

# **User Manual**

For GMoDS Visualizer and Test Driver

Version 1.0

Submitted in partial fulfillment of the requirements of the degree of MSE

Mike Fraka

CIS 895 – MSE Project

Kansas State University

## Table of Contents

1	Introduction.....	3
1.1	References .....	3
2	System Overview .....	3
3	GMoDS Visualizer.....	4
3.1	Installation.....	4
3.2	Usage.....	11
3.3	Common Commands.....	11
4	GMoDS Test Driver.....	18
4.1	Installation.....	18
4.2	Usage.....	18
4.3	Common Commands.....	20
4.4	Event Script Format .....	23
4.5	Log Messages.....	24
4.6	Error Messages.....	24

# 1 Introduction

This document is the user manual for the GMoDS Test Driver and Visualizer components.

## 1.1 References

1. “Vision Document 2.0” available at <http://people.cis.ksu.edu/~mfraka/FrakaMSE.html>.

# 2 System Overview

The Goal Model for Dynamic Systems (GMoDS) is a means for specifying system requirements for agent-oriented software at design time and tracking their achievement at run time. The specification goal tree represents the specified requirements at design time. The instance goal tree represents the run time achievement profile of the specified goals.

The GMoDS Visualizer can optionally display both the specification and instance goal trees and all appropriate relations between the goals. In addition, the visualizer will display goal parameters in both the specification and instance trees and the status and parameter values of instance goals.

The GMoDS Test Driver can test the operation of GMoDS and the Visualizer. The Test Driver will execute event scripts from a file or by random generation compatible with the GMoDS model being used. The Test Driver will operate in automated or manual mode.

The system context is shown in Figure 1 below.

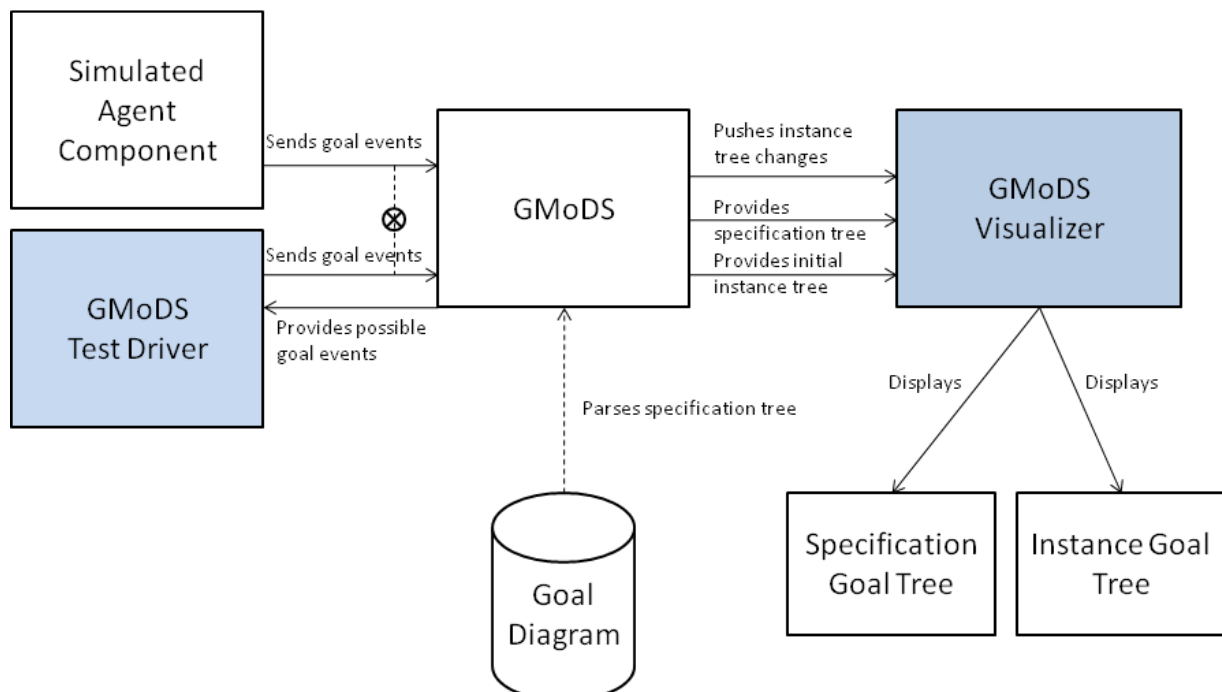


Figure 1 GMoDS Test Driver and Visualizer system context

The figure shows that either the GMoDS Test Driver or a simulated agent component (but not both) send goal events to GMoDS. GMoDS provides the possible goal events to the GMoDS Test Driver. GMoDS pushes instance tree changes to the GMoDS Visualizer using a variant of the Observer design pattern called ChangeManager. The GMoDS component provides the specification goal tree and initial instance goal tree to the GMoDS Visualizer. The GMoDS Visualizer uses the specification goal tree and instance goal tree as part of the “model” for its Model/View/Controller architecture. The GMoDS Visualizer will display the specification goal tree and initial instance tree and await changes from GMoDS. The GMoDS Visualizer does not import any layout information from the goal diagram since GMoDS goal models can be programmatically built rather than through parsing a goal diagram.

GMoDS Test Driver relies on GMoDS to define the possible goal events in order to issue random goal events. In addition, the GMoDS Test Driver can check user-provided event scripts for legality using GMoDS interfaces.

GMoDS parses the goal model diagram, if the specification tree is not programmatically built.

More detail on the system context and use cases is available in [1] (see 1.1 above).

### **3 GMoDS Visualizer**

The GMoDS Visualizer can be used as an optional visualization tool with any application using GMoDS or will be used in the GMoDS Test Driver application.

#### **3.1 Installation**

The GMoDS Visualizer is available as an Eclipse project via the Kansas State University Multiagent & Cooperative Robotics Laboratory (MACR) CVS repositories. This section describes how to install the GMoDS Visualizer as an Eclipse project from this source and assumes you have gained permission to do so.

##### **3.1.1 Check out the GMoDS Visualizer**

The first step is to open the CVS Repository Exploring perspective. Figure 2 below shows how to open an “Other” perspective. Select the CVS Repository Exploring perspective as shown in Figure 3 below.

