



Viskit: Visual Simkit Editor

Don Brutzman and Terry Norbraten

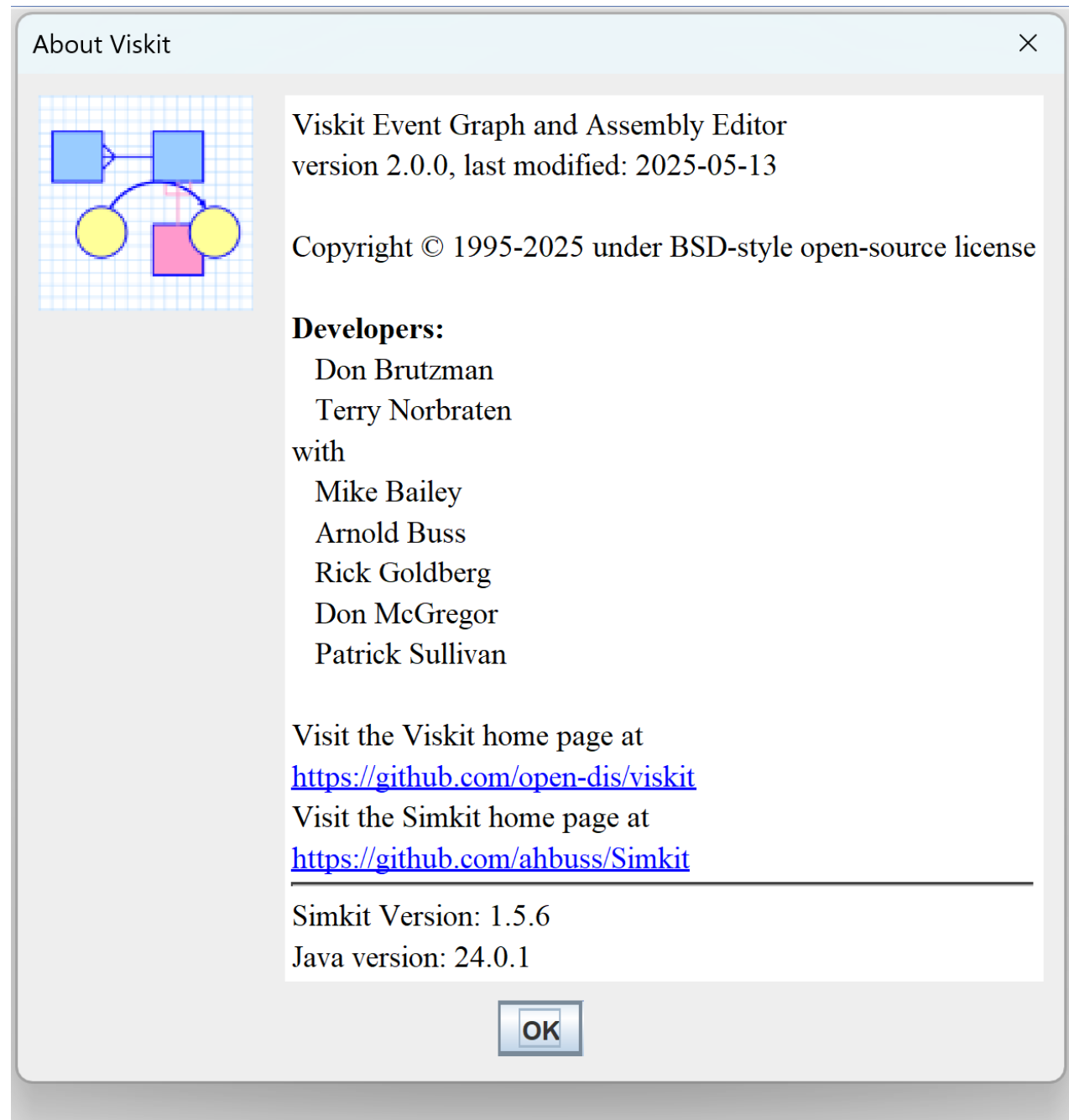
Modeling Virtual Environments Simulation (MOVES) Institute
Naval Postgraduate School (NPS)

13 May 2025

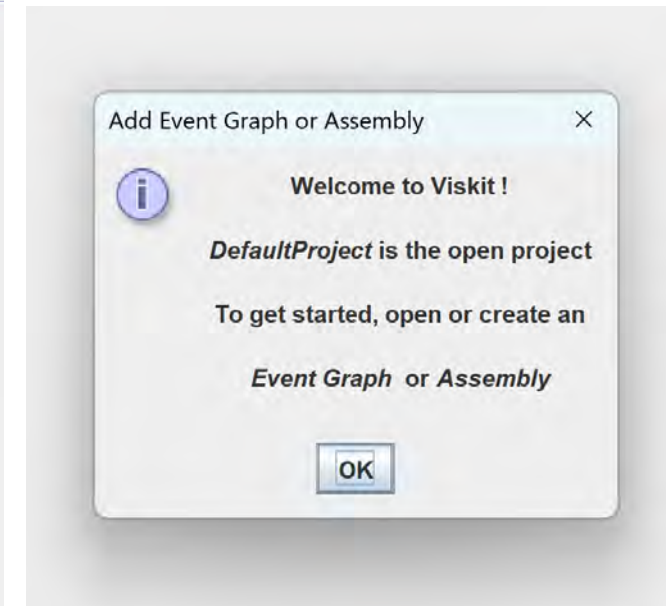
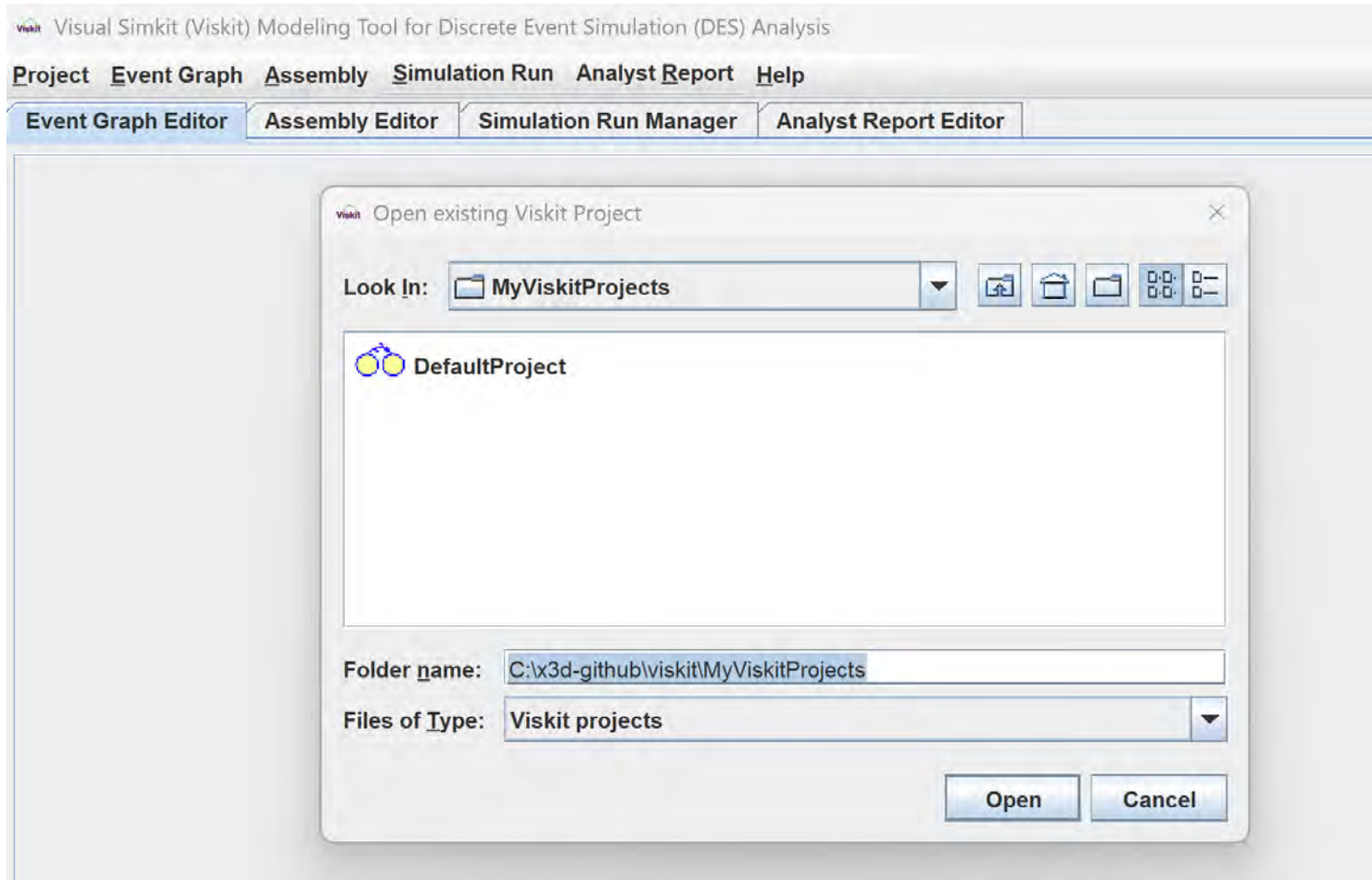
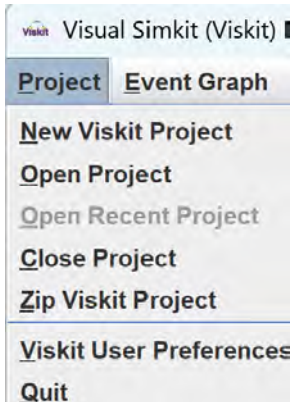
Overview

- Viskit is an open-source application for defining Discrete Event Simulation (DES) models that autogenerate Java source using the Simkit library.
- The Viskit application allows you to build a collection of event graphs that describe the behavior of various entities in a simulation environment.
- Capabilities include Simkit Event Graphs, Assemblies, simulation tools, and an Analyst Report generator suitable for thesis and research efforts.
- The current release includes a thorough refactoring of the underlying Viskit source code, facilitating ongoing maintenance and development.
- Efforts are ongoing.

