

ANDREI BORSHCHEV

my

Multimethod Modeling with AnyLogic 6

**AnyLogic North America
2013**

Contents

CHAPTER 1. MODELING AND SIMULATION MODELING	25
1.1. TYPES OF MODELS	26
1.2. ANALYTICAL vs. SIMULATION MODELING	27
<i>The limits of analytical modeling: queuing theory</i>	29
<i>Advantages of simulation modeling</i>	34
1.3. APPLICATIONS OF SIMULATION MODELING. LEVEL OF ABSTRACTION. METHODS	34
CHAPTER 2. THE THREE METHODS IN SIMULATION MODELING	37
2.1. SYSTEM DYNAMICS	38
<i>Example 2.1: New product diffusion</i>	39
<i>Underlying mathematics and simulation engine</i>	44
<i>Abstraction level</i>	44
<i>Software tools</i>	45
2.2. DISCRETE EVENT MODELING	45
<i>Example 2.2: Bank</i>	46
<i>Abstraction level</i>	47
<i>Underlying mathematics and simulation engine</i>	48
<i>Software tools</i>	49
2.3. AGENT BASED MODELING	49
<i>Example 2.3: Agent based epidemic model</i>	50
<i>Abstraction level</i>	54
<i>For those who have read books and papers on agent based modeling</i>	54
<i>Underlying mathematics and simulation engine</i>	55
<i>Software tools</i>	55
CHAPTER 3. AGENT BASED MODELING. TECHNOLOGY OVERVIEW	57
3.1. WHO ARE THE AGENTS?	57
<i>Who are the agents in an American automotive market model?</i>	58
3.2. AGENT BASED MODELING AND OBJECT-ORIENTED DESIGN	60
<i>OO modeling in AnyLogic</i>	S3
3.3. TIME IN AGENT BASED MODELS	67
3.4. SPACE IN AGENT BASED MODELS	69
3.5. DISCRETE SPACE	71
<i>Example 3.1: Schelling segregation</i>	74
<i>Example 3.2: Conway's Game of Life</i>	76
<i>Example 3.3: Wildfire</i>	78

The Big Book of Simulation Modeling · Multimethod Modeling with AnyLogic 6

<i>Discrete space API</i>	85
3.6. CONTINUOUS 2D AND 3D SPACE	86
<i>Movement in continuous space</i>	88
<i>Example 3.4: Air defense system</i>	90
<i>Example 3.5: Agent leaving a movement trail</i>	105
<i>Continuous space API</i>	109
3.7. NETWORKS AND LINKS	110
<i>Standard networks</i>	///
<i>Example 3.6: Periodic repair of a standard network</i>	113
<i>Example 3.7: Custom network built using standard connections</i>	116
<i>Fully connected networks</i>	118
<i>Network and layout-related API</i>	118
<i>Unidirectional, temporary, and other custom types of links</i>	119
<i>Example 3.8: Kinship modeled using custom links</i>	122
<i>A note on vertical links in hierarchical models</i>	127
<i>Using ports to connect agents</i>	127
3.8. COMMUNICATION BETWEEN AGENTS. MESSAGE PASSING	128
<i>Synchronous and asynchronous communication</i>	128
<i>API for message passing</i>	129
<i>Message handling</i>	131
<i>Other types of inter-agent communication</i>	131
3.9. DYNAMIC CREATION AND DESTRUCTION OF AGENTS	132
3.10. STATISTICS ON AGENT POPULATIONS	134
<i>Example 3.9: Kinship model with standard statistics</i>	134
<i>Example 3.10: Kinship model with dynamic histograms</i>	136
<i>Customized high performance statistics</i>	138
<i>Example 3.11: Kinship model with customized statistics</i>	139
3.11. CONDITION-TRIGGERED EVENTS AND TRANSITIONS IN AGENTS	140
CHAPTER 4. HOW TO BUILD AGENT BASED MODELS. FIELD SERVICE EXAMPLE	142
4.1. THE PROBLEM STATEMENT	143
4.2. PHASE 1. CAN BE DONE ON PAPER	144
<i>Who are the agents?</i>	144
<i>Equipment unit agent</i>	145
<i>Service crew agent</i>	147
<i>Agent communication. Message sequence diagrams</i>	150
<i>Space and other things shared by all agents</i>	153
4.3. PHASE 2. THE MODEL IN ANYLOGIC. THE FIRST RUN	154
<i>The model structure and the top level object Main</i>	154

