Cadena 2.0: Scripting Architecture

May 15, 2006

SAnToS Laboratory, Kansas State University, USA

http://cadena.projects.cis.ksu.edu/

Todd Wallentine
John Hatcliff

Support

US Army Research Office (ARO)
US National Science Foundation (NSF)
US Department of Defense
Advanced Research Projects Agency (DARPA-PCES)

Rockwell-Collins ATC
Lockheed-Martin [Eagan] – (PCES sub-contract)
Honeywell Technology Center and NASA Langley
IBM, Intel, Sun Microsystems

Overview

- Scripting Framework
- Filtering
- Generation

Scripting Framework

- Cadena includes a scripting framework
  - This allows users of Cadena to
    - Quickly prototype new features without having to write Eclipse plugins
    - Share small snippets of business logic
    - Make use of the filtering and generation capabilities currently implemented
  - The current framework uses the Python language with the Jython interpreter
    - Provides access to the Cadena object model

Filtering

- Filtering allows the user to customize the view of a Cadena model
- Filters can be combined into a “pipeline”
- For example, we may want to show only components (filtering out all connections)
- For example, we may want to show only components, and the associated connections (filtering out all sub-scenarios)

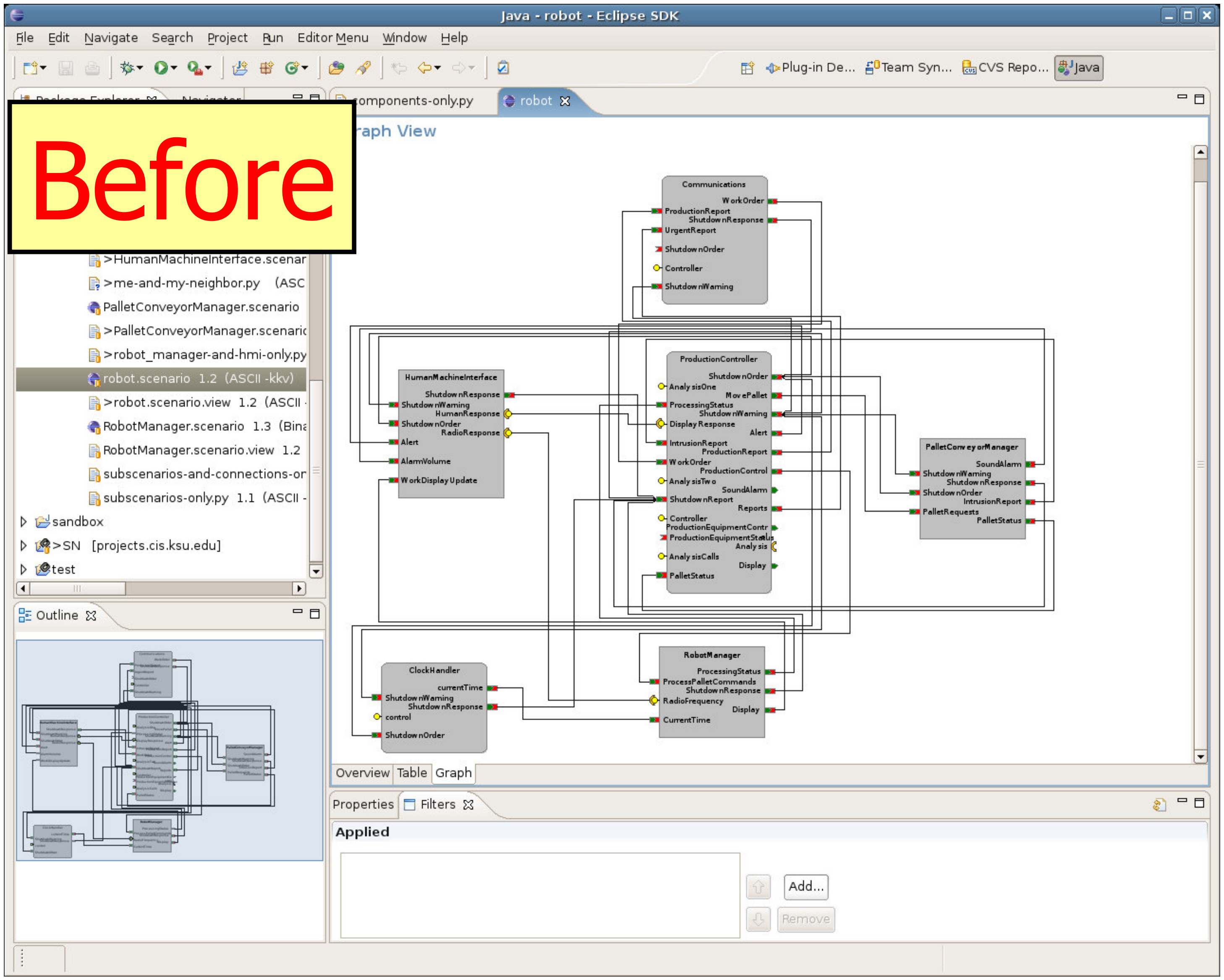
Filtering - Just Components

- In this example we are only interested in seeing components. Put another way, we want to filter out all of the connections.

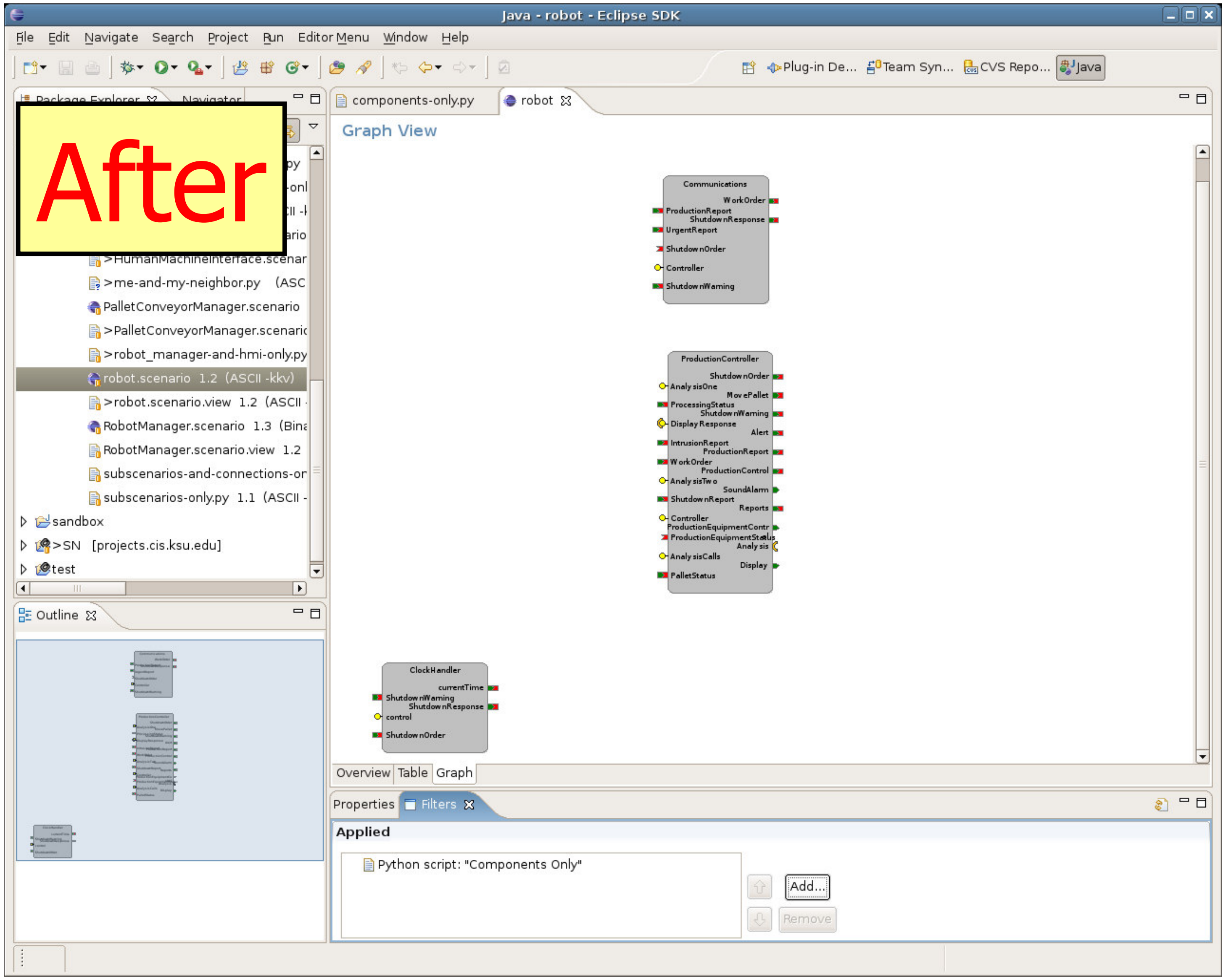
```
def filterScenario(scenario, prevStageResults):
        comps = scenario.componentInstances
        return {}.fromkeys([] + comps, FilterAction.SHOW)
```