About

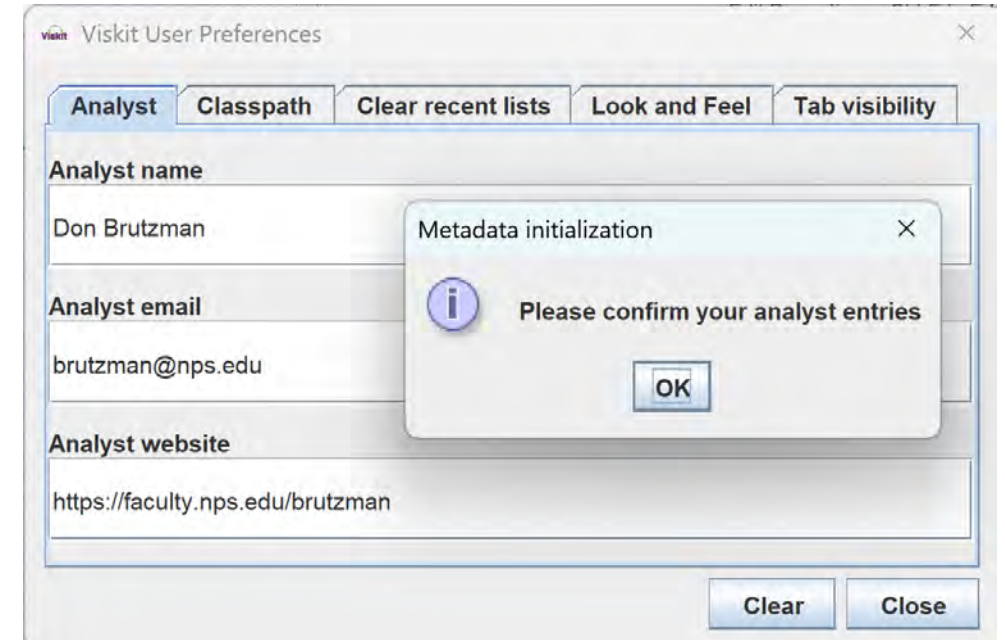


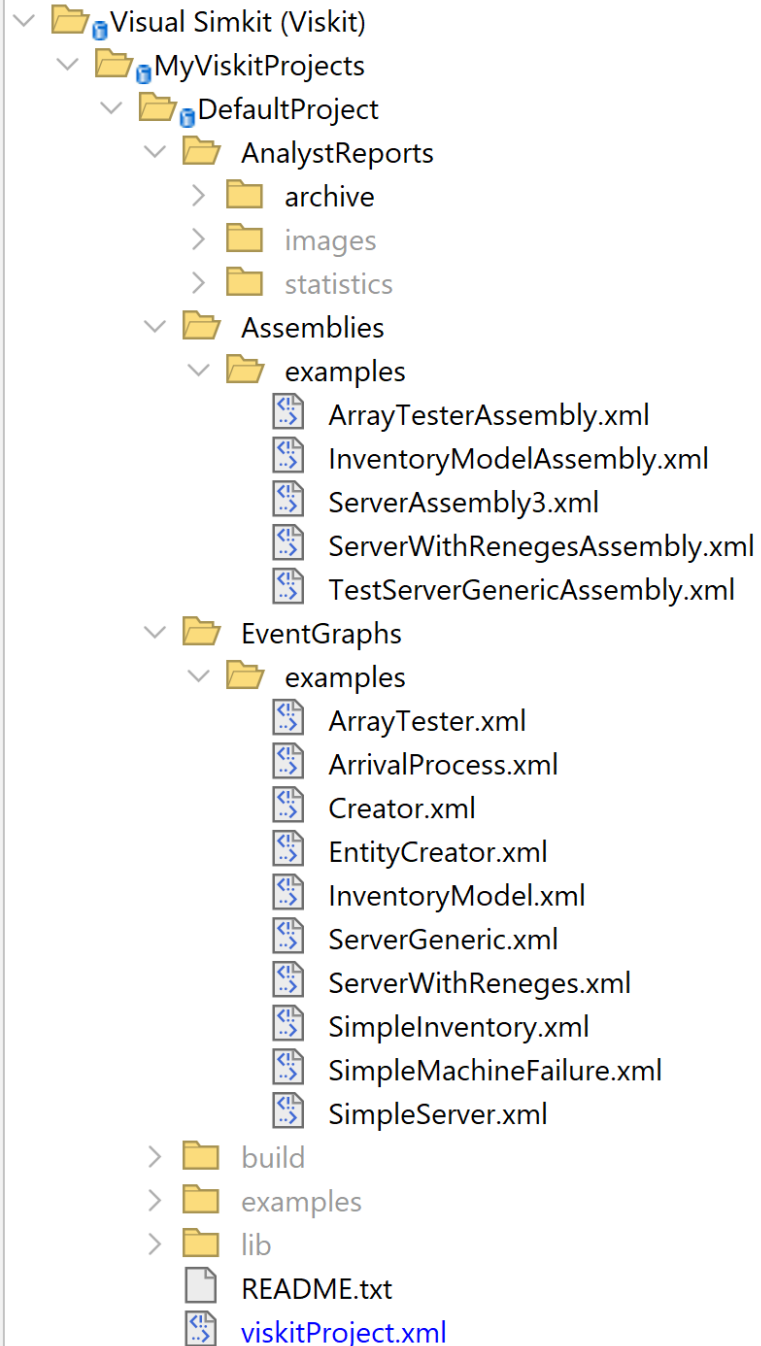
Viskit is organized according to Viskit Projects



Viskit User Preferences

User information is
applied when creating
an Analyst Report



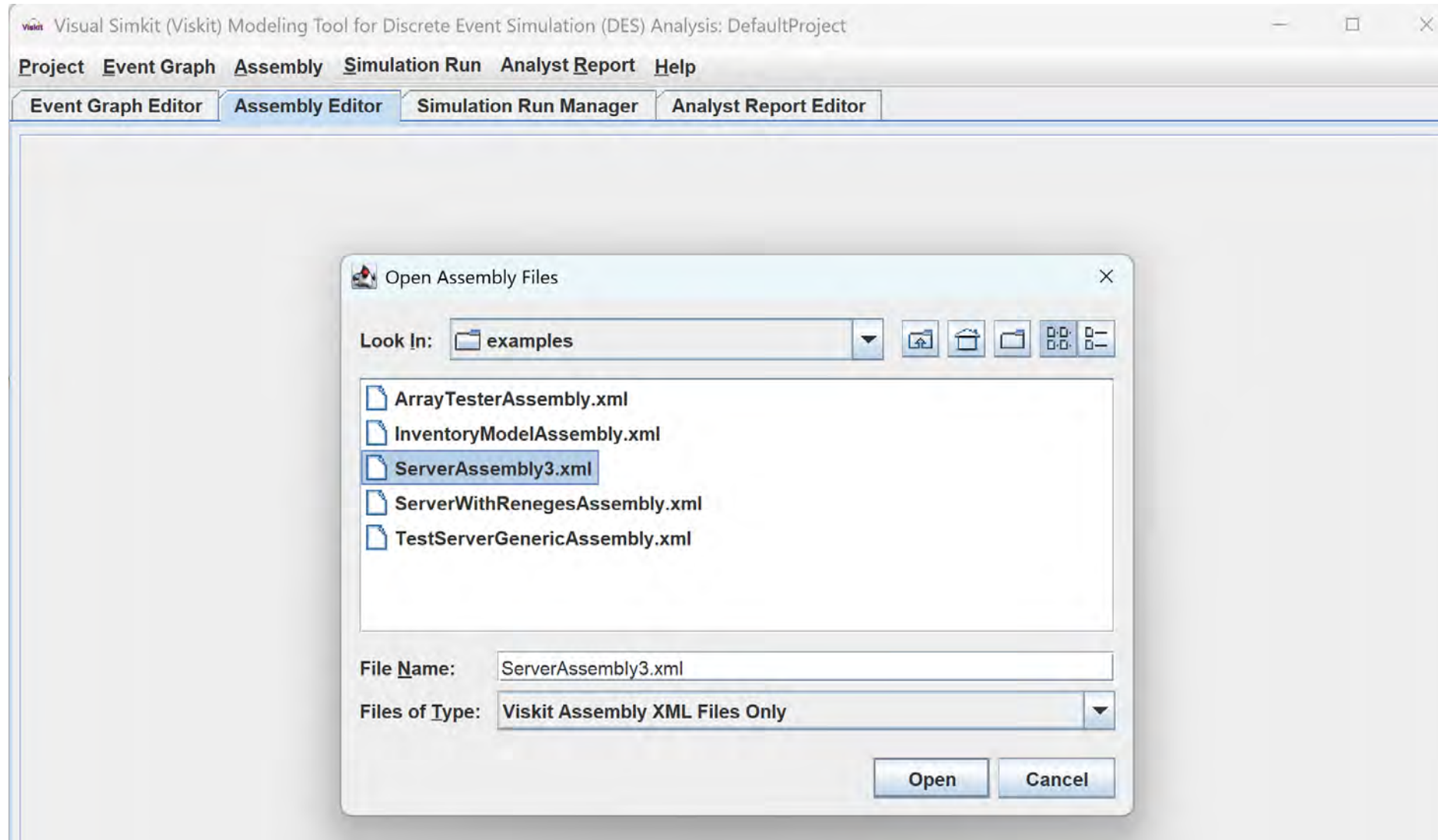


Viskit project files contain multiple assets

The screenshot shows the `viskitProject.xml` file open in a code editor. The XML content is as follows:

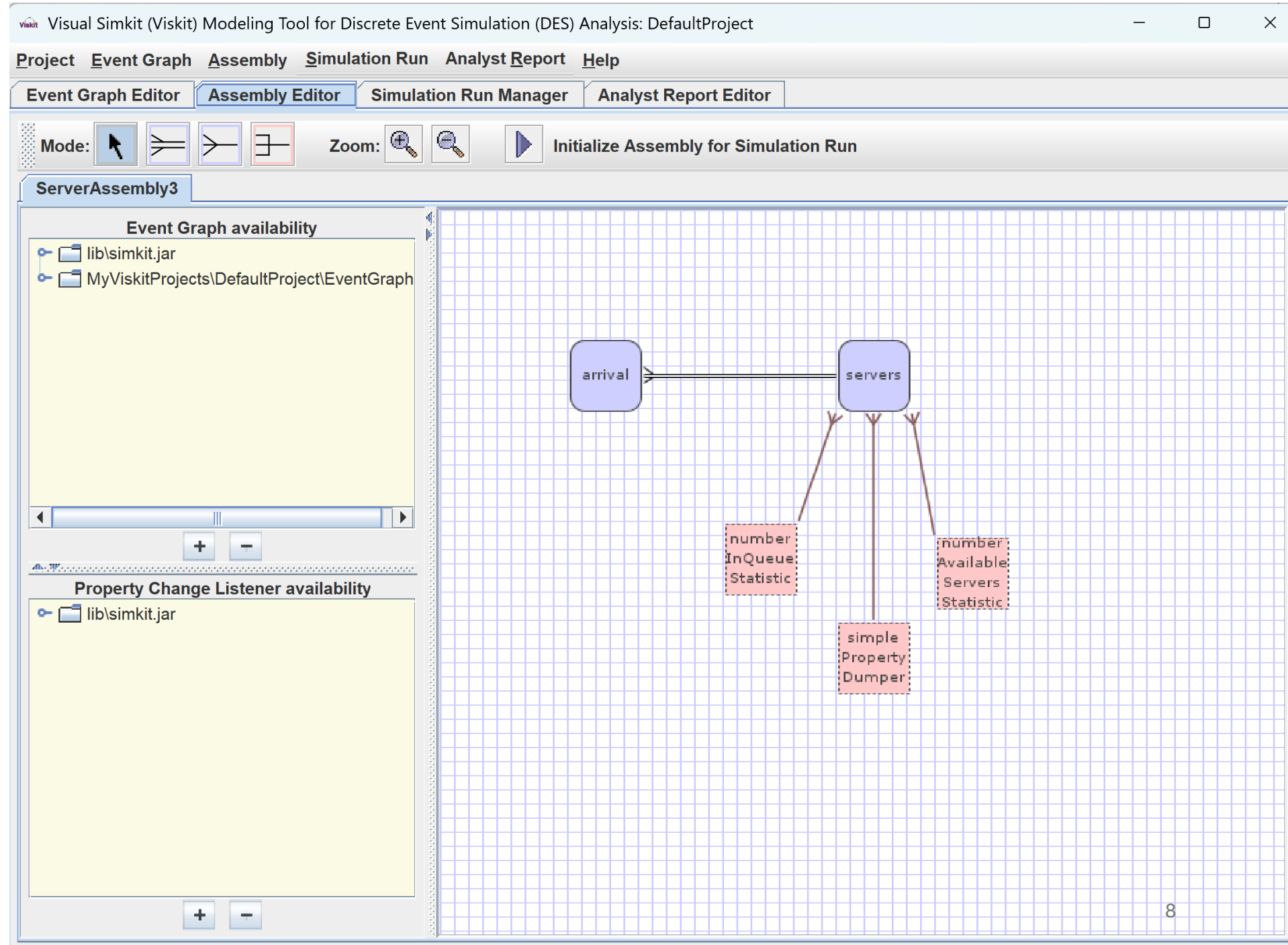
```
1 <?xml version="1.0" encoding="UTF-8"?><ViskitProject>
2   <!-- viskitProject.xml -->
3   <AnalystReports name="AnalystReports"/>
4   <AssembliesDirectory name="Assemblies"/>
5   <EventGraphsDirectory name="EventGraphs"/>
6   <BuildDirectory name="build">
7     <ClassesDirectory name="classes"/>
8     <SourceDirectory name="src"/>
9   </BuildDirectory>
10  <DistDirectory name="dist"/>
11  <LibDirectory name="lib"/>
12  <Project name="DefaultProject"/>
13  <Cached>
25 </ViskitProject>
```

