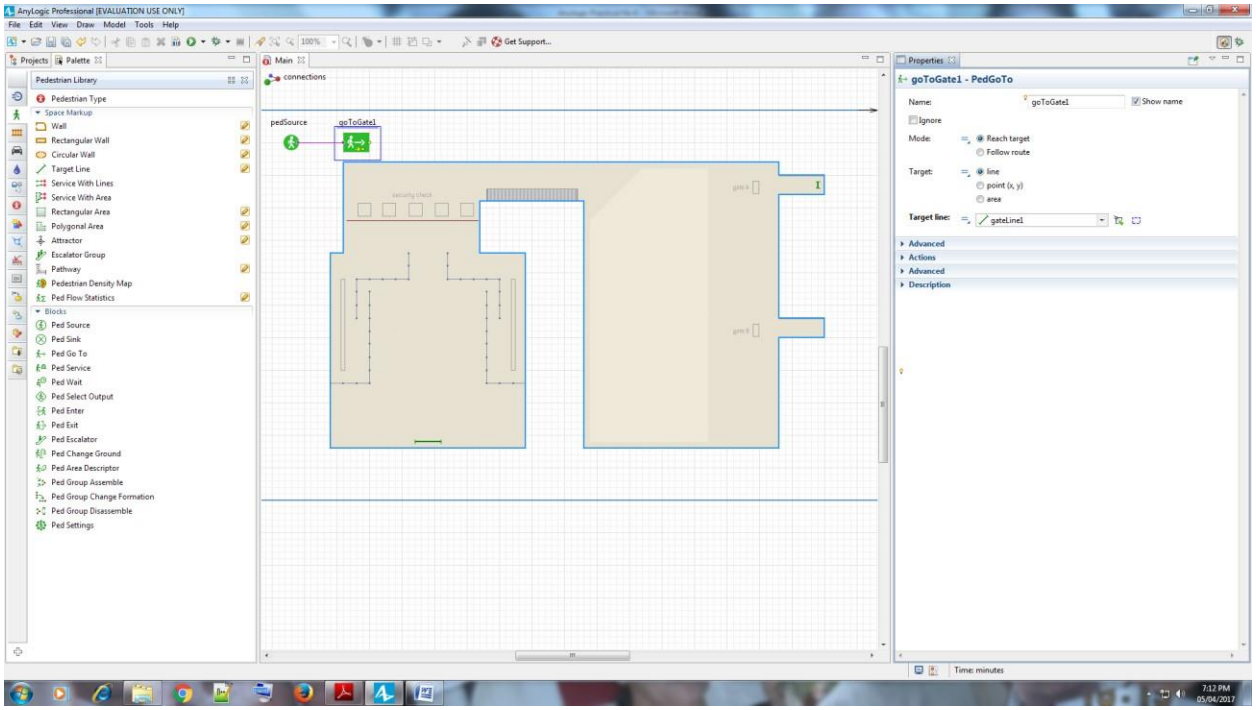

Since we want passengers to arrive randomly at an average rate of 100 passengers per hour, go to the pedSource block properties and then type 100 in the Arrival rate box.

Specify the location where the passengers appear in the simulated system by clicking arrivalLine in the Target line list.

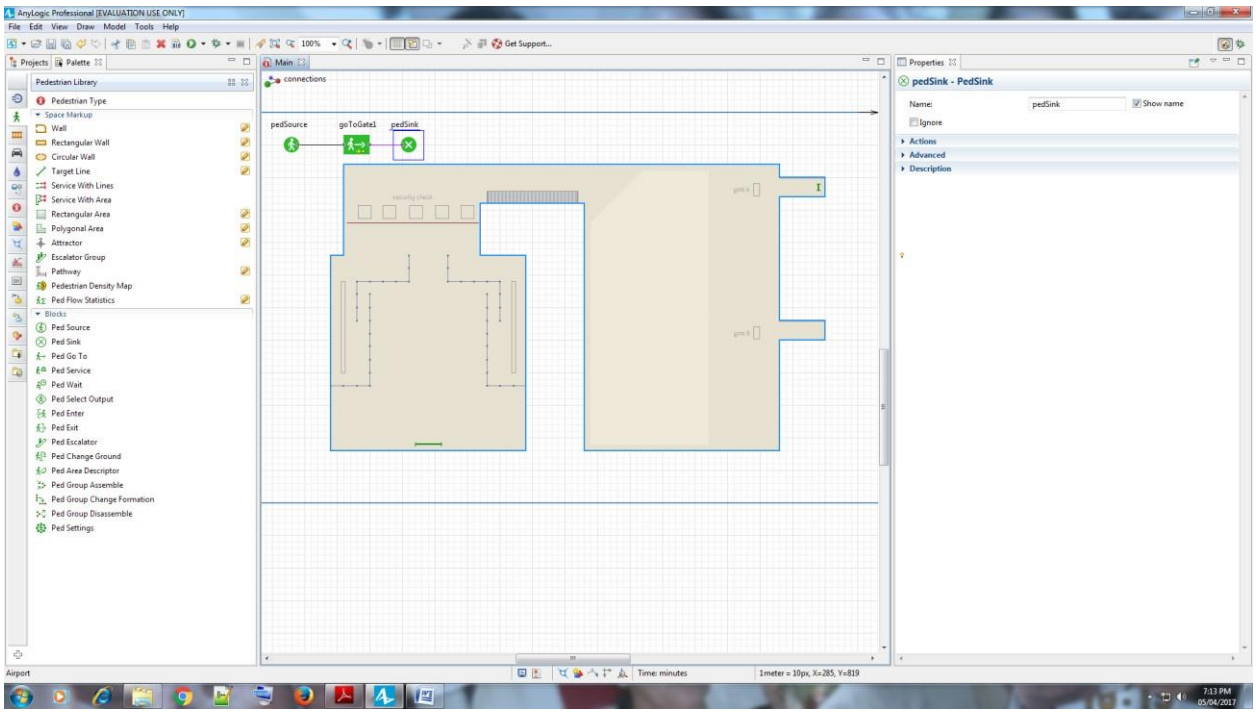
Add a PedGoTo block to simulate pedestrian movement to the specified location and then connect it to pedSource. Since we want our passengers to go to the gate, name the object goToGate1.



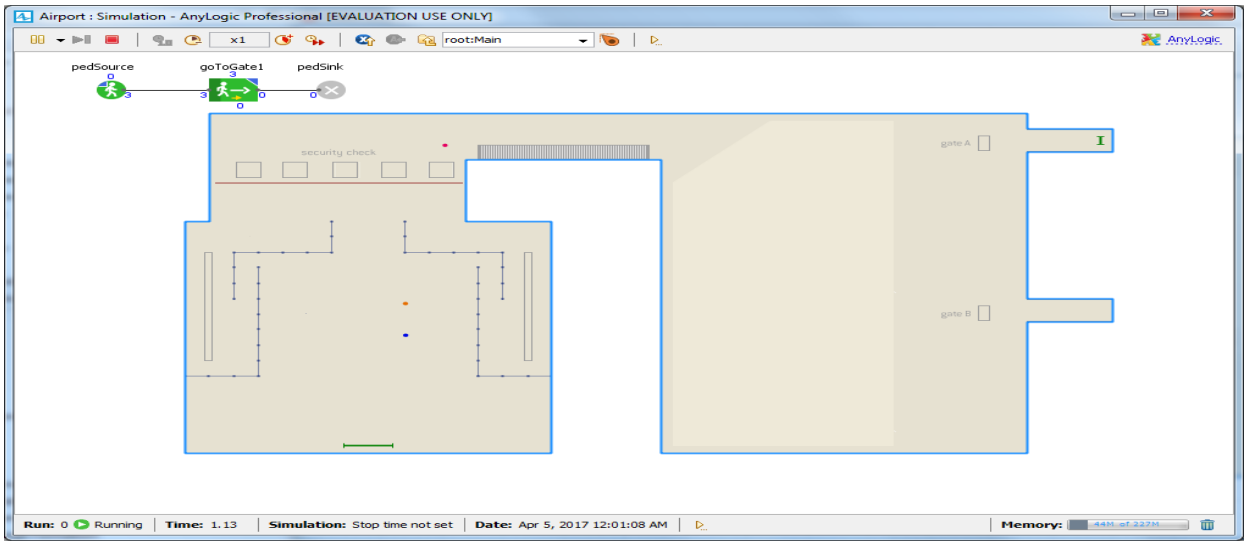

Specify the movement destination by selecting gateLine1 from the Target line combo box.

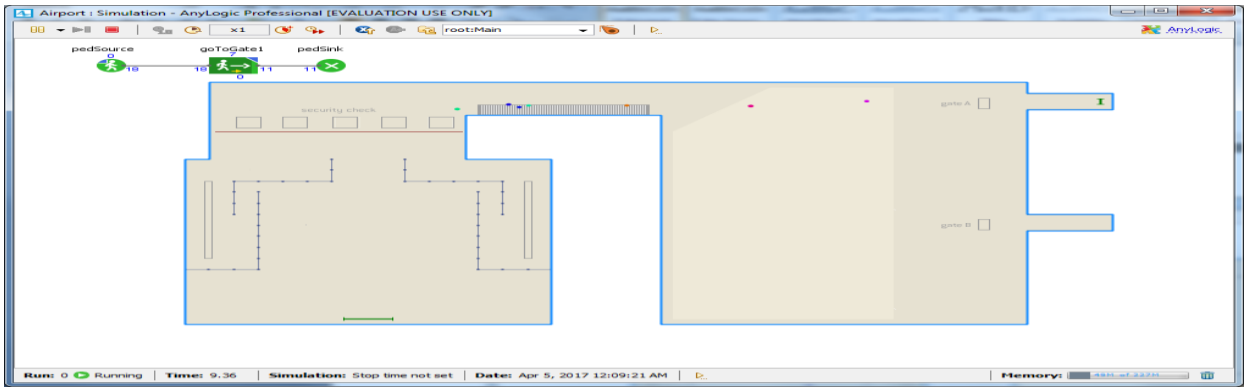



Add a PedSink block to discard incoming pedestrians. Pedestrian flowcharts typically start with a PedSource block and end with a PedSink block.