Filtering - Just Components (contd)

Before



Filtering - Just Components (contd)



After
Java - robot - Eclipse SDK
File Edit Navigate Search Project Run Editor Menu Window Help
Plug-in De... Team Syn... CVS Repo... Java
Package Explorer Navigator
>HumanMachineInterface.scenar
>me-and-my-neighbor.py (ASC
PalletConveyorManager.scenario
>PalletConveyorManager.scenario
>robot_manager-and-hmi-only.py
robot.scenario 1.2 (ASCII -kkv)
>robot.scenario.view 1.2 (ASCII
RobotManager.scenario 1.3 (Bina
RobotManager.scenario.view 1.2
subscenarios-and-connections-or
subscenarios-only.py 1.1 (ASCII
sandbox
>SN [projects.cis.ksu.edu]
test
Outline
components-only.py
robot
Graph View
Communications
WorkOrder
ProductionReport
ShutdownResponse
UrgentReport
ShutdownOrder
Controller
ShutdownWarning
ProductionController
ShutdownOrder
AnalysisOne
MovePallet
ProcessingStatus
ShutdownWarning
DisplayResponse
Alert
IntrusionReport
ProductionReport
WorkOrder
ProductionControl
AnalysisTwo
SoundAlarm
ShutdownReport
Reports
Controller
ProductionEquipmentContr
ProductionEquipmentStatus
Analysis
AnalysisCalls
Display
PalletStatus
ClockHandler
currentTime
ShutdownWarning
ShutdownResponse
control
ShutdownOrder
Overview Table Graph
Properties Filters
Applied
Python script: "Components Only"
Add...
Remove