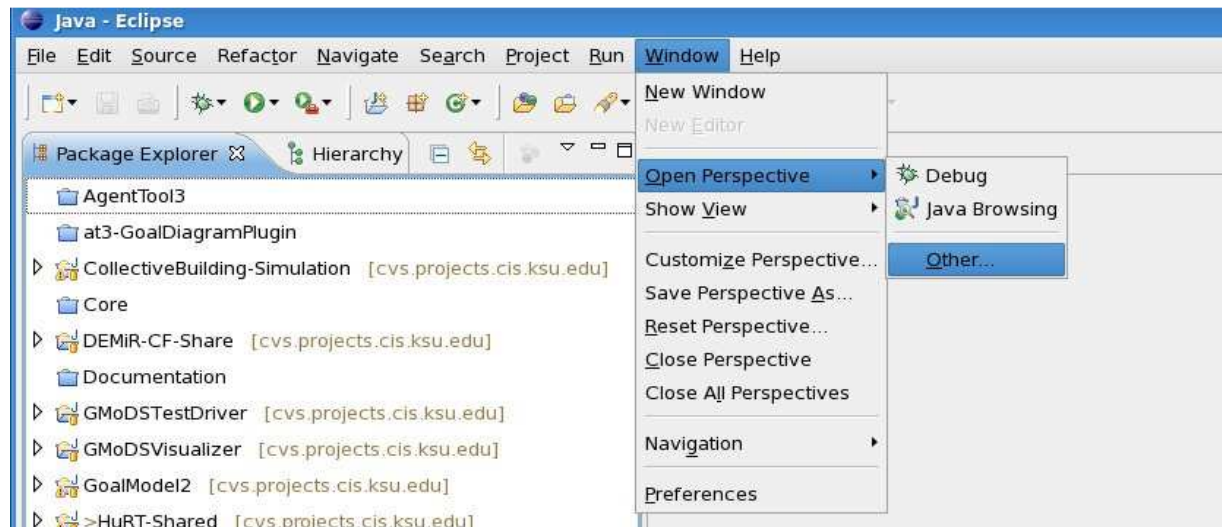


Figure 2 Window | Open Perspective | Other



Figure 3 Open Perspective | CVS Repository Exploring

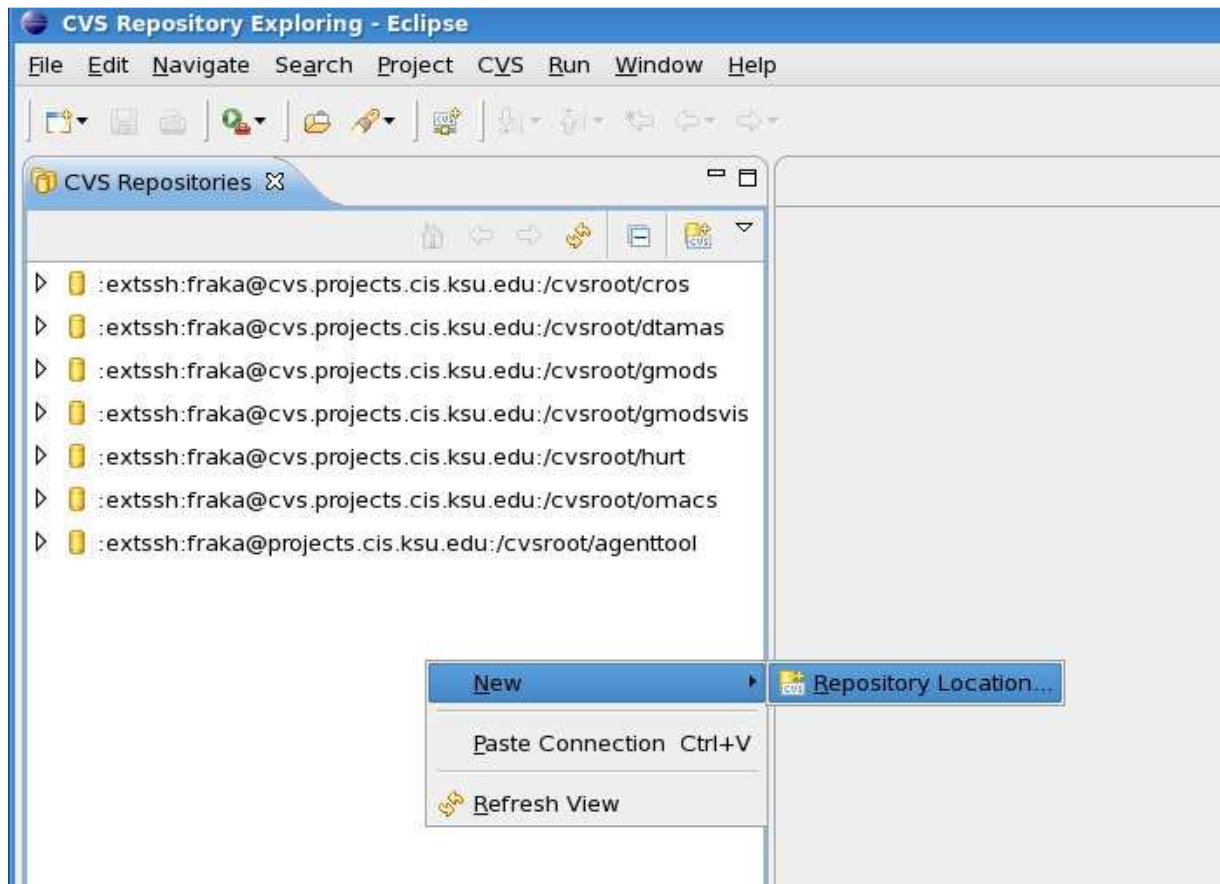
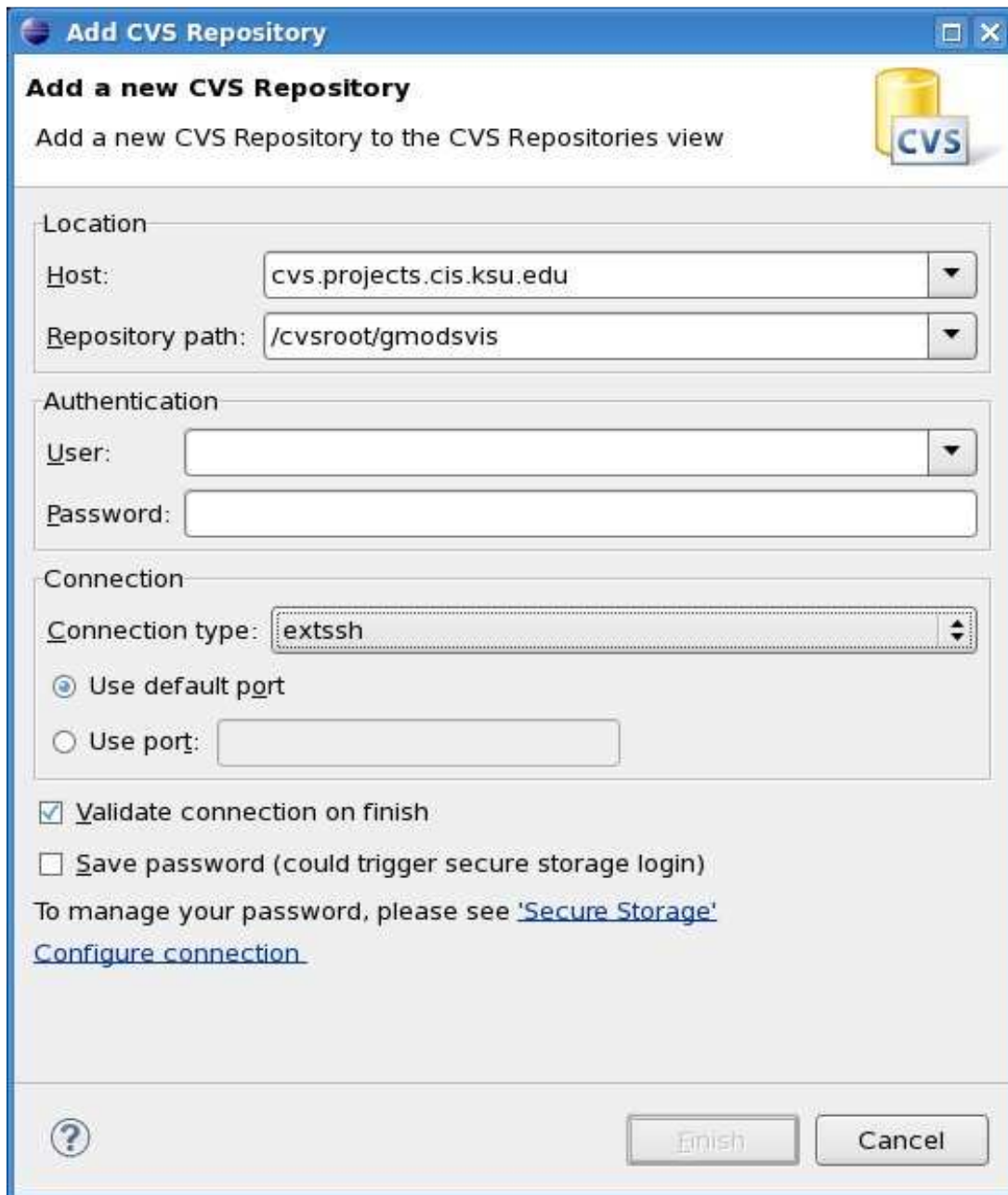


Figure 4 New Repository Location

Establish a new CVS repository location by right clicking in the “CVS Repositories” tab and choosing “New | Repository Location...” as shown above in Figure 4. Add a new CVS repository location for “/cvsroot/gmodsvis” using the host “cvs.projects.cis.ksu.edu” and connection type “extssh” as shown below in Figure 5.



**Add CVS Repository**

**Add a new CVS Repository**

Add a new CVS Repository to the CVS Repositories view

**Location**

**Host:** cvs.projects.cis.ksu.edu

**Repository path:** /cvsroot/gmodsvs

**Authentication**

**User:**

**Password:**

**Connection**

**Connection type:** extssh

☒ Use default port

☐ Use port:

☒ **Validate connection on finish**

☐ **Save password (could trigger secure storage login)**

To manage your password, please see '[Secure Storage](#)'  
[Configure connection](#)

**Finish** **Cancel**

Figure 5 Add a new CVS Repository

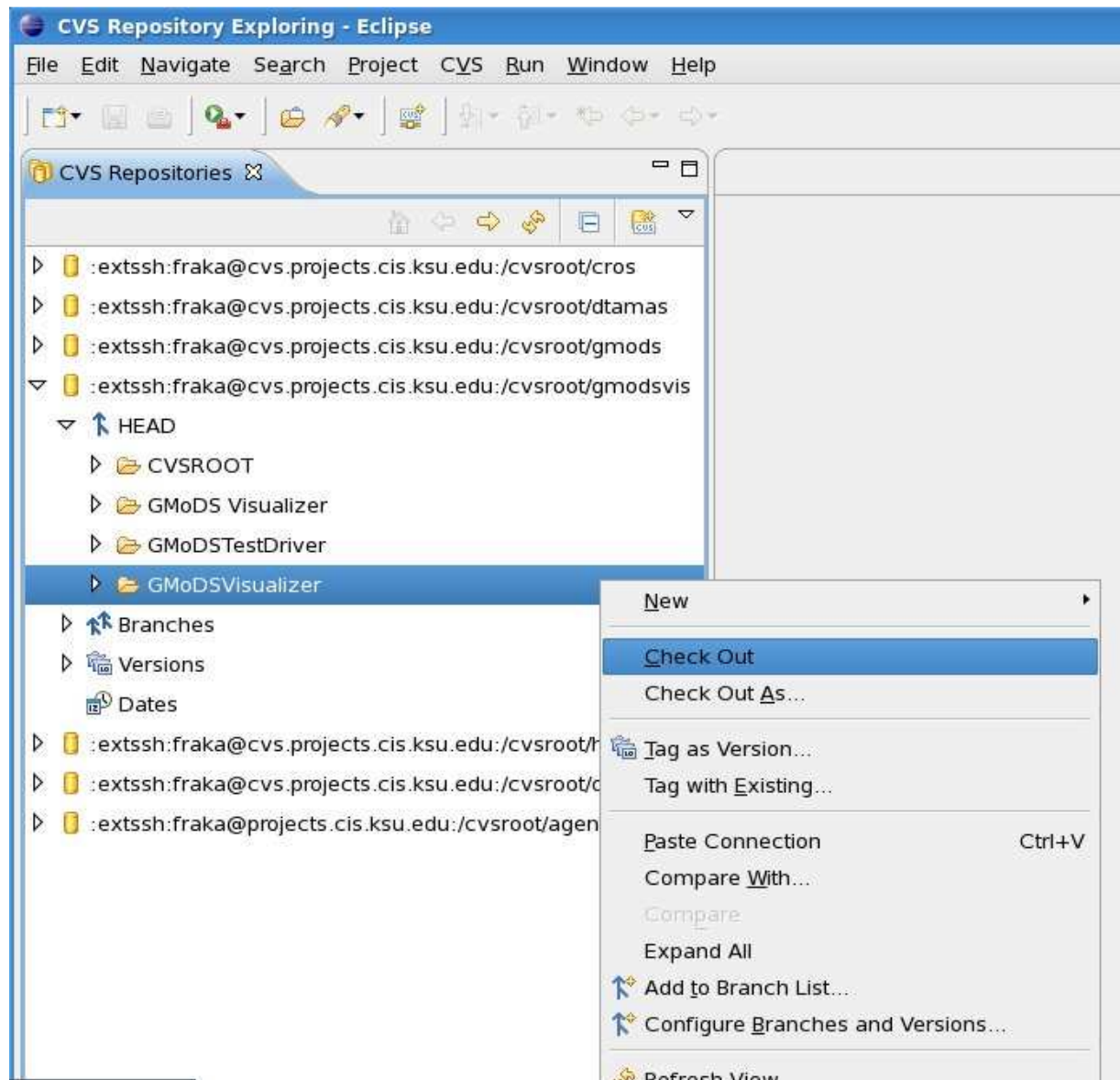


Figure 6 Check Out GMoDSVisualizer

Click the arrows to open “/cvsroot/gmodsvis” and “HEAD” then right click on “GMoDSVisualizer” (not “GMoDS Visualizer” an empty project) as shown in Figure 6 above. Choose “Check Out” or “Check Out As”.

### 3.1.2 Check out the projects the GMoDS Visualizer depends on

The GMoDS Visualizer depends on two other Eclipse projects: GMoDS and the Organization Model for Adaptive Complex Systems (OMACS). If you do not have these projects already checked out you must do so now.

#### 3.1.2.1 GMoDS

Follow the steps above to define the repository location for /cvsroot/gmods and then check out GoalModel2.



### 3.1.2.2 OMACS

Follow the steps above to define the repository location for /cvsroot/omacs and then check out OrganizationModel.