Next step: open an assembly



For example

Server
Assembly3
.xml











Editing

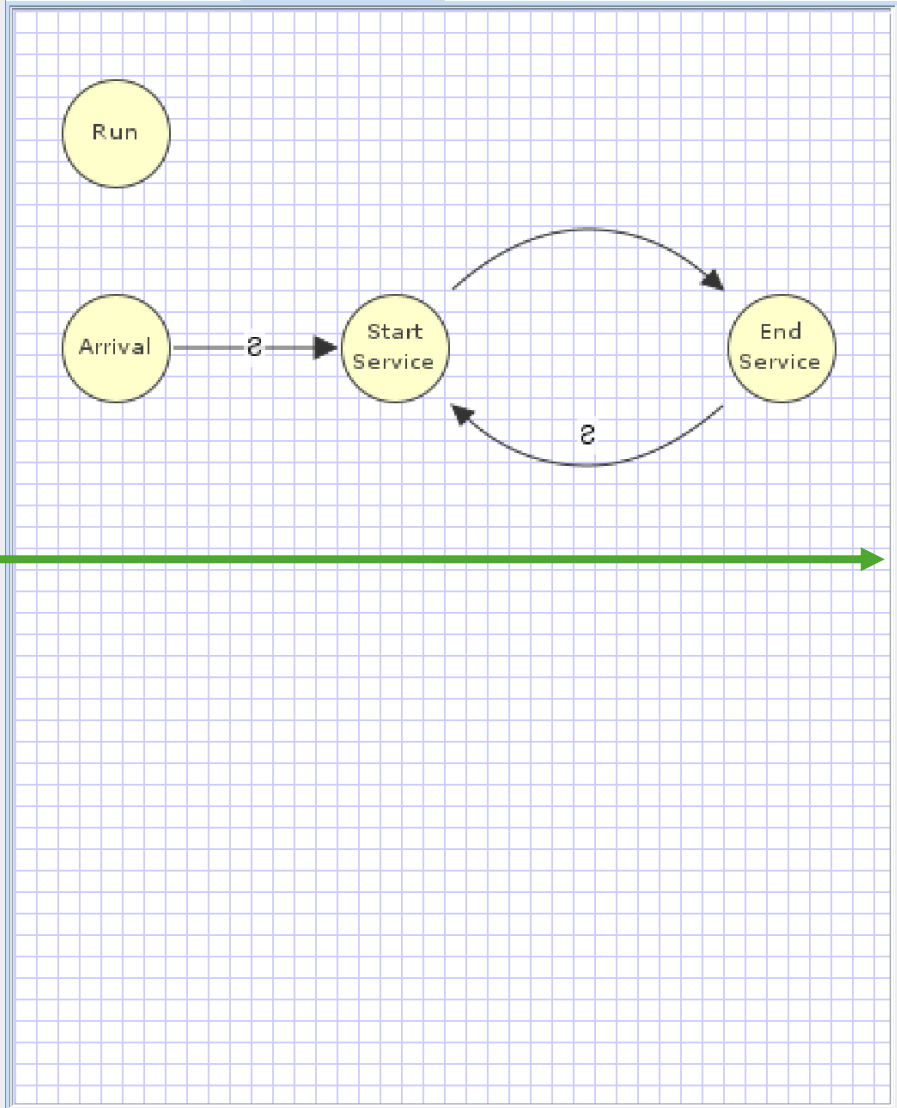
Visual Simkit (Viskit) Modeling Tool for Discrete Event Simulation (DES) Analysis: DefaultProject

Project Event Graph Assembly Simulation Run Analyst Report Help

Event Graph Editor Assembly Editor Simulation Run Manager Analyst Report Editor

Add:    Mode:    Zoom:  

ArrivalProcess SimpleServer



```
graph LR; Run((Run)) --> Arrival((Arrival)); Arrival -- "2" --> StartService((Start Service)); StartService --> EndService((End Service)); EndService -- "2" --> StartService;
```

Event Graph Description
Edit Properties or Ctrl-E to Edit

One of the most fundamental examples, a simple server that handles client customers

Simulation Parameters
Double click a row to edit.

name	type	description
serviceTime	simkit.random.RandomVariate	time duration to perform service
totalNumberServers	int	how many servers are provided

+ -

State Variables
Double click a row to edit.

name	type	description
numberInQueue	int	number of customers in queue
numberAvailableServers	int	number of available servers at a given time

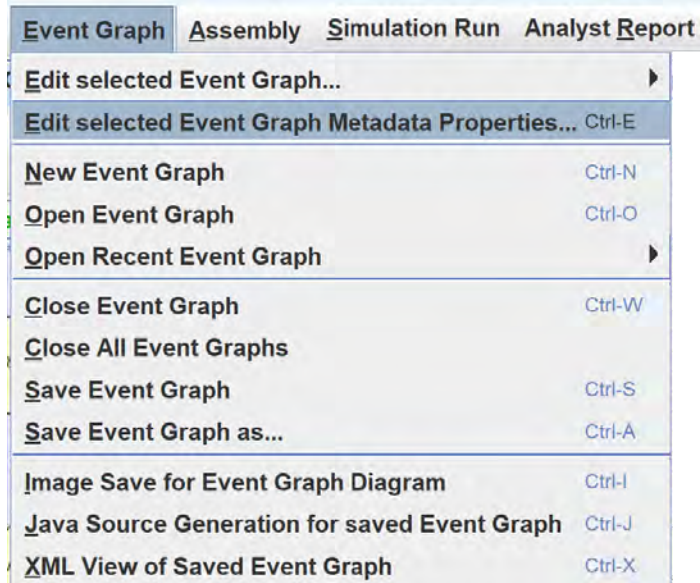
+ -

Global Code Block

10

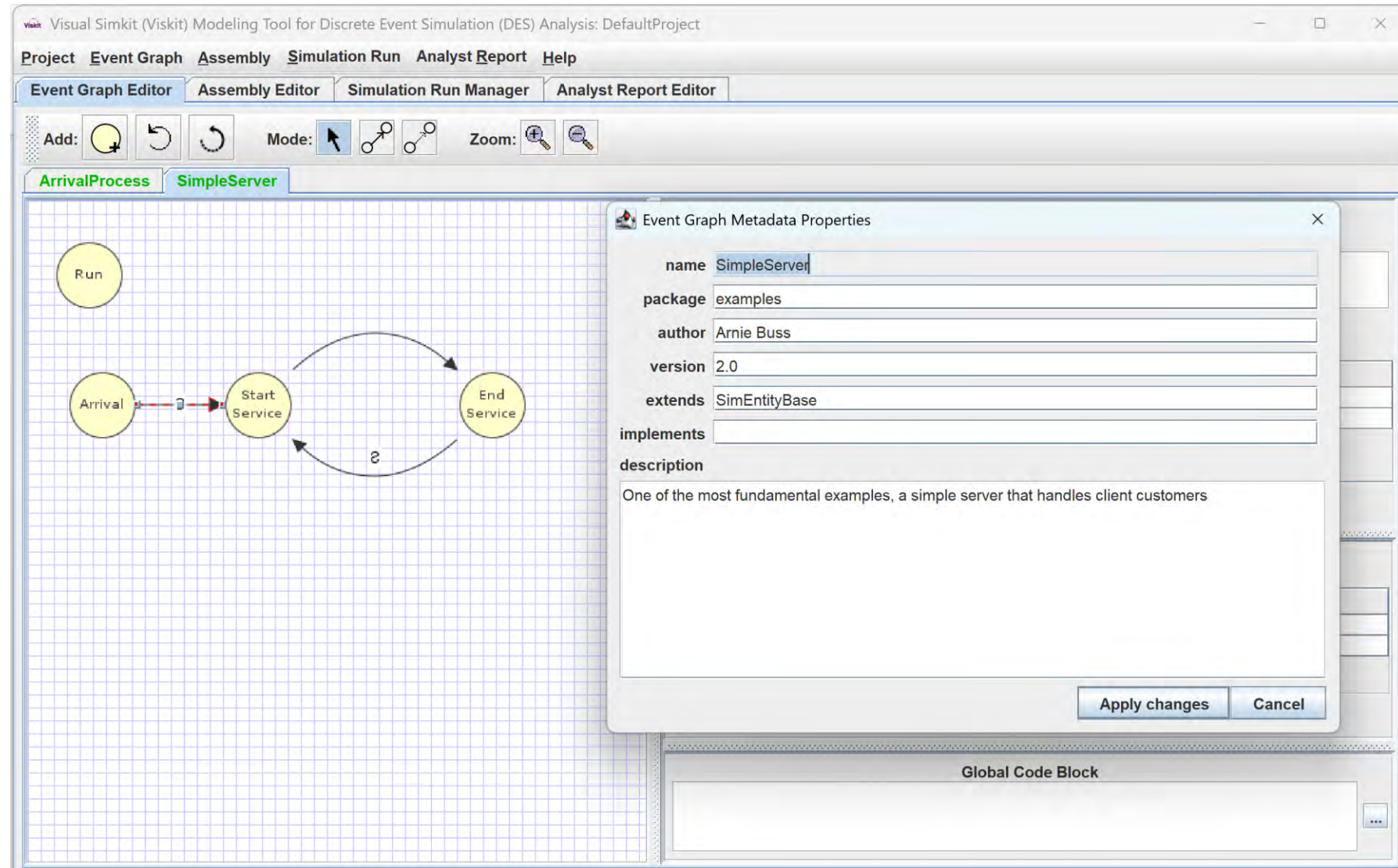
Check that
corresponding
Event Graphs
were opened

Check out
Event Graphs
Simulation
Parameters and
State Variables



Editing an Event Graph

Inspect Metadata



Editing an Event Graph

Inspect State Nodes

Visual Simkit (Viskit) Modeling Tool for Discrete Event Simulation (DES) Analysis: DefaultProject

Project Event Graph Assembly Simulation Run Analyst Report Help

Event Graph Editor Assembly Editor Simulation Run Manager Analyst Report Editor

Add: [Add] [Undo] [Redo] Mode: [Select] [Link] [Unlink] Zoom: [Zoom In] [Zoom Out]

ArrivalProcess SimpleServer

Run

Arrival

Start Service

End Service

Event Graph Description

Edit Properties or Ctrl-E to Edit

One of the most fundamental examples, a simple server that handles client customers

Simulation Parameters

Double click a row to edit.

name	type	description
serviceTime	simkit.random.RandomVariate	time duration to perform service
totalNumberServers	int	how many servers are provided

State Variables

Double click a row to edit.

name	type	description
numberInQueue	int	number of customers in queue
numberAvailableServers	int	number of available servers at a given time

Global Code Block

Event Node Inspector: Arrival

Event name

Arrival

description

Upon getting activated by a scheduling event, Arrival adds a customer to the queue

State transitions

Double click a row to edit.

numberInQueue=numberInQueue + 1

add

Event node arguments Local variables Local code block

Apply changes Cancel

Editing an Event Graph

Inspect Scheduling (Cancelling) Edge Transitions

Visual Simkit (Viskit) Modeling Tool for Discrete Event Simulation (DES) Analysis: DefaultProject

Project Event Graph Assembly Simulation Run Analyst Report Help

Event Graph Editor Assembly Editor Simulation Run Manager Analyst Report Editor

Add: Mode: Zoom:

ArrivalProcess SimpleServer

The diagram shows an event graph with four nodes: 'Run', 'Arrival', 'Start Service', and 'End Service'. 'Arrival' is connected to 'Start Service' by a red edge with a small blue square in the middle. 'Start Service' and 'End Service' are connected by two curved edges forming a loop. The top edge of the loop is labeled with a double-headed arrow, and the bottom edge is labeled with a double-headed arrow and the Greek letter epsilon (ϵ).