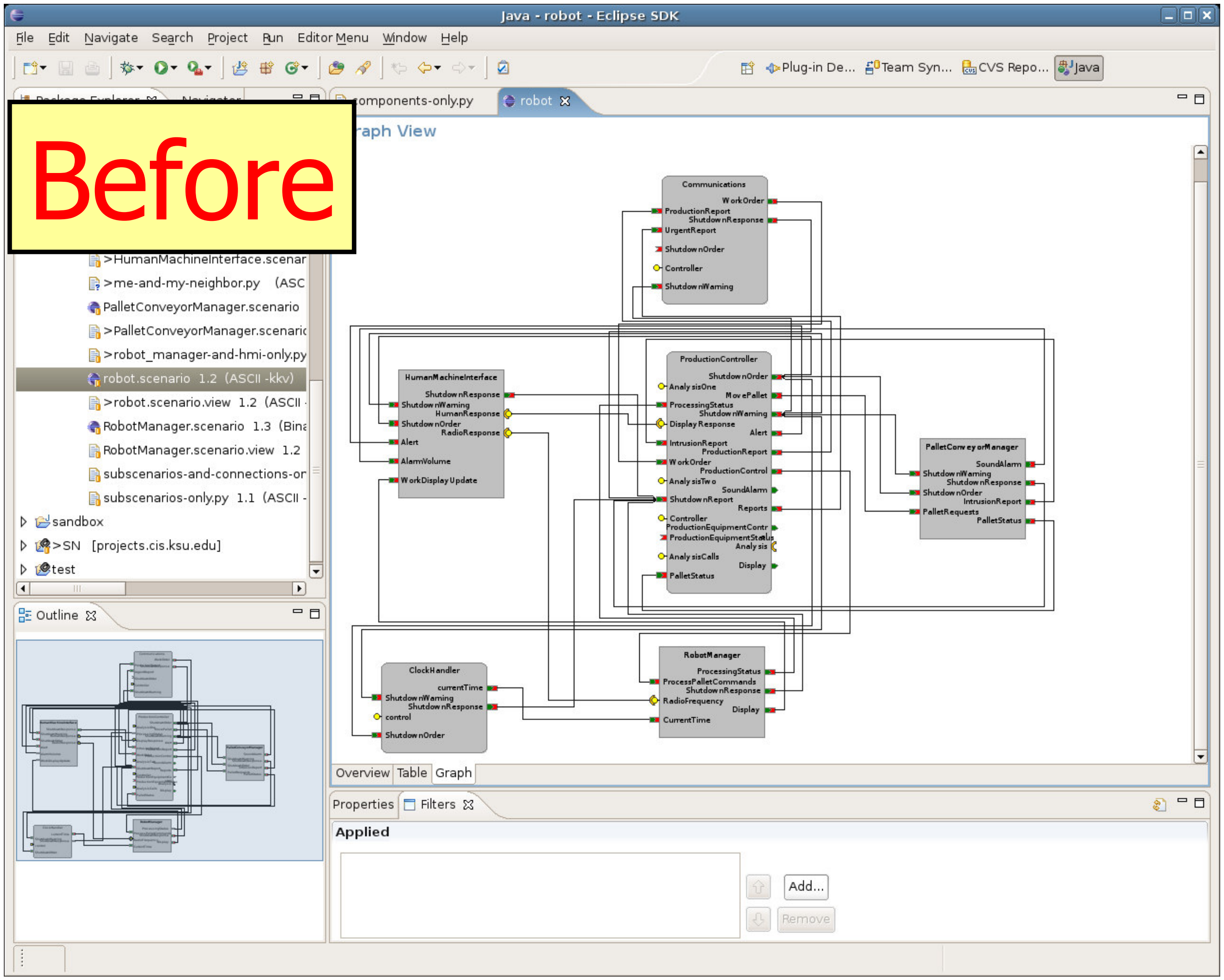
Filtering - No Sub-Scenarios

- In this example, we are only interested in seeing components and their connections and do not wish to see any sub-scenarios.

