AnyLogic 7: New Features Overview

Ilya Grigoryev
Head of Training Services
AL7 New Features Overview. Agenda

• AnyLogic UI & Usability improvement
  – New workspace layout, more space for graphical editor
  – Drop-down lists instead of typing
  – Probability distribution wizard, custom distribution object

• More support for agent based modeling
  – True integration of modeling methods (“The Great Merge”): Entity = Resource unit = Agent
  – Agent population wizard
  – Link object / network visualization
  – Inheritance between agent types

• Space markup
  – Nodes, paths, walls, services, attractors, rail tracks, storages...
  – One unified space for all objects

• New Process Modeling library
  – “Pull” entity flow
  – Flexible resource management / alternative sets, preparation & wrap-up, shifts, breaks, preemption, preferences, ...
  – “Internals” of entities defined graphically

• 3D
  – Gallery of more than 300 3D shapes
  – Painting 3D shapes

• Highly improved Pedestrian library
  – Performance
  – Ease of use / markup

• Rail library improved
  – Rail network construction / auto-connect, curved segments, ...
  – Collection of US rail car 3D models

• Miscellaneous
  – Minor features
  – AnyLogic roadmap
Stays the same:

• Powerful and flexible multi-method simulation environment
  – Agent based / statecharts, decision rules, networks, ...
  – Discrete event / process flowcharts
  – System Dynamics / stock and flow diagrams, numeric solvers

• General-purpose “horizontal” tool with support for several verticals:
  – Pedestrian
  – Rail
  – Health
  – Logistics

• Written in Java /cross-platform, open, exportable, embeddable
  – Windows
  – Linux
  – Mac OS
New version release. What are users afraid of?

I won’t cope with converting my models to AL7

- All AL6 models run in AL7
- All AL6 libraries are still included
- We have developed tools for migration

AL 7 will override AL6

While migrating to AL 7, you can continue work in AL 6. Both versions installed, no conflict!

I have to pay big money for the new version

Having valid Support Services, you can activate AnyLogic 7 with your current activation key FOR FREE, as any other upgrade/new version
User Interface
New Product Logo

AnyLogic 7

AnyLogic 7 Model

© The AnyLogic Company | www.anylogic.com
New splash screen graphics
New Welcome page
New workspace layout

Autoclaved Aerated Concrete Factory

autoclaving time left: 00 : 00

© The AnyLogic Company | www.anylogic.com
Old Palette

- By default, docked on the right
- Numerous tabs take valuable space
- Elements do not fit in the visible area, and you have to scroll palette contents with the scroll bar

Only the small part of the view used to display the palette elements
New Palette

- By default, docked on the left
- No more tabs, just icons in the vertical navigation bar (ribbon)
- More space for palette elements
- Collapsible internal sections (e.g. here: Data, Charts)

Collapse/expand by:
- Clicking the section tab
- Using the context menu: Collapse All, Expand All, Collapse Others
New Palette. Navigation between palettes

1. Hover the mouse over the vertical navigation bar

2. You will see the popup list with the palette names. Just choose the palette name from the list.
Properties in AnyLogic 6

Users constantly had to drag the view border up (to increase properties area) and down (to increase the graphical editor).

Horizontal, docked to the bottom
Properties in AnyLogic 7

- Vertical
  - Docked to the right by default
- One page (no tabs anymore)
- Collapsible sections

Legend:

- Static value
- Dynamically evaluated expression

Small triangle indicates that you can switch between design-time (static) and run-time (dynamic) values

Example:

- Fill color: `line.isVisible() ? red : null`
Problems view

Show view automatically when any errors detected

Number of errors

Show/Hide the view

- Infectious cannot be resolved to a variable
- Hanging transition
- Element is not reachable. Connect it to the statechart defined with Entry Point element.
More space for Graphical Editor

Properties moved to the right
Problems and Console are hidden when not needed
New model window design
Graphical Editor
Define coordinates and dimensions as real numbers

Just one case when you may need this

Having chosen the best scale (say, 5 pixels per 1 meter), you draw the simulated platform, stairs, etc.

New data from the customer. With just integer dimensions, you had to change the scale and redraw all graphical shapes completely.
Hiding categories of elements in the Graphical Editor

• Tons of graphical elements on the same diagram:
  – Presentation and 3D animation shapes
  – Space markup
  – Flowcharts
  – SD stock-and-flow diagrams
  – Statecharts
  – …

• Now you can temporarily hide:
  – Presentation shapes
  – Space markup shapes
  – System Dynamics links
  – 2D shapes
  – 3D shapes
Hide Space Markup Elements / Presentation Shapes
Hide System Dynamics Links

Big stock-and-flow diagram may resemble a bowl of spaghetti

Tons of intersecting links may make impossible selecting some variables
Model window border on the agent diagram

Now you know what gets inside the model window just while drawing in the design time.
Improved Usability
Drop-down lists instead of typing

- Drop-down lists enable you just to choose the element, while previously you had to type the element name.
- Whenever you need to define a dynamically evaluated expression, switch to the code field.

Use drop-down lists
Referring to element by a click in the graphical editor

- To make a reference to a graphical object (such as a path, a node, an area) in a logical object (such as queue or delay) you simply click the object in the editor.

Choose shapes in graphical editor
Event. New controls for Recurrence time

- Now you just choose the required time units from the drop-down list instead of typing:
  - $10 \text{ minute()}$
  - $12 \text{ hour()}$
  - ...
Adding shapes to existing group is simpler

- Two modes for choosing the group where to add the shape(s):
  1. Choose the group from the drop-down list
  2. Click the group icon in the graphical editor
Improved Code Completion

Code Completion lists all element’s functions in alphabetical order

Completion suggests only relevant choices

All the functions are useful!
Flowchart blocks connect automatically

- When you add blocks, the closest ports get automatically connected, significantly reducing model design time.
Option List

• New element for defining agent attributes that have limited choice of alternative options
  – Sex (male, female)
  – Marital status (single, married, separated, divorced, widow)

Create from Projects context menu (New|Option List)

Define options in the dialog

In the project tree option lists appear on the upper level
Probability Distribution Wizard

Click the toolbar button when entering the stochastic expression

Delay time: \texttt{triangular(0.5, 1, 1.5)}

PDF Preview

Distribution Parameters

 Delay time: \texttt{exponential(0.0, 10.0, 5.0, 1.0)}
Custom Distribution

- If none of the provided distributions fits the data, you can use the Custom Distribution to define your own one.

- Distribution can return one of three types:
  - Continuous
  - Discrete
  - Options

Choose Option List and specify probability for each option:

- Type: Options. Defines probabilities for options of the chosen option list.
Custom Distribution. Type: Discrete

- For discrete type, there are two definition modes:
  - Ranges
  - Observed samples

- Ranges
  - Describes the distribution as a series of ranges and corresponding weights (numbers of observations)

- Observed samples
  - Use this mode to create the distribution from a file of observed data values:
    \((1, 2, 7, 4, 4, 6, 1, 5, 6, 1)\)
Custom Distribution. Type: Continuous

• Ranges & Observed samples
  – The same as described on the prev. slide, but distribution returns continuous values

• Frequency table
  – You define the distribution as single values with different probabilities/weights.

• Having loaded the data, choose the interpolation:
  – None (defines single values)
  – Linear (defines sloped ranges)
  – Step (analogous to Ranges)

• Check the result with the Preview
Statechart improvements

Trigger type indicators for transitions:

- Timeout
- Rate
- Condition
- Message
- Agent arrival

Unconnected transitions are highlighted in red
Enhanced Support for Agent Based Modeling
“The Great Merge”

- ActiveObject
- AgentContinuous2D
- AgentDiscrete2D
- Environment

Agent

- Entity
- ResourceUnit
- Pedestrian
- Train
- Car
- AgentContinuous3D
The Great Merge: benefits

• You do not need to learn a dozen of different names and concepts. Forget about active objects! All active objects are agents now!!!
• Entities can have individual behavior in parallel to process they’re in
• Agents can dive into and jump out of the process flowcharts
• Resource units can become pedestrians
• People, vehicles, equipment can interact in the same space
• One unified coordinate system for everybody
Wizard for creating agent populations

Drag
Agent dives into a process and jumps out. Demo

1. When arrived to the port, we put the agent (ship) in the flowchart

2. When finished the process defined by the flowchart, we send the message to the agent that triggers the ServiceFinished transition

```java
Entry action: port.arrival.take( this );
```
Agent is Environment

- No more Environment element
- Every agent acts as environment for agent populations that are on its diagram: Country is environment for agents-companies, Company is environment for agents-employees, etc.
Agent. New API for communication

• New handy functions added!

  sendToAll(Object msg) – sends the message to all agents living in the same environment where this agent lives
  sendToRandom(Object msg)
  sendToAllConnected(Object msg)
  sendToRandomConnected(Object msg)

+ functions for Discrete space only:

  sendToAllNeighbors(Object msg)
  sendToRandomNeighbor(Object msg)

• The old functions are still here:

  send(Object msg, Agent dest)
  send(Object msg, MessageDeliveryType mode) – the recipient(s) are determined by mode argument

• Example:

  people.get(0).sendToRandom(“Hi!”);
Default contact network (connections)

- Each agent has a **non-removable link** connections – this is for standard networks with bidirectional connections.

Here you define the reaction on the message reception:

```
Communication
These actions are executed for messages from all connections
Message type: int
On message received:
  count+=msg;
Forward message to:
  Statecharts
    ✓ statechart
```

Links have customizable visual appearance:

```
Animation
  ✓ Draw line connecting agents
  ✓ Draw behind agents
  ✓ Draw on top of agents
  Line color: black
  Line width: 1pt
  Line style: Solid
  Line arrows: Thin arrow
  Arrow position: End
```
Defining custom inter-agent links (contact networks)

- You can add custom links with custom names and visual appearance
- Link can be:
  - Singular
    *(connection to a single agent)*
  - Multiple
    *(connection to a set of agents)*
- Link can be:
  - Unidirectional
    *(child-parent)*
  - Bidirectional
    *(friend-friend)*
Managing links and connections at runtime

- **Single link:**
  - `Agent getConnectedAgent()` - returns the connected agent
  - `connectTo(Agent a)` – connects to the specified agent
  - `boolean disconnect()` - disconnects this agent from the currently connected agent

- **Collection of links:**
  - `LinkedList<Agent> getConnections()` - returns the list of all connected agents
  - `int getConnectionsNumber()` - returns the number of connected agents
  - `boolean isConnectedTo(Agent a)` - tests if this agent is connected to a given agent
  - `Agent getConnectedAgent(int index)` - returns the connected agent with the given index
  - `Agent getRandomConnectedAgent()` - returns the randomly chosen connected agent
  - `connectTo(Agent a)` - adds a given agent to the connections of this agent
  - `boolean disconnectFrom(Agent a)` - disconnects this agent from a given other agent
  - `disconnectFromAll()` - disconnects the agent from all other agents
Custom Link. Communication API

• Single link:
  – `link.send(Object msg)` – sends the message to the connected agent

• Collection of links:
  – `link.send(Object msg, Agent dest)` – sends the message to the specified connection
  – `link.sendToAllConnected(Object msg)`
  – `link.sendToRandomConnected(Object msg)`
Link to upper level agent

- Agent has non-removable visual link to the upper level agent (to the other agent where this one lives) such as `main`, `city`, `company`.

- This is just a pointer, **not a link** – we need it not to send messages, but to access owner parameters and functions.

- Now you refer to the owner agent just by pointer name (`main`, `distributor`, ...)
Inheritance (visually supported)

The elements of the super class are shown pale (and are not editable here)
Space mark-up
One unified coordinate space

Continuous 2D space
Continuous 3D space
Discrete space
Network based modeling 2D space

Space (3D)

Pedestrian space
Process animation space
Space markup

- Path
- Node
- Point node
- Node with attractors
- Pallet racks
- Walls
- Services and queues
- Target lines
- Virtual pathways
- Waiting areas
- Rail track
- Position on track

© The AnyLogic Company | www.anylogic.com
This is what you can create using the space markup:

Semantically rich floor markup for pedestrian models

Networks of auto-connecting nodes, paths, and attractors for processes and agents

Rail networks with curved tracks segments and auto-created switches
Example: Space markup in the Trauma Center model
Networks

- You draw networks with Space Markup elements:
  - Paths
  - Nodes
    - Polygon Node
    - Rectangle Node
    - Point Node
  - Attractors
    - Define exact target / waiting points inside nodes
  - Pallet Racks

- Network is automatically created when two network elements (a path and a node, or two paths) are connected together
Nodes

• Node defines a place in the network where agents can reside

• There are three types of nodes:
  – Rectangular node
  – Polygonal node
  – Point node. Usually represents a transit node in the network.

• You can define exact target points inside nodes using attractors

• Besides acting as a node in a network, any single node may define an entity location for flowchart blocks (Delay, Combine, Batch, etc.)
Paths

• Path can contain both linear and curved segments. This enables creating more natural movement routes

• Paths can be unidirectional (one-way movement)

• Besides acting as a route in a network, any single path may define an entity location for flowchart blocks (Delay, Queue, etc.)
Drawing paths. Rules and Hints

- **Draw paths using the drawing mode**
  - To add one more linear segment, just click in a new point
  - To add a curved segment, press the left mouse button in a new point and move the mouse with the button being pressed. You will see how the curving radius changes.
  - To get the circular segment, move the mouse exactly along the grid line.

- **Want to adjust the drawn path?** Right-click and choose Edit guiding lines, then drag the guiding lines end points
Path. Continue drawing from start/end point

1. Right-click the path, choose Append line

2. Click the end point where you want to add path segment(s)

3. Continue drawing the path from this point as usual
Path. Curved <-> Linear transformations

Linear -> Curved

1. Right-click the path, choose Edit Using Guiding Lines
2. Drag the guiding line end point

Curved -> Linear

1. Select the path.
2. Ctrl-drag the point
3. The neighboring segments will become linear
Converting network shapes to new space markup

1. Right-click the group of network shapes, choose Convert | Convert to Network Elements
Agent. New API for movement

• You make agent move by calling one of its movement functions
  – moveTo() – agent moves to the specified location (with the agent’s speed)
  – moveToInTime() – agent moves to the specified location. You specify the trip time as the function argument
  – jumpTo() – agent instantly jumps to the specified location in zero model time
  – moveToNearestAgent(agents) – new handy function that starts movement to the nearest agent from the given population

• Related functions:
  – getX(), getY(), getZ()
  – getVelocity(), setVelocity()
  – getTargetX(), getTargetY(), getTargetZ()
  – distanceTo()
  – stop()
Movement functions. New options

• \textit{moveTo()} and \textit{moveToInTime()} functions provide several notations, enabling defining target location in different ways:

• You can define location as:
  – X, Y coordinates:
    – \textit{moveTo(double x, double y)}
  – X, Y, Z coordinates:
    – \textit{moveTo(double x, double y, double z)}
  – Point:
    – \textit{moveTo (Point location)}
  – Attractor:
    – \textit{moveTo (Attractor attractor)}
  – Network node:
    – \textit{moveTo (Node node)}
  – Network node and location inside the node – (x,y,z) point:
    – \textit{moveTo (Node node, Point location)}
Process Modeling Library
New library for discrete event (process) modeling

• "Pull" instead of "Push": natural entity flow
  – Flowchart objects will stop and wait until the next objects are ready
  – "Push" is still available as an option

• Flexible resource management
  – All kinds of resources (regular and network) are unified
  – Request particular resources, allow alternative sets
  – Preparation and wrap-up
  – Preemption, interrupts
  – Shifts, breaks

• “Internals” of entities defined graphically

• Conveyor connections

• Warehouse storages
“Pull” entity flow

- AL 6. Enterprise Library: “Push” protocol. Entities are pushed further regardless the state of the succeeding block.
- In “Pull”, entity goes further only when the next block is ready to take it.

<< things back up | Failure
Pull vs. push
AnyLogic 6. User's "Pull" protocol implementation

Manufacturing model

[Diagram of manufacturing model]
Blocks supporting “pull” protocol

- Source
- Enter
- Delay
- Pickup
- Dropoff
- Seize
- Batch
- Unbatch
- Combine

Selecting the option you turn the PUSH protocol on for this block
Flexible resource management

• Regular and "network" resources are merged into Resource
• Entities can request specific resources, allow alternative sets
  – Seize at once, seize one by one, ...
• Preparation and wrap-up as a process
  – Resource may seize other resources
• Preemption, interrupts
• Shifts, breaks
New resources

• Regular and "network" resources are merged into Resource

  "Regular" resource → Resource

  Network-based resource

  • Static
  • Portable
  • Moving

• Resource Pools do not require graphic connection to the blocks using resources in the flowchart (Seize, Assemble, Release, etc.)
Creating custom entity / resource types

1. Drag Entity type / Resource type

2. Choose the animation and fix the parameters using the wizard

3. You may add any other elements (statecharts, etc.) on the agent diagram
Entity “internals” are now defined graphically
Preparation and wrap-up branches for resources

These processes do not run in parallel, the preparation process starts only when the entity gets to this Seize block.

Main patient flow >>

Nurse prepares portable ultra sound

Preparation flowchart branch for a resource

Nurse returns ultra sound

Wrap-up flowchart branch
Alternative sets of resources
Priorities and task preemption policies

- When some other task is initiated, you may choose what to do with the task being executed currently:
  - No preemption (the current task continues executing)
  - Wait for original resource (the task is interrupted and waits for the same resource unit to finish it)
  - Terminate serving (the task is interrupted and never continued)
  - Seize any resource (the task is interrupted and tries to seize any resource of the specified type)
  - Continue without resource (the task is finished without resources assistance)
Here you choose how the failure time is taken into account in the resource usage statistics.

Alternative to modeling a failure/repair just with a delay, you can model it with a flowchart describing the process. The flowchart start is defined with the specified `ResourceTaskStart` block.
Resource. Breaks

ResourcePool

1. Define the breaks pattern with a schedule
2. Set the priority of the Break task

If selected, break will preempt the currently performed task (if allowed in the task properties & the task priority is lower)

‘Break’ preemption policy:
- **Terminate** – the break is terminated, new task starts
- **No preemption** – no tasks can stop the break
Resource. Maintenance

**ResourcePool**

- **Shifts, breaks, failures, maintenance...**
  - **Maintenance:**
    - Initial time to maintenance: \( \text{uniform}(0, 1000) \)
    - Time to next maintenance: \( \text{triangularAV}(1000, 0.1) \)
  - 'Maintenance' priority: 100
  - 'Maintenance' may preempt:
  - **Maintenance type:** Send to flowchart
  - **Task start block (maintenance):** resourceTaskStart
  - **Usage statistics are:** counted as 'idle'

**Maintenance type:**
- *Delay*
- *Send to flowchart* (that starts with the specified ResourceTaskStart block and ends with ResourceTaskEnd)
Resource. Other tasks

You may need to define some other task that cannot be described by previously mentioned task patterns.

First, define the task(s) with the block(s) **ResourceTask**

Then choose the task(s) in the **ResourcePool**’s **List of tasks**
Modeling warehouse storages

- Special space markup shape: 
  Pallet Rack (ex- flowchart block NetworkStorage)  
  – automatically connects to networks

- Similar flowchart blocks for defining the logic:  
  – RackSystem (ex- NetworkStorageZone)  
  – RackStore (ex- NetworkStoragePut)  
  – RackPick (ex- NetworkStoragePick)

- Due to the changes in internal topology definition mechanism, gained up to 5X PERFORMANCE IMPROVEMENT at the model initialization stage
Pallet Rack

• Three topologies are supported:
  
  a) One rack, one aisle
  b) One rack, two aisles
  c) Two racks, one aisle

• Vertical levels are animated in 3D
Multi Deep Pallet Racks

![Double-deep pallet rack]

**palletRack - Pallet Rack**

- Name: palletRack
- Visible: Yes
- Type: One rack, one aisle
- Cell width: 20
- Number of deep positions: 2
- Number of levels: 1
- Level height: 50
Pallet Flow/Gravity Flow Racking Systems

This system uses a slightly inclined rail with rollers that allow pallets to move easily along the sloped plane.

Valid for *One rack, two aisles* only
- *No direction* – No gravity flow racking
- *Left to right, Right to left* – pallet flow flow racking applied.

The arrow on the graphical diagram shows the pallet flow direction.
True “physical” connection of conveyors

Conveyor

- Grab entity from prev conveyor:
- Change entity length:
  - Entity length: 10
  - Instantly, on full enter

Advanced

- Smoothly, matching speed
- Instantly, on enter
- Instantly, on full enter

Baggage Claim Simulation

© The AnyLogic Company | www.anylogic.com
Animation for entities that are moved by conveyors

Put animation’s front side in (0,0)
Queue. More queuing disciplines

FIFO (First In First Out)

LIFO (Last In First Out)

Priority-based (does not require turning on Preemption anymore)

Entity comparison
SelectOutput with the required number of exits

- Using **SelectOutputIn** and a set of **SelectOutputOut** blocks you can create a quasi-block **SelectOutput** with the required number of exits
  - You do not need to connect these blocks together graphically
Converting models to the new DE library

1. Select the whole flowchart, right-click the selection and choose **Convert to new library**. The command is available if only flowchart blocks and connectors are in the selection.

In the Log view, you can see the list of the things that should be converted manually.
3D Animation
Common Presentation palette for both 2D and 3D

- AnyLogic 6: two palettes: **Presentation** and **3D**
- To bring shape to 3D you had to check the shape’s option **Show in 3D**

- AnyLogic 7: just one palette **Presentation**
- By default **all shapes** are shown **both in 2D and 3D** *

* Text shapes by default are shown in 2D only (since they are used mostly for decoration)
Different animation shapes for 2D & 3D

• By default all shapes are shown both in 2D and 3D
• Sometimes you may need to animate the same object with different 2D and 3D shapes
• You can draw two animations, and set one shape to be shown only in 2D, while another only in 3D (in the shape’s Advanced property section, Show in property)
• Example: pedestrians
More 3D objects

• Hundreds of new ready-to-use 3D objects:
  - People
  - Buildings
  - Road, rail, maritime transport
  - Hospital equipment
  - Manufacturing
  - Energy
  - Warehouses
  - CNC Machines, and so on

• Standard objects optimized for fast rendering
Painting standard and imported 3D objects

You can change the color of any component of 3D model in the Colors property section.

Add and color more objects in the same way.
Run the model. You will see 3D models of different colors both in 2D and 3D.
Creating animated 3D objects

• You can create animated 3D objects by taking several 3D models animating different object states and playing with Visible property.

• Let’s draw the house changing its look when the sun goes down
  – Drag House 3D object. Write in the Visible property:
    `getHourOfDay() > 8 && getHourOfDay() < 20`
  – Drag House Night and place it exactly where the House 3D is. Visible:
    `getHourOfDay() <= 8 || getHourOfDay() >= 20`
Paths in 3D

- Built-in decoration for:
  - Railway track
  - Conveyor
  - Road
Predefined light types

- Street light
- Car headlight
- Daylight
- Moonlight

Example: Car headlight definition
Highly Improved Pedestrian Library
Highly Improved Pedestrian Library

- Performance increase
- Rich pedestrian-specific space markup
- Very compact flowcharts
- Zero coding

These markup and flowchart fully describe pedestrian flows in a large shopping mall.
Which facilities are modeled?

- Railway stations
- Metro stations
- Airports
- Car parks
- Pedestrian passageways

In general all the facilities where the arrangement of physical space for pedestrians affects throughput capacity, quality of service, and safety transport

- Shopping malls
- Museums
- Amusement parks

“attractions”

- Stadiums
- Concert halls
- Worship facilities
- Street events (festivals, rallies, demonstrations)

events

- As well as production, warehouse and even movements of personnel in a kitchen...
Performance Improvement

• Pedestrian Library has upgraded to a new high performance engine. You can now scale pedestrian models into tens of thousands or higher without impacting performance.

• Radical performance increase (7x - 9x)

• Better route management for pedestrians
New Demo Model: MEGA Shopping Mall
Individual Features of a Pedestrian

• Since each pedestrian is modeled as a specific object, individual features can be adhered to them

• These can be added to them:
  – Individual targets (flight, platform, shop)
  – Servicing class (first / business / economy)
  – Citizenship (US, EU, ...)
  – Servicing speed
  – ...

• These features can be checked during the pathway of a pedestrian throughout a process diagram and affect their behavior
Changes in flowchart block set

Pedestrian Library (old)
- Ped Configuration
- Ped Ground
- Ped Source
- Ped Sink
- Ped Go To
- Ped Wait
- Ped Service
- Ped Change Ground
- Ped Enter
- Ped Exit
- Ped Select Output
- Ped Area
- Ped Services
- Ped Attractor
- Ped Flow Statistics
- Ped Density Map Legend
- Ped Group Assemble
- Ped Group Change Form
- Ped Group Disassemble
- Ped Tie
- Ped Cmd Go To
- Ped Cmd Wait
- Ped Cmd Queue
- Ped Cmd Service

Pedestrian Modeling Library
- Pedestrian Type
  - Space Markup
    - Well
    - Rectangular Wall
    - Circular Wall
    - Target Line
    - Service With Lines
    - Service With Area
    - Rectangular Area
    - Polygonal Area
    - Attractor
    - Pathway
    - Ped Density Map
    - Ped Flow Statistics
- Blocks
  - Ped Source
  - Ped Sink
  - Ped Go To
  - Ped Service
  - Ped Wait
  - Ped Select Output
  - Ped Enter
  - Ped Exit
  - Ped Change Ground
  - Ped Area Descriptor
  - Ped Group Assemble
  - Ped Group Change Formation
  - Ped Group Disassemble
  - Ped Settings

Alternative found
Substituted with space markup shapes
Removed blocks modeling commands at the low level
Simplifying the first steps of the model creation

• In AnyLogic 6, you always started with adding mandatory \textit{PedConfiguration} and \textit{PedGround} blocks.

• No more \textbf{PedGround} object. The space markup shapes (walls, services, etc.) compose the ground automatically.

• \textbf{PedConfiguration} is not necessary to create a pedestrian model anymore. Few advanced functions moved to \textit{PedSettings} block that is optional and is used in a limited number of cases.
Space Markup elements for pedestrian simulation

- Wall
- Rectangular Wall
- Circular Wall
- Target Line
- Service With Lines
- Service With Area
- Rectangular Area
- Polygonal Area
- Attractor
- Pathway
- Pedestrian Density Map
- Ped Flow Statistics

- Queues
- Linear service
- Area
- Wall
- Pathway
- Pedestrian density map
- Target line
Walls

- **Wall**
  - Exterior walls, interior walls, guard rails
- **Rectangular Wall**
  - Working spaces
- **Circular Wall**
  - Columns, fountains, skating-rinks

- **Wall. Draw it in the drawing mode, exactly as the path:**
  - Click to draw a linear wall
  - Press the mouse button and move the mouse: draw a curved wall
Services

• Service with lines
  – Turnstiles, cash desk, passport checkpoint

• Service with area
  – Electronic queue (ticket office, bank office, info point)
Group behavior in services

Only the group leader stands in a line, others just wait aside
Tourist groups with guides

All group members stand in a line, only one is serviced
Couples in the cinema tickets line

All group members are serviced individually
Turnstiles
Stairs, Escalators, Roads, Floors

• To set sloped surfaces as well as to modify speed on a surface PedArea with a Sloped option and its ”descriptor” is used

  PedAreaDescriptor
  
  Limits / multiplies speed of pedestrians or sets a moving surface

• Each markup element belongs to a certain level
  – By default it is a ground level, but you may set new ones

• Special object is used to move between the levels:

  PedChangeGround
  
  Moves pedestrians from one level to another
Stairs, etc. Demo
Attractors in Pedestrian Areas

• With attractors you can define exact waiting points inside the area

Add attractors just by dragging them inside the area, or use the Wizard
Pathway. Virtual corridor for pedestrian flow

- With **Pathway** you may tell pedestrians to follow a specified route.
- Example:
  - Corridor in a subway station where passengers go in the opposite directions.
Pedestrian Density Map

• Show and estimate density of pedestrians in the simulated space with Density Map

• Tune up:
  – Attenuation
  – Transparency
  – Critical density level
  – Color scheme (choose from logarithmic, linear, custom)
Pedestrian Flow Statistics

• You can easily monitor the passenger flow passing through the corridor, doors, etc.

• Markup element **PedFlowStatistics** defines the cross-section

• Get numbers using the element functions:
  
  `traffic()`
  
  `intensity()`

Choose the direction of the monitored flow.
Converting models to the new library. Step 1. Walls

1. Right-click the group of shapes defining walls, choose Convert to | Walls

2. If the area inside the walls is drawn hatched, uncheck the Wall option Closed

You can convert only walls and target lines, not services and queues
Converting models to the new library. Step 2. Blocks

1. Select the whole flowchart, right-click the selection and choose Convert to new library.

2. Blocks are substituted with analogs from the new library, but some parameter values are lost.
New Rail Library
What problems can be efficiently solved?

- Classification yards
- Rail car repair facilities
- Rail in container terminals
- Rail transportation in a coal mine
- ...

In general **Rail Library** allows you to efficiently model, simulate and visualize operation of rail yards and rail transportation of any complexity and scale

- Railway stations
- Subways
- Airport shuttle trains
- Trams
Previously you had to draw tracks with polylines, switches with circles, and put “track” polyline end points inside the “switch” circles.

Now the railway tracks are connected automatically.
Graphical elements for drawing rail tracks

- Rail tracks: auto-connect and auto creation of switches
- Curved track segments
3D railways

• Previously users had to draw rails and ties by their own using replicated shapes, etc.

• Now you just drag Railway tracks on the canvas and get 2D and 3D animation automatically
Full collection of rail car 3D models

- All types and sizes of US rail cars
- All 3D models are in the same scale
Miscellaneous
Changes in Evaluation license limitations

<table>
<thead>
<tr>
<th>The maximum number of elements in one model</th>
<th>AnyLogic 6</th>
<th>Anylogic 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent types</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>Flowchart blocks + agent populations</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>Option lists</td>
<td>n/a</td>
<td>5</td>
</tr>
</tbody>
</table>
New SD Healthcare Models

- Chronic Treatment Indication Dynamics
- Clinical Process Concepts
- Pharmaceutical Pipeline And Medicines Use
- Hospital Admission Emergency Elective Interaction
- Diabetes Progression
- Patient Care Process Swimming Pool Metaphor
New time units: months and years

• Essential for System Dynamics modelers

• We use these values:
  – Month contains 30 days
  – Year contains 365 days

Set the time units in the model properties

See them displayed in the status bar
Easy model attach on contacting Support

- Attach your AnyLogic model (you may choose a model on the next page)
- Enter your message and press Submit to send it (Internet connection required):

  Dear AnyLogic Support Team!
  Could you please help me with fixing a runtime error in my model.
  Thanks a lot!

Support Services expiration date: 23 January 2014
Java Editor

Collapsible sections

Line numbers helpful to address the specific line when communicating with Support

```
package aac_factory;

import java.io.Serializable;

public class UnloadingCrane extends Agent {
    // Parameters
    public double speed;

    @AnyLogicInternalCodegenAPI
    public double _speed_DefaultValue_xjal() {
        final UnloadingCrane self = this;
        return 16/minute;
    }

    public void set_speed(double speed) {
        if (speed == this.speed) {
            return;
        }
        double _oldValue_xjal = this.speed;
        this.speed = speed;
        onChange_speed_xjal(_oldValue_xjal);
        onChange();
    }

    /**
     * Calls "On change" action for parameter speed.
     * Note that 'oldValue' in that action will be unavailable if this method is called twice
     * (current parameter value will be passed as 'oldValue').
     * Please call <code>set_speed()</code> method instead.
     *
     */
    void onChange_speed() {
        onChange_speed_xjal(speed);
    }
```
Customizable model report

New Document Formats

Customize the report contents: choose only the elements and categories you need
Copy any graphical elements as image

1. Select the elements
2. Choose Edit | Copy as Image from the menu
3. Paste the image
Scale

• Two systems of measurement supported:
  – Metric (cm, m, km)
  – English (miles, inches)

• Scale element

  – The default scale is 10 pixels in 1 meter
  – Now can define the scale for the agent animation. It will be taken into account when drawing agent on environment animation
  – Later it will enable users to define the scale for the whole model, so you will be able to refer to physical length units, not pixels while defining sizes and distances
triangularAV() probability distribution function

- Generates a sample of the Triangular distribution with mode set to average.
- This is just a helper for specifying distribution in the form like "roughly this, +/-20%“.
- Parameters:
  - **average**: the most likely x value
  - **variability**: the percent [0...1] of average representing the half of distribution range, where the generated sample falls
- Is equivalent to:
  \[
  \text{triangular} \left( \text{average} \ast (1 - \text{variability}), \text{average} \ast (1 + \text{variability}) \right)
  \]
Improved API for schedule initialization on startup

• AL 6: just one constructor, the most ugly piece of code

```java
new Schedule<ValueType>(Utilities owner, boolean calendarType, int firstDayOfWeek, long period, long timeUnits, Long snapTo, <ValueType> defaultValue, long[] starts, long[] ends, Object[] values, boolean glueIntervals, boolean[] exceptionsAnnually, boolean singleThreadMode)
```

**Example:** `Schedule<Integer> NursePlan = new Schedule<Integer>(this, true, MONDAY, 1L * TIME_UNIT_DAY, TIME_UNIT_DAY, null, 0, varNursePlanTime, null, varNursePlanNumber, false, null, true );`

• AL 7: new constructor and handy methods

  – **Constructor:**
    - `Schedule ( )` – creates empty schedule with no data
  – **Methods that initialize the schedule:**
    - `addInterval(), addMoment(), addException()`
    - `setFirstDayOfWeek(), setSnapTo(), setDefaultValue(), setPeriod(), setTimeUnits(), setCalendarType(), setGlueIntervals()`
• All element functions described in the Help in a handy way
New Standard Training Program

- New program takes only the best exercises from both existing courses (and some new are added)

**DAY 1**
- Introduction to multimethod simulation modeling – theory and demo
- Getting familiar with AnyLogic model development environment – demo
- Discrete Event (Process-Centric) modeling – theory
- Factory model – Discrete Event exercise
- System Dynamics – theory
- Bass Diffusion model – System Dynamics exercise

**DAY 2**
- Agent Based modeling – theory
- SIR Model. Calibration
- Java basics for AnyLogic modelers – theory
- Windmill turbines maintenance model. Optimization
- Supply Chain GIS model – exercise

**DAY 3**
- Pedestrian modeling – theory
- Small airport model – exercise
- Call Center model - self-paced Discrete Event exercise
- Hump Yard – rail model
- Questions & Answers
Fixes and minor improvements

• User Interface
  – Welcome page restores the last opened tab/section (extremely helpful when working with example models)
  – Fixed the bug with disappearing toolbar on Welcome minimization

• Graphical editor
  – You can select the elements that are behind the locked shape
  – Filled polyline can now be selected by clicking not just its outline, but also any its internal point

• Agents
  – Added deleteSelf() function
  – Improved performance of agent networks creation and agent connect/disconnect operations
  – Added function setParameter() which allows to set parameter by its name (e.g. for initializing model from custom data source by using java code)

• Connectivity
  – Query: Fixed reading dates from some types of databases (time wasn't read in some cases)

• System Dynamics
  – Shadow variable has the same color as its parent
Fixes and minor improvements

• **General**
  – Parameter. In *On change* code, old value of the parameter is accessible as *oldValue*.
  – Collection. Added *Initial contents* field

• **Presentation**
  – Rectangle. Added *getCenter()* function
  – Oval and Arc. “Circular” option

• **3D**
  – Fixed strange 3D scene lighting when there are more than 7 sources of light

• **Controls**
  – File chooser now initially opens the current working folder (usually this is the model folder)

• **Numerical methods settings** moved from experiment to model properties

• **Engine**
  – Added function for executing model action from custom (e.g. UI) thread: `Presentation.executeModelActionLater()`

• **Clock element** moved to *Pictures* palette
GIS: Alternative Solution
While we are preparing to improve our built-in GIS...

• ... AnyLogic Consulting Team have developed new, alternative GIS solution.

• This solution enables the users to:
  – Display Tile map on model presentation with pan and zoom. Downloaded tiles are cached in model folder.
  – Switch different tile sources in runtime.
  – Display animation of agents living in GIS space shown on top of map.
  – Perform geocoding/reverse geocoding by parsing http query response.
  – Get routing data by parsing http query response.

• Requires Internet connection to download tiles and data from OpenStreetMap
Implementation

- **TileGIS Library**: takes empty radio buttons as a placeholder for the displayed map.
- **TileSourceSelector**: should be linked to TileGIS object and allows switching sources for tiles using its animation.
Display routes and points on the map

- Supports two types of map specific objects displayed on top of the map:
  - PointOverlay: used to display points on the map as a circle with some text
  - PathOverlay: designed to display routes on top of the map
- TileGIS component is also able to display AnyLogic agents living in GIS space
  - (You will need AnyLogic GIS map in your model and set GIS space for the agent populations).
Example: Searching for Manhattan pharmacies

• Here you can see the “pharmacy” search results being displayed on the map:
Some useful links

• OpenStreetMap:
  – http://wiki.openstreetmap.org/

• Routing:
  – http://open.mapquestapi.com/directions/

• Alternative routing with offline capabilities (cached routing graph, fast and offline):
  – http://graphhopper.com/

• Geocoding/Reverse geocoding/Map search:
  – http://open.mapquestapi.com/nominatim/
  – http://open.mapquestapi.com/xapi/