Edge Inspector

Type: Scheduling

Source event: Arrival

Target event: StartService

description

Conditional Expression

if (
numberAvailableServers > 0
) then schedule target event

Priority

DEFAULT

Time Delay

0.0

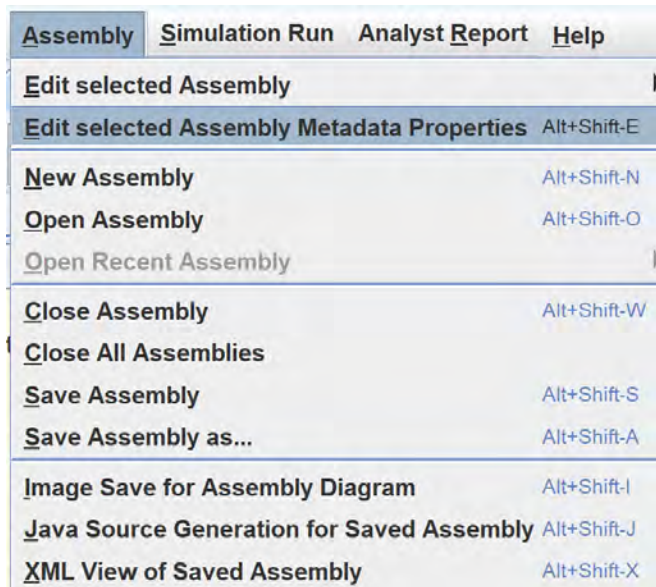
Edge Parameters passed to StartService

Double click a row to edit.