### 3.1.3 Make your Eclipse project depend on the GMoDS Visualizer

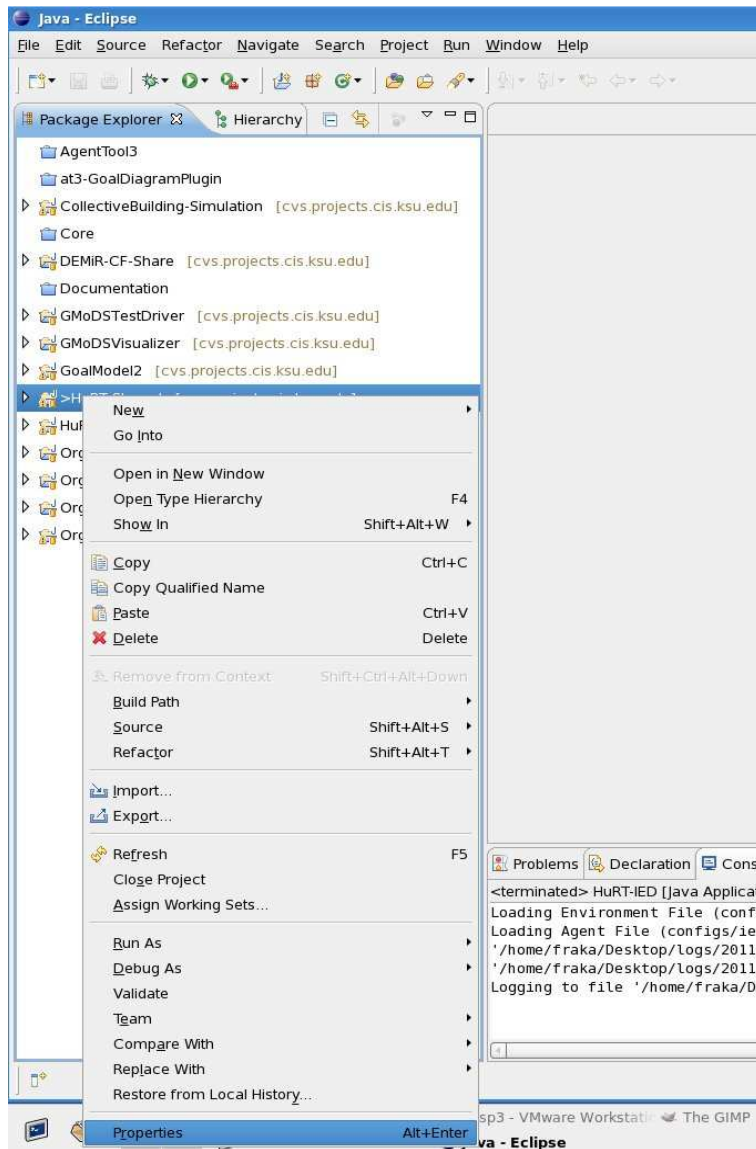


Figure 7 Project | Properties

In the Java perspective, right click the project that needs to use the GMoDS Visualizer and select “Properties” as shown above in Figure 7. In the dialog that pops up select “Java Build Path” as show in Figure 8 below. Click “Add” and select “GMoDSVisualizer” as shown in Figure 9 below.

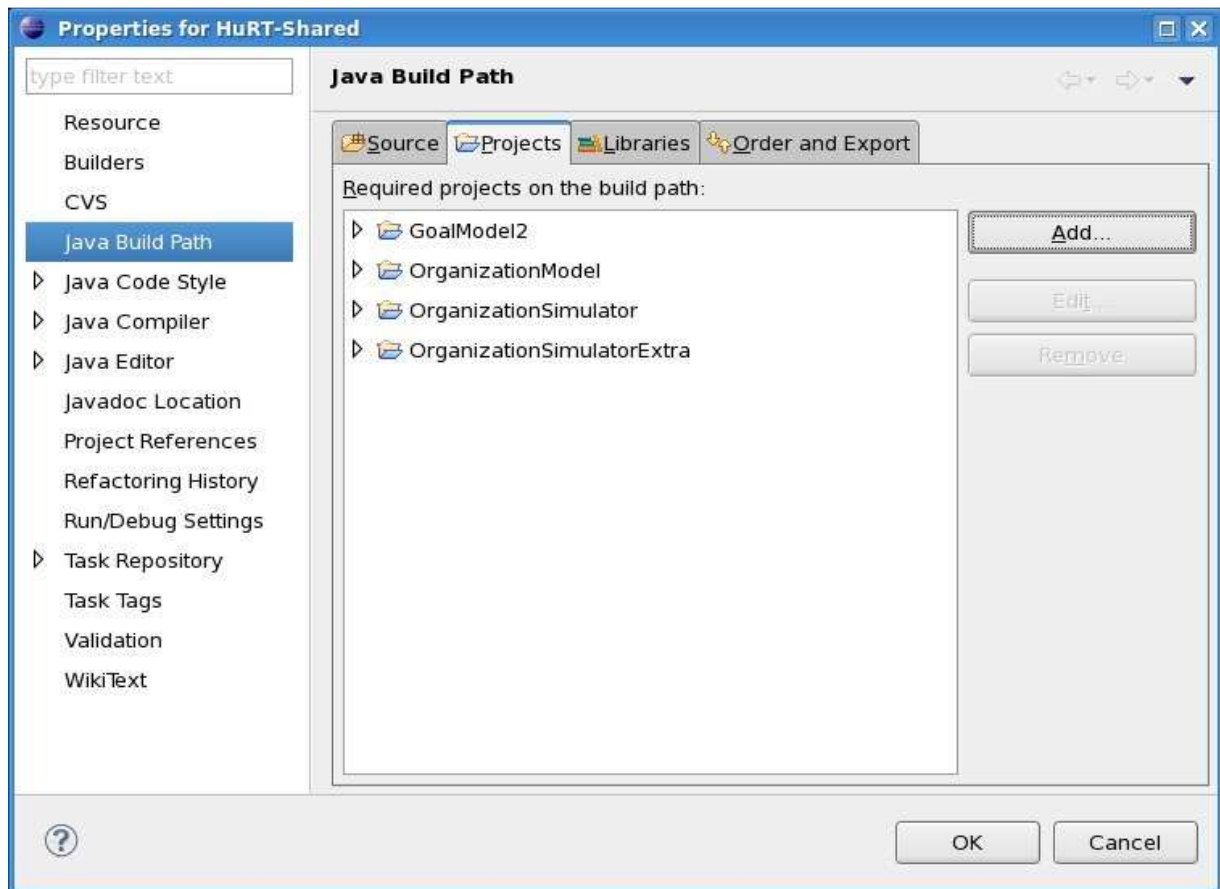


Figure 8 Java Build Path

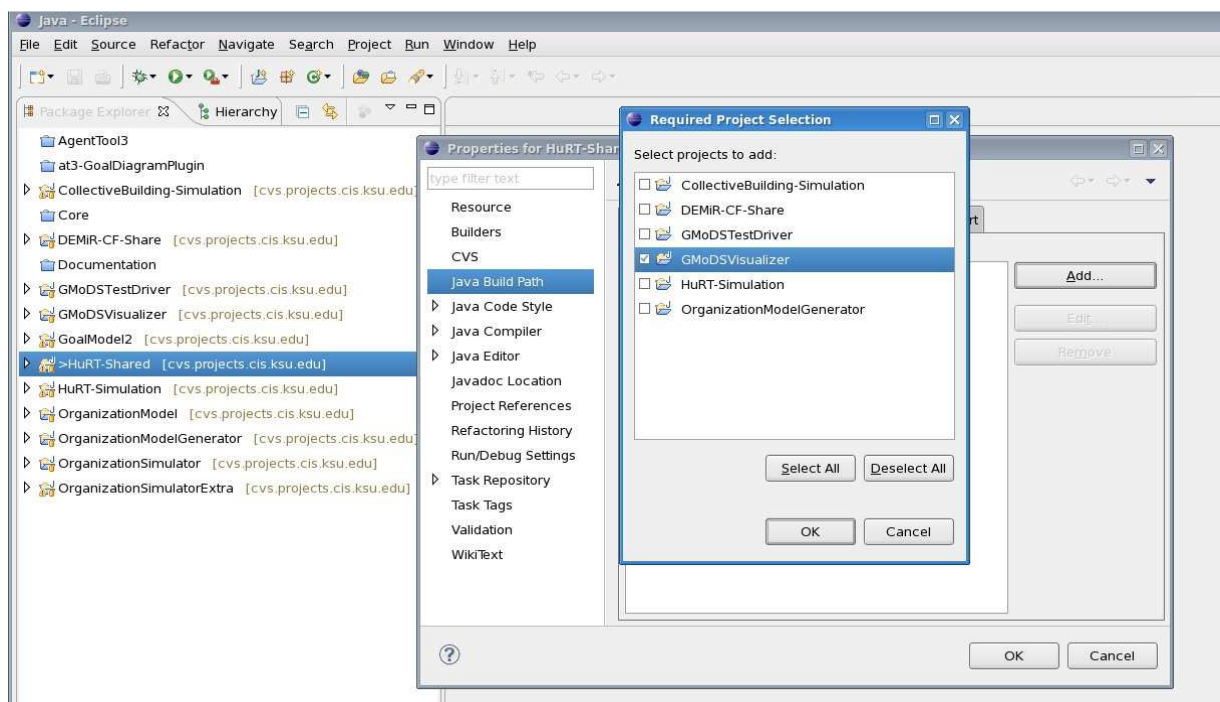


Figure 9 Required Project Selection

### 3.2 Usage

To use the GMoDS Visualizer in an application, one must construct GMoDSVisualizerImpl and then call its initialize() method. GMoDSVisualizerImpl takes 3 parameters:

- 1) SpecificationTree,
- 2) InstanceTree, and
- 3) Test Driver.

An application that uses GMoDS should have no trouble identifying objects that implement parameters 1 and 2 above. **The InstanceTree object must have its “initialize()” method called before being passed to the GMoDSVisualizerImpl constructor or an IllegalArgumentException will be thrown.** The application should send in “null” for the Test Driver.

To use the GMoDS Visualizer with the GMoDS Test Driver see 4 below.