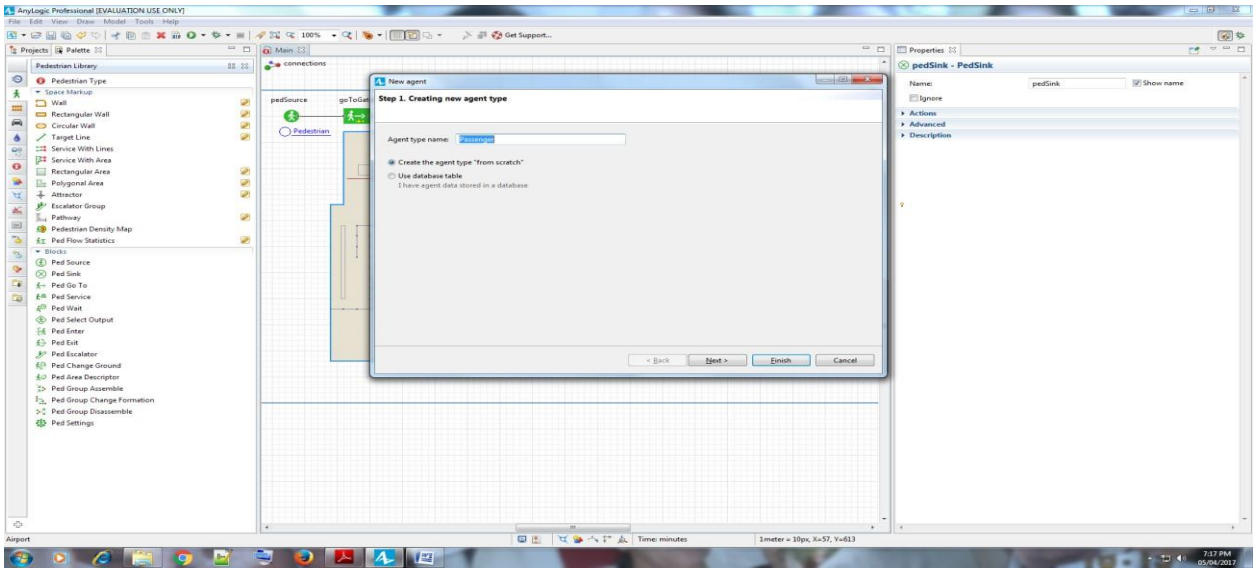

Run the model. In the 2D animation, you'll see the pedestrians move from the airport entrance to the gate.



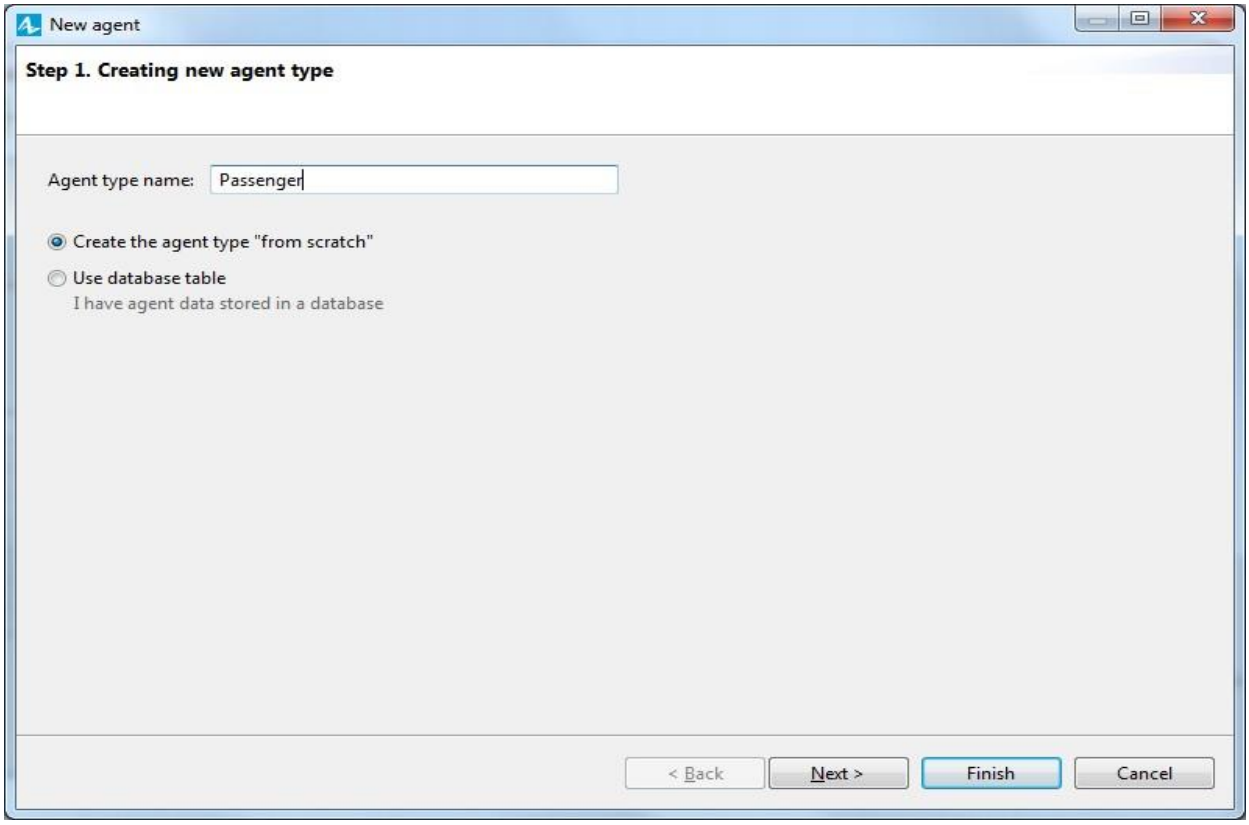




Drawing 3D animation

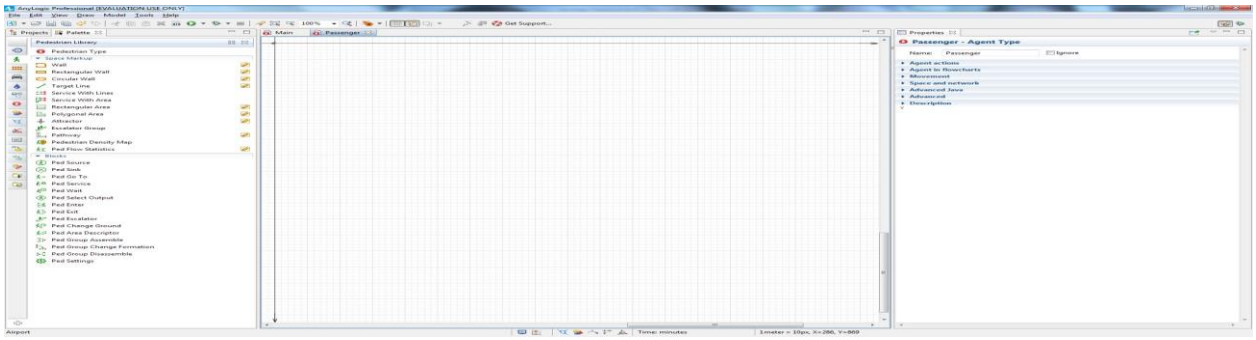
From the Pedestrian Library palette, drag the Pedestrian Type element on to the Main diagram.



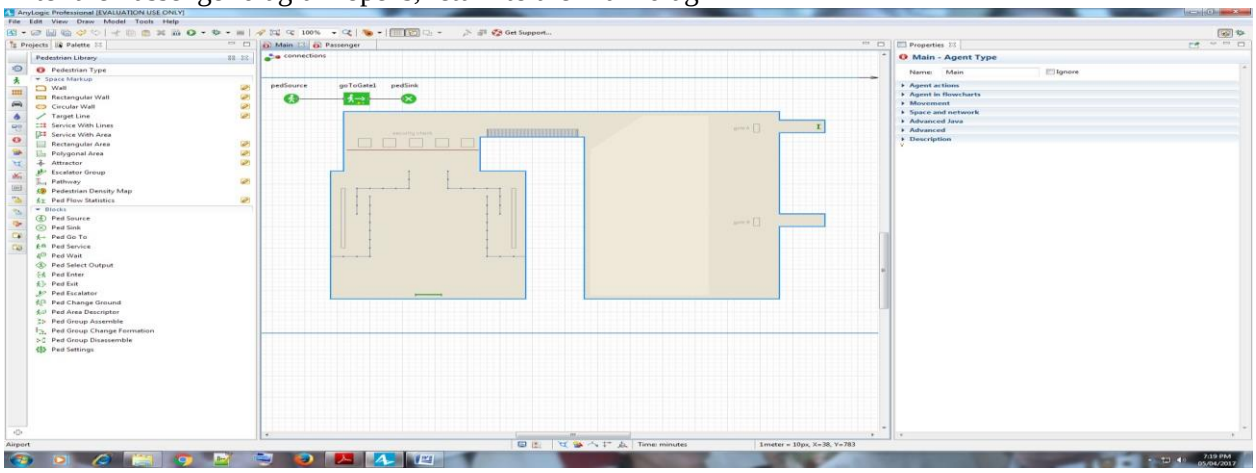

In the New agent wizard, enter the new pedestrian type's name – Passenger – and then click Next.