event argument	value
arg_0 (int)	numberAvailableServers

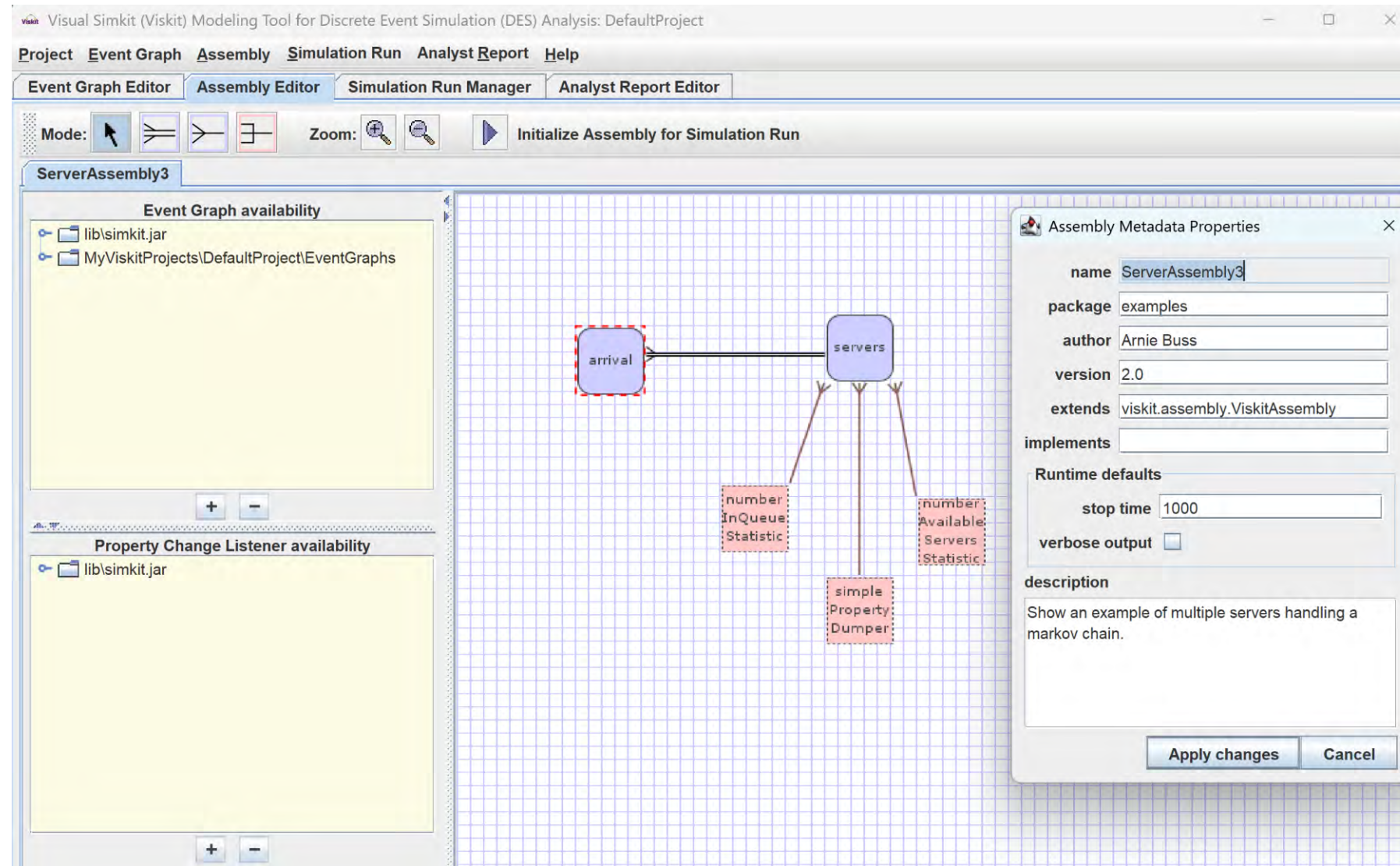
add

Apply changes Cancel



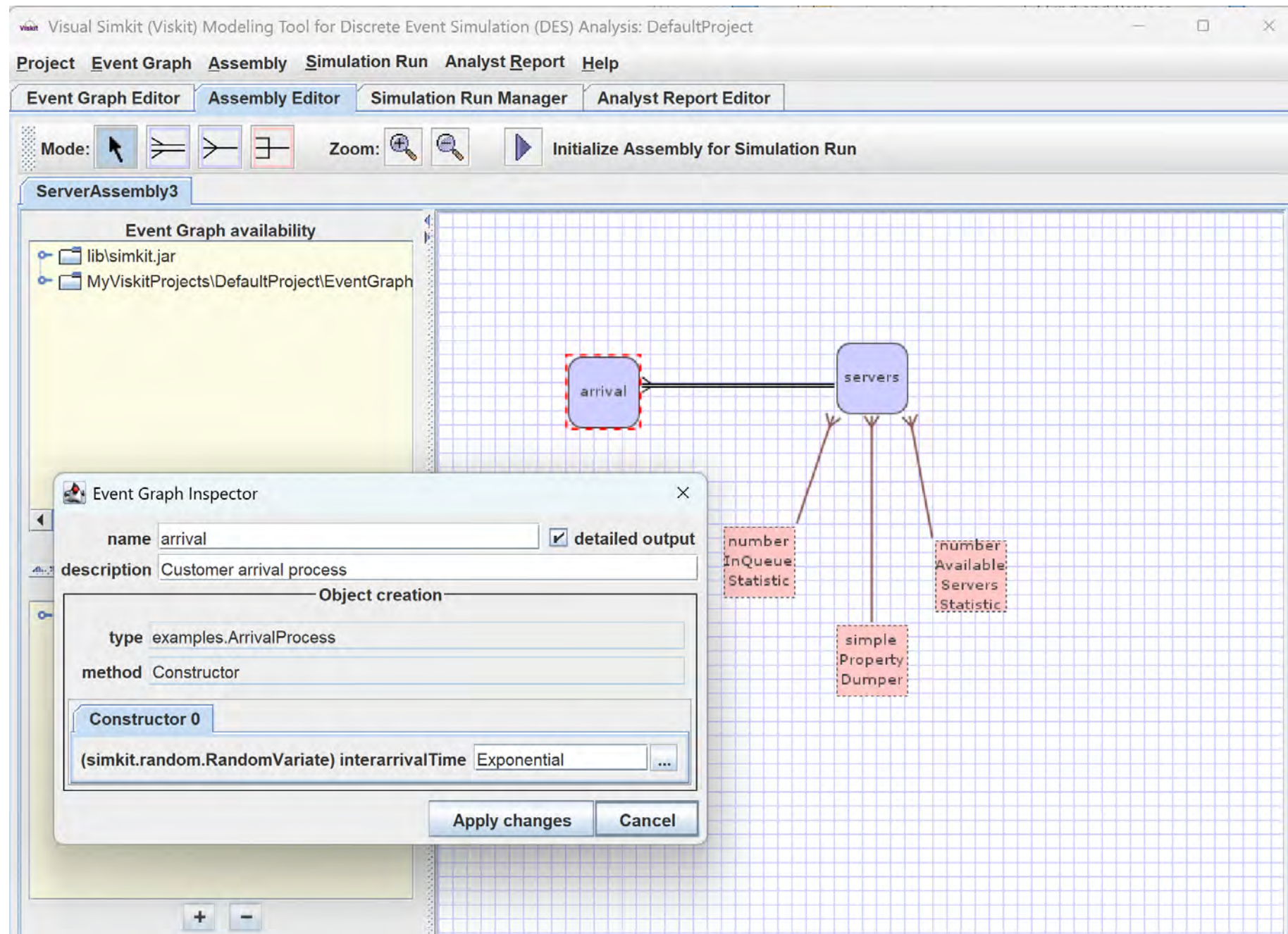
Editing an assembly

Inspect Metadata



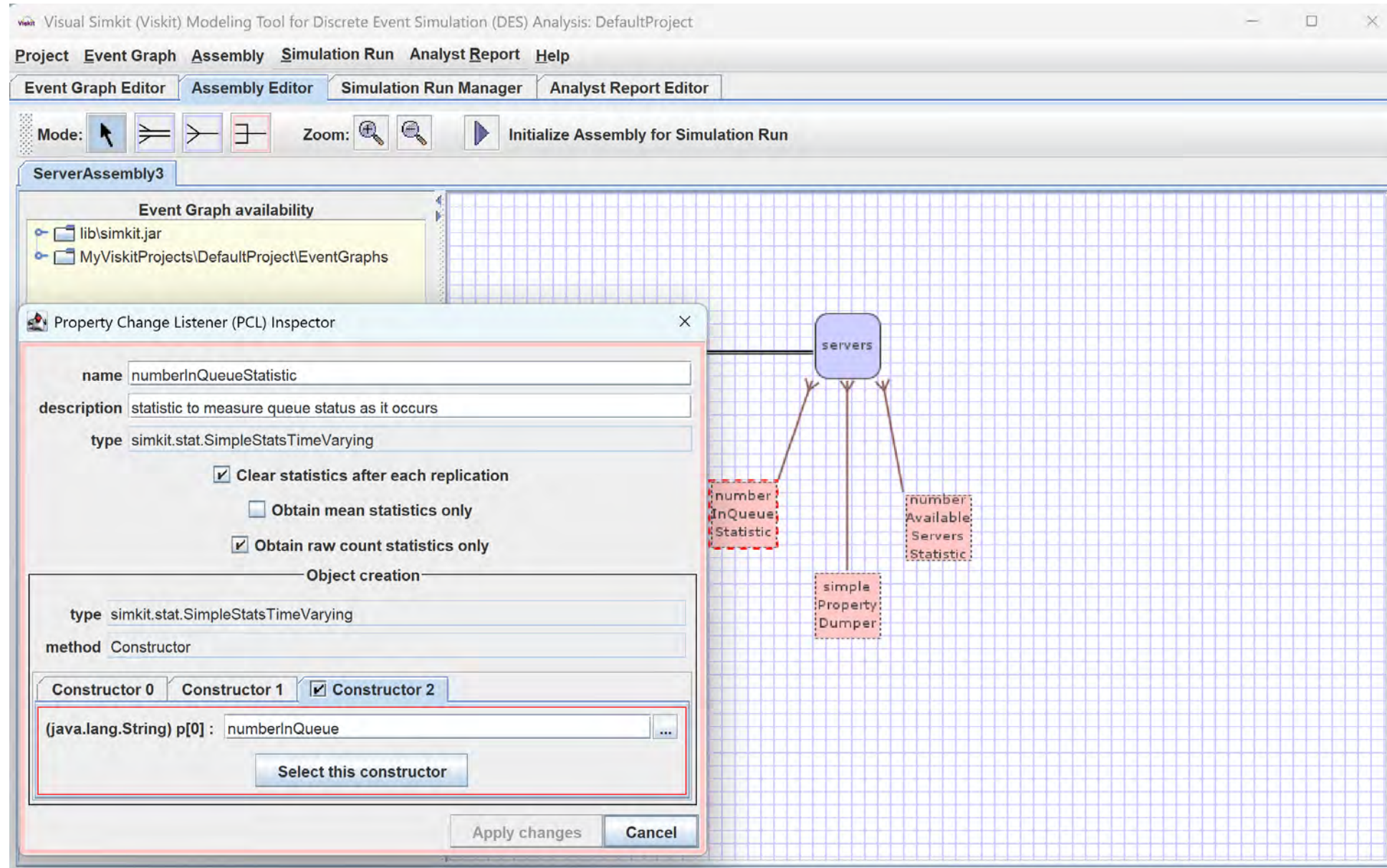
Editing an
assembly

Inspect
Event Graph
nodes



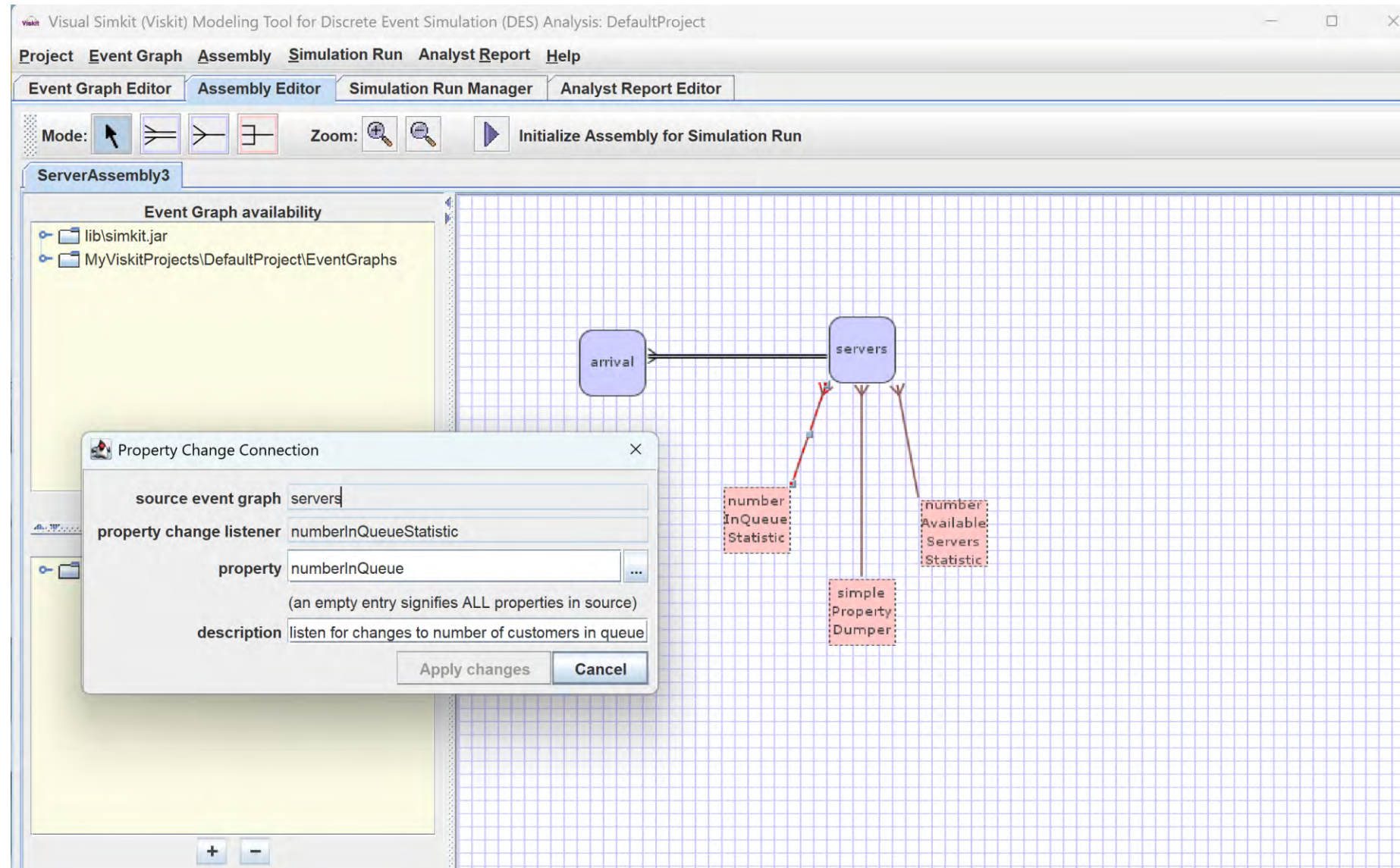
Editing an assembly

Inspect Property Change Listeners PCLs



Editing an assembly

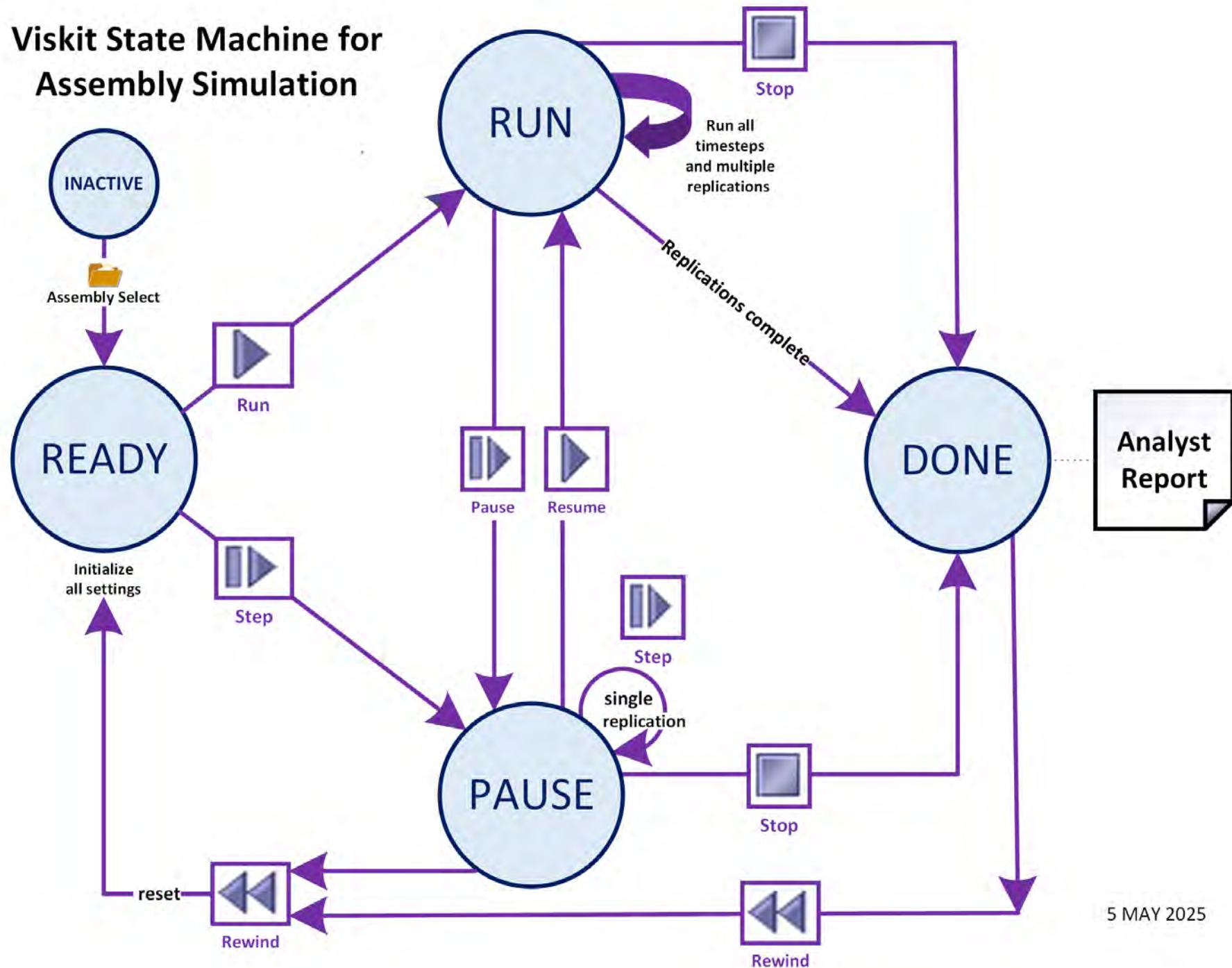
Inspect Property Change Connection



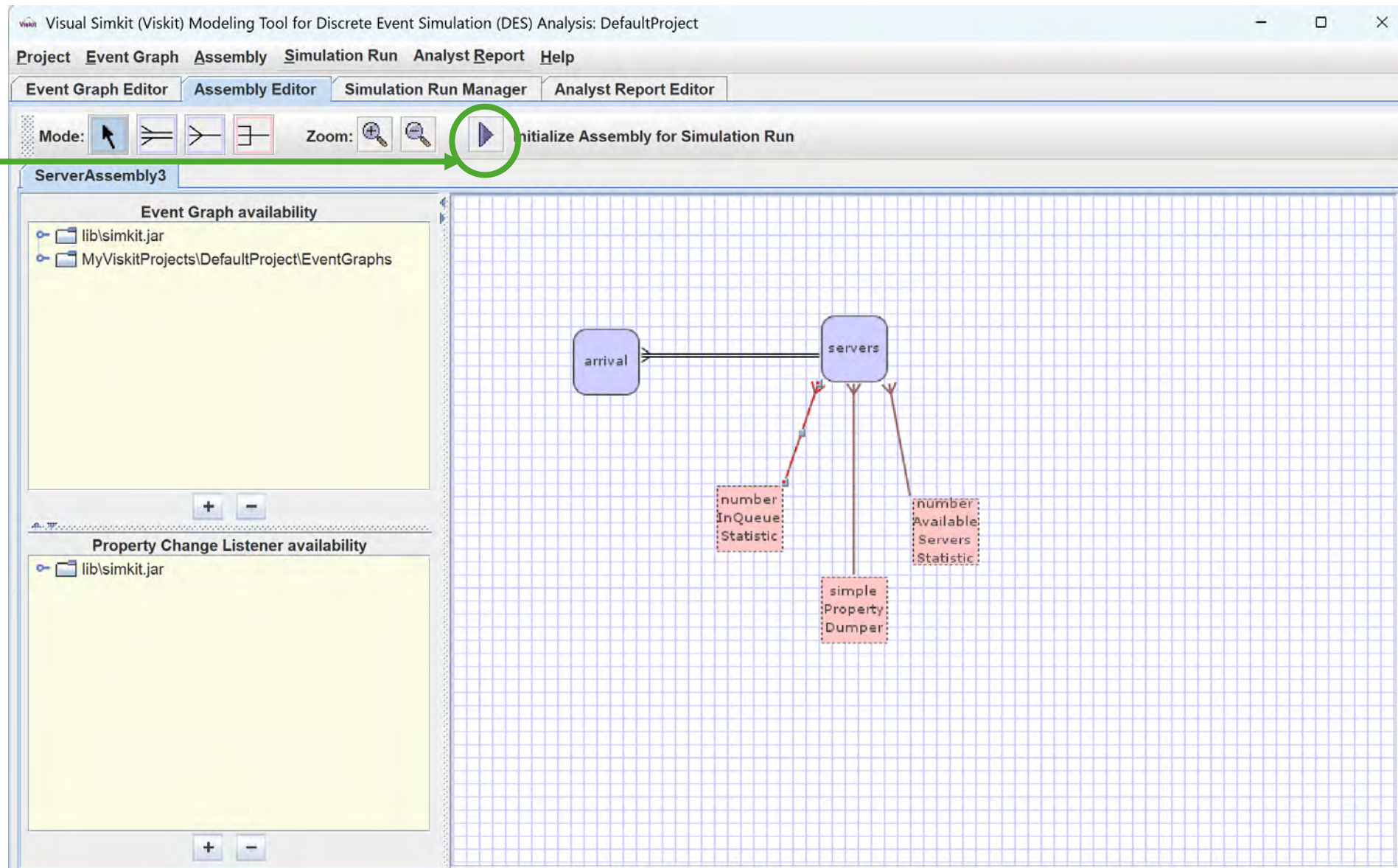
Run Simulation Replications

Rehearse and repeat for several or numerous repetitions

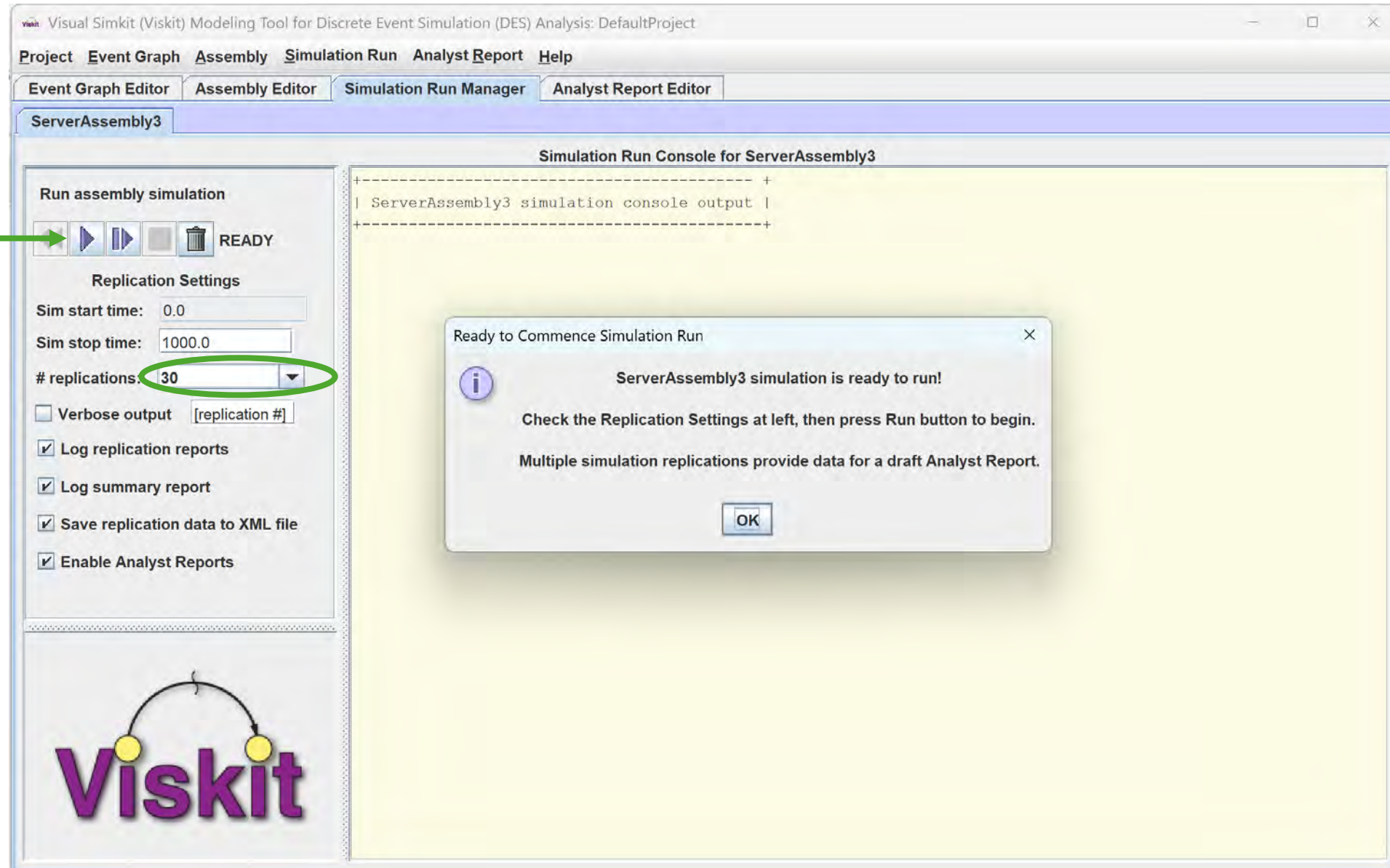
Viskit State Machine for Assembly Simulation



Ready
to run
simulations



Get set
to run
simulations



Go!
Run
simulations

Visual Simkit (Viskit) Modeling Tool for Discrete Event Simulation (DES) Analysis: DefaultProject

Project Event Graph Assembly Simulation Run Analyst Report Help

Event Graph Editor Assembly Editor Simulation Run Manager Analyst Report Editor

