

Introduction

AnyLogic

Author: Alejandro Rodriguez Grau

Tutor: Prof. Dr. Angel A. Juan

Full Course Index




- Introduction to AnyLogic (First Model)
- Basic models
- Heterogeneous model
- Advanced model

Introduction to AnyLogic

- **Download & Install**
- **Execute AnyLogic**
- **AnyLogic Interface**
- **First Model**
 - Definition
 - Building a New Model
 - Execution
 - Results
 - Experiments and randomness

1. Download & Install

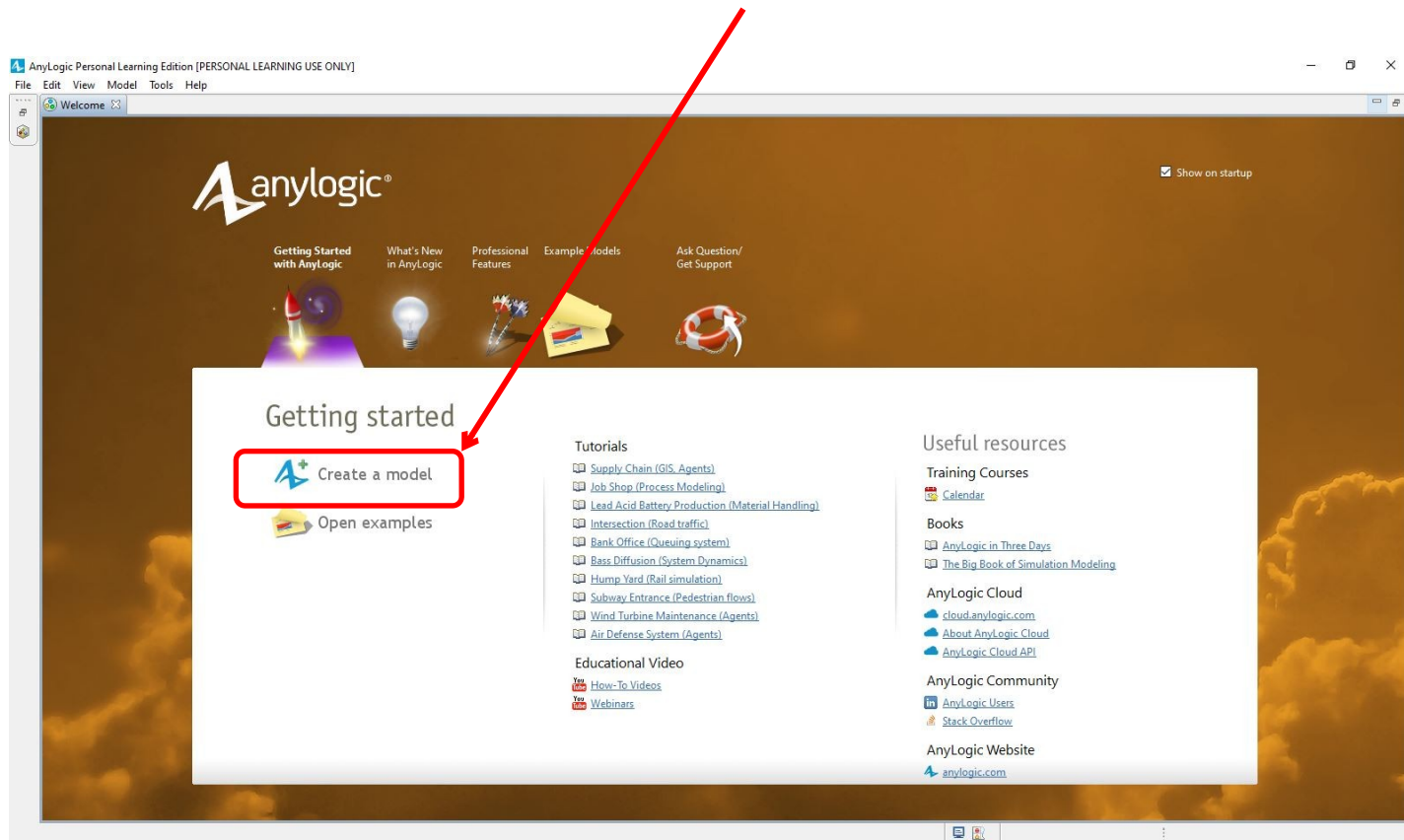
The screenshot shows the 'Download AnyLogic' page. At the top, it states 'latest version: 8.7.4' and 'released on: May, 4, 2021 *'. Below this are links for 'legal information', 'activation guides', 'version history', and 'system requirements'. It also indicates the software is 'available for' Windows, macOS, and Linux. The main content is divided into three columns:

Personal Learning Edition for beginners and students	University Researcher for public research in universities	Professional for companies and government organizations
		