### 3.3 Common Commands

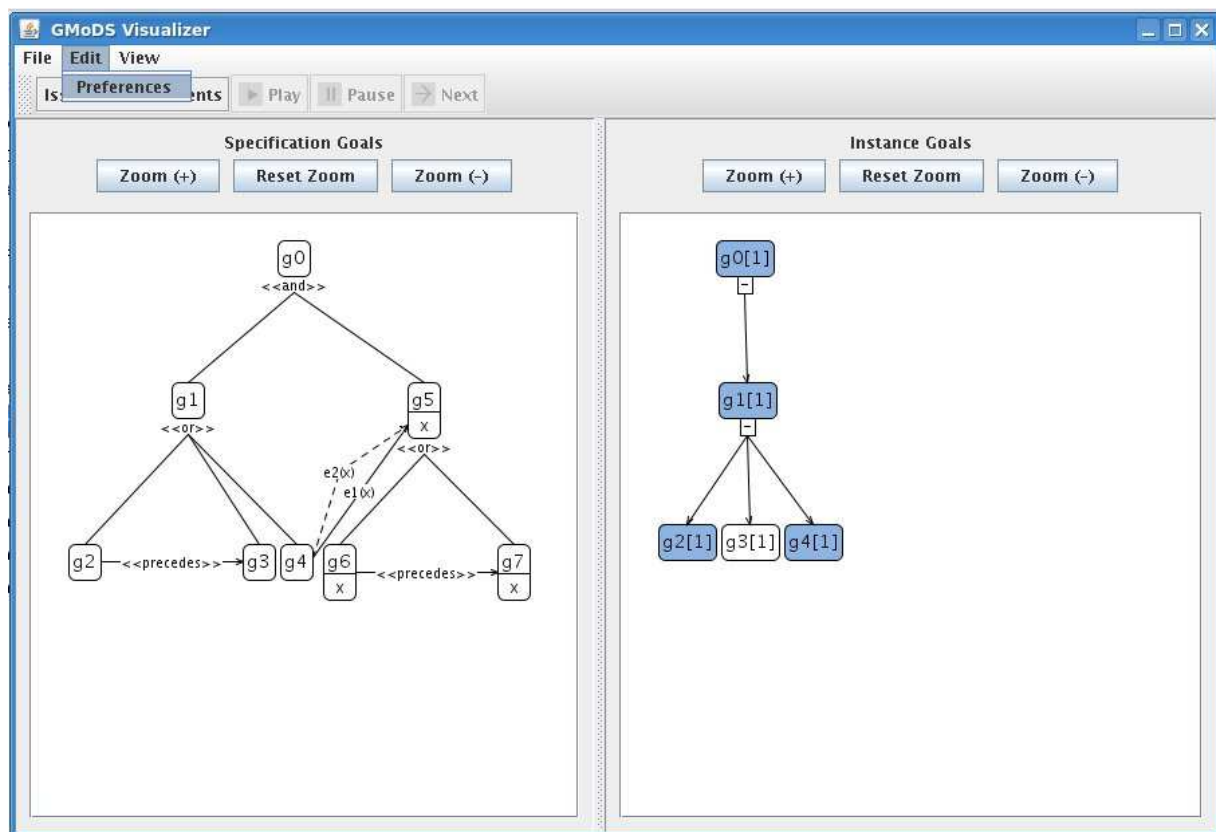


Figure 10 Edit | Preferences

Figure 10 above shows sample specification and instance trees for the goal diagram “VisionDocument2.goal”. One can zoom the specification or instance tree using the applicable buttons. Figure 10 shows that one can “Edit | Preferences”. Figure 11 below shows that one

can “Edit | Preferences | States” for the GMoDS Visualizer. This allows one to pick the total time a changed goal will flash, the time for a single flash, and select colors for any goal state.