ServerAssembly3

Simulation Run Console for ServerAssembly3

Run assembly simulation

Now Running Replication 30 of 30...

Replication Settings

Sim start time: 0.0

Sim stop time: 1000.0

replications: 30

☐ Verbose output [replication #]

☒ Log replication reports

☒ Log summary report

☒ Save replication data to XML file

☒ Enable Analyst Reports

Simulation Run Data Collected, Analyst Report Ready

30 total replications performed, with data saved.

ServerAssembly3 Analyst Report

is now loaded and ready for further analysis.

Output Report following Replication #30

Name	Count	Minimum	Maximum	Mean	Standard Deviation	Variance
numberInQueue	1184	0.0	45.0	13.4026	13.7788	189.8545
numberAvailableServers	1148	0.0	2.0	0.1069	0.3861	0.1491

Summary Output Report: ServerAssembly3

numberInQueue.count (TALLY)

30 1,097.000 1,233.000 1,161.067 776.754 27.870

numberAvailableServers.count (TALLY)

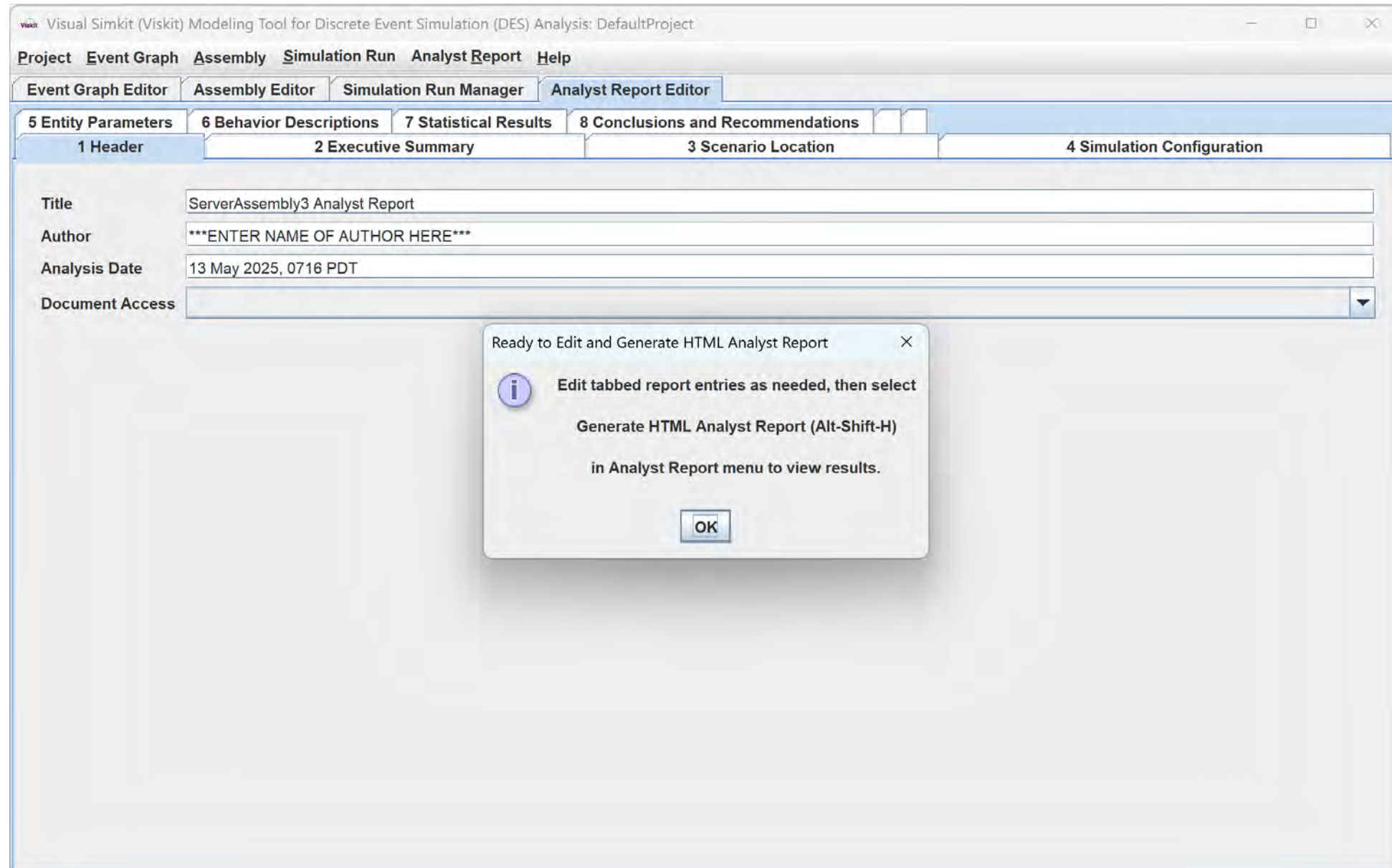
30 1,096.000 1,179.000 1,138.833 433.799 20.828