FREE VERSION DOWNLOAD	DOWNLOAD ASK FOR A QUOTE	DOWNLOAD ASK FOR A QUOTE

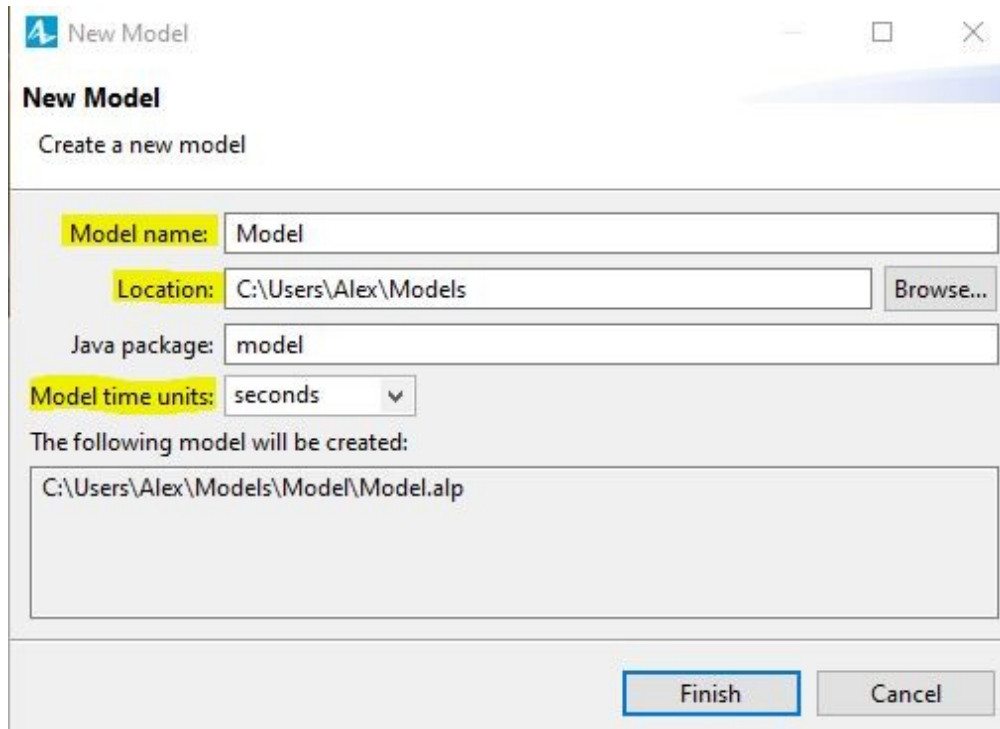
From website: <https://www.anylogic.com/downloads/>

2. Execute (1/2)

- Once installed, we can execute the software. We will be greeted with the following intro screen. Here we can find online tutorials for further learning if desired, after or along this course.
- Lets get started and select Create a Model.



2. Execute (2/2)



New Model
Create a new model

Model name: Model

Location: C:\Users\Alex\Models

Java package: model

Model time units: seconds

The following model will be created:

C:\Users\Alex\Models\Model\Model.alp

- **Model name** □ When choosing a name for the model, make it easy and choose a short name. Beware of special characters as they could become a problem when executing a model.
- **Location** □ It's suggested to create a folder near to the course slides and save there your models.
- **Models time units** □ May differ between seconds, hours, days... depending on what are we simulating.

3. AnyLogic Interface (1/2)

The screenshot displays the AnyLogic software interface, specifically the Process Modeling Library (PML) and the main workspace. The interface is divided into several key areas:

- 1. Menu Bar:** Located at the top, containing File, Edit, View, Draw, Model, Tools, and Help.
- 2. Toolbar:** A horizontal bar below the menu bar containing various icons for file operations, navigation, and modeling tools.
- 3. Process Modeling Library (PML):** A large vertical panel on the left side, organized into categories:
 - Agent Type / Resource Type:** Includes Space Markup (Path, Point Node, Rectangular Node, Polygonal Node, Attractor) and Pallet Rack.
 - Blocks:** A collection of modeling blocks such as Source, Sink, Delay, Queue, Select Output, Select Outputs, Hold, Match, Split, Combine, Assembler, Move To, Conveyor, Resource Pool, Seize, Release, Service, Resource Send To, Resource Task Start, Resource Task End, Downtime, Schedule, Enter, Exit, Batch, Unbatch, Dropoff, Pickup, Restricted Area Start, Restricted Area End, Time Measure Start, Time Measure End, Resource Attach, Resource Detach, Rack System, Rack Store, Rack Pick, and PML Settings.
 - Auxiliary:** Includes Wait, Select, and Plain.
- 4. Main Workspace:** A large central grid area where the model is built. It contains a few blue lines representing a simple layout.
- 5. Properties Panel:** Located on the right side, titled "Main - Agent Type". It shows the Name field set to "Main" and a list of expandable sections: Agent actions, Agent in flowcharts, Dimensions and movement, Space and network, Advanced Java, and Description.
- 6. Status Bar:** At the bottom, it displays "Time: seconds" and "1meters = 10px, X=-6, Y=72".