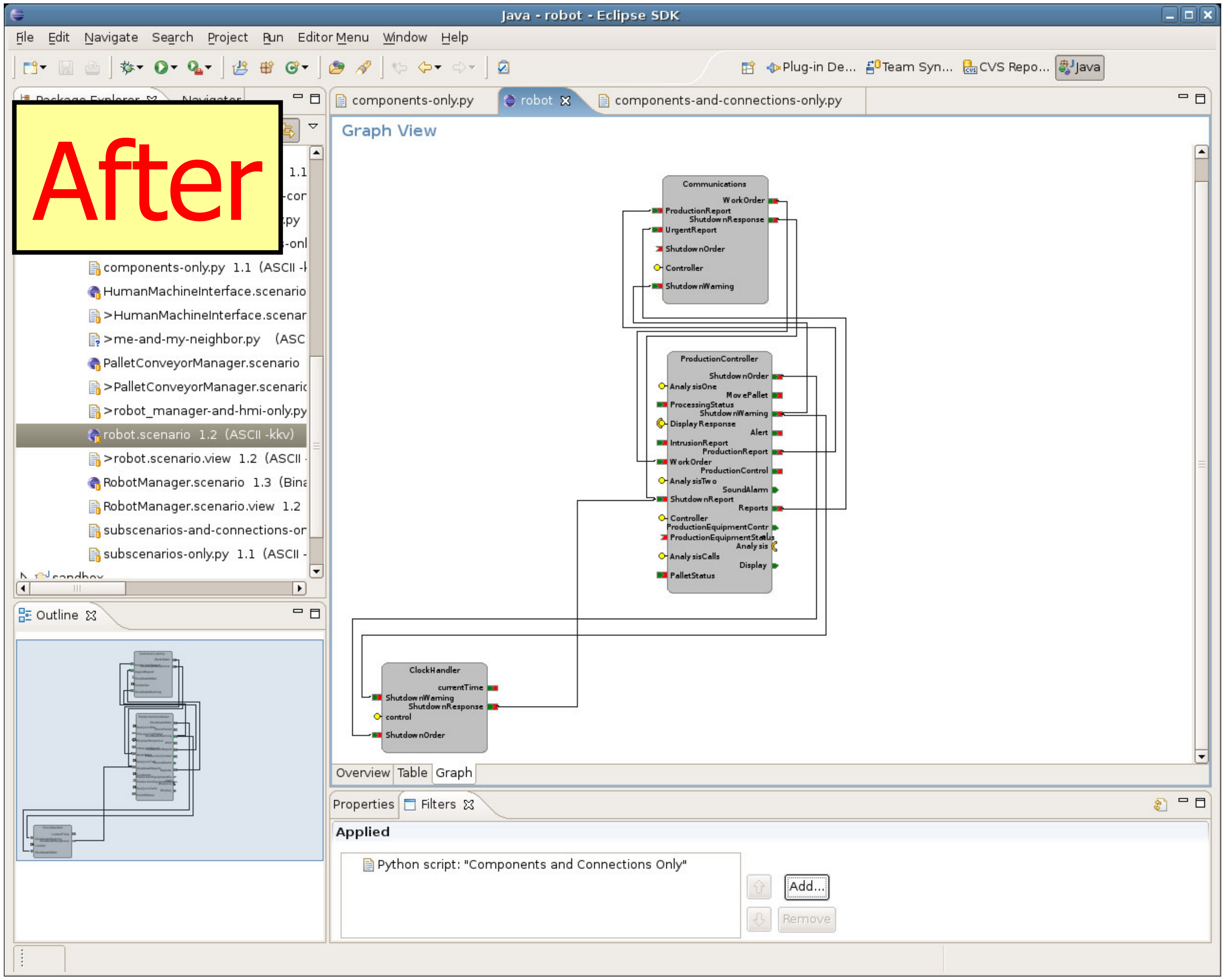
```
def forall(set, predicate): return reduce(lambda r, x: r and predicate(x), set, True)

def filterScenario(scenario, prevStageResults):
        comps = scenario.componentInstances
        conns = filter(
                lambda conn:
                        forall(conn.portBindings, lambda pb: pb.instance in comps),
                scenario.getTypedConnectors())
        pbs = reduce(
                lambda x, y: x +y,
                map(lambda conn: filter(lambda pb: pb.instance in comps, conn.portBindings),
                conns), [])

        return {}.fromkeys([] + comps + conns + pbs, FilterAction.SHOW)
```

Filtering - No Sub-Scenarios (contd)

Before



Filtering - No Sub-Scenarios (contd)