ServerAssembly3 simulation replications DONE

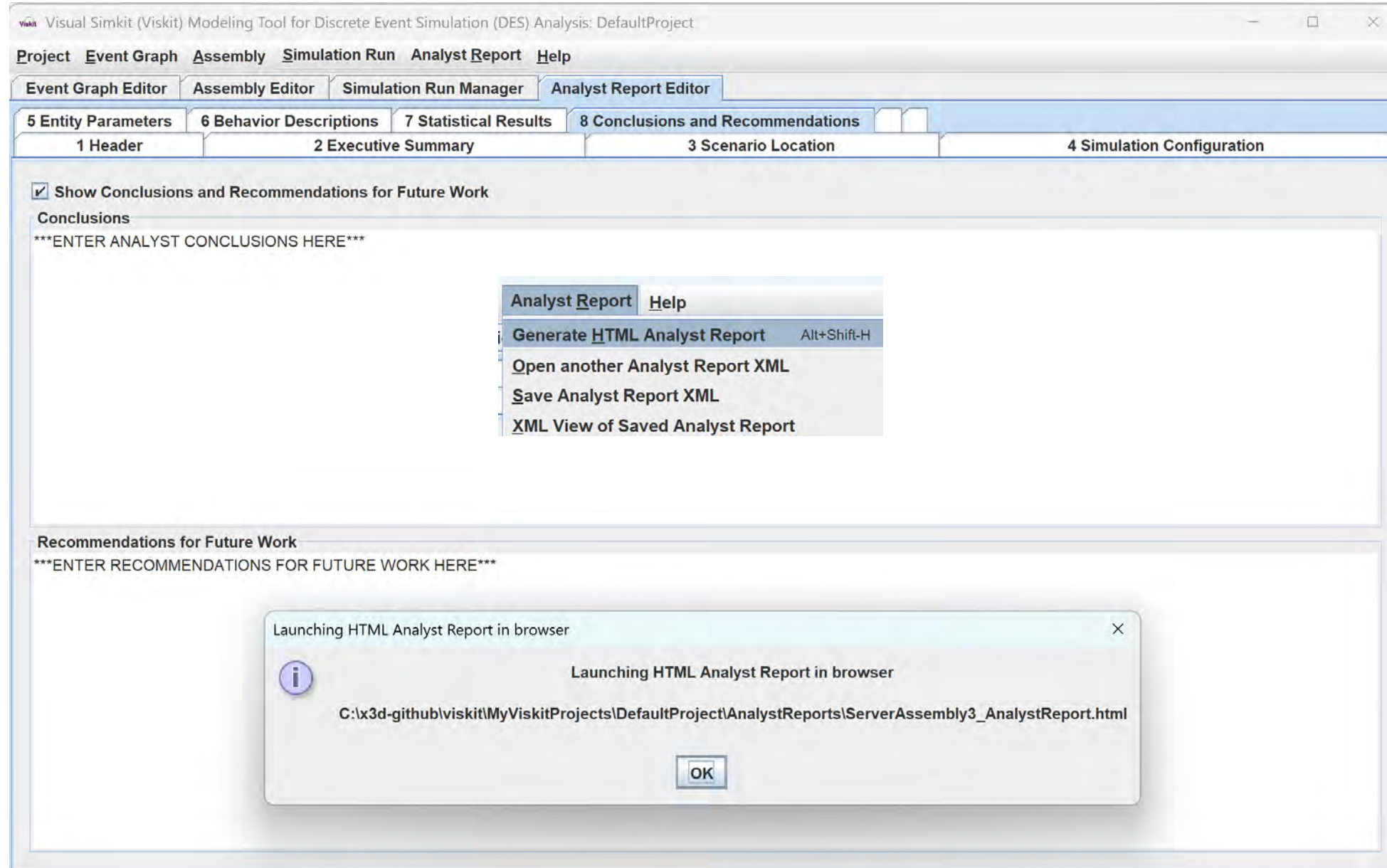
Viskit

Simulation
replications
complete

Edit
Analyst
Report data



When
ready,
launch
HTML
Analyst
Report
in browser





File

C:/x3d-github/viskit/MyViskitProjects/DefaultProject/AnalystReports/ServerAssembly3_AnalystReport.html



ServerAssembly3 Analyst Report

Analyst: ***ENTER NAME OF AUTHOR HERE***

Analysis date: 13 May 2025, 0716 PDT

[Executive Summary](#) | [Scenario Location](#) | [Simulation Configuration](#) | [Entity Parameters](#) | [Behavior Descriptions](#) | [Statistical Results](#) | [Conclusions and Recommendations](#)

Executive Summary

Assessment Overview

ENTER EXECUTIVE SUMMARY HERE

[Back to top](#)

Scenario Location for the Simulation

Description of Scenario Location Features

ENTER SCENARIO LOCATION DESCRIPTION HERE

Production Notes

ENTER SCENARIO LOCATION PRODUCTION NOTES HERE

All units are meters and degrees unless otherwise noted.

Post-Experiment Analysis of Significant Scenario Location Features

ENTER SCENARIO LOCATION CONCLUSIONS HERE

Simulation Configuration: Viskit Assembly Preparation for ServerAssembly3

Simulation configuration is defined by the Viskit Assembly which collects, lists, initializes, and connects all Event Graphs for participating entity models within a single scenario. The runnable assembly is then ready for repeated simulation replications, either for visual validation of behavior or statistical analysis of Measures of Effectiveness (MoEs).

Description: not provided

Entity Parameters for configuring Event Graphs

Entity parameters configure the behaviors of connected event graphs.

Entity Initialization Parameters for this Simulation Assembly

Initialization parameters are applied to individualize generic behavior models. These parameters customize the event-graph models.

Entity Parameters Conclusions: Post-Experiment Analysis of Entity Behaviors

ENTER ENTITY PARAMETERS CONCLUSIONS HERE

Behavior Descriptions

Description of Behavior Design

ENTER ENTITY BEHAVIOR DESCRIPTION HERE

Statistical Results for the Simulation

This simulation included 30 replication repetitions for the assembly of interest.

Statistical Results for the Simulation

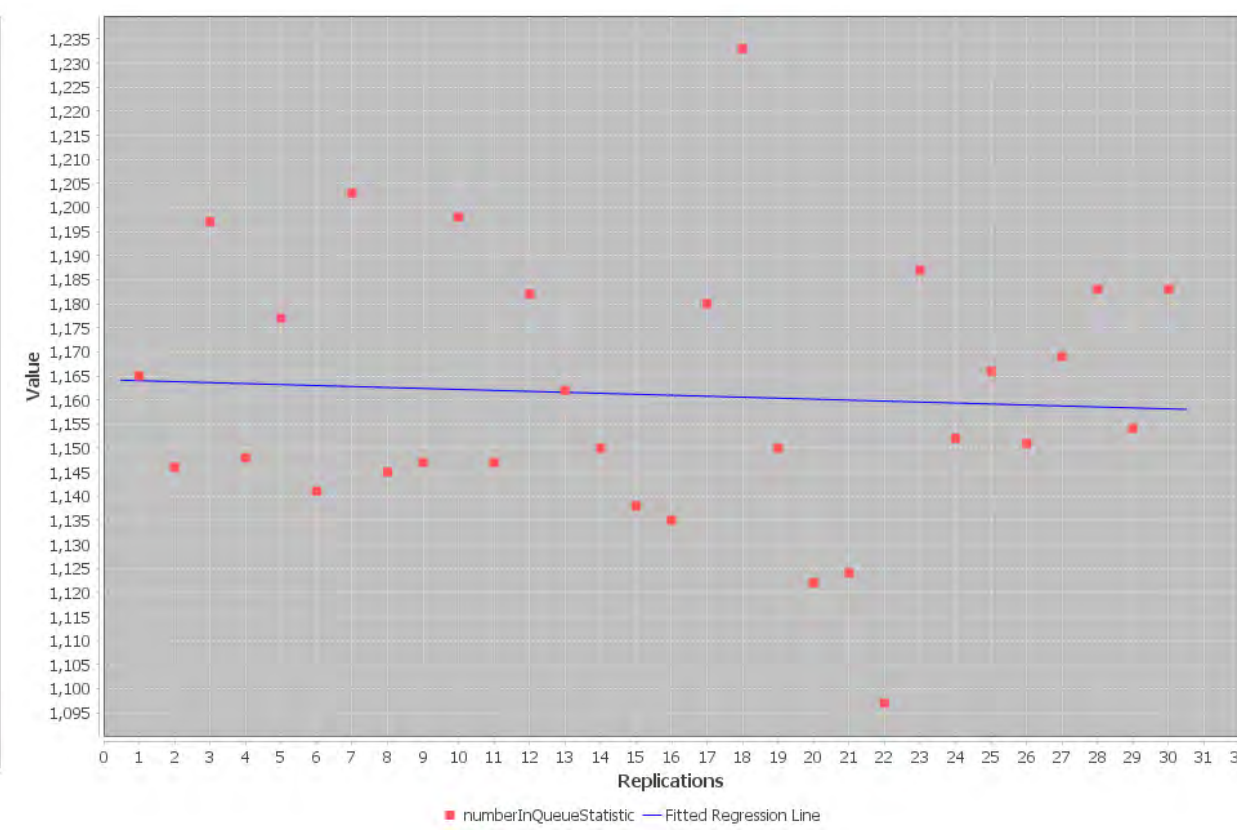
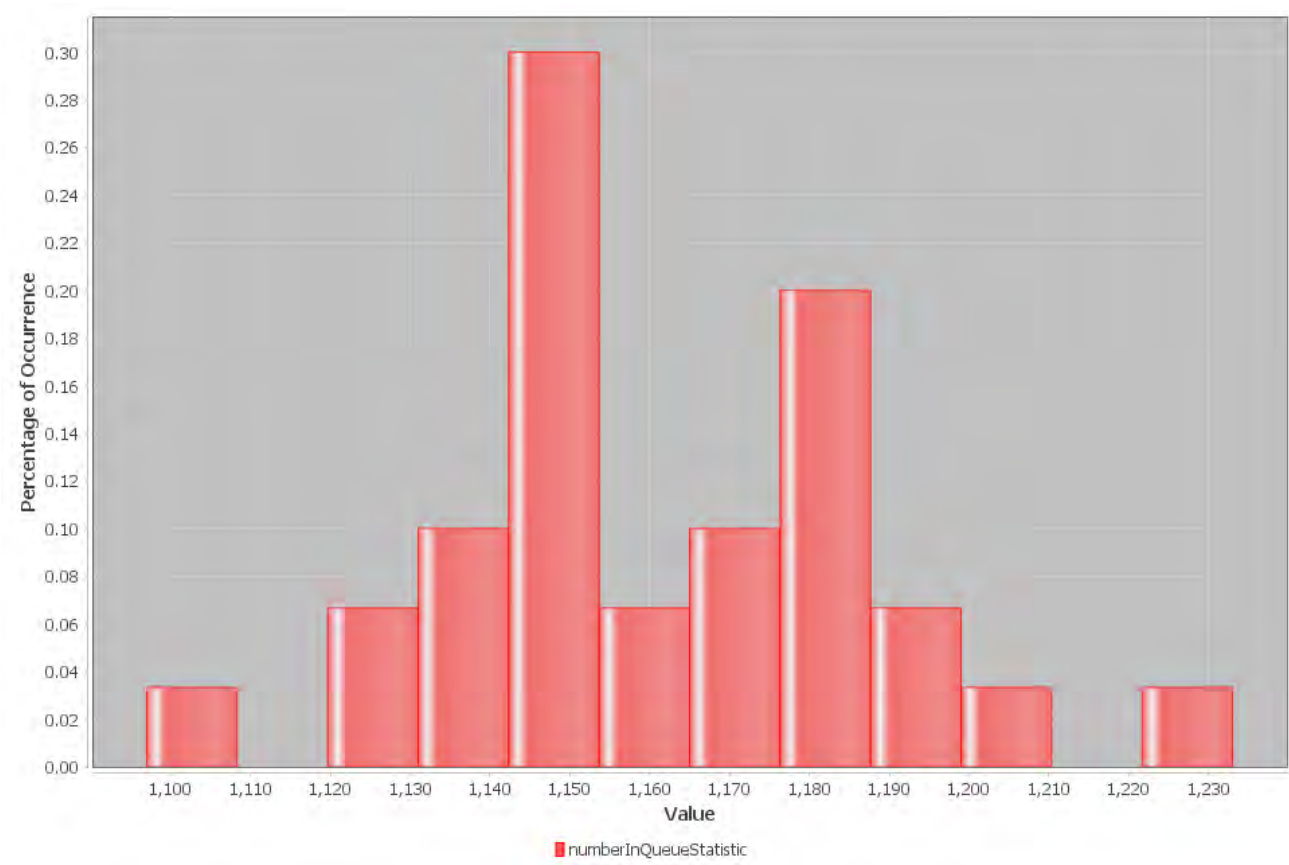
This simulation included 30 replication repetitions for the assembly of interest.

Statistical results are collected and produced by Property Change Listener (PCL) definitions in the Assembly model.

Assembly Simulation Replication Report

Measure of Effectiveness (MoE)

Property: numberInQueueStatistic



Replication #	numberAvailableServersStatistic	Min	Max	Mean	StdDev	Variance
1	1133	0	2	0.054	0.274	0.075
2	1143	0	2	0.100	0.350	0.123
3	1155	0	2	0.027	0.199	0.039
4	1131	0	2	0.028	0.189	0.036
5	1145	0	2	0.032	0.204	0.042
6	1131	0	2	0.175	0.465	0.216
7	1133	0	2	0.006	0.105	0.011
8	1143	0	2	0.187	0.508	0.258
9	1143	0	2	0.147	0.448	0.201
10	1161	0	2	0.056	0.258	0.067
11	1135	0	2	0.080	0.341	0.116
12	1179	0	2	0.071	0.336	0.113
13	1131	0	2	0.063	0.331	0.110
14	1145	0	2	0.113	0.393	0.155
15	1097	0	2	0.127	0.424	0.180
16	1115	0	2	0.160	0.474	0.225
17	1159	0	2	0.150	0.445	0.198
18	1173	0	2	0.019	0.164	0.027
19	1145	0	2	0.177	0.493	0.243
20	1113	0	2	0.093	0.356	0.127
21	1115	0	2	0.087	0.344	0.118
22	1096	0	2	0.233	0.552	0.305
23	1177	0	2	0.045	0.256	0.066
24	1137	0	2	0.030	0.210	0.044
25	1123	0	2	0.003	0.079	0.006
26	1115	0	2	0.117	0.401	0.161
27	1159	0	2	0.102	0.371	0.138
28	1147	0	2	0.056	0.293	0.086
29	1139	0	2	0.059	0.290	0.084
30	1147	0	2	0.107	0.386	0.149

Analyst Report Completion

Conclusions and Recommendations

Conclusions

ENTER ANALYST CONCLUSIONS HERE

Recommendations for Future Work

ENTER RECOMMENDATIONS FOR FUTURE WORK HERE

[Back to top](#)

This report was autogenerated by the Viskit Event Graph and Assembly modeling tool using Simkit discrete-event simulation (DES) libraries. Online at <https://github.com/open-dis/viskit> and <https://github.com/ahbuss/Simkit>.

- Thus well suited for iterative analysis and experimentation
- Report files can be saved and reopened as needed – archival.
- Improvements ongoing and suggestions always welcome

Infrastructure

Open-source license

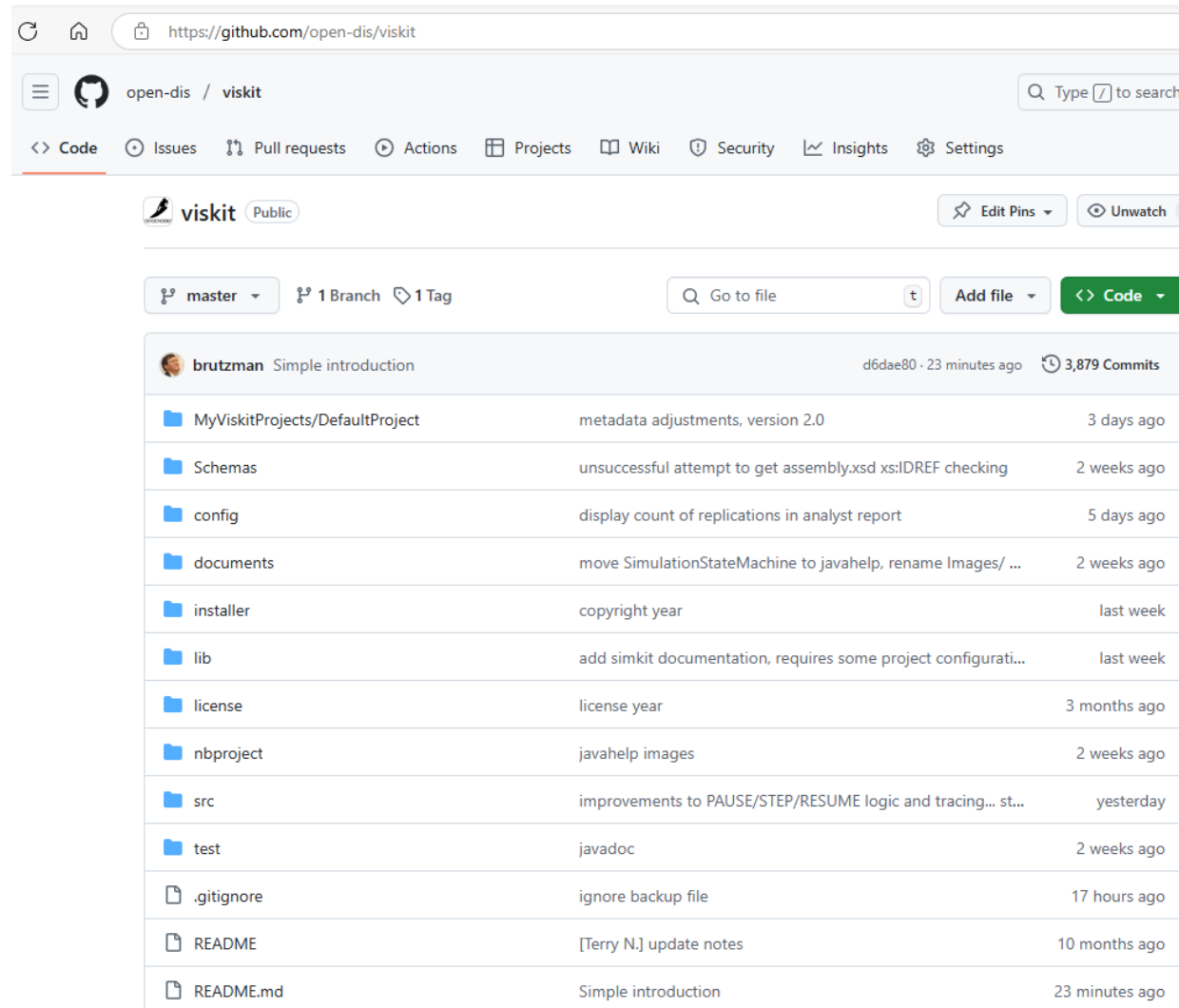
Copyright (c) 1995-2025 held by the author(s). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of the [Naval Postgraduate School \(NPS\) Modeling Virtual Environments and Simulation \(MOVES\) Institute](#) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
- THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Open-source repository has moved to github

<https://github.com/open-dis/viskit>



- Previously hosted at <https://gitlab.nps.edu/Savage/viskit> to all NPS personnel
 - (U.S. and international)
- Inclusion under open-dis project will hopefully encourage greater use and contributions

MV3500 Networked Graphics for Simulation

<https://gitlab.nps.edu/Savage/NetworkedGraphicsMV3500>

The screenshot shows the GitLab web interface for the project `NetworkedGraphicsMV3500`. The browser address bar displays `https://gitlab.nps.edu/Savage/NetworkedGraphicsMV3500`. The interface includes a left sidebar with navigation options: **Project**, `NetworkedGraphicsMV3500`, **Manage**, **Code**, **Deploy**, and **Analyze**. The main content area shows the project name `NetworkedGraphicsMV3500` with a globe icon, a dropdown menu set to `master`, and buttons for **Find file** and **Code**. Below this, a commit summary is shown: `omit 20203 from SISO-REF-010, update some links` by `brutzman` 2 months ago, with commit ID `98408467` and a **History** button. A table lists the project's directory structure and the last commit for each:

Name	Last commit	Last update
archive	remove build-impl.xml NetBeans a...	9 months ago
assignments	omit broken links in javadoc	2 months ago
conferences	Update file README.md	1 year ago
documentation	page title icon	2 years ago
examples	Security warnings in comments	2 months ago
lib	updated external jar files	2 months ago
nbproject	[Terry N.] enable viskit run from ID...	7 months ago
presentations	touchups	8 months ago
specifications	omit 20203 from SISO-REF-010, u...	2 months ago
viskit	[Terry N.] update	7 months ago

Course slidesets and examples soon to migrate to github as well

Viskit demo video

<https://savage.nps.edu/Savage/videos/Viskit-demo-2023MAR4.mp4>

Visual
Simkit
(VISKIT)
Demo

0:01 / 3:23

Visual Simkit (Viskit) Demo

Source: History

Mode: []

Output: []

Debugger Console

Event Graph

```
graph LR
    Run((Run)) -- "t_a {q.clear(), S=k, N=0, R=0}" --> Arrival((Arrival))
    Arrival -- "t_s (S > 0)" --> StartService((Start Service))
    StartService -- "t_e (S = S + 1, h = q.removeFirst())" --> EndService((End Service))
    EndService -- "t_r (Q > 0)" --> StartService
    StartService -- "h {R = R + 1, q.remove(c)}" --> Renege((Renege))
    Renege -- "t_r {N = N + 1, q.add(N)}" --> Arrival
```

Figure 4-10. Multiple Server Queue with Customers who Renege

As shown by the Event Graph in Figure 4-10, an arriving customer schedules the Renege event upon arrival, so that whenever it occurs, the queue is decremented. If the StartService event occurs first, however, then the Renege event that corresponds to that customer is removed. Note that the expression $M + R$ that is on the cancelling edge from StartService to Renege gives the number of the customer who has just started service.

The state variables M , R , and N can be used to compute the proportion of customers who renege (R/N) and the proportion of customers who receive service (M/N). The time-varying averages of S and Q can be used to estimate the average number in the queue and the average utilization of the servers.

Note that Little's formula cannot be applied in this situation to estimate the delays in queue or the time in the system for customers who received service. This is because the queue count includes all customers and can't distinguish (before a Renege occurs) between those who will eventually renege and those who will receive service. Applying Little's formula would require two state variables, one for served customers and one for renege customers. Since this cannot be done until after the fact, a model that explicitly computes these times is needed (see below).

4.5. Containers

For certain situations it is convenient to model using containers to hold data, as the previous model illustrated. In Event Graph models a container can be thought of as a state variable with a value that can go beyond simple numerical variables.

Contact

Don Brutzman

brutzman@nps.edu and don.brutzman@gmail.com

Terry Norbraten

tdnorbraten@nps.edu and terry.norbraten@gmail.com