3. AnyLogic Interface (2/2)

- 1 □ Standard Main Ribbons [Used mainly to *Save as* and change the *View*]
- 2 □ Specific Ribbons [Similar purpose as 1]
- 3 □ Project Libraries [All objects are sorted in different *Libraries* depending on type]
- 4 □ Objects in the selected Project Library [Drag & Drop objects to the centre to add them to your model]
- 5 □ Properties of selected object/model. [When something is selected, *Properties* will show up here, also this will be where we will do most of our work changing numbers]

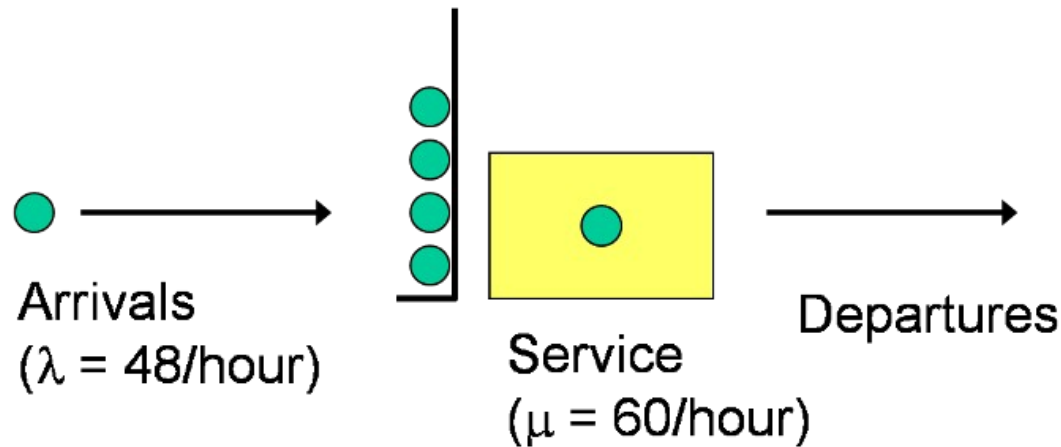
4. First Model

▪ Goal: to illustrate the **simulation model-building** process and the statistical **analysis** of simulation **output** result.

▪ Example: We'll use a simple **single-server queueing system** with Poisson arrival rate (average 48 entities/hour) and exponential service time (average 60 entities/hour).

▪ Further assumptions: (a) interarrival times and service times are assumed to be **independent**; (b) queue has **infinite capacity**; (c) queue discipline will be **first-in-first-out** (FIFO).

▪ Questions: **number** of entities in the queue? **time** an entity spends in the queue? **utilization** of the server? etc.



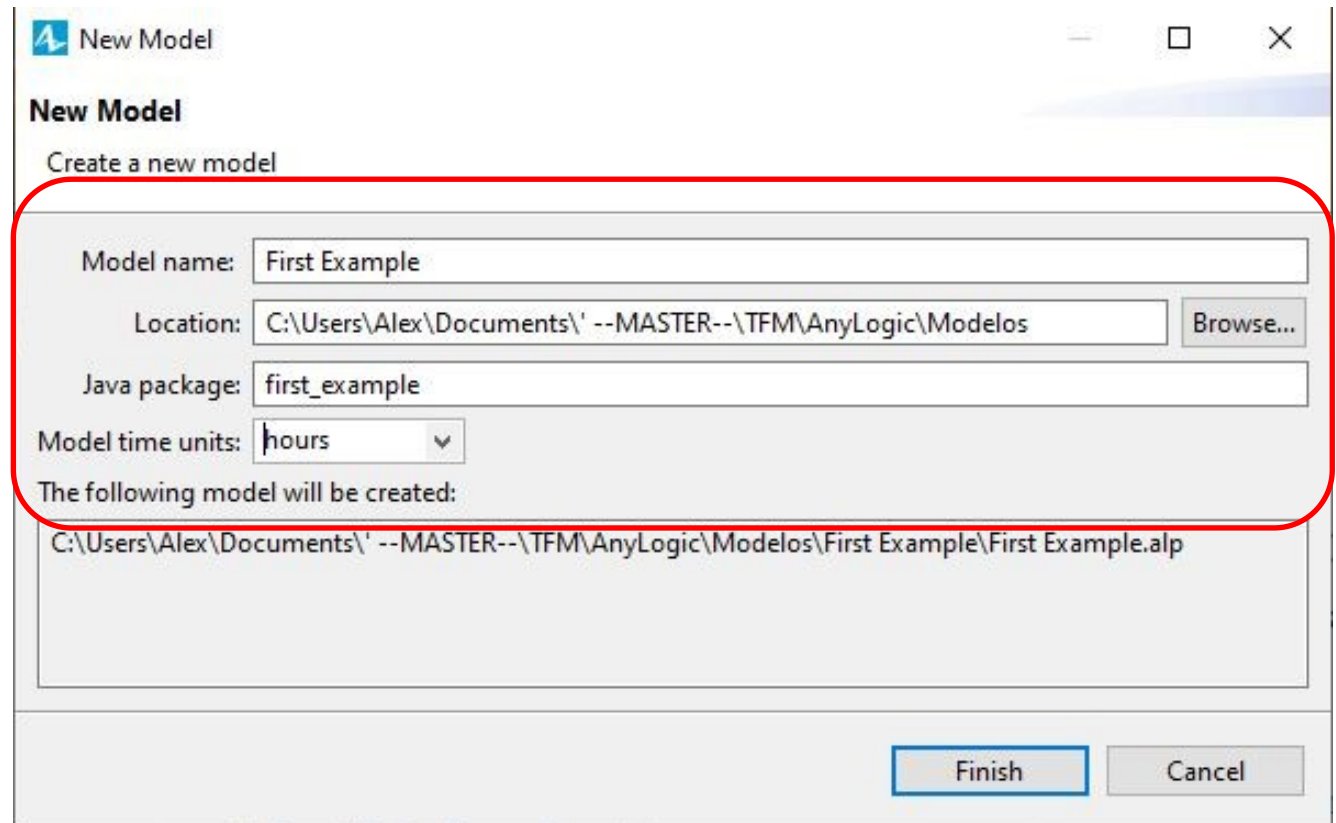
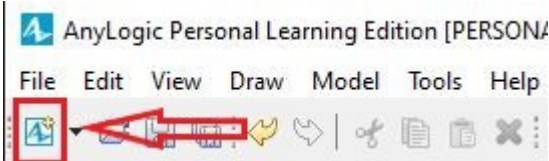
4.1. Definition