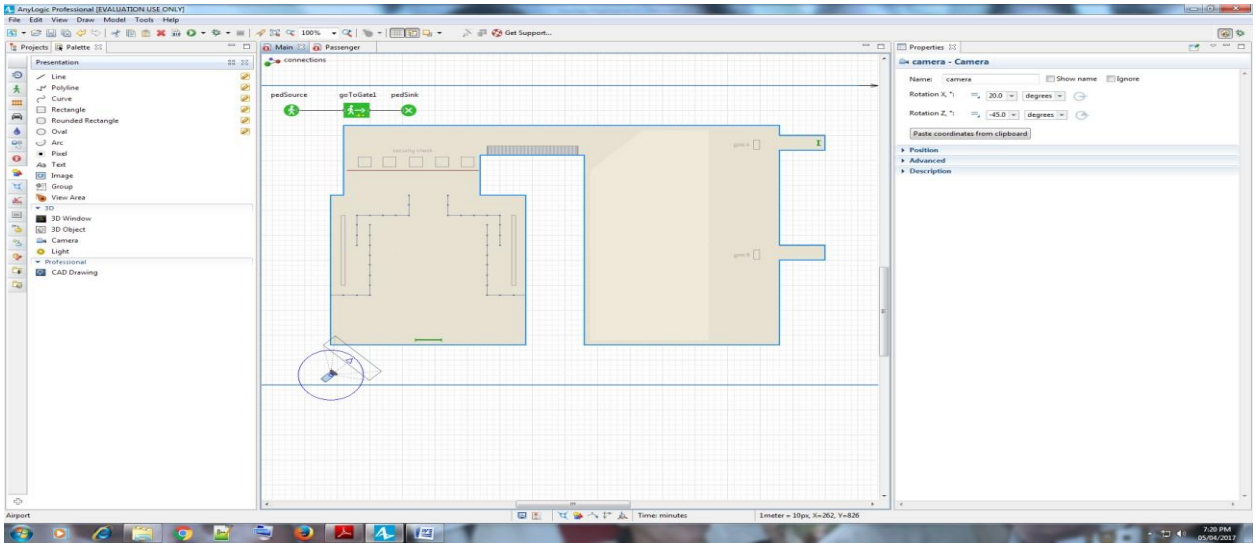

On the Agent animation page, select the General list's first item: Person, and click Finish.



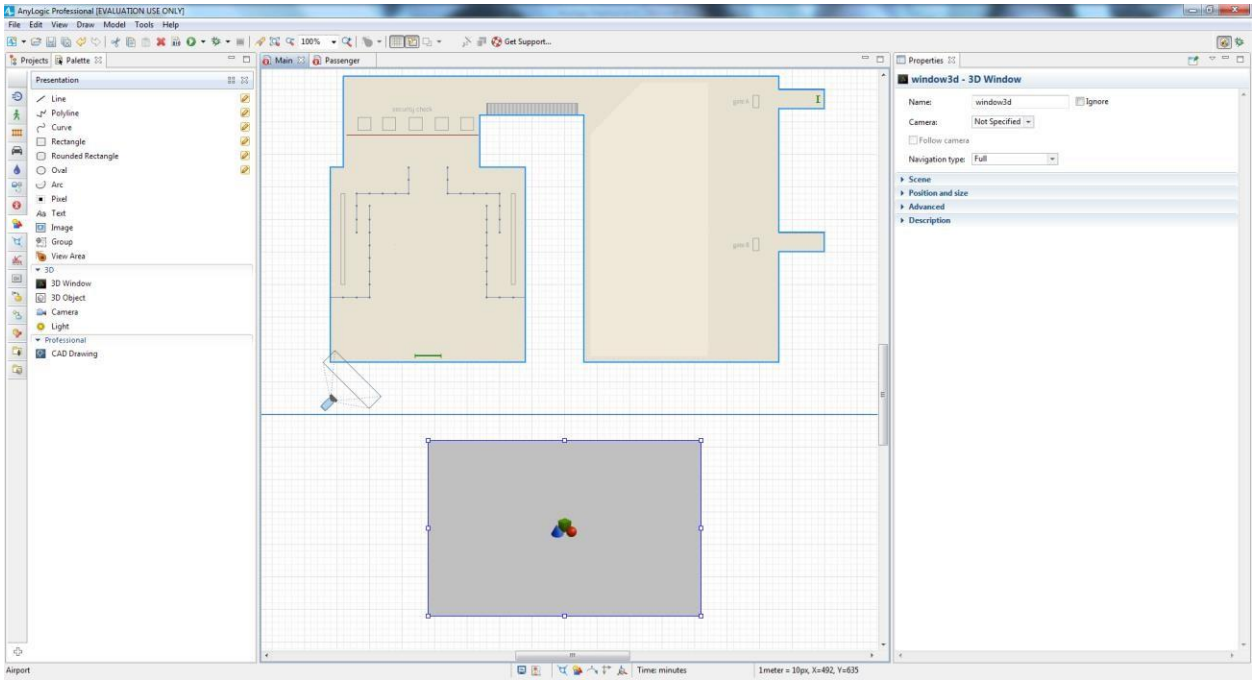
After the Passenger diagram opens, return to the Main diagr



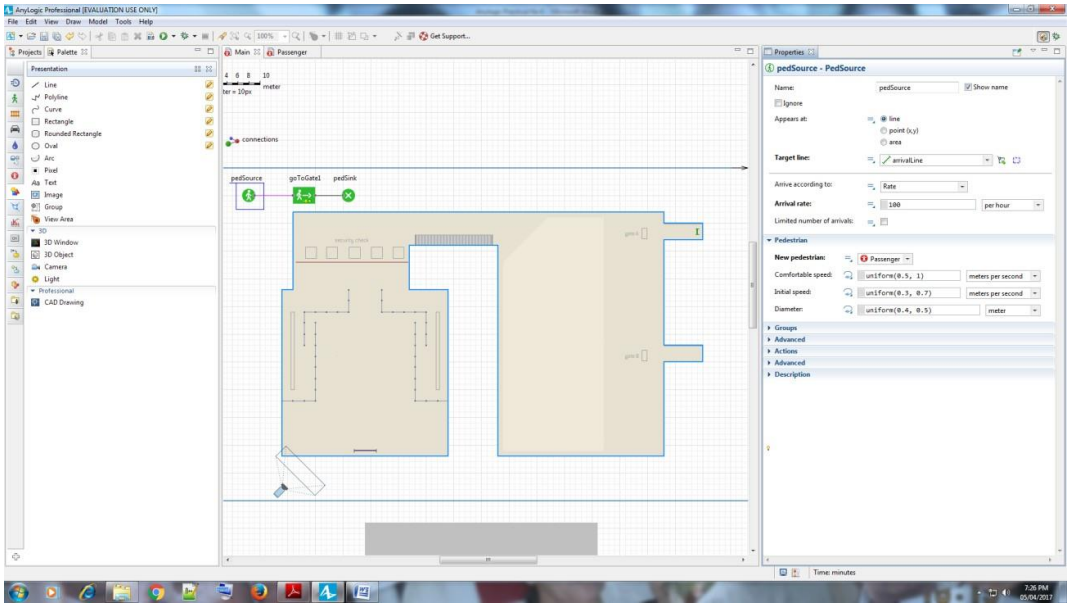
From the Presentation palette, drag the Camera on to the Main diagram and place it so it faces the terminal.



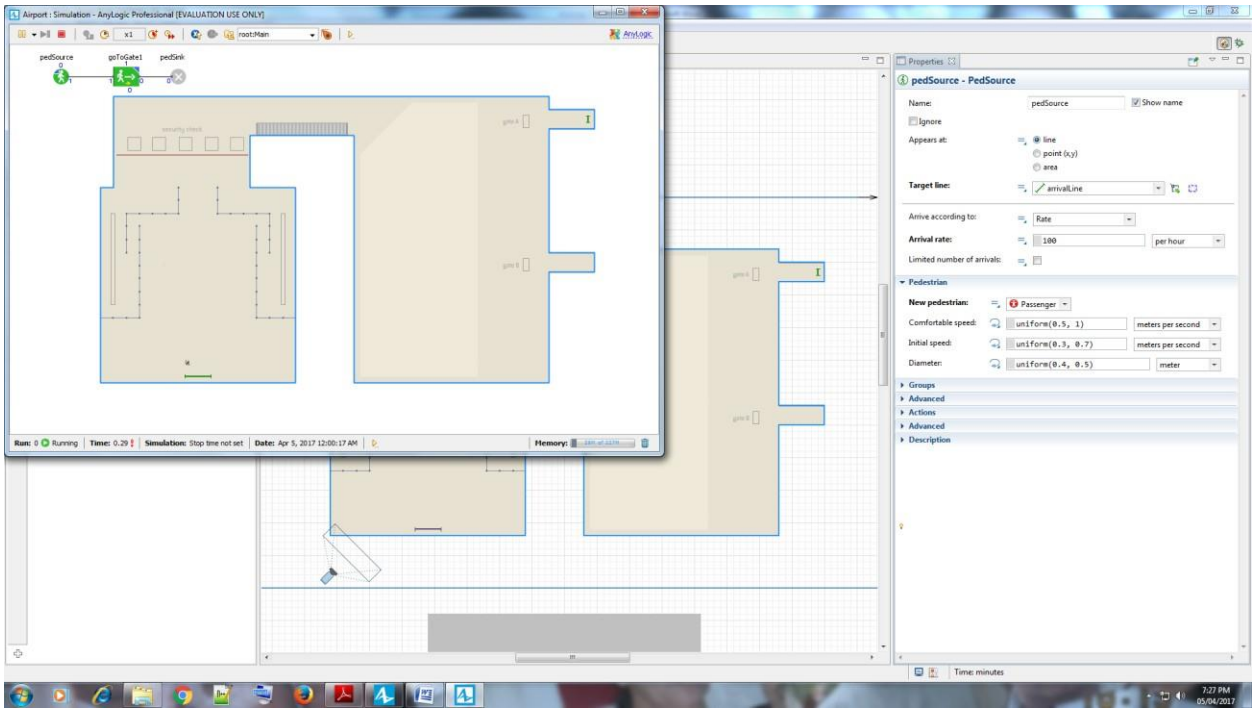

Drag the 3D Window on to the Main diagram and place it below the terminal layout image.