Java - robot - Eclipse SDK
File Edit Navigate Search Project Run Editor Menu Window Help
Plug-in De... Team Syn... CVS Repo... Java
After
components-only.py 1.1 (ASCII -
HumanMachineInterface.scenario
>HumanMachineInterface.scenar
>me-and-my-neighbor.py (ASC
PalletConveyorManager.scenario
>PalletConveyorManager.scenario
>robot_manager-and-hmi-only.py
robot.scenario 1.2 (ASCII -kkv)
>robot.scenario.view 1.2 (ASCII
RobotManager.scenario 1.3 (Bina
RobotManager.scenario.view 1.2
subscenarios-and-connections-or
subscenarios-only.py 1.1 (ASCII -
Outline
components-only.py
robot
components-and-connections-only.py
Graph View
Communications
WorkOrder
ProductionReport
ShutdownResponse
UrgentReport
ShutdownOrder
Controller
ShutdownWarning
ProductionController
ShutdownOrder
AnalysisOne
MovePallet
ProcessingStatus
ShutdownWarning
DisplayResponse
Alert
IntrusionReport
ProductionReport
WorkOrder
ProductionControl
AnalysisTwo
SoundAlarm
ShutdownReport
Reports
Controller
ProductionEquipmentContr
ProductionEquipmentStatus
Analysis
AnalysisCalls
Display
PalletStatus
ClockHandler
currentTime
ShutdownWarning
ShutdownResponse
control
ShutdownOrder
Overview Table Graph
Properties Filters
Applied
Add...
Remove

Python script: "Components and Connections Only"

Generation

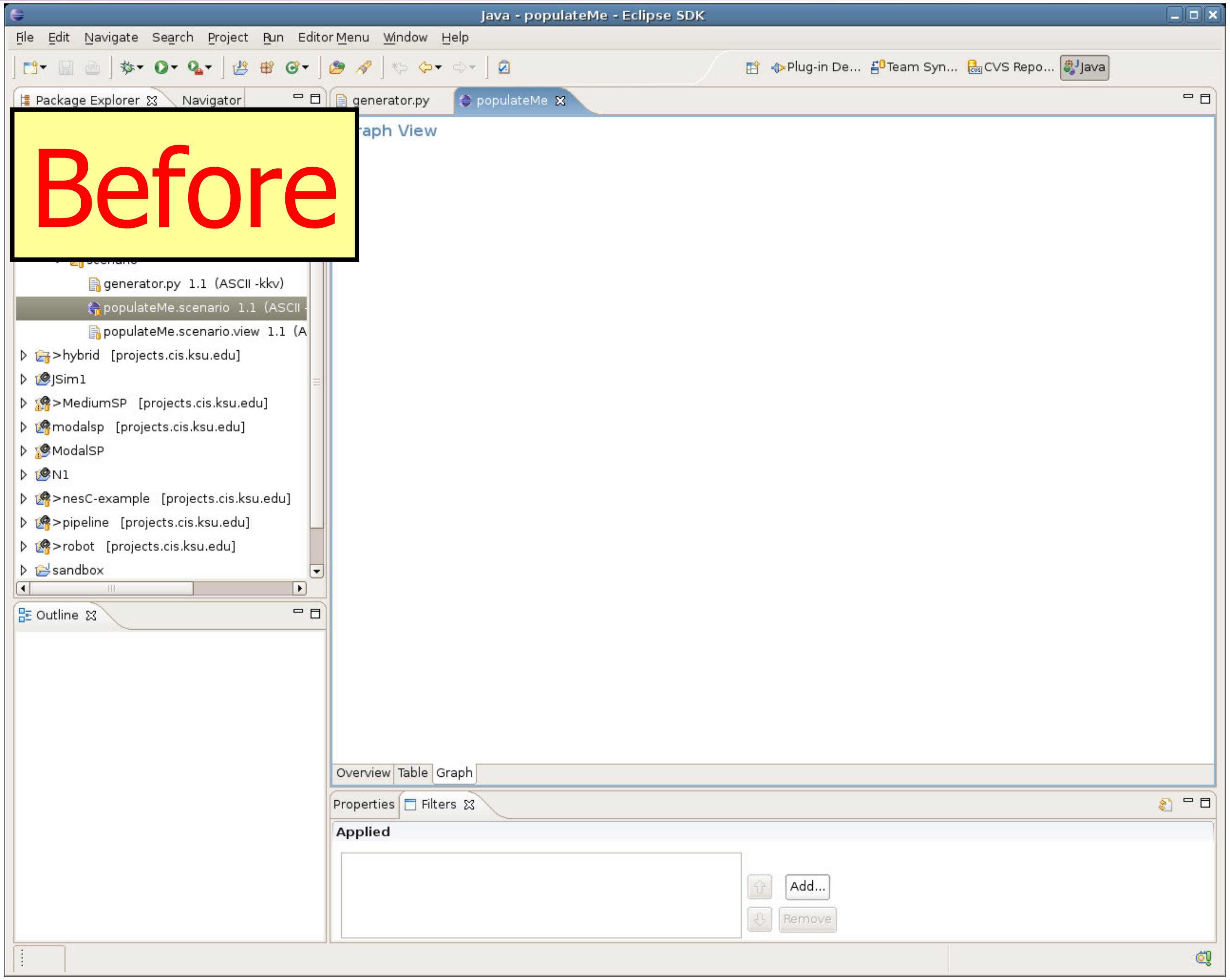
- Generation allows the user to automate the generation of models
- For example, a user may want to generate a certain number of components of a certain type with a certain set of connections. This may be useful when testing or experimenting with a set of modules/components.