<i>The EquipmentUnit agent</i>	155
<i>The ServiceCrew agent</i>	158
<i>Animation</i>	159
<i>The first run</i>	160
<i>Discussion and next steps</i>	161
4.4. PHASE 3. THE MISSING FUNCTIONALITY	162
<i>Maintenance, age, and failure rate</i>	162
<i>Scheduling maintenance. Handling requests of two types</i>	164
<i>Discussion. Code in the model</i>	167
4.5. PHASE 4. MODEL OUTPUT. STATISTICS. COST AND REVENUE CALCULATION	169
<i>Equipment availability and service crew utilization</i>	169
<i>Cost and revenue</i>	174
4.6. PHASE 5. CONTROL PANEL. RUNNING THE FLIGHT SIMULATOR	177
<i>Design of control panel</i>	177
<i>Changing the number of service crews</i>	178
<i>Equipment replacement policy</i>	179
<i>Running the flight simulator</i>	180
4.7. PHASE 6. USING THE OPTIMIZER TO FIND THE BEST SOLUTION	182
<i>Preparing the model for optimization</i>	182
<i>Setting up the optimization experiment</i>	183
<i>Optimization run</i>	185
4.8. ASSUMPTIONS	187
4.9. BONUS PHASE. 3D ANIMATION	188
4.10. BONUS DISCUSSION. COULD WE MODEL THIS IN DISCRETE EVENT STYLE?	191
CHAPTER 5. SYSTEM DYNAMICS AND DYNAMIC SYSTEMS	195
5.1. How TO DRAW STOCK AND FLOW DIAGRAMS	196
<i>Drawing stocks and flows</i>	196
<i>Drawing variables, dependency links, polarities, and loop types</i>	197
<i>Naming conventions for system dynamics variables</i>	199
<i>Layout of large models. "Sectors" and shadow variables</i>	199
5.2. EQUATIONS	202
<i>Using Java in SD equations</i>	204
<i>"Constant variables" and parameters</i>	205
<i>Units and unit checking</i>	206
5.3. EXAMPLE: POPULATION AND CARRYING CAPACITY	208
<i>Phase 1: Unlimited resources. Positive feedback. Exponential growth</i>	208
<i>Customizing the dataset collection</i>	212
<i>Phase 2: Crowding affects lifetime. Negative feedback. S-shaped growth</i>	213

<i>Phase 3: Crowding affects births</i>	216
<i>Phase 4: Negative feedback with delay. Overshoot and oscillation</i>	218
<i>Specifying units and performing unit checking</i>	220
5.4. OTHER TYPES OF EXPERIMENTS. INTERACTIVE GAMES	221
<i>Example 5.1: New product diffusion - compare runs</i>	221
<i>Example 5.2: New product diffusion - sensitivity analysis</i>	224
<i>Example 5.3: Epidemic model-calibration</i>	225
<i>Example 5.4: Epidemic model - instant charts</i>	229
<i>Example 5.5: Stock management game</i>	232
5.5. EXPORTING THE MODEL AND PUBLISHING IT ON THE WEB	237
CHAPTER 6. MULTI-METHOD MODELING	240
6.1. ARCHITECTURES	241
<i>The choice of model architecture and methods</i>	243
6.2. TECHNICAL ASPECT OF COMBINING MODELING METHODS	244
<i>Examples 5.1 - 5.21: Combining modeling methods</i>	244
<i>System dynamics -> discrete elements</i>	244
<i>Discrete elements -> system dynamics</i>	246
<i>Agent based <-> discrete event</i>	253
<i>Referencing model elements located in different active objects</i>	257
<i>The simulation performance of multi-method models</i>	259
6.3. EXAMPLES	259
<i>Example 6.22: Epidemic and clinic</i>	259
<i>Example 6.23: Consumer market and supply chain</i>	267
<i>Example 6.24: Product portfolio and investment policy</i>	272
6.4. DISCUSSION	285
CHAPTER 7. DESIGNING STATE-BASED BEHAVIOR: STATECHARTS	287
7.1. WHAT IS A STATECHART?	287
<i>Example 7.1: A laptop running on a battery</i>	287
<i>How do statecharts differ from action charts and flowcharts?</i>	288
7.2. DRAWING STATECHARTS	289
<i>Simple states</i>	290
<i>Transitions</i>	290
<i>Statechart entry point</i>	291
<i>Composite states</i>	292
<i>History state</i>	293
<i>Final state</i>	294
7.3. STATE TRANSITIONS: TRIGGERS, GUARDS, AND ACTIONS	294