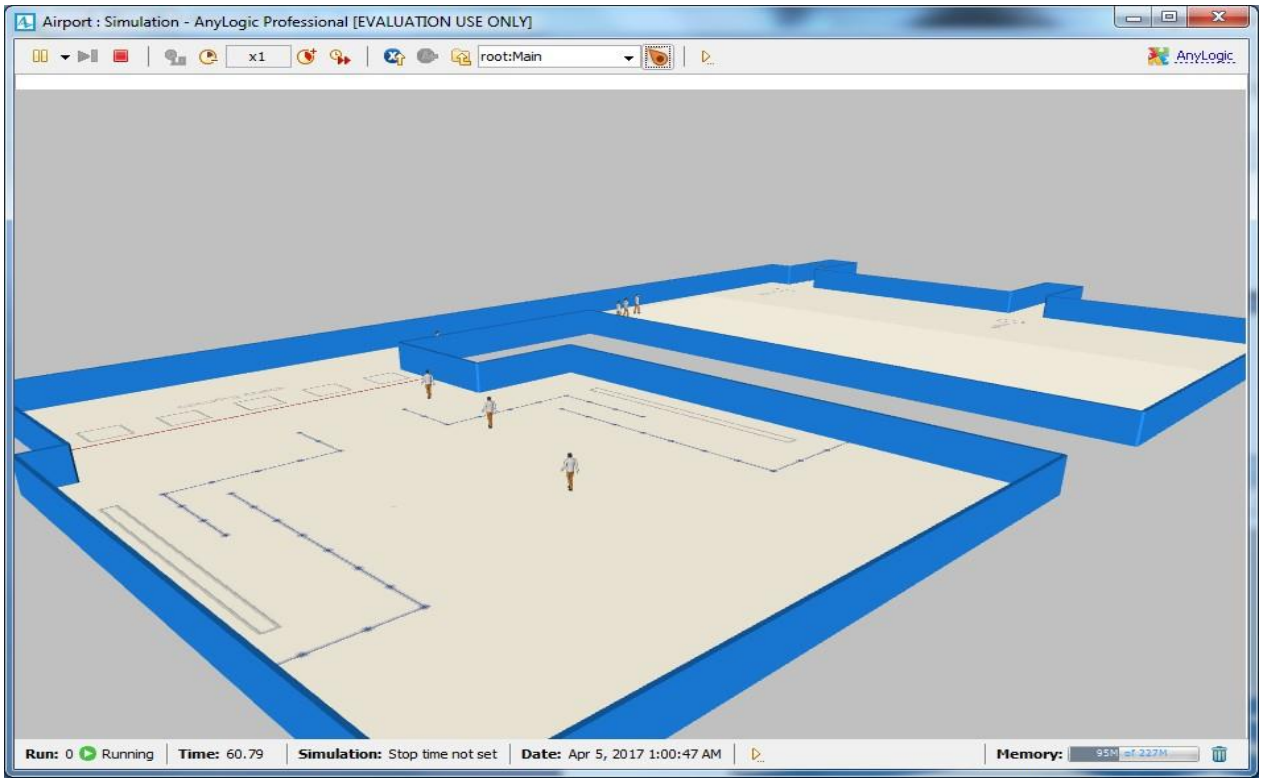



Open the 3D Window properties, and then select camera from the Camera list. We want our flowchart block pedSource to create pedestrians of our custom
Passenger type. Open the pedSource properties, and then select Passenger
from the New pedestrian box in the Pedestrian section.



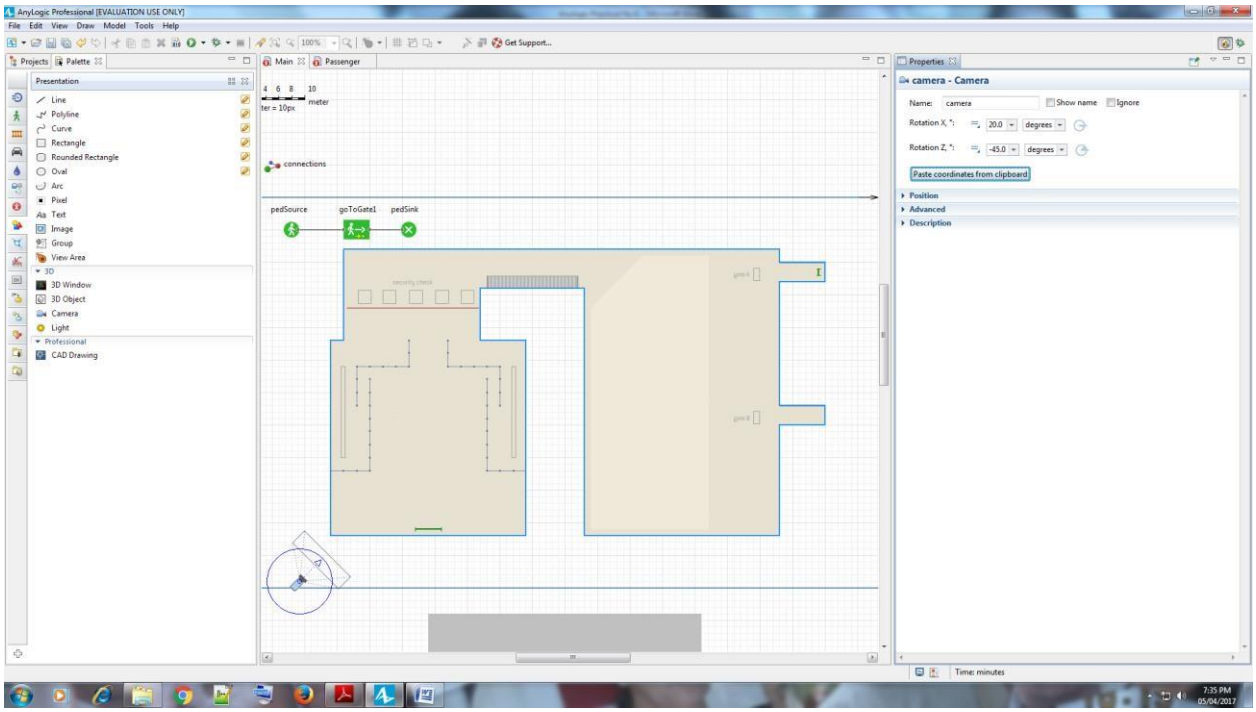
Run the model, and you'll see pedestrians move from the entry to the gate inside the building. You can switch to a 3D view by clicking the toolbar's Navigate to view area… button and then selecting [window3d] from the list.



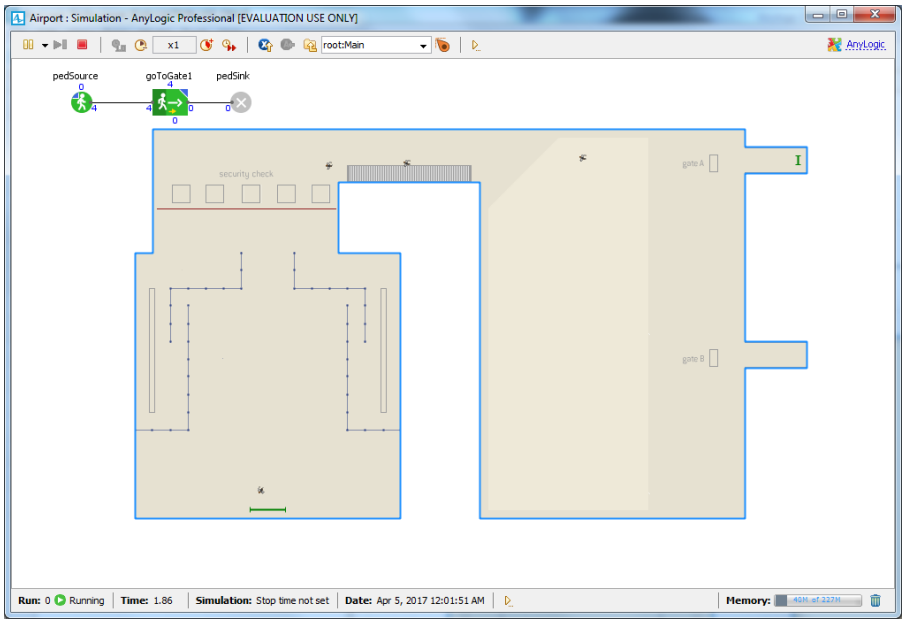


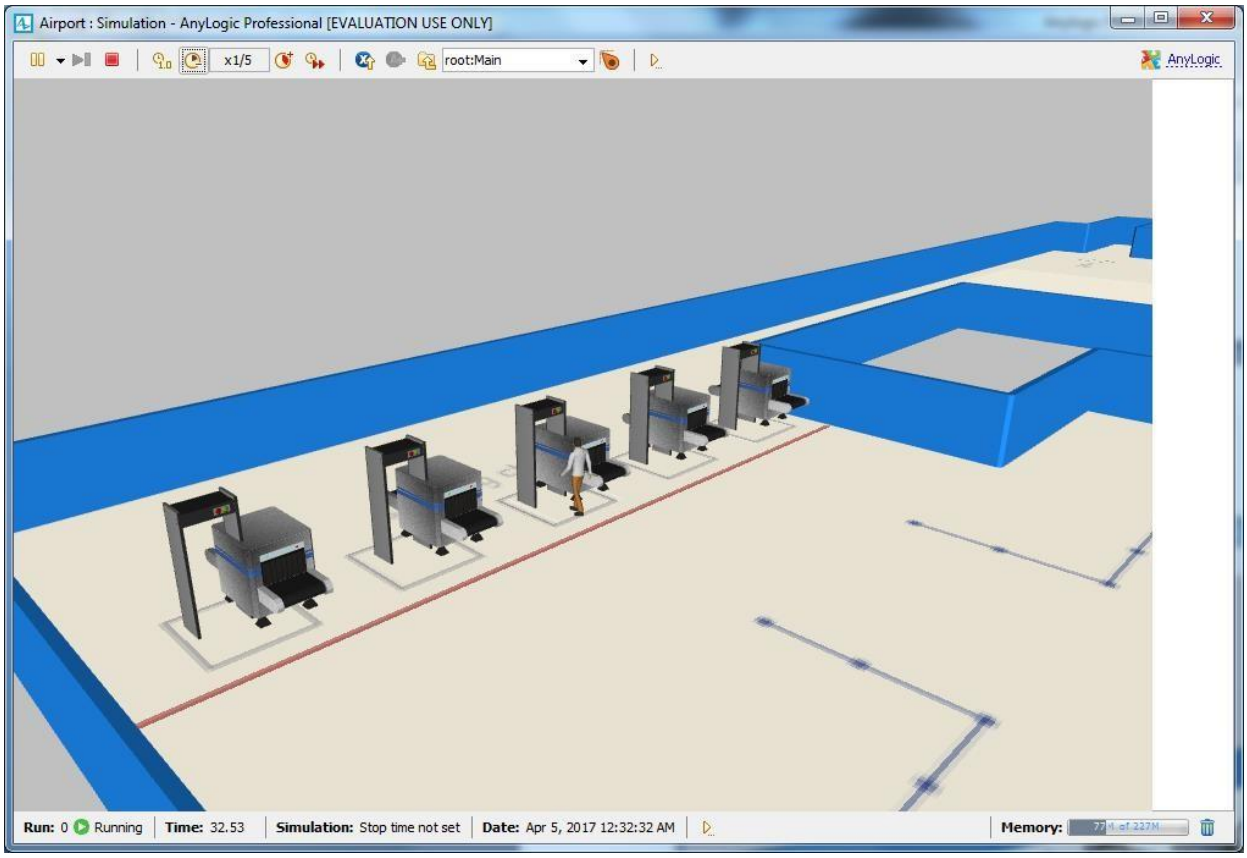

Navigate the scene to get the best view, right-click
inside the 3D scene, and then click Copy the camera's location.