Generation Example - Pipeline

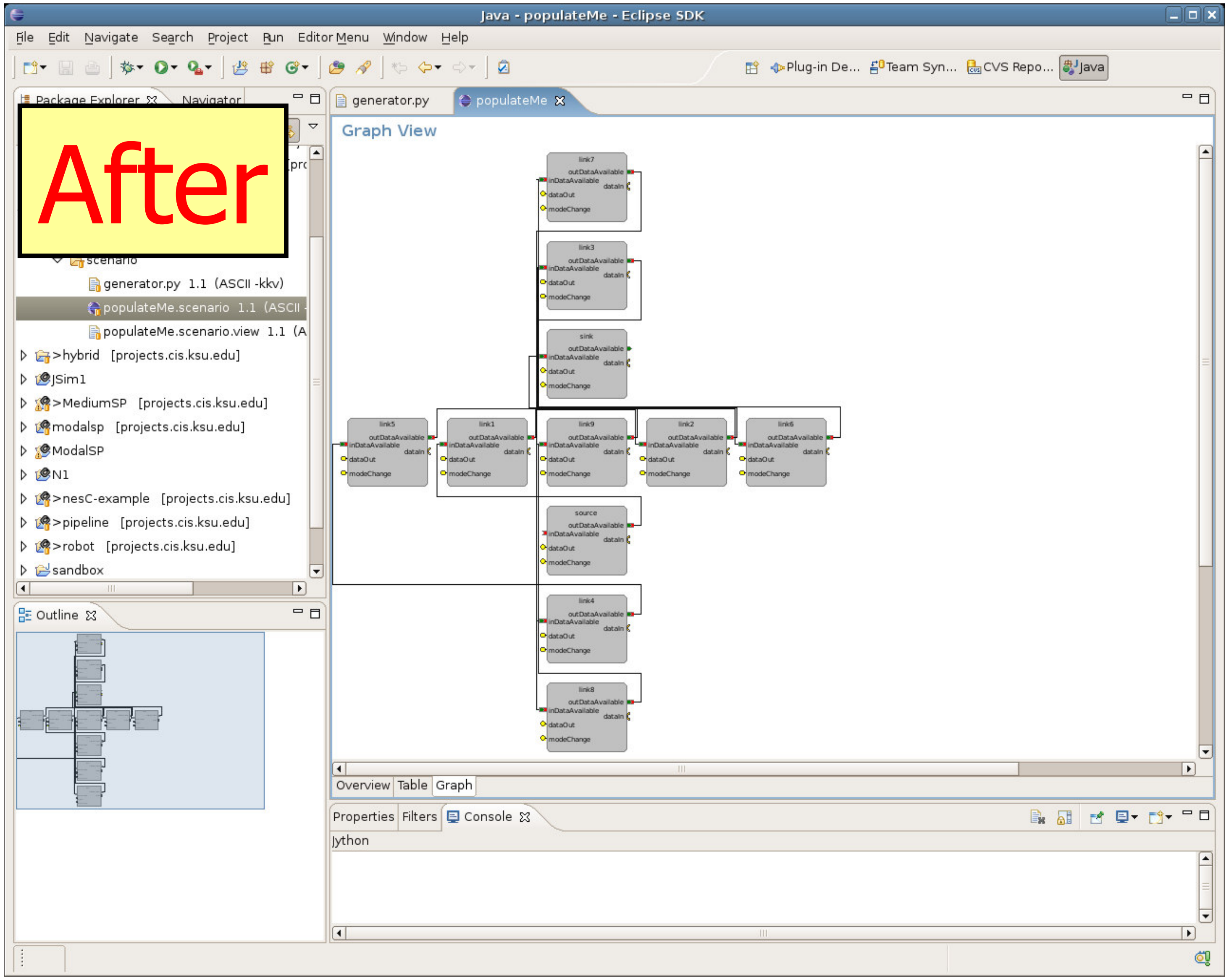
- In this example we simply want to generate a source and sink along with 10 intermediate links (named link0-9) for the information to traverse through.

```
sink = createComponentInstance(bmModal, “sink”)
source = createComponentInstance(bmModal, “source”)
self.addComponent(scenario, sink)
self.addComponent(scenario, source)
self.addConnector(scenario,
        createConnector(
                prismEventConnector,
                {
                        “usesSide”: (source, “outDataAvailable),
                        “providesSide”: (sink, “inDataAvailable”)
                }))
```