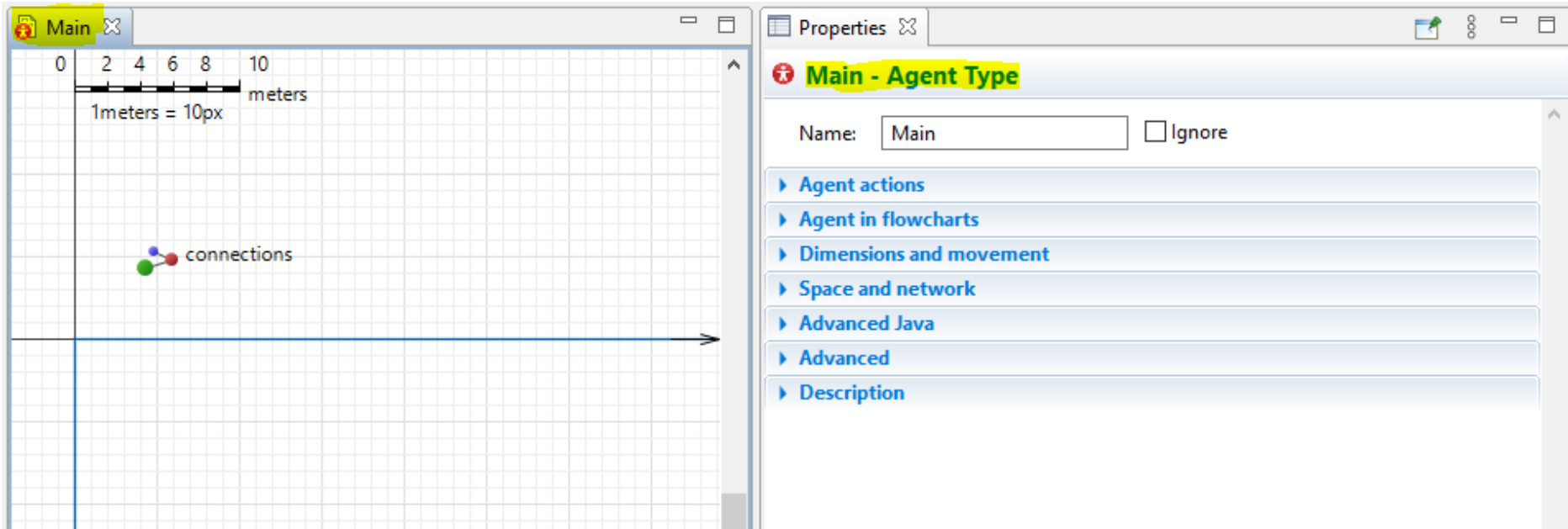
As shown on the next slides, we will start using the **Objects** from our **Libraries**, such as “Source”, “Sink”, “Service” and “Connectors”. Drag & drop any **object** from the selected **library** to the main centre screen to add them to your model.

Select the **objects** in your model and change their **properties** according to our queue example model shown in the previous slide, and use the numbers suggested in the captions of each screenshot to come.

We will also change some names to make it easier to identify each **object**, in the following slide we can follow step by step this procedure.

4.2. Building a New Model





One Agent is created by default ("Main").

Changes will be made in the Properties tab.

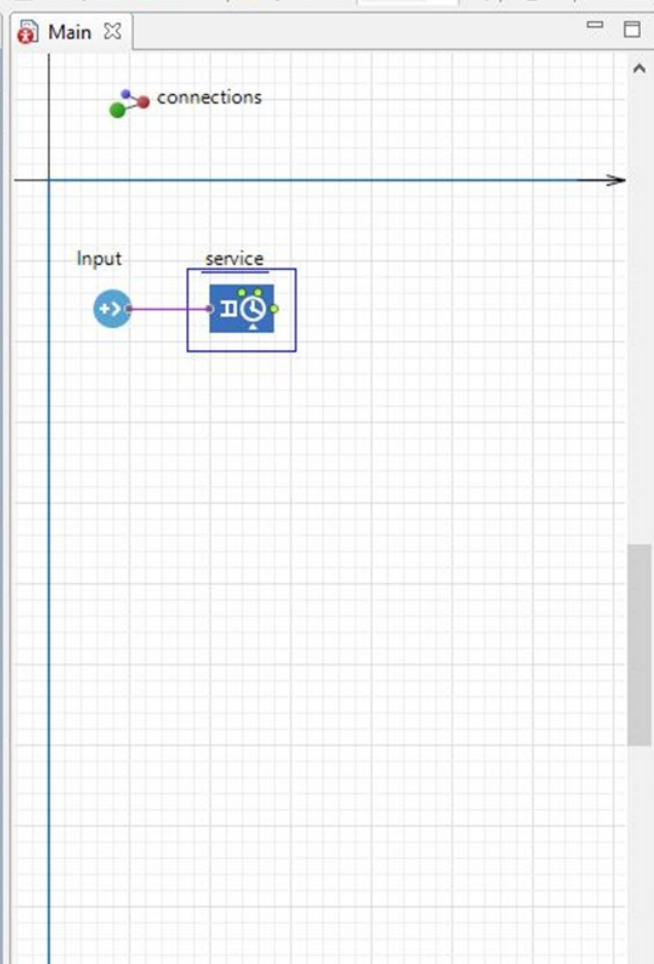
The screenshot displays the AnyLogic software interface. On the left is a palette with various modeling elements categorized into 'Process Modeling Li...', 'Agent Type', 'Resource Type', 'Space Markup', and 'Blocks'. The 'Source' block is highlighted in yellow. The main canvas shows a 'connections' block and an 'Input' block. A blue arrow points from the 'Input' block to the 'connections' block. On the right, the 'Properties' panel for the 'Input - Source' block is visible. The 'Name' field is set to 'Input' and 'Show name' is checked. The 'Arrivals defined by' dropdown is set to 'Interarrival time'. The 'Interarrival time' is set to 'exponential(1.25)' with a unit of 'minutes'. The 'First arrival occurs' dropdown is set to 'After timeout'. The 'Location of arrival' dropdown is set to 'Not specified'. Below the main settings are expandable sections for 'Agent', 'Advanced', 'Actions', 'Advanced', and 'Description'.