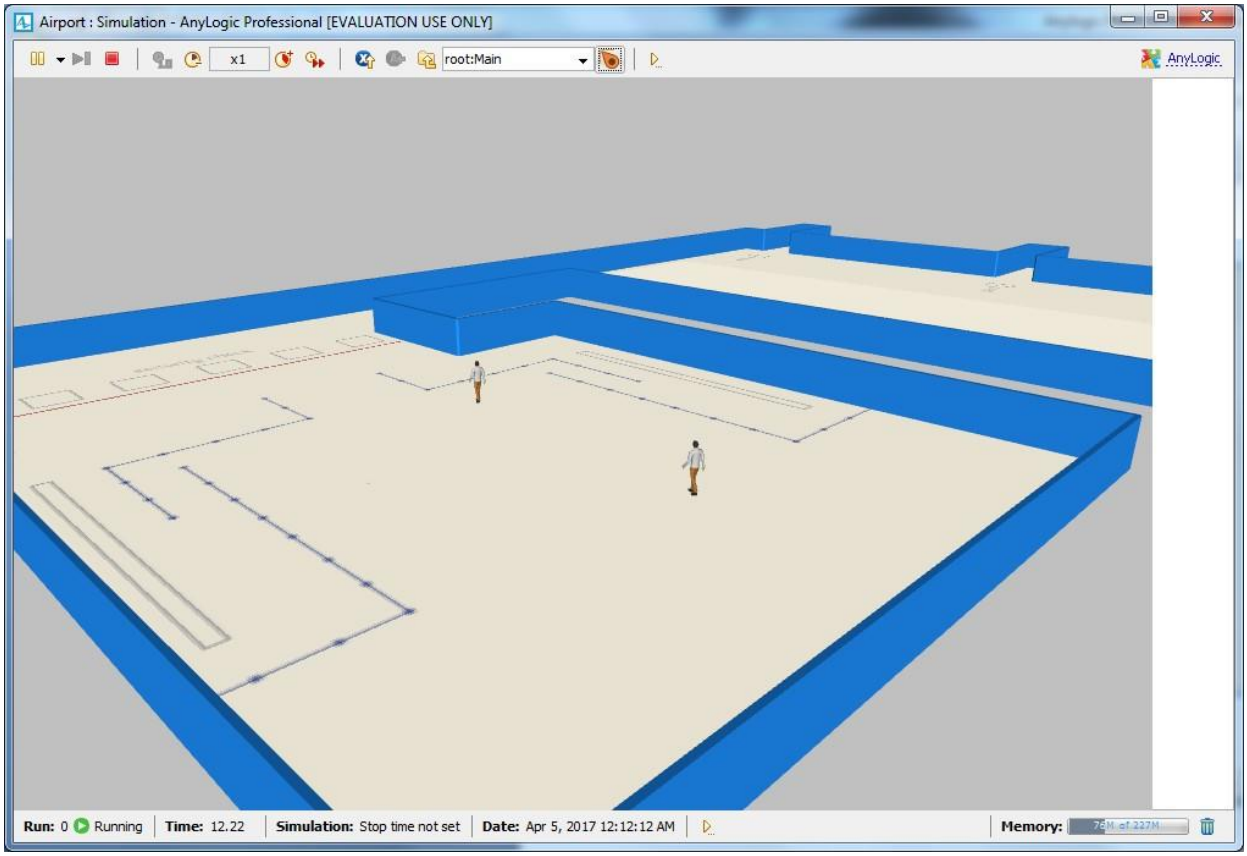
Close the model's window, open the camera's properties, and then apply the optimal camera you selected during the previous step by clicking Paste coordinates from clipboard.



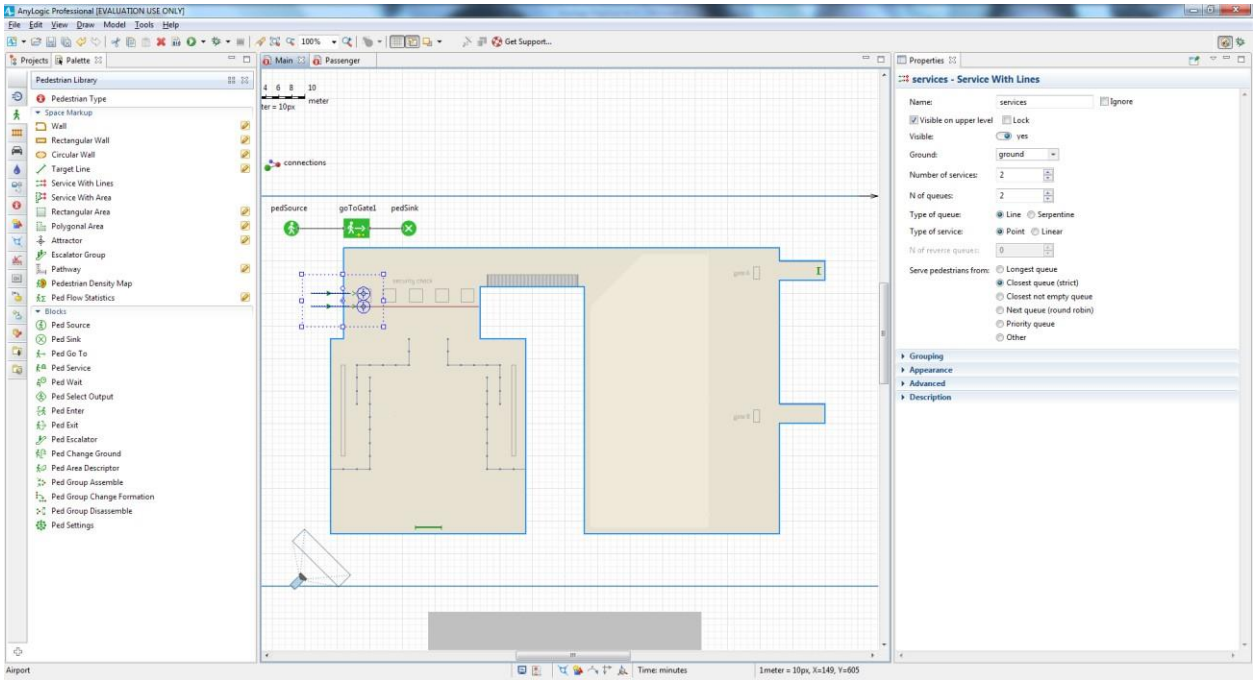

Run the model a second time and view the 3D view that the new camera position provides


Airport : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x1
root:Main
AnyLogic
pedSource
goToGate1
pedSink
security check
gate A
gate B
Run: 0 Running
Time: 1.86
Simulation: Stop time not set
Date: Apr 5, 2017 12:01:51 AM
Memory:

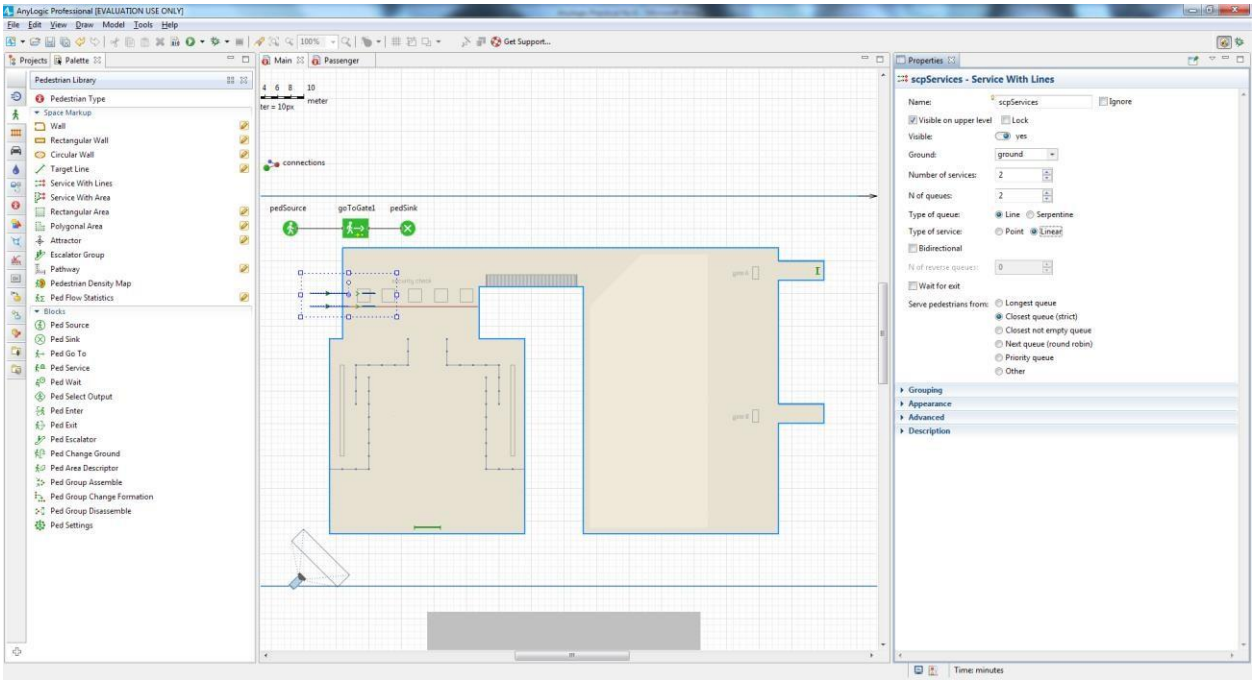

Airport : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x1/5
root:Main
AnyLogic
Run: 0 Running
Time: 32.53
Simulation: Stop time not set
Date: Apr 5, 2017 12:32:32 AM
Memory:



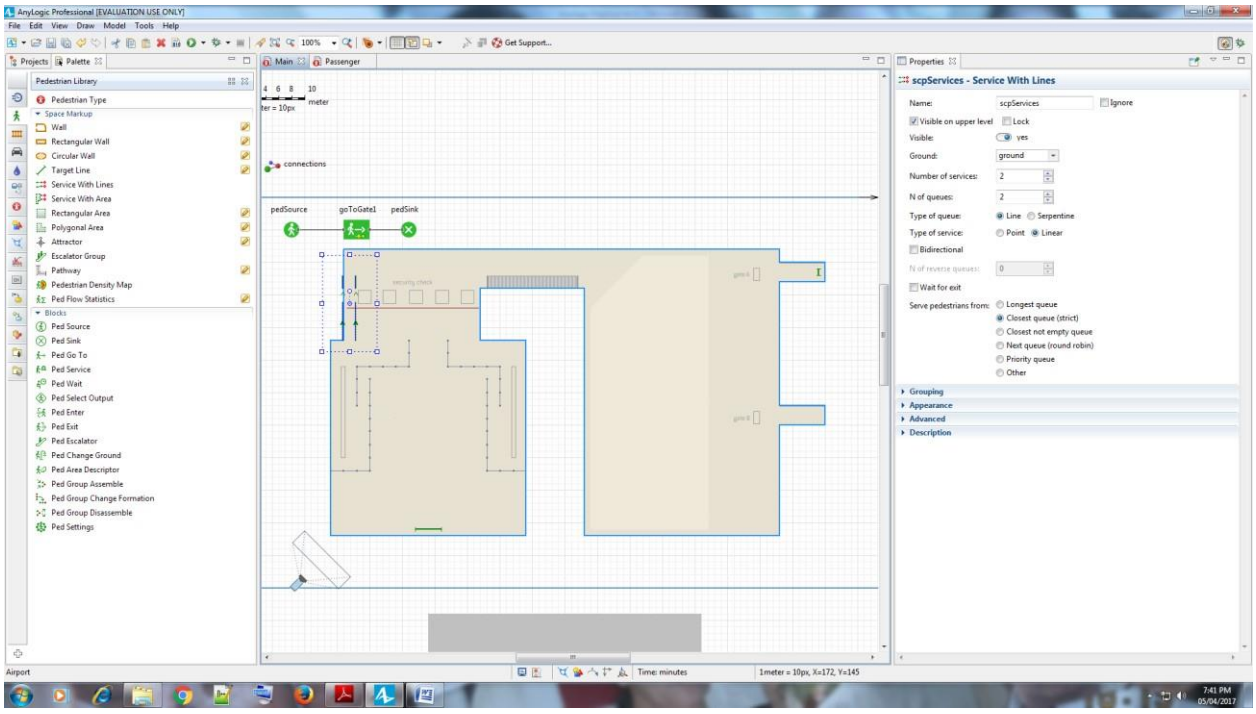

Adding security checkpoints
Drag the Service with Lines element from the Pedestrian Library palette on to the terminal layout. By default, a service will have two service points and two queue lines that lead to the service points.



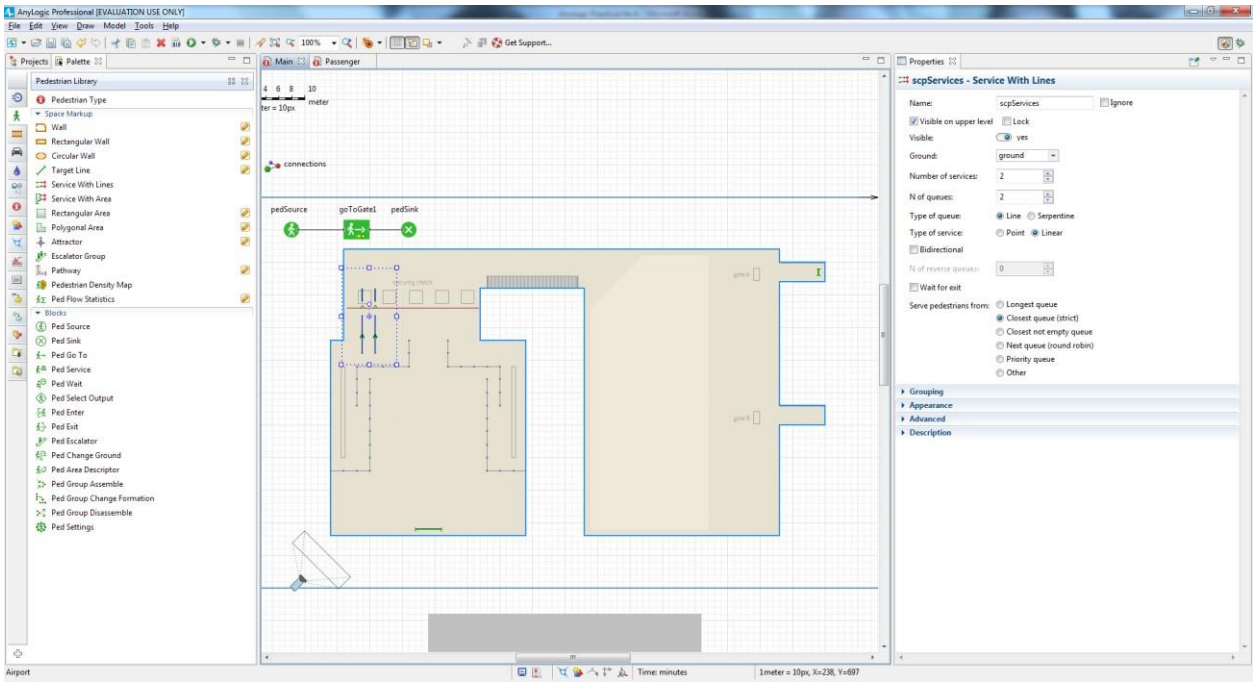

Open the Service With Lines properties area, use the Name box to name the shape scpServices - in this case, “scp” stands for security checkpoints - and then change the Type of service to Linear.



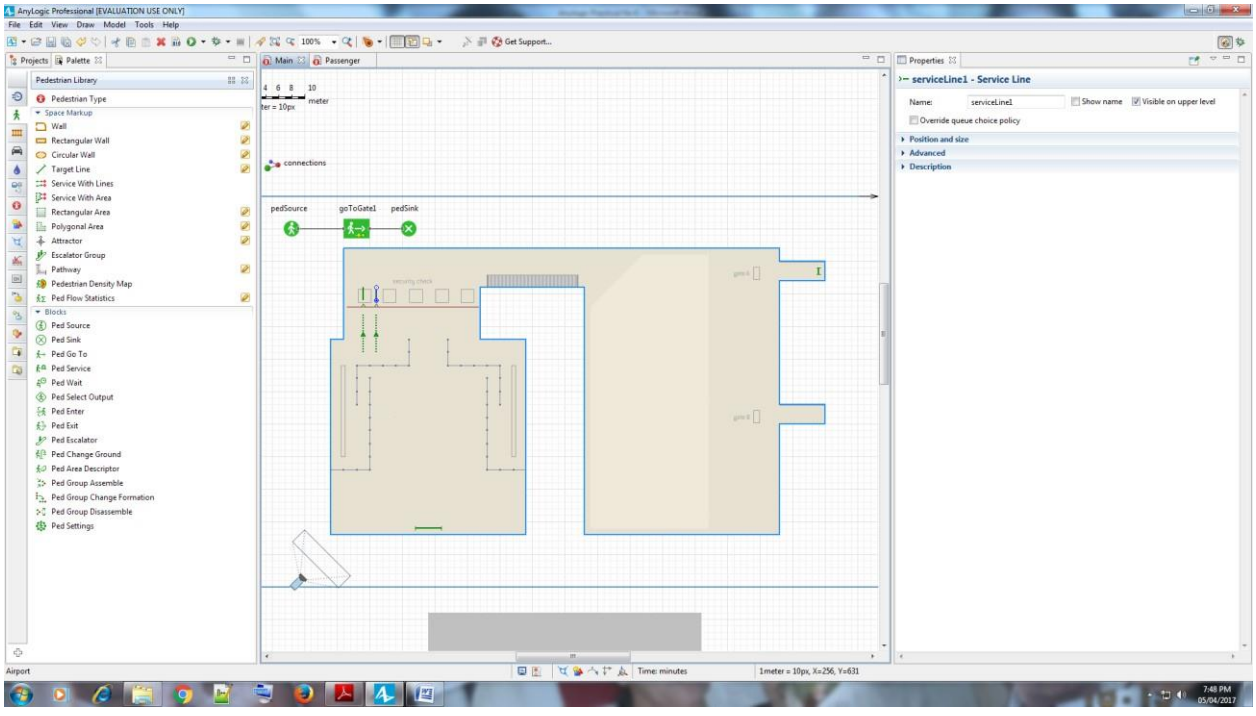

Use the round handle above the shape's center to rotate the service. So that the arrow points upward



