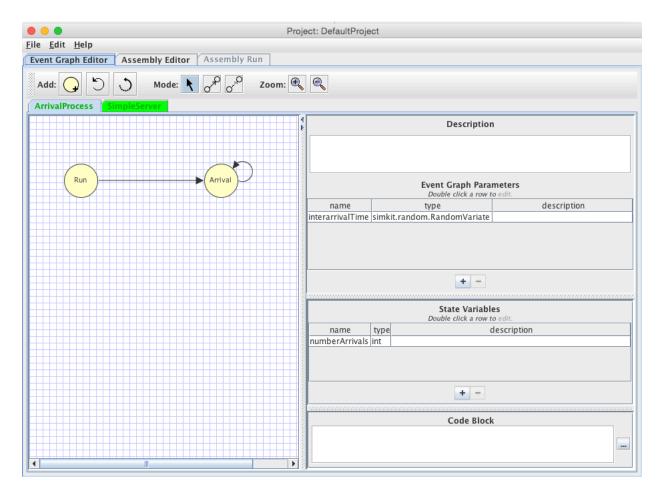
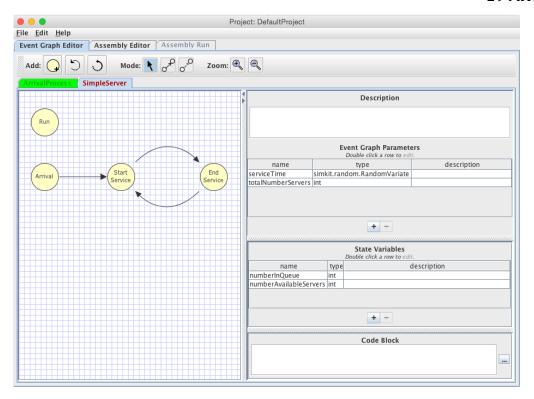
Viskit is the graphical user interface (UI) that leverages Simkit, an application programmer interface (API) that supports creation and analysis of component-based DES models.

Two modes of Viskit are utilized to achieve the goal of creation and analysis of DES models, (1) component design mode, and (2) analysis mode.

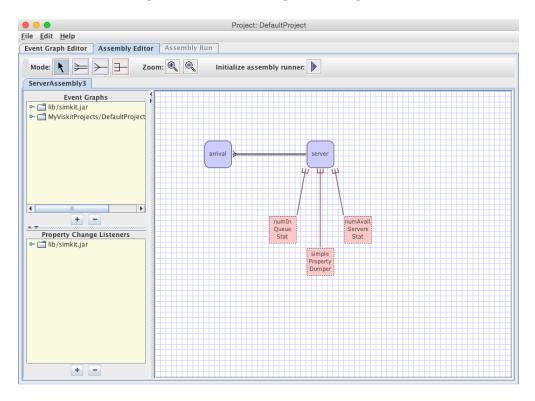
In the component design mode, a non-programmer can graphically "draw" one or more event graphs, and apply appropriate parameters to those graphs. In the component construction (assembly) mode, components (event graphs) are "hooked" together to create a model. In the analysis mode, models are exercised (simulation run) where their various outputs are compared to expected results based on an experiment design.



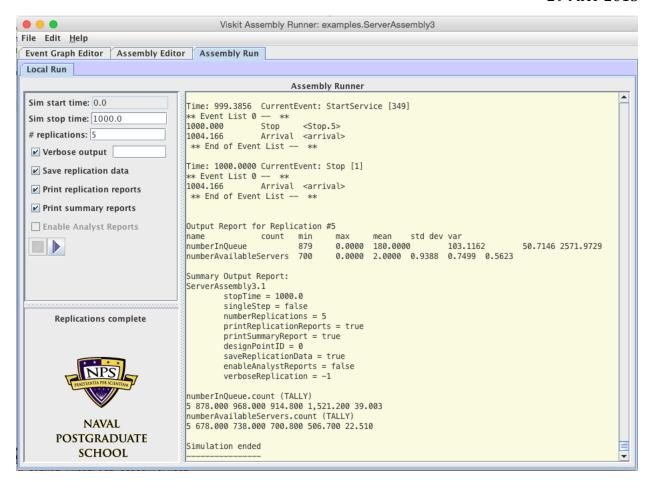
1. Arrival Process Event Graph showing parameters and state variables