Add a "Source", change it's name, and specify an *Interarrival Time* as an **exponential distribution** with a mean time of 1.25 minutes.

Projects Palette

Process Modeling Library

- Split
- Combine
- Assembler
- Move To
- Conveyor
- Resource Pool
- Seize
- Release
- Service**
- Resource Send To
- Resource Task Start
- Resource Task End
- Downtime
- Schedule
- Enter
- Exit
- Batch
- Unbatch
- Dropoff
- Pickup
- Restricted Area Start
- Restricted Area End



Properties

service - Service

Name: Show name Ignore

Seize: (alternative) resource sets
 units of the same pool

Resource sets (alternatives):

Queue capacity:

Maximum queue capacity:

Delay time:

Send seized resources:

Agent location (queue):

Agent location (delay):

▸ Priorities / preemption

▸ Advanced

▸ Actions

▸ Advanced

▸ Description

Add a "Service" and specify a *Delay Time* as another **exponential distribution** with a mean time of 1 minute.

The screenshot displays the AnyLogic Personal Learning Edition software interface. The main workspace shows a simulation model with three blocks: a blue 'Source' block with a plus sign, a blue 'service' block with a clock icon, and a blue 'Sink' block with a plus sign inside a circle. A blue arrow connects the Source to the service, and another blue arrow connects the service to the Sink. A red box highlights the Sink block. Above the Sink block, a blue arrow points from the left towards the Sink. The 'Process Modeling Library' is visible on the left, with the 'Sink' block highlighted. The 'Properties' panel on the right shows the 'Output - Sink' block with the name 'Output' in a yellow box, and checkboxes for 'Show name' (checked) and 'Ignore' (unchecked). The 'Actions', 'Advanced', and 'Description' sections are collapsed.

- The last piece we need is a sink to send the entities that have finished the route, and can be discarded from our simulation.
- This is now a defined environment, with a start, an end, and a service between these points that simulates the theoretical distribution process time for the users/entities.
- Let's run this model so we can see where the results such as "time in system" or other relevant information.

4.3. Execution (1/3)

Set the model time units to Minutes (if you use seconds, simulation will be slow, if you choose hours, the simulation will run fast), this depends on what are we simulating, how long can we wait for the results, and how detailed should the results be.

The screenshot displays the AnyLogic Personal Learning Edition interface. The main workspace shows a model diagram with three components: 'Input' (a blue circle with a right-pointing arrow), 'service' (a blue square with a clock icon), and 'Output' (a blue circle with an 'X' icon). These are connected by arrows. A 'connections' component is also visible at the top. The 'Properties' panel on the right is titled 'First Example - Model' and shows the 'Name' as 'First Example' and 'Model time units' set to 'minutes'. The 'Dependencies' section is expanded, showing 'System Dynamics solver', 'Advanced', and 'Description'.

4.3. Execution (2/3)

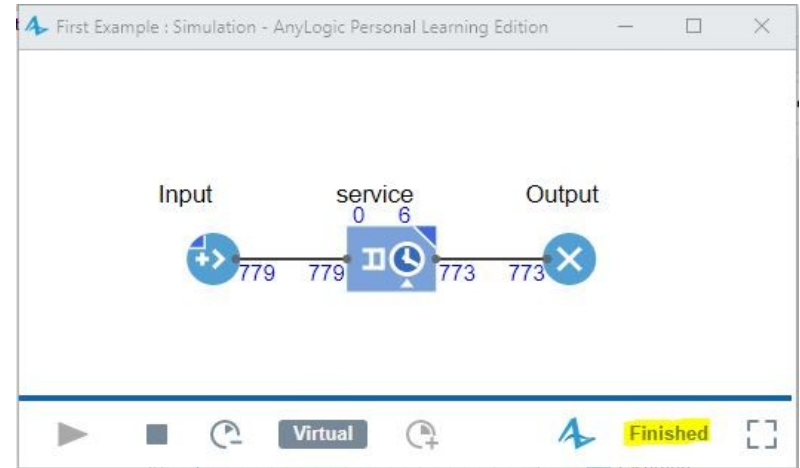
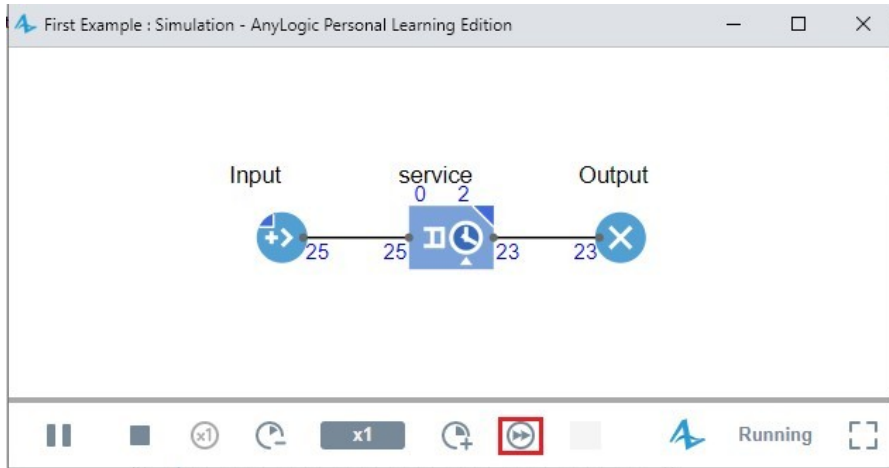
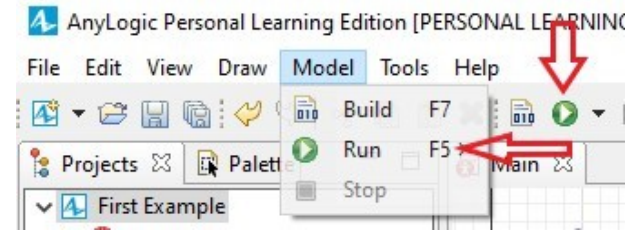
The screenshot displays the AnyLogic Personal Learning Edition interface. The main workspace shows a simulation model with three components: 'Input' (a blue circle with a right-pointing arrow), 'service' (a blue square with a clock icon), and 'Output' (a blue circle with an 'X' icon). These are connected by arrows in a linear sequence. A 'connections' palette is visible at the top of the workspace. The 'Properties' panel on the right is titled 'Simulation - Simulation Experiment' and contains the following settings:

- Name: Simulation
- Top-level agent: Main
- Maximum available memory: 512 Mb
- Skip experiment screen and run the model
- Execution mode: Real time with scale 1
- Stop: Stop at specified time
- Start time: 0
- Stop time: 600
- Start date: 04/06/2021
- Stop date: 04/06/2021
- Time format: 0:00:00
- Time format: 10:00:00

You must also remember that we don't want the simulation to run forever, so let's change the *Simulation properties* so it stops after 10 virtual hours.

4.3. Execution (3/3)

You can find the “start simulation” button in 3 or 4 places, they all do the same, here we show a couple. Simulation will open a new window where we can follow the simulation at the rate we just specified.



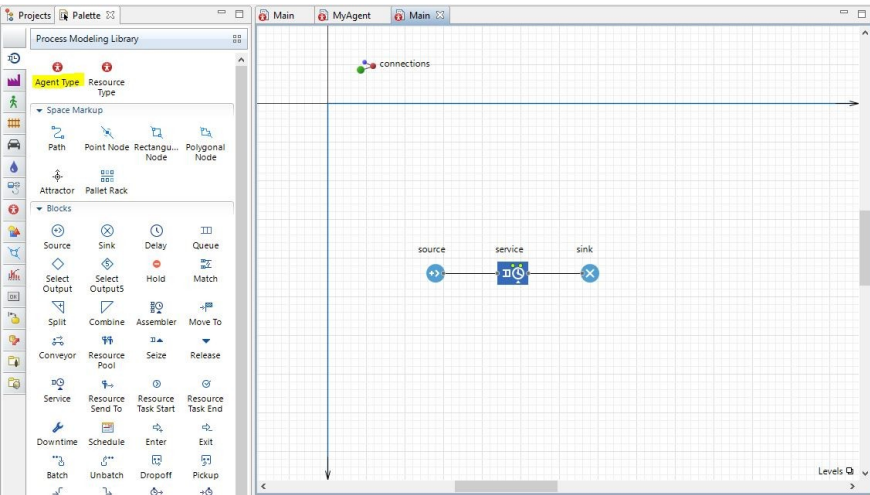
5. Results

- Up to this point, simulation can be completed, but we will only see changes in the numbers as seen in the images in the previous slide, lets add a new **Agent**, and lets give it some **parameters**, so we can see more information on the simulation.
- The **simulation environment** has a special developer button, that when clicked, opens a console. Consoles are used to either input data, or show results during the simulation, here we can print our output chosen parameters/variables.



- We will make our model show an output in the console of "*time in system*" for each individual agent created in the simulation.

5.1 New Agent



Drag & Drop Agent Type

New agent
Step 1. Creating new agent type

Agent type name:

Create the agent type "from scratch"
 Use database table
I have agent data stored in a database

< Back Next > Finish Cancel

New agent
Step 2. Agent animation

Choose animation: 3D 2D None

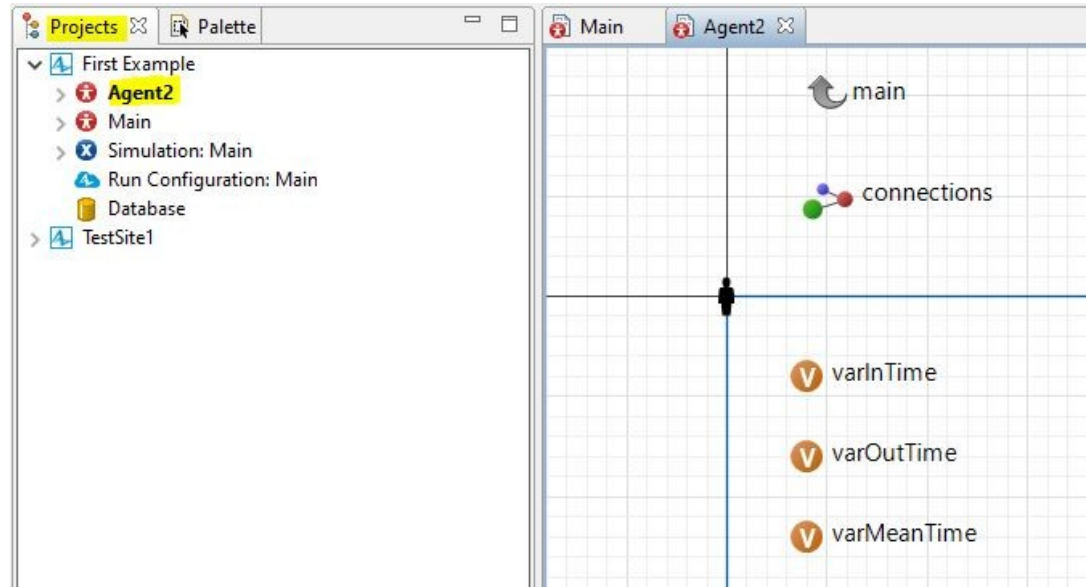
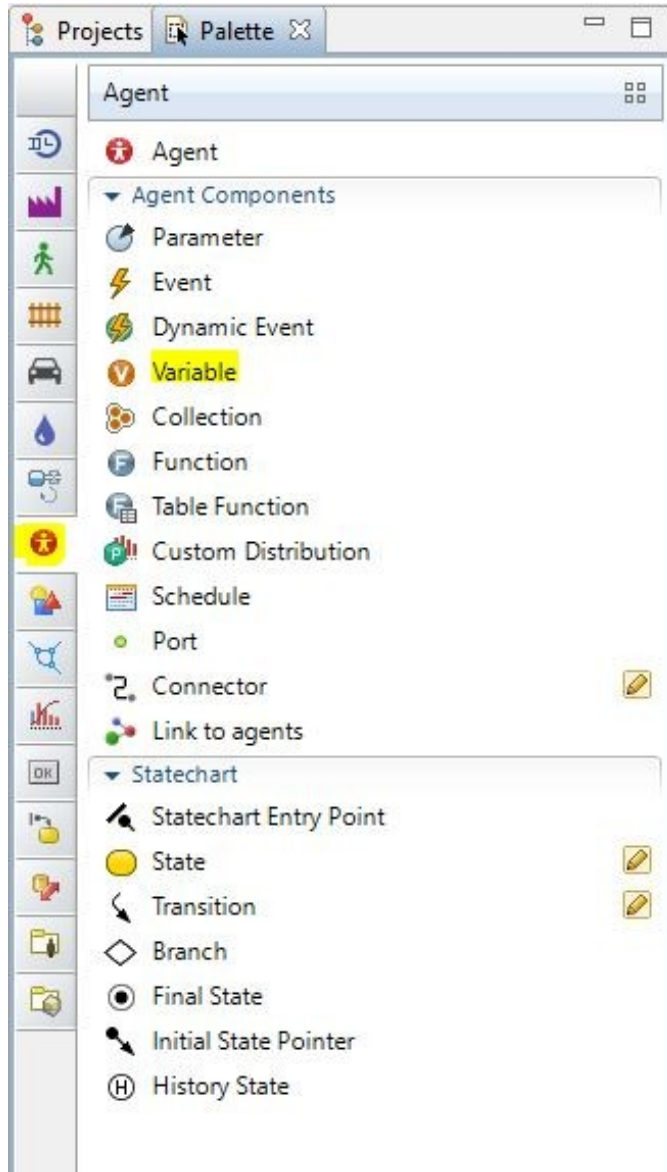
General
Person
Nurse
Doctor
Patient
USA Map
Lorry

Preview window showing a black silhouette of a person.

< Back Next > Finish Cancel

Skip step 3 for now, parameters will come later.

5.2 New Variables



Drag & Drop and choose the variable name and units (minutes)

5.3 Interaction Between Agents

multiple agents per arrival:

Limited number of arrivals:

Location of arrival:

Agent

Advanced

Actions

On before arrival:

On at exit:

On exit:

Advanced

Agent type:

Single agent Population of agents

Model/library: [\(change...\)](#)

Visible: yes

Properties

sink - Sink

Name: Show name Ignore

Actions

On enter:

Advanced

Agent type:

Single agent Population of agents

Model/library: [\(change...\)](#)

Visible: yes

5.4 Result Simulation

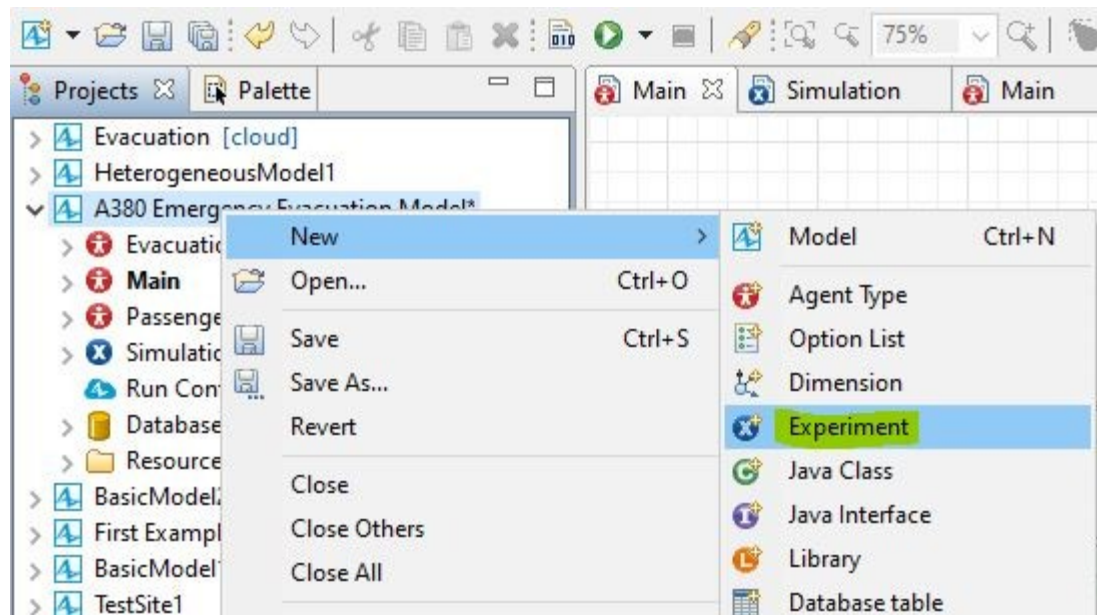
- In red, we see the button that shows us the console.
 - In green we have the first results printed in the console that come from the variable “*varMeanTime*”, which means that the result shown is the “mean time in system” for each agent (in minutes as specified).
- If you followed all the steps, the console will also display time of entry to the system of the entity .

The screenshot displays the AnyLogic simulation environment. On the left, a process flow diagram shows a sequence of three entities: 'source' (a blue circle with a plus sign), 'service' (a blue square with a play button), and 'sink' (a blue circle with an X). The 'service' entity is highlighted with a blue box. Above the 'service' entity, the text reads 'agent2 root.agent2(parInTime = 0.0, parOutTime = 0.0, parMeanTime = 0.0)'. The console window on the right shows a list of numerical values, with the first few lines highlighted in green. The console also displays a 'WARNING' message and a 'Running' status at the bottom right, which is highlighted with a red box.

```
21.80 min 06/05/2021 00:21 :
0.032775383188984764
0.8584682651373523
0.036970258423049795
0.004635486592592386
0.025841496762097904
0.3784305578686835
0.20714277333072406
0.9723267147315386
1.4017029808085404
0.3639182857115255
0.29573509619801186
0.7303750135414742
0.06350991831569885
0.4726249649353935
1.2858806649969203
0.026514410890943196
0.2647981602319689
0.9300831456916896
0.11074990372259208
0.8393137651853486
0.11663634774203757
```

6. Create Experiments

- **Experiments** run several times the proposed simulation, and is used to compare results between simulations and detect bottlenecks or mistakes.
- In AnyLogic there are three types of randomness:
 - ✓ The Fixed seed (runs the same simulation repeatedly)
 - ✓ The Random seed (Introduces RNG)
 - ✓ The custom generator (not used in this slides, custom made)



7. Introducing Randomness (RNG)

Where are we using random functions?
In the Source and in the Server:

But if executed twice, results will be exactly the same in the console, in the same sequence. This is because we haven't told the software to make unique or random repetitions yet.

The screenshot shows the NetLogo environment with a main window titled 'Main' and 'Agent2'. The main window displays a diagram with three components: 'source', 'service', and 'sink', connected by arrows. The 'source' component is highlighted with a yellow box. The right-hand pane shows the 'Properties' window for the 'source' component. The 'Interarrival time' is set to 'exponential(1.25)' minutes, which is highlighted with a red box. Other settings include 'Arrivals defined by: Interarrival time', 'First arrival occurs: After timeout', and 'Agent' parameters.

The screenshot shows the NetLogo environment with a main window titled 'Main' and 'Agent2'. The main window displays a diagram with three components: 'source', 'service', and 'sink', connected by arrows. The 'service' component is highlighted with a yellow box. The right-hand pane shows the 'Properties' window for the 'service' component. The 'Delay time' is set to 'exponential(1)' minutes, which is highlighted with a red box. Other settings include 'Seize: (alternative) resource sets', 'Resource sets (alternatives):', 'Queue capacity: 100', and 'Send seized resources:'. The 'Agent' parameters are also visible at the bottom.