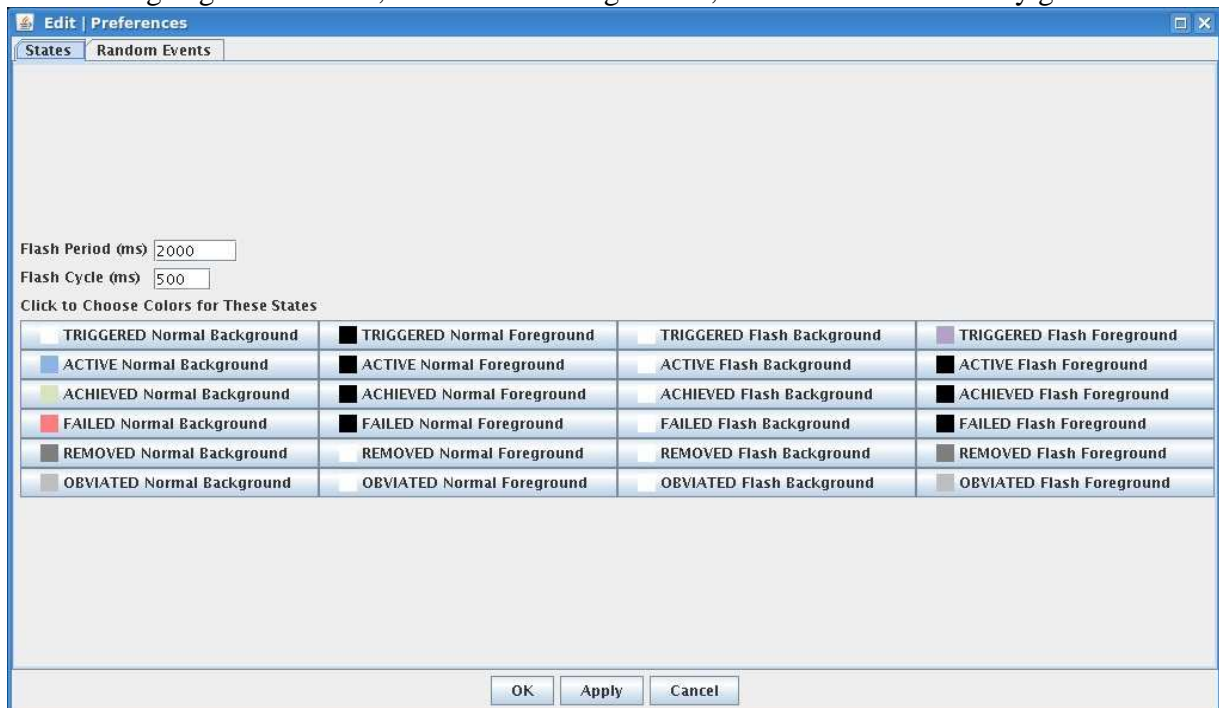


Figure 11 Edit | Preferences | States

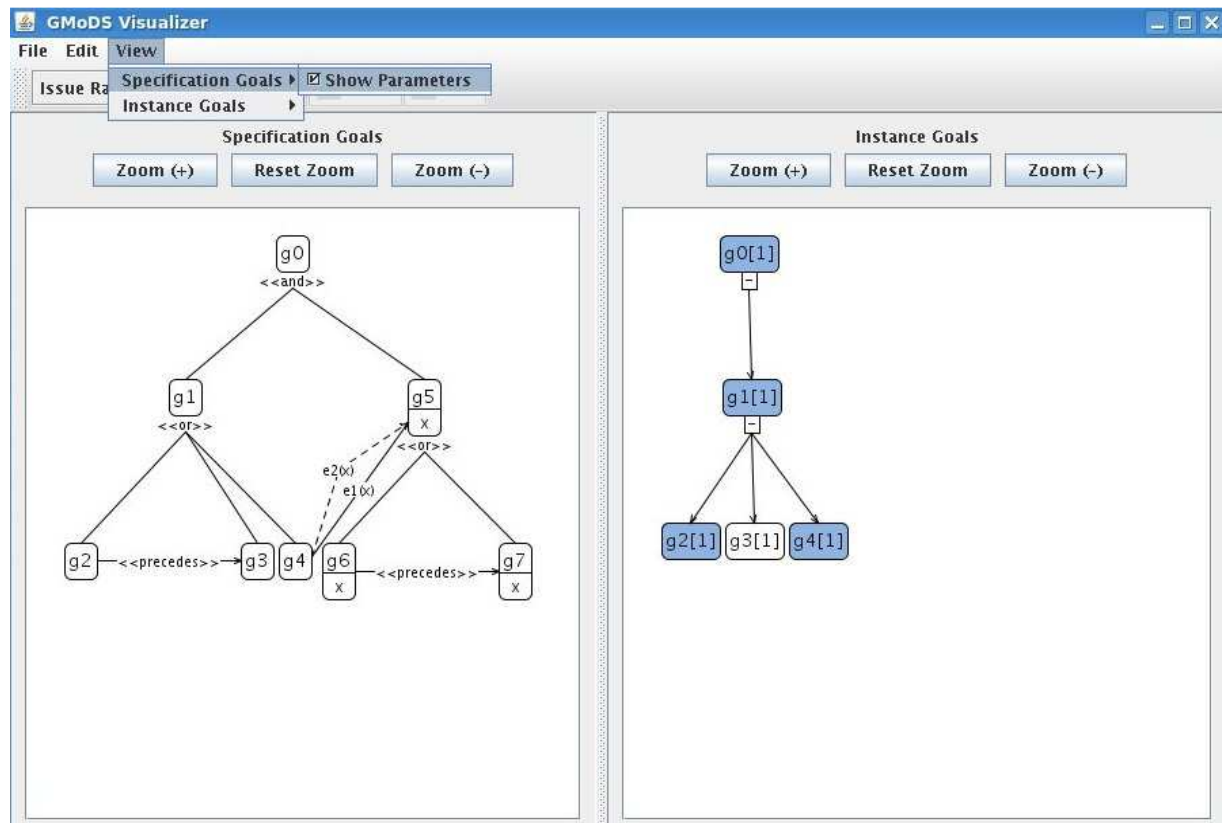


Figure 12 View | Specification Goals | Show Parameters

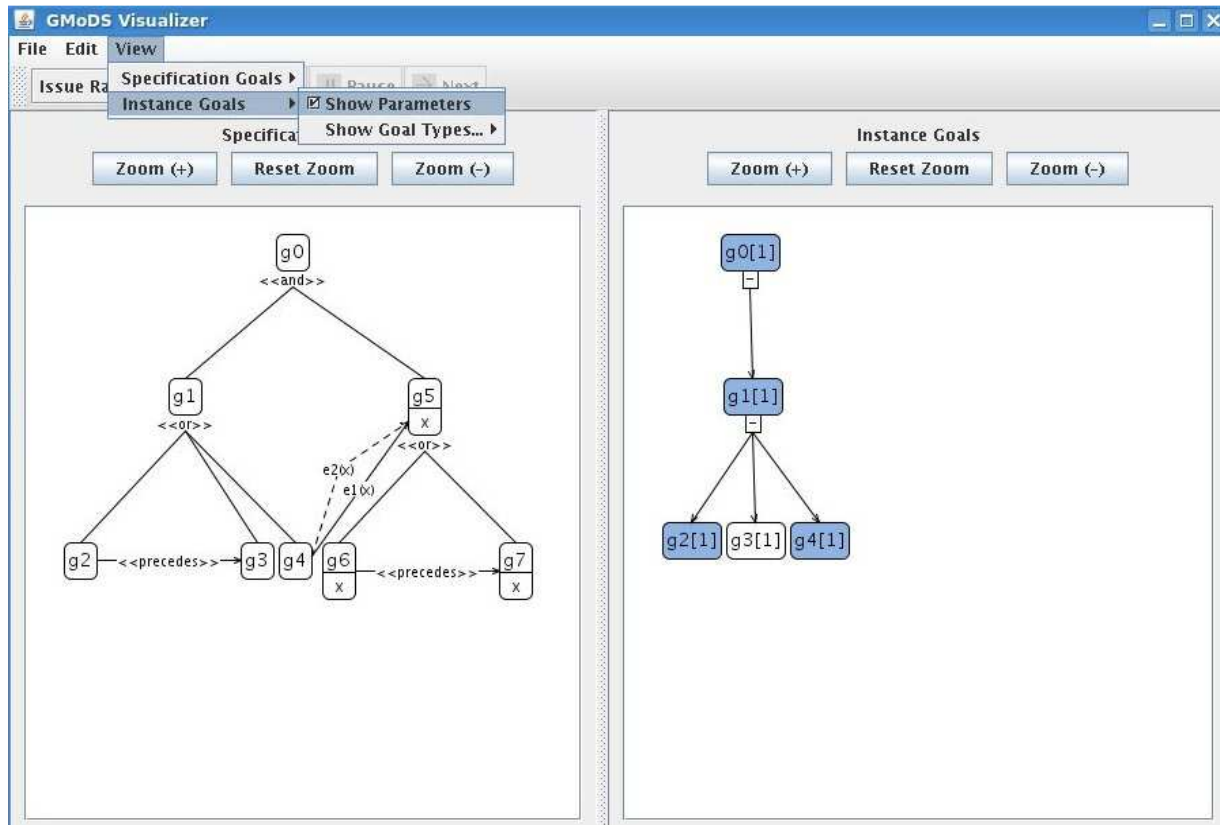


Figure 13 View | Instance Goals | Show Parameters

Figure 12 and Figure 13 above show one can view or not view the parameters throughout the specification or instance trees. Figure 14 below shows that one can view or not view all instance of any particular specification goal type. If a goal type is not viewed all descendants' instances are hidden.

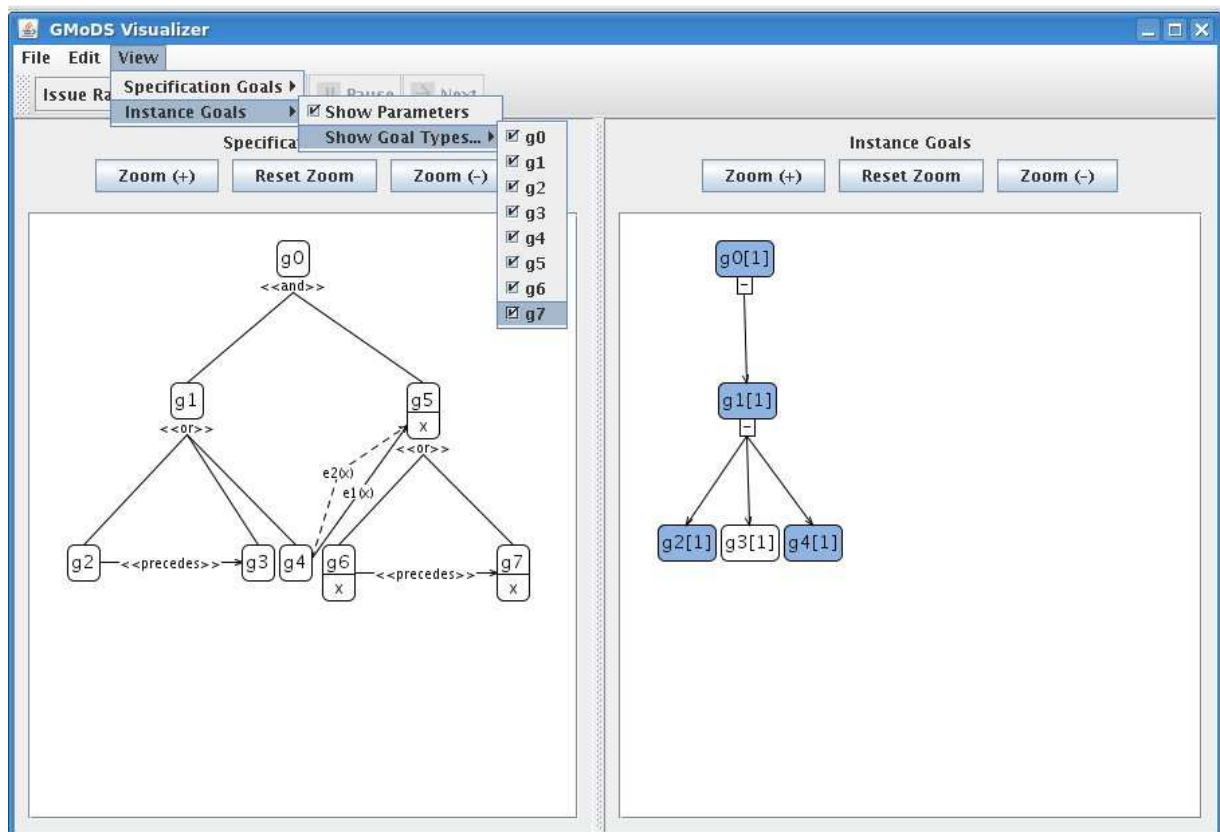


Figure 14 View | Instance Goals | Show Goal Types

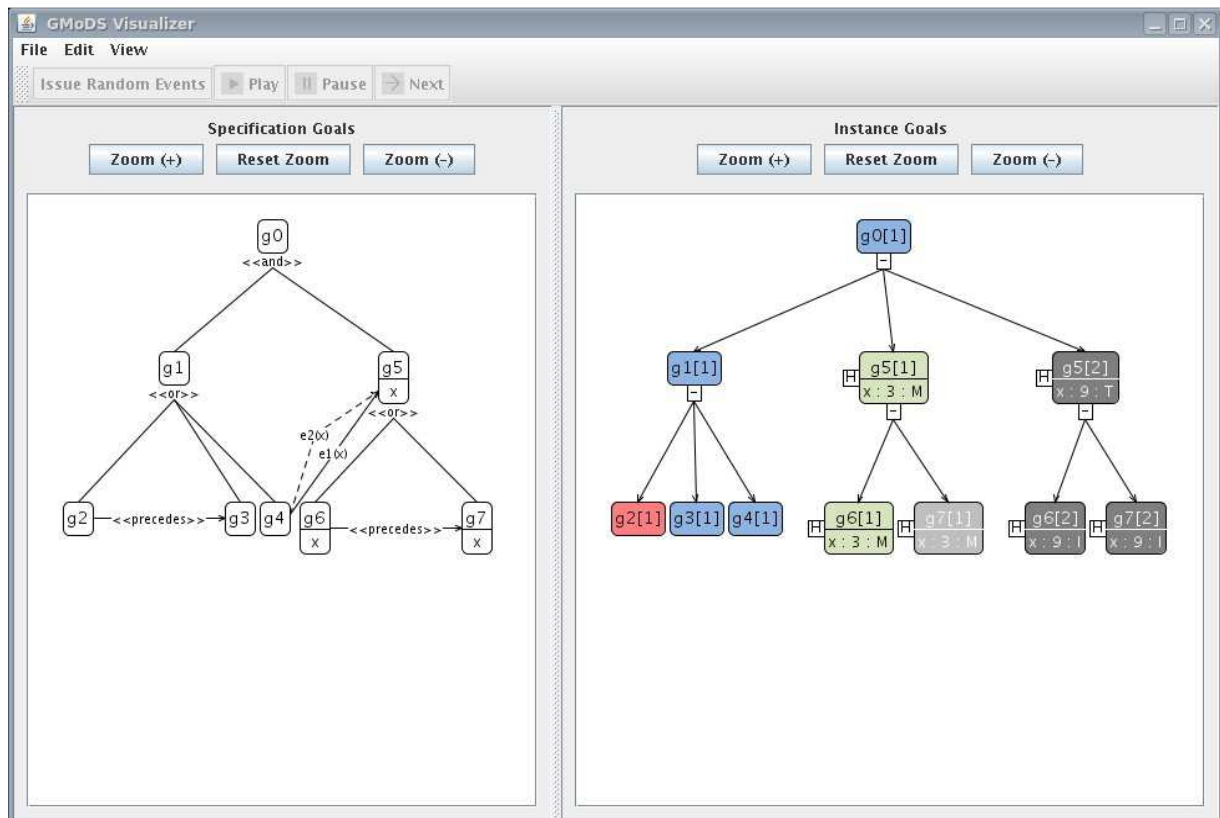


Figure 15 Valid events executed

Figure 15 above shows the same specification and instance trees after the script shown in Figure 25 below is executed. Compare Figure 16 and Figure 17 below to this figure. To collapse all child goals of a particular instance goal click the rectangle containing the “minus” sign below the instance goal. To expand the child goals, click the rectangle containing “plus” sign. To hide the parameters of a particular instance goal click the rectangle containing “H”. To show the parameters, click the rectangle containing “S”.



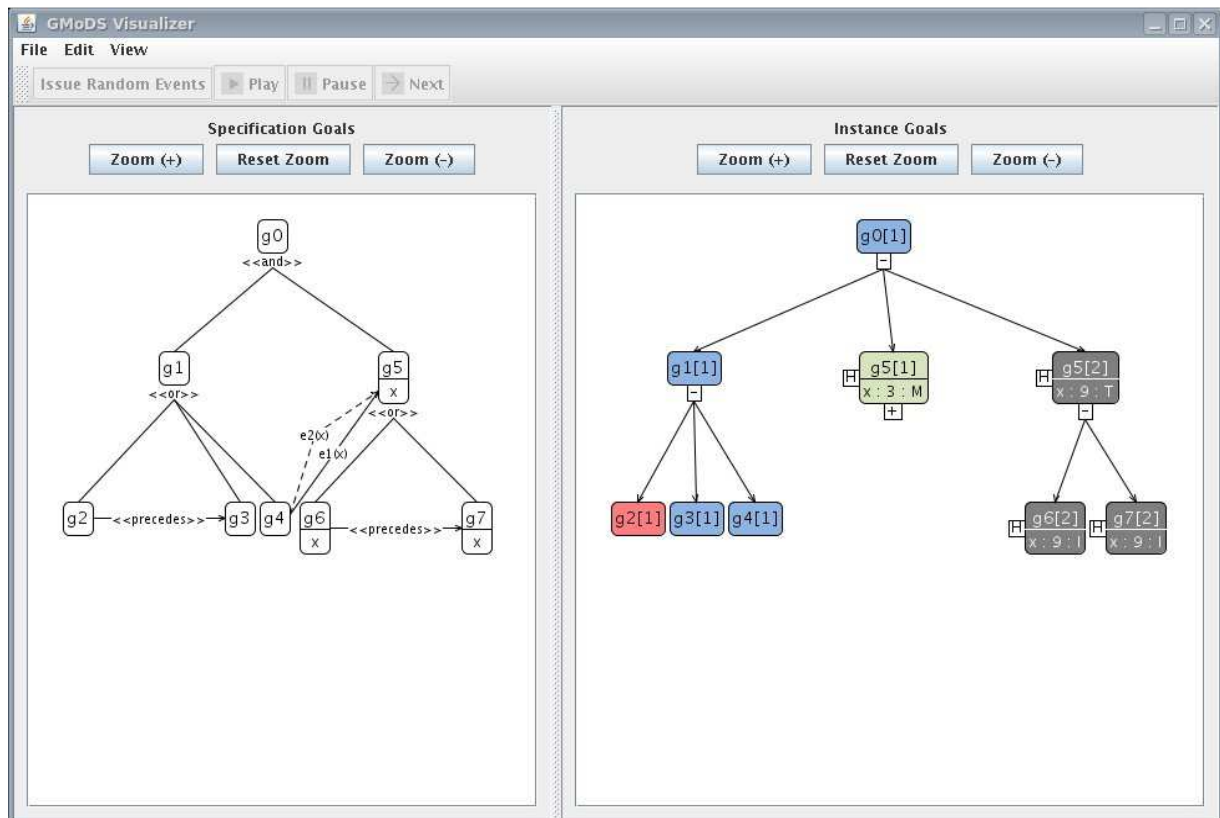


Figure 16 Collapse sub-goals

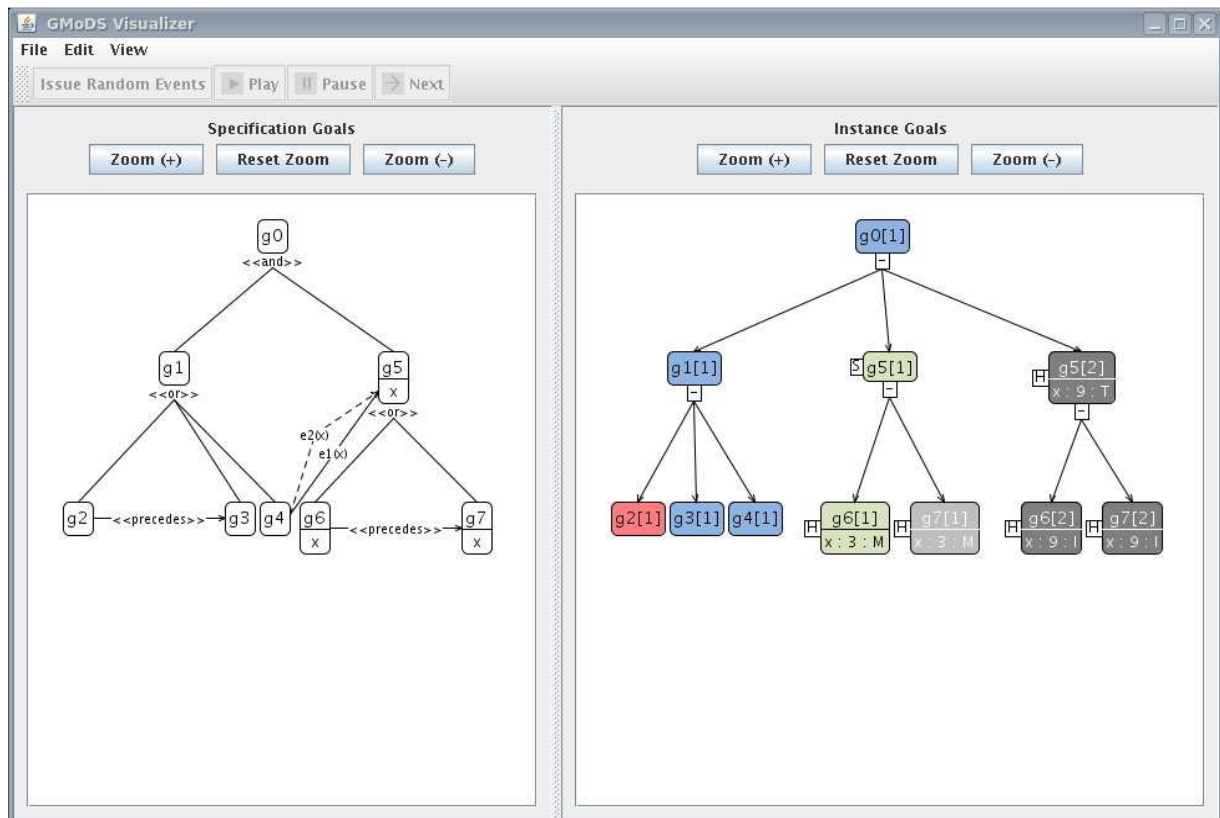


Figure 17 Hide parameters

## 4 GMoDS Test Driver

The GMoDS Test Driver can be used to test GMoDS or the GMoDS Visualizer using scripts or random events in manual or automatic mode.

### 4.1 Installation

#### 4.1.1 Install the GMoDS Visualizer

Follow the steps in section 3.1 above, skipping section 3.1.3. Section 3.1.3 is skipped because the GMoDS Test Driver project already depends on the GMoDS Visualizer project.

#### 4.1.2 Install the GMoDS Test Driver

Follow the steps in section 3.1 above except the CVS repository location /cvsroot/gmodsvi will already be defined and instead of “GMoDSVisualizer” check out “GMoDSTestDriver”.

### 4.2 Usage

The project GMoDSTestDriver includes the folders “eventscripts” and “goalmodels” for sample event scripts and GMoDS goal models. The GMoDS Test Driver is launched using the class edu.ksu.cis.macr.goal.model.testdriver.launcher.Launcher. To run the GMoDS Test Driver, first define a run configuration as shown in Figure 18, Figure 19, and Figure 20 below. Then click the “Run” button.

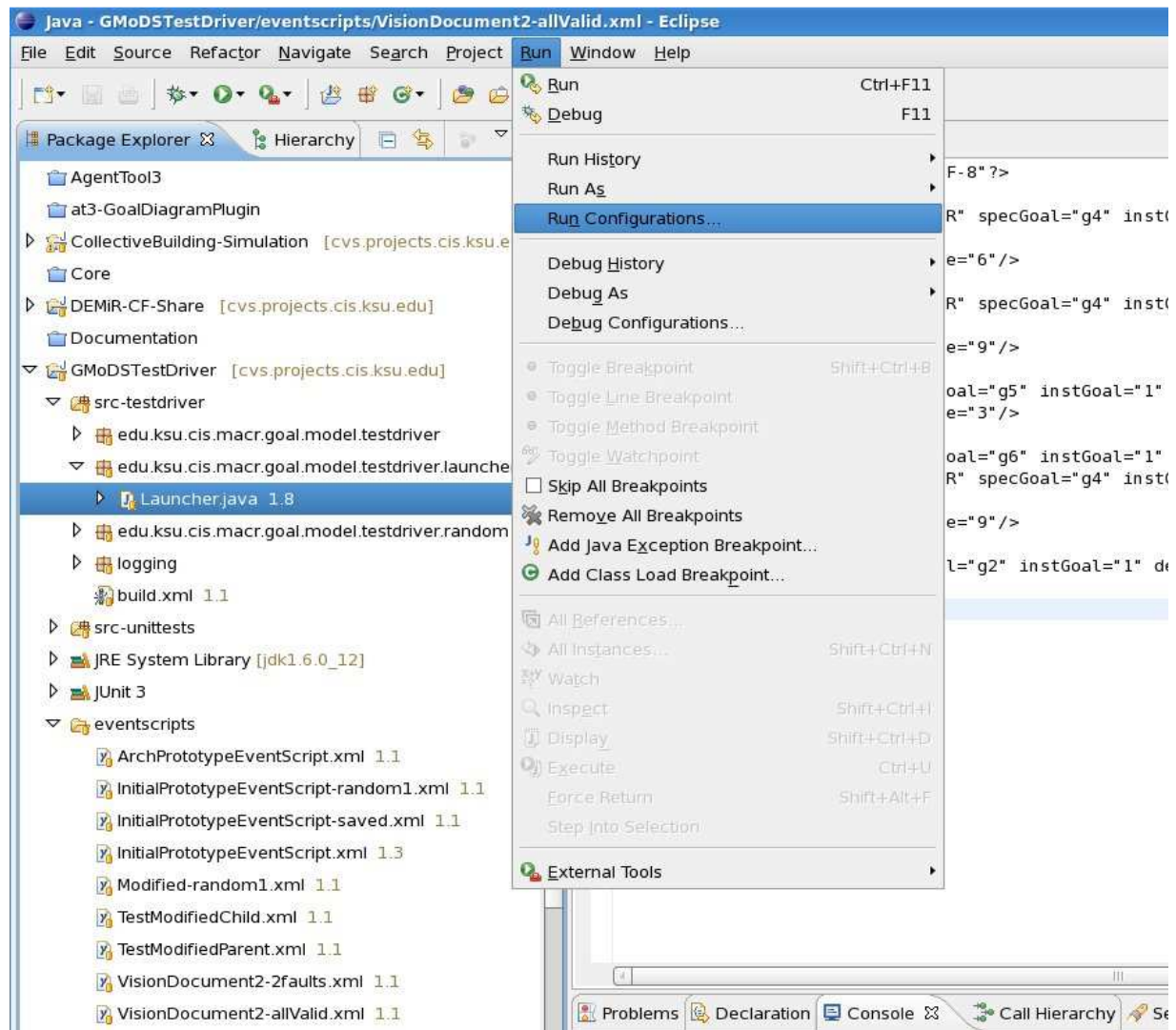


Figure 18 Run Configurations

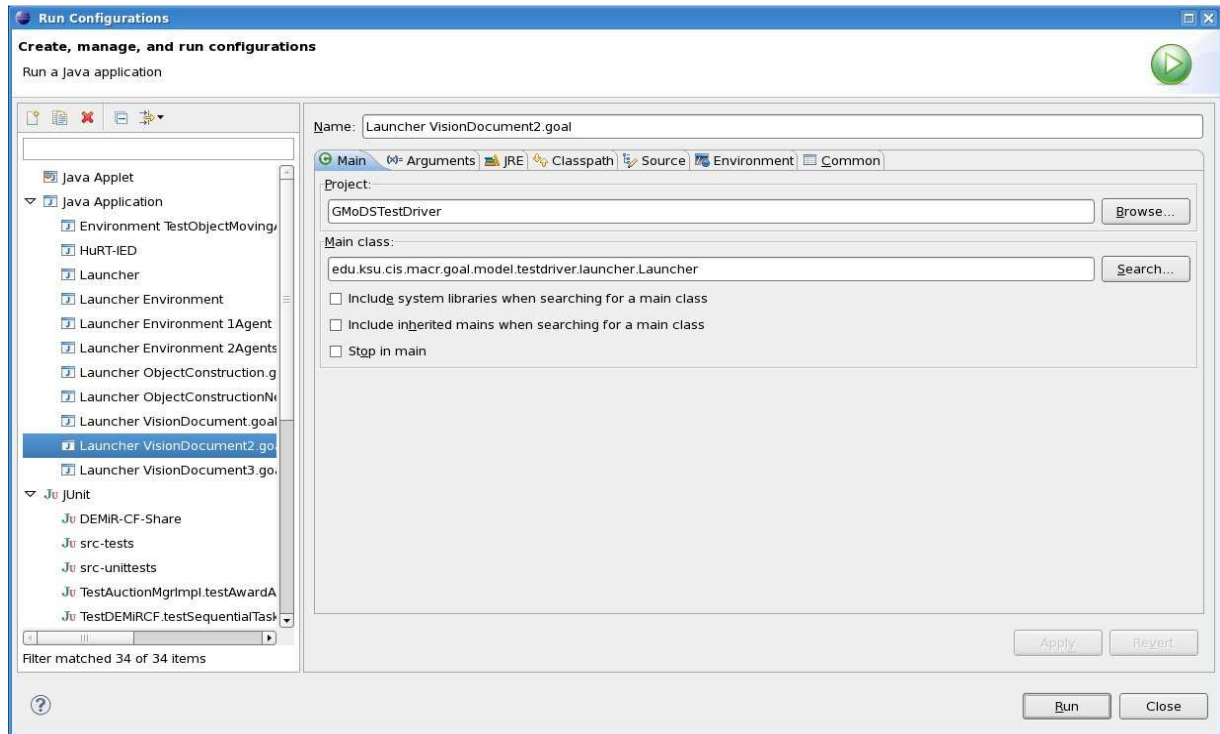


Figure 19 GMoDS Test Driver Main

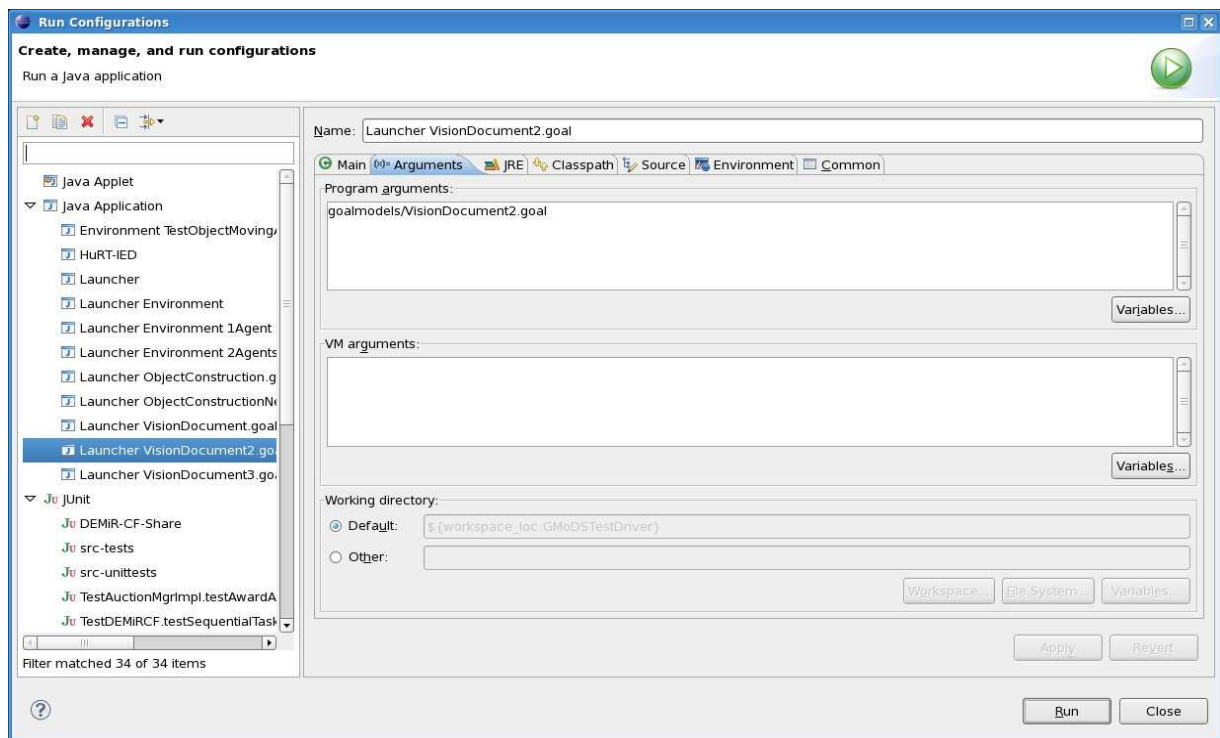


Figure 20 GMoDS Test Driver Arguments

### 4.3 Common Commands

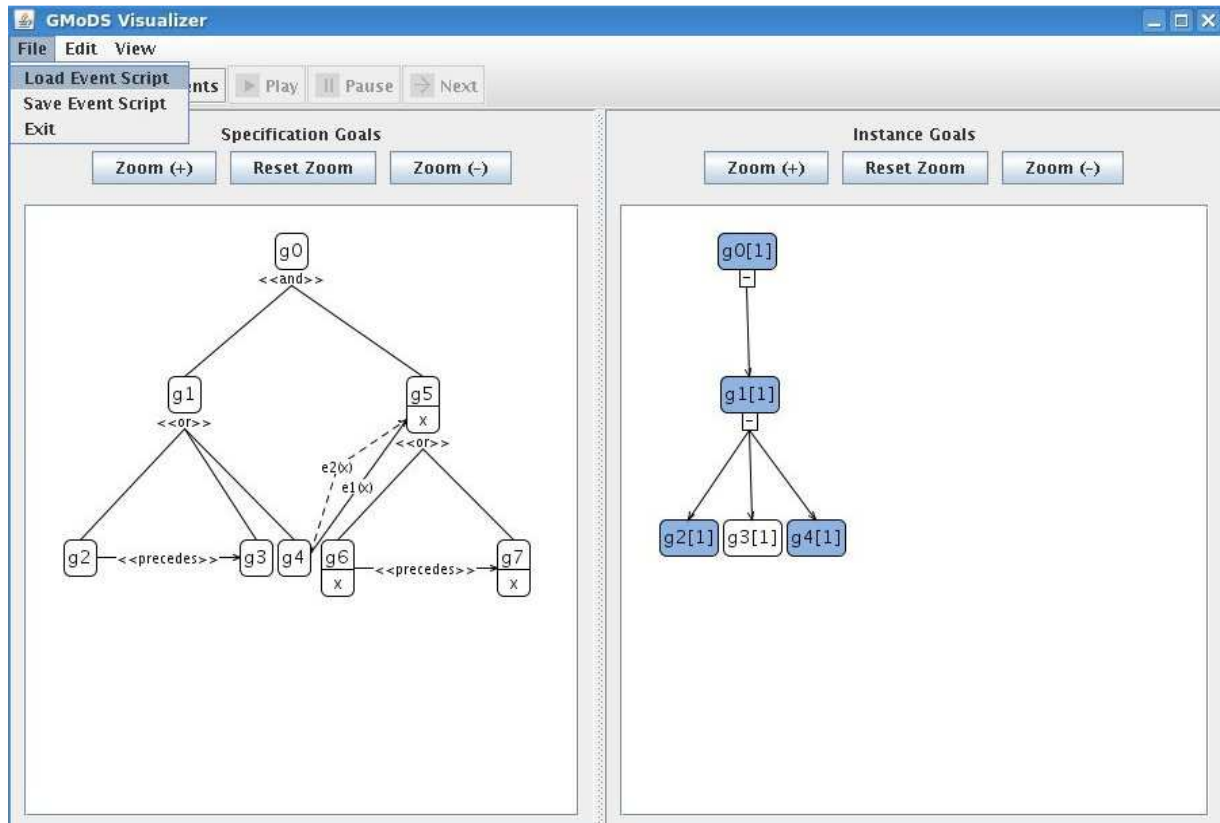


Figure 21 Load Event Script

Figure 21 above shows that the GMoDS Test Driver allows one to load or save an event script. Figure 22 below shows that one can issue random events (replacing any previously loaded event script), play an event script in automated mode, pause an event script and enter manual mode, or issue the next event manually. Figure 23 below shows that one can “Edit | Preferences | Random Events” to control the behavior of the GMoDS Test Driver when issuing random events.

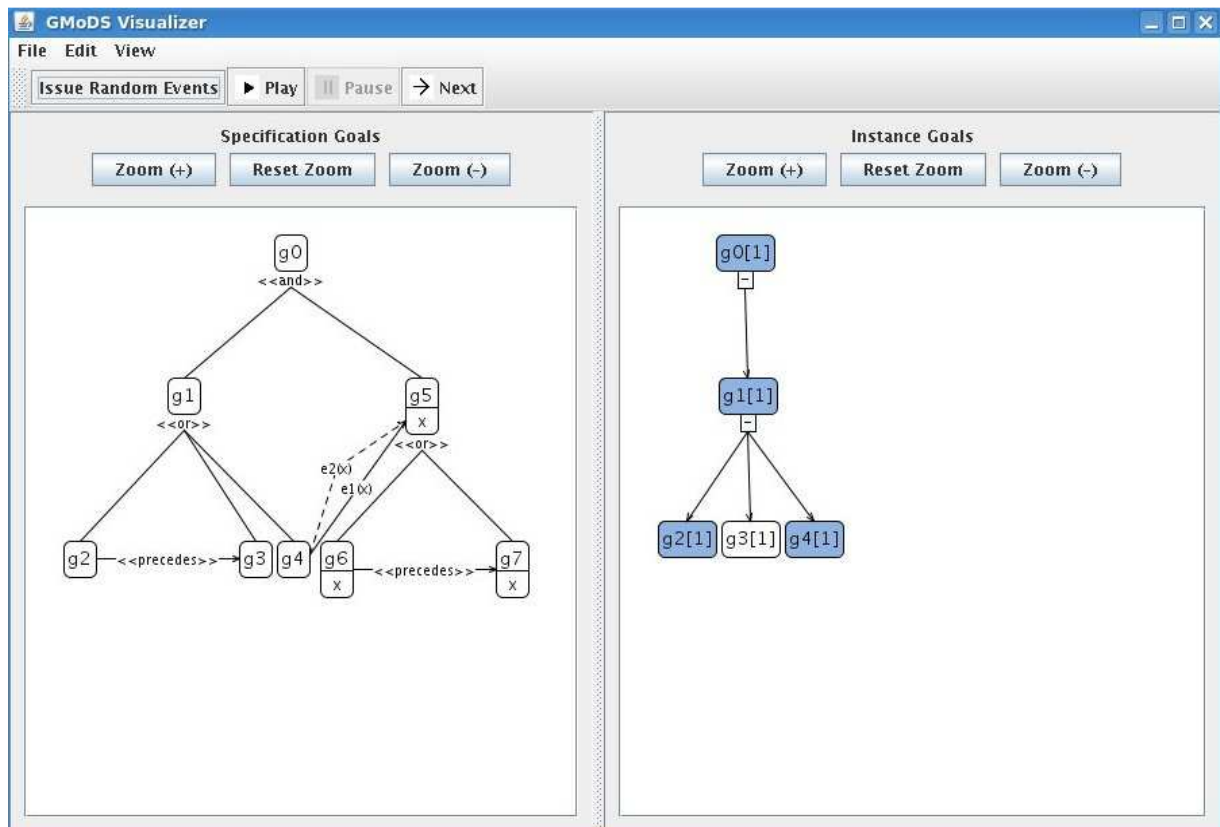


Figure 22 Test Driver controls

The screenshot shows the Edit | Preferences | Random Events dialog box. The dialog has a tabbed interface with 'States' and 'Random Events' tabs. The 'Random Events' tab is selected, showing the following settings:

- Number of Events: 25
- Min Delay Time (ms): 3000
- Max Delay Time (ms): 4000
- Min String Length: 1
- Max String Length: 10

At the bottom of the dialog are buttons for OK, Apply, and Cancel.

Figure 23 Edit | Preferences | Random Events

## 4.4 Event Script Format

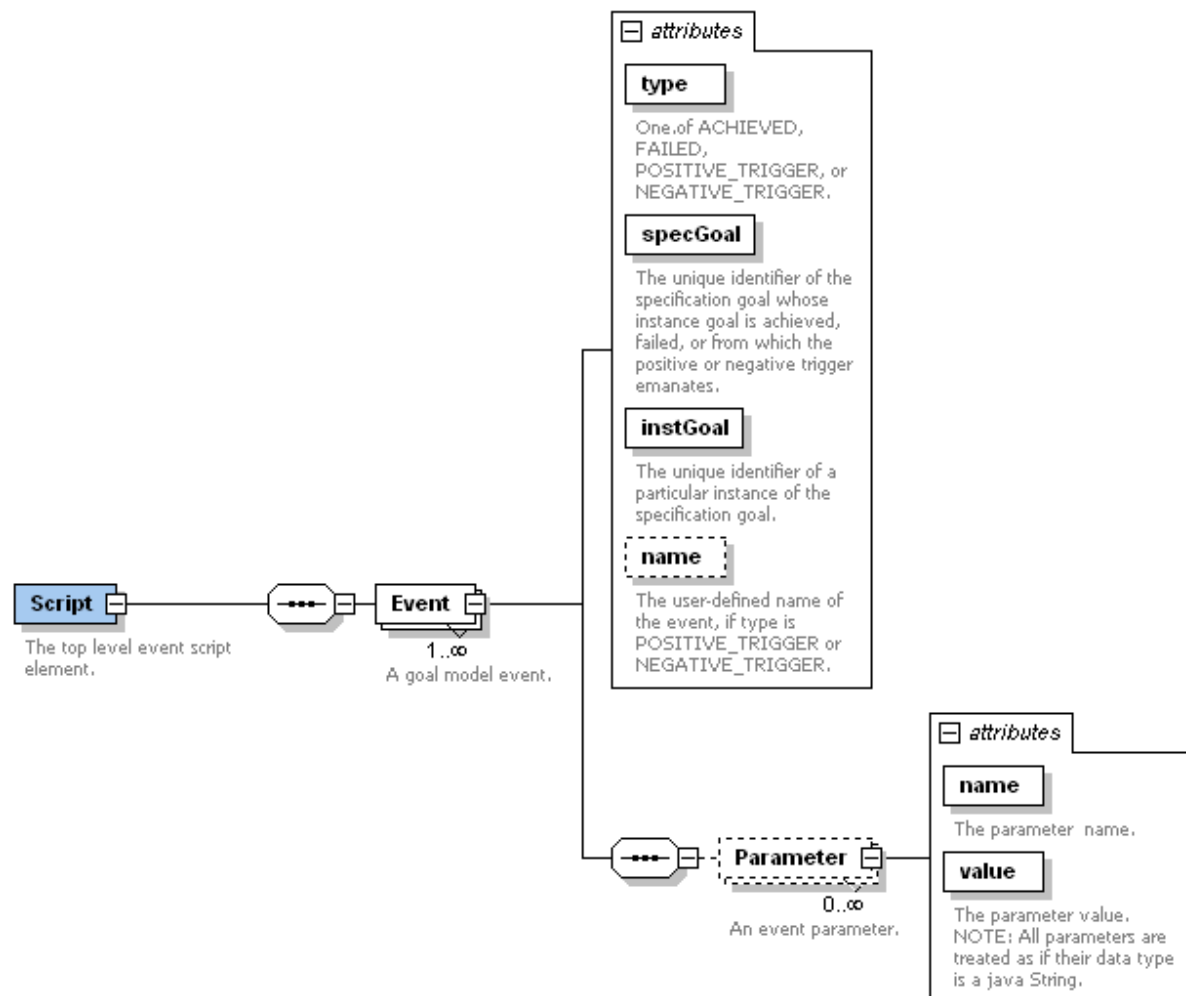



Figure 24 GMoDS Test Driver Event Script XML Schema

Figure 24 above shows the XML Schema defining legal XML format for event script files. The GMoDS Test Driver does not validate event scripts using this schema so it is the user's responsibility to follow these formats. The XML parser will notify the user of badly formed XML by throwing an exception halting the load of the script (and requiring a restart of the GMoDS Test Driver).

Figure 25 below shows as a sample legal event script for the goal model "VisionDocument2.goal" using all event types.





```

1<?xml version="1.0" encoding="UTF-8" ?>
2<Script>
3  <Event type="POSITIVE_TRIGGER" specGoal="g4" instGoal="1" name="e1"
4    delay="3000">
5    <Parameter name="x" value="6" />
6  </Event>
7  <Event type="POSITIVE_TRIGGER" specGoal="g4" instGoal="1" name="e1"
8    delay="3000">
9    <Parameter name="x" value="9" />
10 </Event>
11 <Event type="MODIFIED" specGoal="g5" instGoal="1" delay="2100">
12   <Parameter name="x" value="3" />
13 </Event>
14 <Event type="ACHIEVED" specGoal="g6" instGoal="1" delay="3000" />
15 <Event type="NEGATIVE_TRIGGER" specGoal="g4" instGoal="1" name="e2"
16   delay="3000">
17   <Parameter name="x" value="9" />
18 </Event>
19 <Event type="FAILED" specGoal="g2" instGoal="1" delay="3000" />
20</Script>
21

```

Figure 25 Sample Event Script

## 4.5 Log Messages

Table 1 Log Messages

Message	Meaning
addEvent,addEvent	A legal event was added to the current event script.
addEvent,Exception	An invalid event with respect to the SpecificationTree in GMoDS was not added to the current event script.
issueToGMoDS,event	A legal event was issued to GMoDS.
issueToGMoDS,modifyInstanceGoal	A legal MODIFIED event was issued to GMoDS.
next,Exception	The next command found an invalid event with respect to the current InstanceTree in GMoDS.

## 4.6 Error Messages

Table 2 Error Messages

Message	Meaning
"%s - improper specification event id %s"	The GMoDS Test Driver did not assign the correct internal specification event id to an ACHIEVED or FAILED event.
"%s - specification goal %s not defined"	The event's "specGoal" attribute does not reference a legal specification goal.
"%s - specification goal %s is not a leaf goal"	The event's "specGoal" attribute does not reference a leaf goal for an ACHIEVED or FAILED event type.



Message	Meaning
"%s - specification event %s not defined for specification goal %s"	The specification event referenced by the "name" attribute is not defined for the specification goal referenced by the "specGoal" attribute.
"%s - parameter names %s do not match those specified %s for specification goal %s"	The MODIFIED goal events' Parameter "name" attribute does not match any specified for the specification goal referenced by "specGoal".
"%s - parameter names %s do not match those specified %s for specification event %s for specification goal %s"	A POSITIVE_TRIGGER or NEGATIVE_TRIGGER goal events' Parameter "name" attribute does not match any specified for that trigger in the specification tree for "specGoal".
"%s - specification event %s not a negative trigger defined for specification goal %s"	A NEGATIVE_TRIGGER goal event's "name" attribute does not refer to a negative trigger in the specification tree for "specGoal".
"%s - specification event %s not a positive trigger defined for specification goal %s"	A POSITIVE_TRIGGER goal event's "name" attribute does not refer to a positive trigger in the specification tree for "specGoal".
"%s - instance goal %s not defined"	The instance goal referenced by "specGoal" and "instGoal" does not exist in the instance tree (yet).
"%s - instance goal %s not active"	The instance goal referenced by "specGoal" and "instGoal" is not an active goal for an event "type" other than MODIFIED.
"%s - instance goal %s's negative trigger parameter values %s do not match any instance goal."	No instance goal exists in the instance tree that is negatively triggered by the negative trigger referenced by "name" on "specGoal" with instance parameter names that match the Parameter "name" attributes in the goal event.
"%s - unspecified triggering instance goal: %s"	The instance goal referenced by the goal event was not found during issueToGMoDS.