2. SimpleServer Event Graph showing status as not valid



3. Component Construction (Assembly) showing Listener Event Graph Object (LEGO) connections

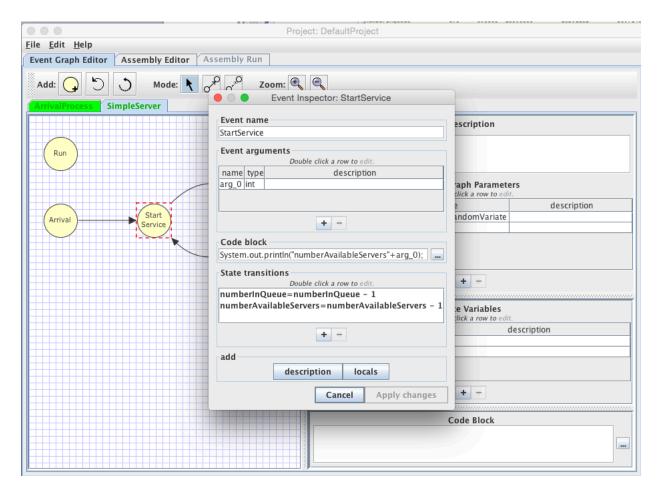


4. Assembly Run output showing individual replication and a summary reports

DES methodology is a way of modeling a situation as a series of events that take place during, or within a given situation. Two elements of DES are noteworthy. First, DES models advance time according to the Next Event rule. A list of future events (the "Event List") hold the pending list of scheduled future events at any point in time. Rather than advancing time in discrete, uniform increments, the simulation is advanced to the next scheduled event pending on the Event List. Events can be scheduled to occur according to a sophisticated pseudo-random distribution of choice, or at fixed intervals of time. The second identifying element is that defined state variables stay constant between events, but can change instantaneously during the next scheduled event. It is the change in value of these state variables that occur during state transitions between events that can be tracked and collected over simulated time that reveal interesting insights of statistical significance.

The backbone of how Viskit operates is that once components are authored graphically, that model is saved out in Extensible Markup Language (XML) format. That XML is then parsed to generate Java source code. Once the Assembly is constructed, to include LEGO listening patterns, the Assembly is also serialized to XML and Java source code is generated.

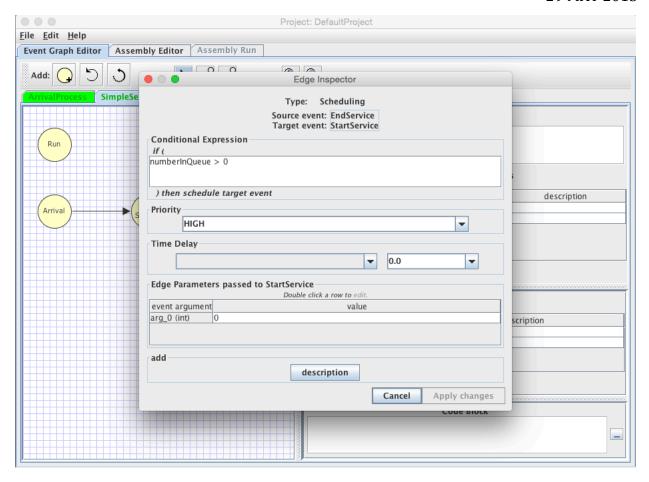
When the simulation is ready to be run for however many replication desired, the compiled Java source code is executed to produce the results seen in Figure 4.



5. Event Inspector showing method argument and state transition editors

Visual Discrete Event Simulation (DES) Toolkit (VISKIT)

29 MAY 2015



6. Scheduling Edge Inspector showing conditional, priority and time delay editors

```
• • •
                                                                              SimpleServer.xml

    SimEntity

          name = SimpleServer
          package = examples
version = 0.1
extend = SimEntityBase
          noNamespaceSchemaLocation = http://diana.nps.edu/Simkit/simkit.xsd  
• Parameter
              name = serviceTime
               type = simkit.random.RandomVariate
          Parameter
            name = totalNumberServers
type = int
          StateVariable
              name = numberInQueue
type = int
          StateVariable
              name = numberAvailableServers
              type = int
              name = Run

→ ■ StateTransition

                state = numberInQueue
     <Parameter name="serviceTime" type="simkit.random.RandomVariate" />
<Parameter name="totalNumberServers" type="int" />
<StateVariable name="numberInQueue" type="int" />
<StateVariable name="numberAvailableServers" type="int" />
     <Event name="Run">
  <StateTransition state="numberInQueue">
        <Assignment value="0" />
</StateTransition>
        <StateTransition state="numberAvailableServers">
<Assignment value="totalNumberServers" />
        </StateTransition>
<Coordinate x="20.0" y="30.0" />
     </Event>
<Event name="Arrival">
                                                                                                                                                                       Close
```

7. Event Graph serialization to XML format

```
Font: Larger Smaller Print Find Find next
          protected int numberAvailableServers;
          @viskit.ParameterMap
24:
25:
26:
27:
              names = {"serviceTime",
"totalNumberServers"},
              types = {"simkit.random.RandomVariate",
    "int"}
29:
30:
31:
          /** Creates a new default instance of SimpleServer */
32:
33:
34:
          public SimpleServer() {
35:
36:
37:
          setServiceTime(serviceTime):
38:
39:
               setTotalNumberServers(totalNumberServers);
40:
41:
42:
43:
44:
          @Override
          public void reset() {
              super.reset();
numberInQueue = 0;
45:
46:
47:
               numberAvailableServers = totalNumberServers;
48:
49:
          /** Bootstraps the first simulation event */
          public void doRun() {
50:
51:
               fire Property Change ("number In Queue", getNumber In Queue()); fire Property Change ("number Available Servers", getNumber Available Servers()); \\
52:
53:
54:
55:
56:
          public void doArrival() {
                                                                               Compile test
```

8. Generated Java source code from XML

29 MAY 2015

Conclusion:

The requirement for rapid development and implementation of DES models has been present for quite some time and will continue to persist. Effective tools are required to support this effort. Simkit is one such API that supports this requirement, is open source and has proven useful as a DES tool of choice utilized to conduct experiments in numerous research programs, as a teaching aid in the classroom, and in order to complete many Masters theses at the Naval Postgraduate School (NPS).