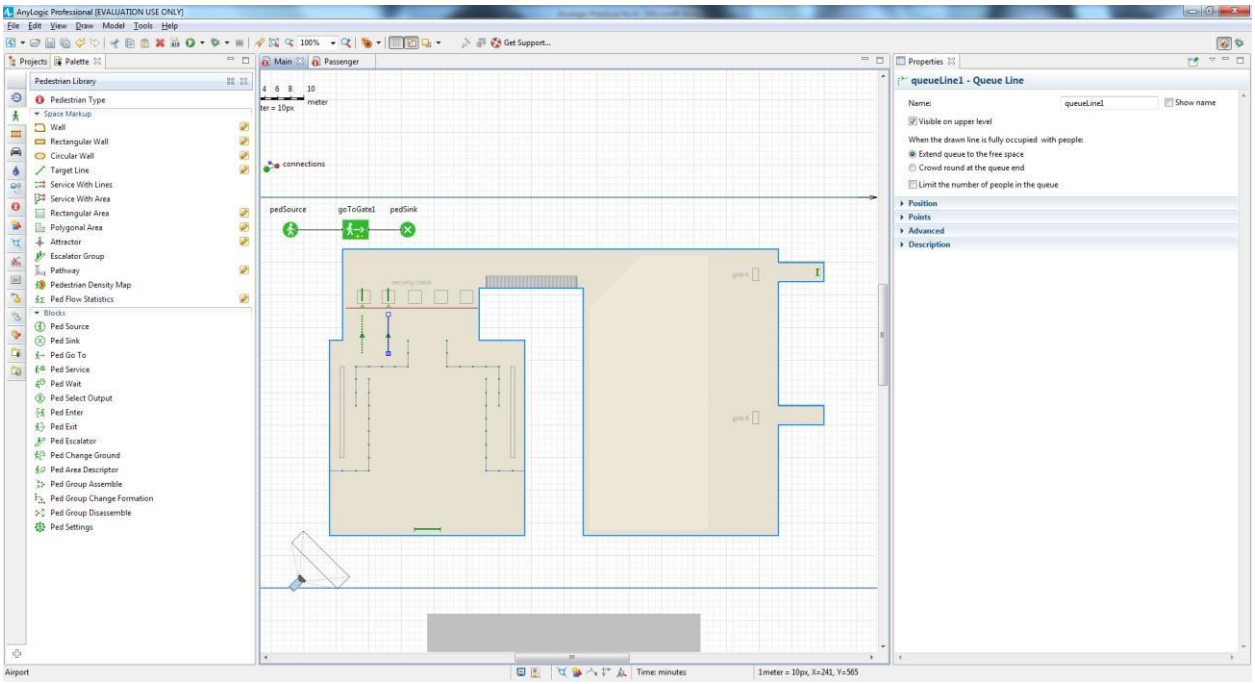

Move the service in a way that ensures the first linear service crosses the rectangle that represents the metal detector frame.



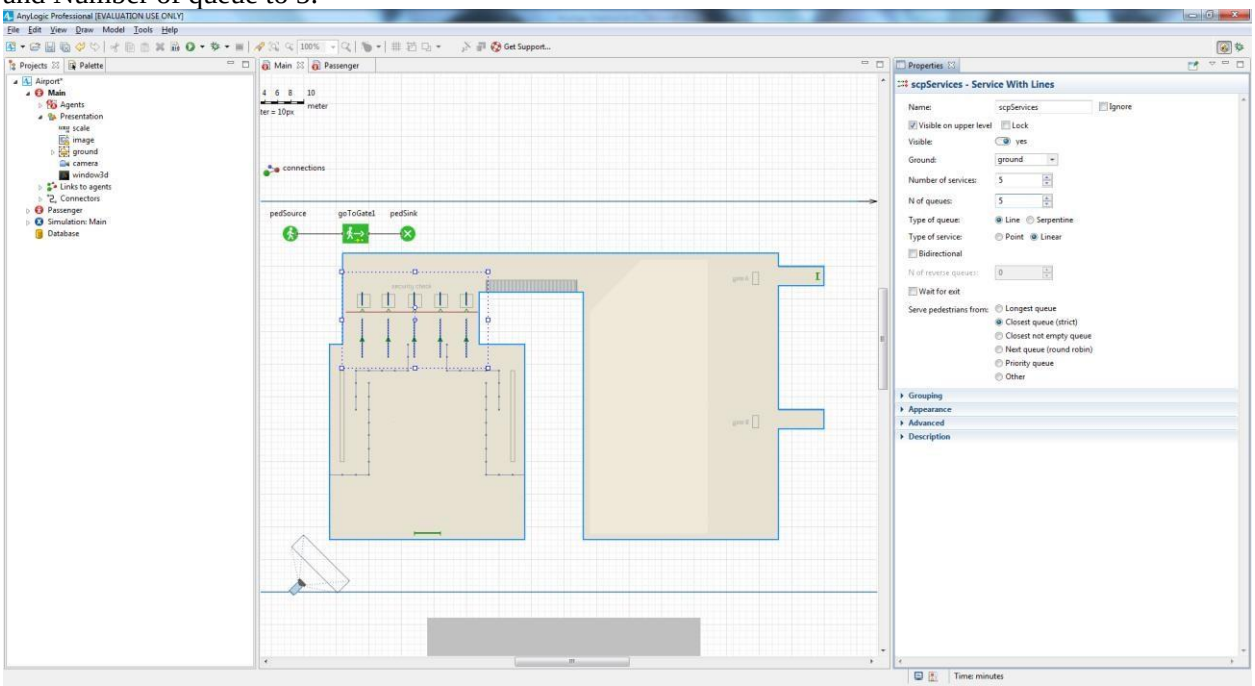

Select the next service line.



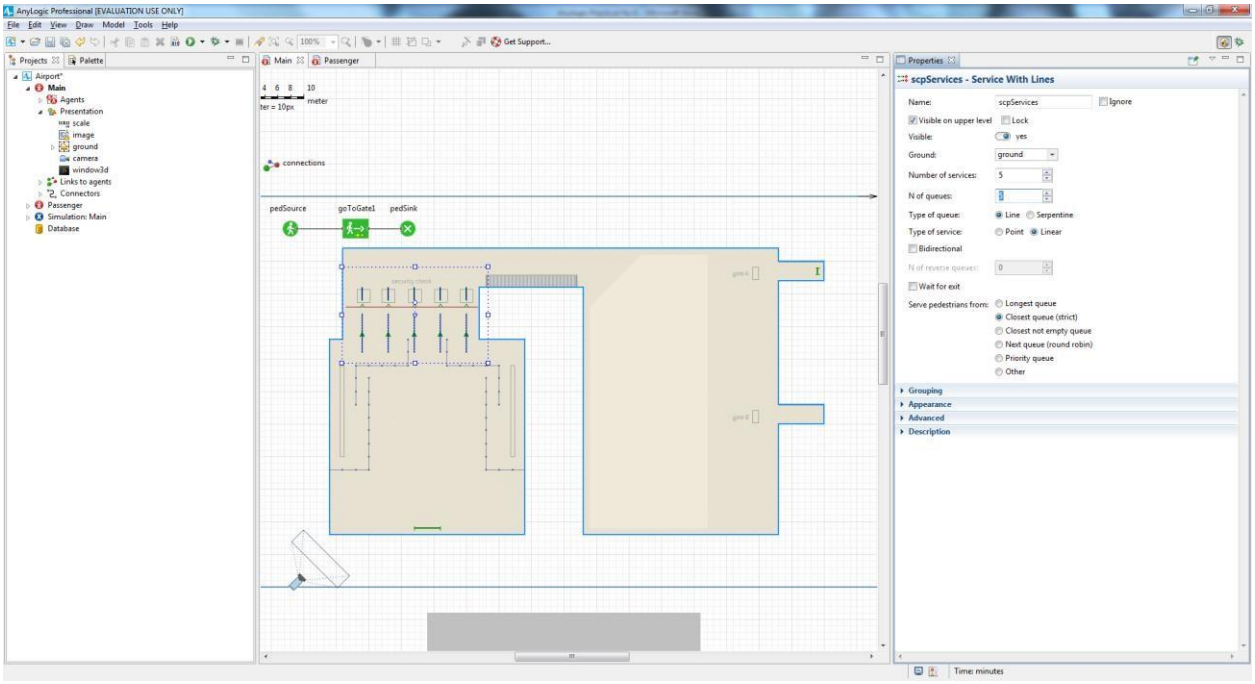

Accurately place the service line on top of the second security checkpoint placeholder and then adjust the queue location.



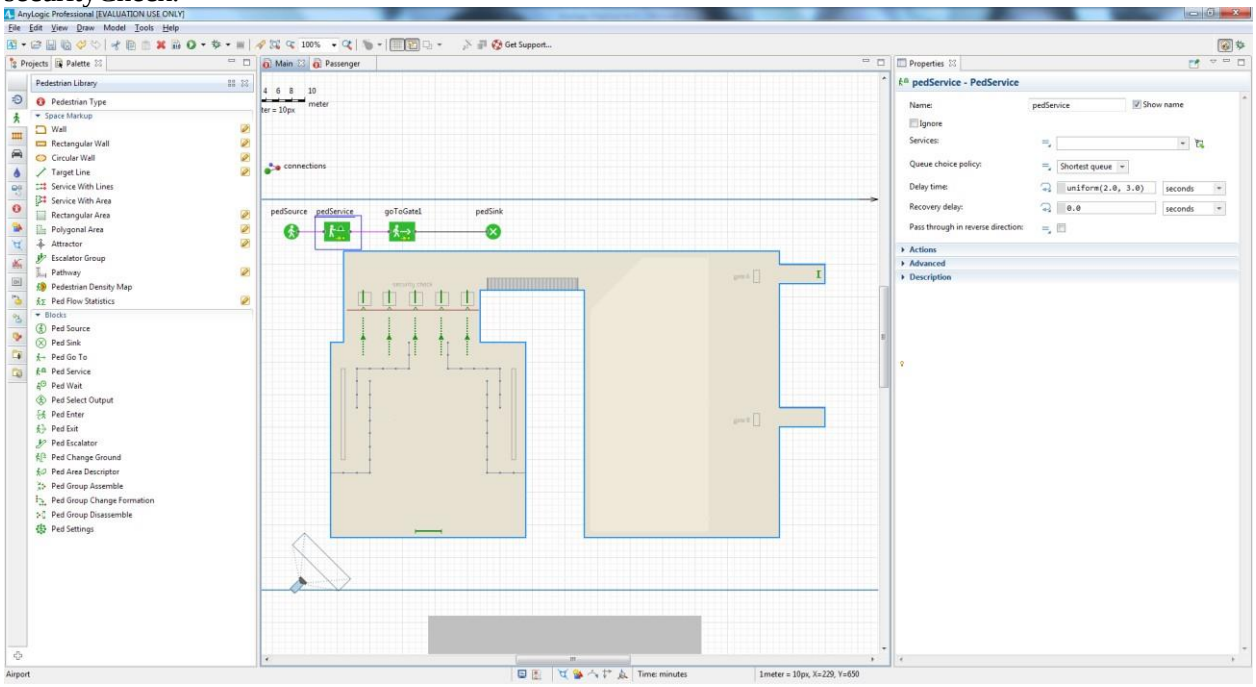
Navigate to the Service with Lines shape’s properties and then change both the Number of services and Number of queue to 5.