Generation Example - Pipeline (contd)


Java - populateMe - Eclipse SDK
File Edit Navigate Search Project Run Editor Menu Window Help
Plug-in De... Team Syn... CVS Repo... Java
Package Explorer Navigator
generator.py populateMe
raph View
Before
generator.py 1.1 (ASCII -kkv)
populateMe.scenario 1.1 (ASCII
populateMe.scenario.view 1.1 (A
>hybrid [projects.cis.ksu.edu]
JSim1
>MediumSP [projects.cis.ksu.edu]
modalsp [projects.cis.ksu.edu]
ModalSP
N1
>nesC-example [projects.cis.ksu.edu]
>pipeline [projects.cis.ksu.edu]
>robot [projects.cis.ksu.edu]
sandbox
Outline
Overview Table Graph
Properties Filters
Applied
Add...
Remove

Generation Example - Pipeline (contd)


Java - populateMe - Eclipse SDK
File Edit Navigate Search Project Run Editor Menu Window Help
Plug-in De... Team Syn... CVS Repo... Java
Package Explorer Navigator
After
scenario
generator.py 1.1 (ASCII -kkv)
populateMe.scenario 1.1 (ASCII
populateMe.scenario.view 1.1 (A
>hybrid [projects.cis.ksu.edu]
JSim1
>MediumSP [projects.cis.ksu.edu]
modalsp [projects.cis.ksu.edu]
ModalSP
N1
>nesC-example [projects.cis.ksu.edu]
>pipeline [projects.cis.ksu.edu]
>robot [projects.cis.ksu.edu]
sandbox
Outline
generator.py
populateMe
Graph View
link7
link3
sink
link5
link1
link9
link2
link6
source
link4
link8
outDataAvailable
inDataAvailable
dataIn
dataOut
modeChange
Overview Table Graph
Properties Filters Console
Jython

Conclusion

- The scripting framework allows a user to quickly develop business logic
- The framework could be extended to handle different languages (e.g., JavaScript, Groovy)
- More information can be found in the Cadena Manual Chapter 6