<i>Which transitions are active?</i>	294
<i>Trigger types</i>	295
<i>Timeout expressions</i>	296
<i>Transitions triggered by messages</i>	297
<i>Sending messages to a statechart</i>	299
<i>Guards of transitions</i>	300
<i>Transitions with branches</i>	301
<i>Internal transitions</i>	302
<i>Order of action execution</i>	303
<i>Synchronous vs. asynchronous transitions</i>	304
7.4. STATECHART-RELATED API	306
7.5. VIEWING AND DEBUGGING THE STATECHARTS AT RUNTIME	307
7.6. STATECHARTS FOR PEOPLE'S LIVES AND BEHAVIOR	308
<i>Example 7.2: Life phases</i>	308
<i>Example 7.3: Adoption and diffusion</i>	309
<i>Example 7.4: Disease diffusion</i>	310
<i>Example 7.5: Purchase behavior with a choice of two competing products</i>	311
<i>in</i> . STATECHARTS FOR PHYSICAL OBJECTS	313
<i>Example 7.6: Generic resource with breakdowns and repairs</i>	313
<i>Example 7.7: Delivery truck</i>	314
<i>Example 7.8: Aircraft maintenance checks</i>	314
7.8. STATECHARTS FOR PRODUCTS AND PROJECTS	316
<i>Example 7.9: Product life cycle, including NPD</i>	316
<i>Example 7.10: Pharmaceutical NPD pipeline</i>	318
7.9. STATECHARTS FOR TIMING	319
<i>Example 7.11: Statechart for shop working hours</i>	319
CHAPTER 8. DISCRETE EVENTS AND EVENT MODEL OBJECT	321
8.1. DISCRETE EVENTS	321
<i>The terminology</i>	321
<i>Discrete events: approximation of real world continuous processes</i>	321
<i>Discrete event management inside AnyLogic engine</i>	323
8.2. EVENT-THE SIMPLEST LOW LEVEL MODEL OBJECT	324
<i>Example 8.1: Event writes to the model log every time unit</i>	326
<i>Example 8.2: Event generates new agents</i>	328
<i>Events triggered by a condition</i>	329
<i>Example 8.3: Event waits on a stock reaching a certain level</i>	330
<i>Example 8.4: Automatic shutdown after a period of inactivity</i>	331
<i>Example 8.5: Event slows down the simulation on a particular date</i>	333

<i>Event API</i>	334
8.3. DYNAMIC EVENTS	335
<i>Example 8.6: Product delivery</i>	336
<i>API related to dynamic events</i>	338
CHAPTER 9. RAILS AND TRAINS	340
9.1. DEFINING THE RAIL TOPOLOGY	341
<i>Example 9.1: A very simple rail yard</i>	342
<i>3D animation of rail yards</i>	345
<i>Creating rail yards programmatically</i>	345
<i>Example 9.2: Creating a rail yard by code</i>	346
<i>Java class Track</i>	347
<i>Java class Switch</i>	349
9.2. DEFINING THE OPERATION LOGIC OF THE RAIL MODEL	350
<i>Example 9.3: Train stop</i>	352
<i>Example 9.4: Ensuring safe movement of trains</i>	357
<i>Example 9.5: Simple classification yard</i>	363
<i>Example 9.6: Airport shuttle train (featuring AnyLogic Pedestrian Library)</i>	372
<i>Java class Train (subclass of Entity)</i>	375
<i>Java class RailCar</i>	377
CHAPTER 10. JAVA FOR ANYLOGIC USERS	380
10.1. PRIMITIVE DATATYPES	381
10.2. CLASSES	382
<i>Class as grouping of data and methods. Objects as instances of class</i>	382
<i>Inheritance. Subclass and super class</i>	383
<i>Classes and objects in AnyLogic models</i>	385
10.3. VARIABLES (LOCAL VARIABLES AND CLASS FIELDS)	387
<i>Local (temporary) variables</i>	387
<i>Class variables (fields)</i>	388
10.4. FUNCTIONS (METHODS)	389
<i>Standard and system functions</i>	390
<i>Functions of the model elements</i>	392
<i>Defining your own function</i>	394
10.5. EXPRESSIONS	396
<i>Arithmetic expressions</i>	396
<i>Relations and equality</i>	398
<i>Logical expressions</i>	398
<i>String expressions</i>	399

<i>Conditional operator ?:</i>	400
10.6. JAVA ARRAYS AND COLLECTIONS	400
<i>Arrays</i>	401
<i>Collections</i>	403
<i>Replicated active objects are collections too</i>	407
10.7. NAMING CONVENTIONS	408
10.8. STATEMENTS	410
<i>Variable declaration</i>	411
<i>Function call</i>	412
<i>Assignment</i>	412
<i>If-then-else</i>	413
<i>Switch</i>	414
<i>For loop</i>	415
<i>While loop</i>	416
<i>Block (...) and indentation</i>	417
<i>Return statement</i>	418
<i>Comments</i>	419
10.9. WHERE AM I AND HOW DO I GET TO...?	421
10.10. VIEWING JAVA CODE GENERATED BY ANYLOGIC	423
10.11. CREATING YOUR JAVA CLASSES WITHIN ANYLOGIC MODEL	425
<i>Inner classes</i>	428
10.12. UNLINKING EXTERNAL JAVA MODULES (JAR FILES)	429
CHAPTER 11. EXCHANGING DATA WITH EXTERNAL WORLD	430
11.1. TEXT FILES	431
<i>Example 11.1: Using textfile as a log</i>	432
<i>Example 11.2: Reading tablefunction from a textfile</i>	433
<i>Example 11.3: Reading agent parameters from a CSVfile</i>	435
11.2. EXCEL SPREADSHEETS	437
<i>Example 11.4: Reading data of various types from fixed cells in Excel</i>	439
<i>Example 11.5: Reading model parameters from Excel using Java reflection</i>	441
<i>Example 11.6: Displaying the model output as a chart in Excel</i>	443
11.3. DATABASES	446
<i>SQL queries</i>	448
<i>AnyLogic database connectivity objects</i>	451
<i>Example 11.7: Loading data from a database and using ResultSet</i>	453
<i>Example 11.8: Creating agent populations parameterized from a database</i>	455
<i>Example 11.9: Dumping simulation output into a database table</i>	459
<i>Example 11.10: Using prepared statement when writing to databases</i>	462

The Big Book of Simulation Modeling • Multimethod Modeling with AnyLogic 6

11.4.	WORKING WITH THE CUPBOARD	464
	<i>Example 11.11: Working with clipboard</i>	464
11.5.	STANDARD OUTPUT, THE MODEL LOG, AND COMMAND LINE ARGUMENTS	466

CHAPTER 12. PRESENTATION AND ANIMATION: WORKING WITH SHAPES, GROUPS, COLORS

468

12.1.	DRAWING AND EDITING SHAPES	469
	<i>Polylines and curves</i>	470
	<i>Arcs</i>	471
	<i>Text</i>	471
	<i>Images</i>	472
	<i>Z-Order</i>	474
	<i>Selecting hidden shapes</i>	475
	<i>Coordinates and the grid</i>	476
	<i>Copying shapes</i>	477
	<i>Locking shapes – preventing selection by mouse</i>	477
	<i>General properties of graphical shapes</i>	478
	<i>Advanced properties of graphical shapes</i>	479
12.2.	GROUPING SHAPES	480
	<i>3D Groups</i>	485
	<i>Working with the group contents dynamically using API</i>	485
	<i>On draw extension point – execute custom code on each frame</i>	486
	<i>Groups in the project tree</i>	486
	<i>Top level groups for active object presentation and icon</i>	487
12.3.	ANIMATION PRINCIPLES. DYNAMIC PROPERTIES OF SHAPES	487
	<i>Dynamic properties of shapes</i>	488
	<i>Example 12.1: Commodity price change animation</i>	489
	<i>Example 12.2: Elevator doors animation</i>	491
	<i>Example 12.3: Stock of money animation</i>	493
	<i>Example 12.4: Missile attack animation</i>	495
	<i>Animation frames</i>	497
12.4.	REPLICATED SHAPES	498
	<i>Example 12.5: Drawing seats in a movie theater</i>	499
	<i>Example 12.6: Selling seats in the movie theater.</i>	501
	<i>Example 12.7: Drawing a flower</i>	502
	<i>Example 12.8: Product portfolio bubble chart (BCG chart)</i>	504
12.5.	SHAPES' API	507
	<i>Example 12.9: Using color to show the current state of a statechart</i>	507
	<i>Example 12.10: Show/hide a callout</i>	508

<i>Example 12.11: Read graphics from a textfile</i>	510
<i>Example 12.12: Find all red circles</i>	512
<i>Example 12.13: Resize the red circles</i>	513
<i>API of non-persistent shapes</i>	514
<i>AnyLogic Java class hierarchy for shapes</i>	514
12.6. COLORS AND TEXTURES	514
<i>Example 12.14: Choosing appropriate colors for an arbitrary number of objects.</i>	516
<i>Transparency</i>	517
<i>Example 12.15: Using transparency to show coverage zone</i>	517
<i>Example 12.16: Show population density using color interpolation</i>	518
CHAPTER 13. DESIGNING INTERACTIVE MODELS: USING CONTROLS	521
<i>Example 13.1: Slider linked to a model parameter</i>	522
<i>Example 13.2: Buttons changing the parameter value</i>	523
<i>Example 13.3: Edit box linked to a parameter of embedded object</i>	525
<i>Example 13.4: Radio buttons changing the view mode</i>	526
<i>Example 13.5: Combo box controlling the simulation speed</i>	527
<i>Example 13.6: File chooser for text files</i>	529
<i>Indivisibility of control actions and model events</i>	530
13.1. DYNAMIC PROPERTIES OF CONTROLS	530
<i>Example 13.7: Radio buttons enabling/disabling other controls</i>	530
<i>Example 13.8: Keeping controls in the top left corner of the window</i>	531
<i>Example 13.9: Replicated button</i>	532
13.2. CONTROLS'API	533
13.3. HANDLING MOUSE CLICKS	534
<i>Example 13.10: Hyper link menu to navigate between view areas</i>	535
<i>Example 13.11: Creating dots at the click coordinates</i>	536
<i>Example 13.12: Catching mouse clicks anywhere on the canvas</i>	537
CHAPTER 14. 3D ANIMATION	539
<i>Example 14.1: A very simple model with 3D animation</i>	540
14.1. PRIMITIVE 3D SHAPES	542
14.2. 3D GROUPS AND ROTATION	545
<i>Example 14.2: Rotation in 3D—a sign on two posts</i>	545
<i>Example 14.3: Bridge crane 3D</i>	546
14.3. STANDARD AND IMPORTED 3D GRAPHICS	549
<i>Using standard 3D graphics</i>	549
<i>Using external 3D graphics</i>	550
14.4. HIERARCHICAL 3D ANIMATIONS. EMBEDDED 3D PRESENTATION	551

14.5.	3D WINDOWS	552
	<i>Navigation in the 3D scene at runtime</i>	552
	<i>Multiple 3D views</i>	553
14.6.	CAMERAS	554
	<i>Example 14.4: A very simple model with multiple 3D windows and cameras</i>	554
	<i>Example 14.5: Camera on a moving object</i>	556
14.7.	LIGHTS	559
	<i>Example 14.6: Examples of Lights in 3D Scene</i>	561
CHAPTER 15. RANDOMNESS IN ANYLOGIC MODELS		565
15.1.	PROBABILITY DISTRIBUTIONS	565
	<i>Probability distribution functions</i>	567
	<i>Distribution fitting</i>	570
	<i>Custom (empirical) distributions</i>	570
15.2.	SOURCES OF RANDOMNESS IN THE MODEL	573
	<i>Randomness in process models</i>	574
	<i>Randomness in agent based models</i>	576
	<i>Example 15.1: Agents randomly distributed within a freeform area</i>	578
	<i>Example 15.2: Agents randomly distributed over a finite set of locations</i>	579
	<i>Randomness in system dynamics models</i>	581
	<i>Example 15.3: Stock price fluctuations in a system dynamics model.</i>	581
	<i>Randomness in AnyLogic simulation engine</i>	582
15.3.	RANDOM NUMBER GENERATORS. REPRODUCIBLE AND UNIQUE EXPERIMENTS	583
	<i>Random number generators</i>	583
	<i>The seed. Reproducible and unique experiments</i>	585
	<i>Example 15.4: Reproducible experiment with a stochastic process model</i>	585
CHAPTER 16. MODEL TIME, DATE AND CALENDAR. VIRTUAL AND REALTIME		589
16.1.	THE MODEL TIME	589
	<i>Time units</i>	590
	<i>Developing models independent of time unit settings</i>	591
16.2.	DATE AND CALENDAR	592
	<i>Finding out the current date, day of week, hour of day, etc</i>	594
	<i>Constructing dates. Converting the model date to the model time and vice versa</i>	595
	<i>Specifying timeouts and delays in days, months, years</i>	596
16.3.	VIRTUAL AND REAL-TIME EXECUTION MODES	597
	<i>Execution mode API</i>	599
REFERENCES		601