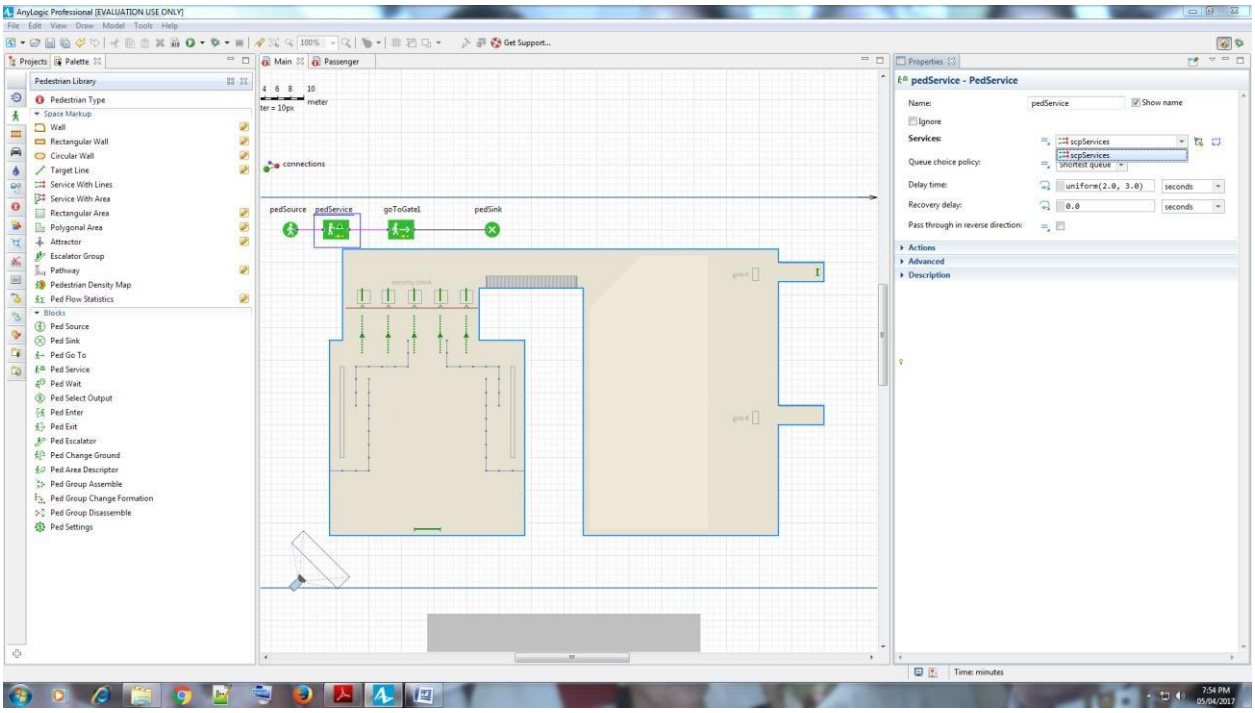
If necessary, adjust the new service and queue lines. After you’ve completed this step, the service shapes should look like those in the figure below.




Add the PedService block on to the flowchart between the PedSource and PedGoTo blocks to make pedestrians pass through the service we defined using the referenced Service with Lines shape, and then name it securityCheck.

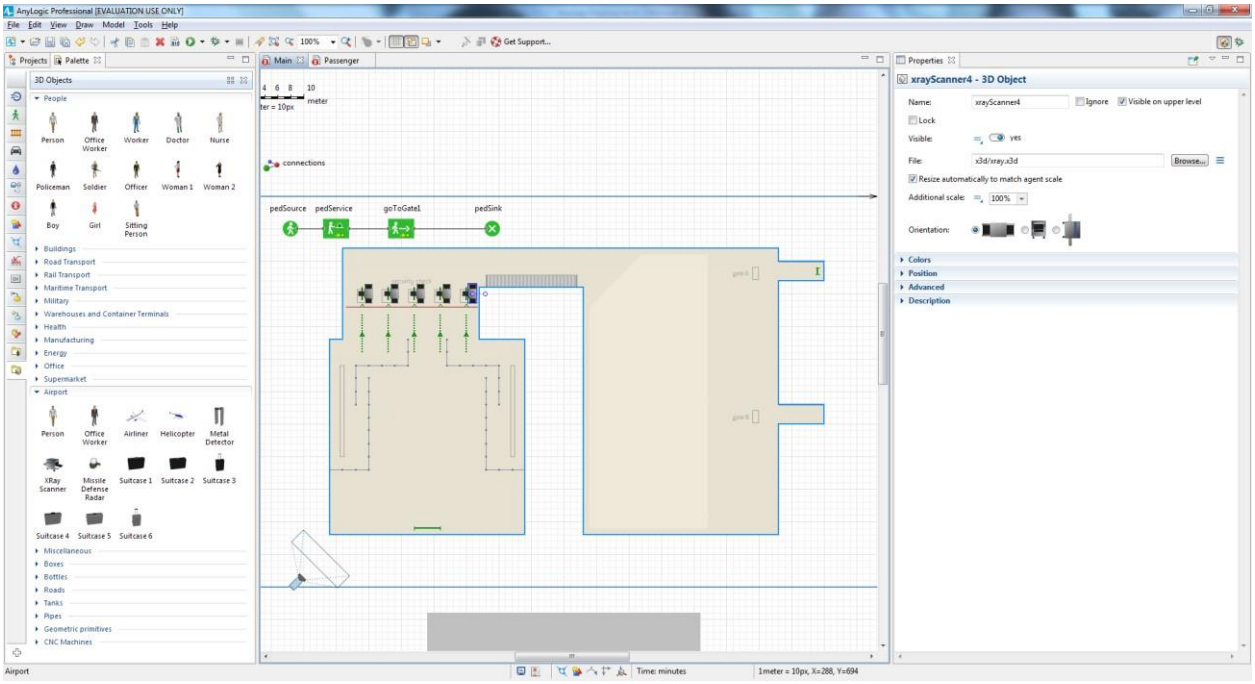



Go to the securityCheck block's properties. Select the services scpServices as Services.

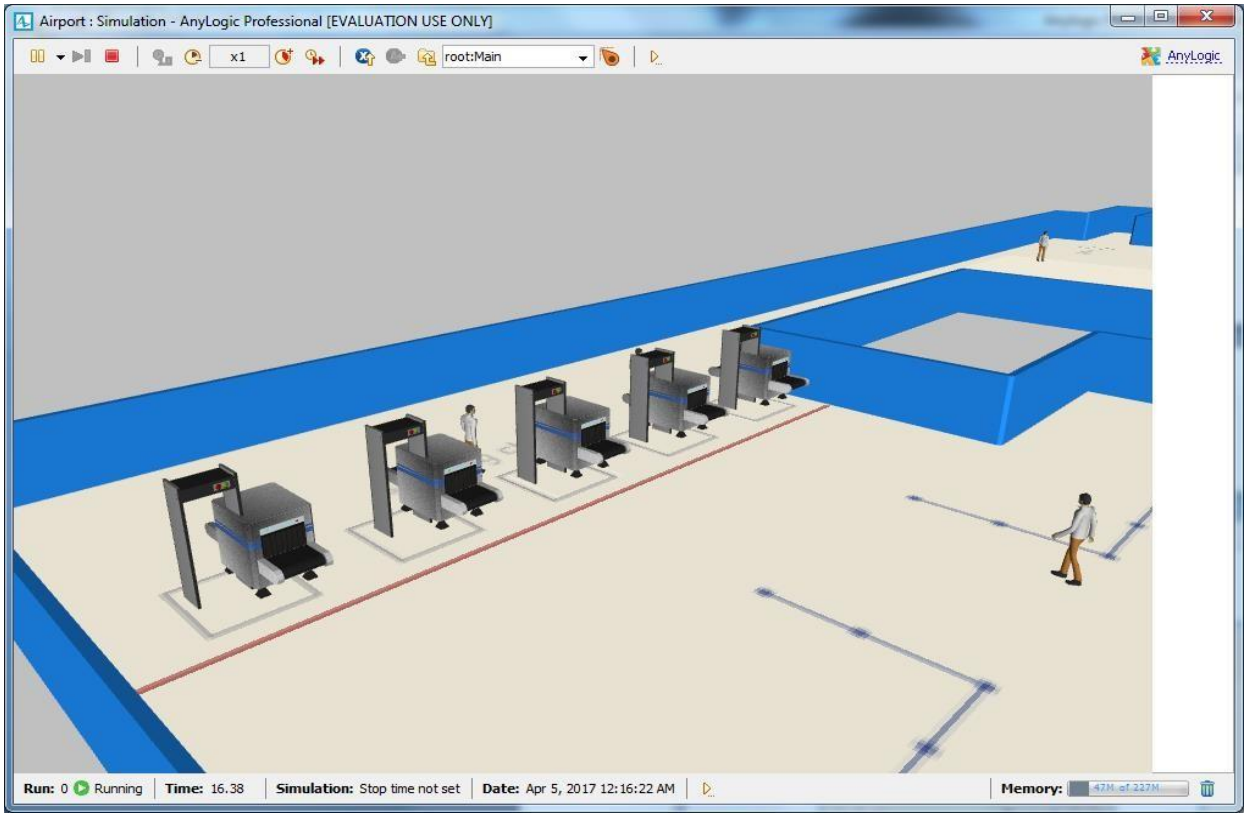



Since we assume it takes between 1 to 2 minutes to pass through the security checkpoint, type uniform(1, 2) minutes as the Delay time.

Now let's add 3D models of the security checkpoints. Using the 3D Objects palette, Airport section's Metal Detector and XRay Scanner elements, draw five security checkpoints. You will see the message box, prompting you to change the scale of 3D object. Select the option Do not ask me again and click OK.




Run the model. You'll see that passengers are now scanned at the security checkpoints.


Airport : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x1
root:Main
AnyLogic
Run: 0 Running
Time: 16.38
Simulation: Stop time not set
Date: Apr 5, 2017 12:16:22 AM
Memory: