Vision Document

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 **Motivation** .................................................................................................................. 3
1.2 **GMoDS** ..................................................................................................................... 3
1.3 **Terms and Definitions** ............................................................................................... 4
1.4 **References** ................................................................................................................ 5

2. **Project Overview** ......................................................................................................... 5
2.1 **Project Goal** ............................................................................................................... 5
2.2 **System Context** ......................................................................................................... 6

3. **Project Requirements** ................................................................................................. 7
3.1 **GMoDS Test Driver** ................................................................................................. 7
3.1.1 **Use Case GTD-1 Issue Scripted Goal Events** ..................................................... 7
3.1.2 **Use Case GTD-2 Issue Random Goal Events** ..................................................... 12
3.1.3 **Use Case GTD-3 Save Goal Event Script** ......................................................... 14
3.2 **GMoDS Visualizer** .................................................................................................. 15
3.2.1 **Use Case GV-1 Display Specification Tree** ....................................................... 17
3.2.2 **Use Case GV-2 Display Instance Tree** .............................................................. 19
3.2.3 **Use Case GV-3 Update Instance Tree** ............................................................... 22
3.2.4 **Use Case GV-4 Change Specification Tree View** ............................................. 22
3.2.5 **Use Case GV-5 Change Instance Tree View** ................................................... 23
3.2.6 **Use Case GV-6 Change Instance Goal View** .................................................... 24
3.2.7 **Use Case GV-7 Change Instance Sub-tree View** .............................................. 25

4. **Assumptions** ................................................................................................................. 26

5. **Constraints** .................................................................................................................. 26

6. **Environment** ................................................................................................................. 26
1 Introduction

This paper provides the background, motivation, and specific system requirements for a graphical visualization tool employable by software incorporating the Goal Model for Dynamic Systems (GMoDS) [1] component. In addition, the paper specifies requirements for a test driver for GMoDS that can substitute for simulation components as a client of GMoDS. Finally, the paper concludes with assumptions, constraints, and the proposed development environment.

1.1 Motivation

Several simulations of agent-oriented systems using the GMoDS system exist but users of these simulations have limited access to the state of GMoDS at run time. Developers of these simulations would find GMoDS run time information invaluable as a debugging tool, but should not be required to use this tool. The GMoDS Visualizer will provide a graphical representation of GMoDS specification goals and run time instance goals with loose coupling to GMoDS allowing for optional use.

Running an agent simulation is more complex and computationally expensive than is necessary to test the Visualizer. The GMoDS Test Driver will test the GMoDS Visualizer by stimulating GMoDS to in turn stimulate the Visualizer, thus allowing an alternative to simulation clients as test mechanisms.

1.2 GMoDS

The GMoDS system represents system requirements as goals and their relationships using an a priori (i.e., prior to run time) specification tree. See Figure 1 [1] below. Higher level goals can be decomposed into lower level goals using a parent/child relation, forming the tree. Parent goals are related to child goals using either an AND or OR connective. Child goals connected via AND must all be achieved to achieve the parent goal. Child goals connected via OR require only one child to be achieved for the parent goal to be achieved. The specification tree goals can be parameterized and are a template for goals created at run time; such run time goals are called instance goals. Relationships between specification goals represent causal events ("triggers" that create instance goals), negating events ("negative triggers" that cancel instance goals), and ordering relations ("precedes" relations that force certain goals to be achieved before others).
The GMoDS instance goal tree represents run time instances of the specification goal templates, with any template parameters instantiated with values. See Figure 2 [1] below. Instance goals are created in response to events that occur during execution of tasks that fulfill goals and as a result of the relationships specified in the goal specification tree. A special “init event” bootstraps the system, creating the initial instance goals that have no other trigger specified. An instance goal tree represents parent/child relationships and can be color coded to represent the status of a goal (triggered, active, achieved, failed, removed, or obviated) (see 1.3 below, terms and definitions, for explanation of these statuses).

1.3 Terms and Definitions

- Instance goal state – the state of the instance goal within GMoDS (one of triggered, active, achieved, failed, removed, or obviated).
- Triggered – a goal is triggered by the “init event” if it has no other trigger. Otherwise, the event associated with a “triggers” relation must occur while a task associated with the
goal at the source of the triggers relation is being executed for the goal to become triggered.

- Active – a goal becomes active if it is triggered and no precedes relation exists that points to the triggered goal (or its ancestors) from an unachieved goal.
- Achieved – a leaf goal is achieved when the agent executing the goal notifies GMoDS of its achievement. A parent of child goals joined by “AND” is achieved when all of its child goals are achieved. A parent of child goals joined by “OR” is achieved if any of its child goals are achieved.
- Failed – a leaf goal enters the failed status if the agent executing it notifies GMoDS of failure to fulfill it.
- Removed – a goal is removed from the instance tree as if it never existed if the event associated with a negative trigger occurs during the execution of the goal that is the source of the negative trigger and that trigger points to the specification goal that is the template for the removed instance goal.
- Obviated – a goal is obviated (made unnecessary to successful completion of its parent goal) if a sibling goal is achieved when those siblings are connected by OR to their parent goal.
- Parameter value origin – the means by which the parameter value was established in GMoDS (one of inherited, trigger, or modification). An inherited value comes from its parent goal. A value with origin trigger comes from the triggering event. The origin “modification” indicates the parameter value was modified.

1.4 References


2 Project Overview

2.1 Project Goal

The goal of this project is to provide an optional GMoDS run time information visualizer that can be tested by multiple means. An additional project goal is to provide a GMoDS test driver component that can test the visualizer by directly stimulating GMoDS substituting for a simulation application component.
2.2 System Context

Figure 3 above shows the system context for the components developed in this project. The project goal is to develop the GMoDS Test Driver and GMoDS Visualizer components shown shaded in light blue in this figure.

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. GMoDS already implements the client portion of this design pattern. 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 will 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.
3 Project Requirements

3.1 GMoDS Test Driver

Figure 4 above shows use cases for the GMoDS Test Driver. When invoking the GMoDS Test Driver, the user specifies the goal diagram that will be passed to GMoDS to build the specification goal tree. The GMoDS Test Driver instantiates the GMoDS component, populates the specification tree, and initializes the instance goal tree. The GMoDS Test Driver then constructs the GMoDS Visualizer passing in a reference to the GMoDS component and to itself. The reference to the GMoDS Test Driver causes the GMoDS Visualizer to add UI components to control the GMoDS Test Driver.

3.1.1 Use Case GTD-1 Issue Scripted Goal Events

3.1.1.1 Description
The user selects “Load Event Script”. The GMoDS Test Driver prompts the user to provide a goal event script that defines the goal events that will be issued to GMoDS and the delay time between these events. It is the user’s responsibility to assure that the goal events are consistent with the goal diagram but the GMoDS Test Driver will check the script for correct event and parameter names. Upon initialization, the GMoDS Test Driver enters manual mode and waits for user interaction. If the user clicks “Play”, the GMoDS Test Driver enters automatic mode and executes the goal event script pausing for the specified delay time between events. If the user clicks “Pause” in automatic mode, the GMoDS Test Driver event execution pauses and the GMoDS Test Driver enters manual mode. If the user clicks “Next” in manual mode, the next event is executed. The goal events are issued to the GMoDS component which alters the instance tree based on the event and specification tree definition and informs the GMoDS Visualizer.
3.1.1.2 Associated Functional Requirements

3.1.1.2.1 SR.GTD-1.1 (Non-critical Requirement)
The GMoDS Test Driver shall prompt the user for a goal event script if the user selects “Load Event Script” and if such a script is provided, the GMoDS Test Driver shall enter scripted event mode (“Use Case GTD-1 Issue Scripted Goal Events”).

3.1.1.2.2 SR.GTD-1.2 (Critical Requirement)
The GMoDS Test Driver shall parse the goal event script to generate goal events, their parameters, and the time delay relative to the previous goal event.

3.1.1.2.2.1 SR.GTD-1.2.1 (Non-critical Requirement)
The GMoDS Test Driver shall log errors and drop the corresponding goal event from the script if a goal event or parameter name does not match a legal name defined in the goal diagram. In addition, the GMoDS Test Driver shall visually inform the user of these errors.

3.1.1.2.2.2 SR.GTD-1.2.2 (Critical Requirement)
The GMoDS Test Driver shall support a scripted events language with the following event types: ACHIEVED and FAILED events for each active instance goal, and positive and negative trigger events defined by the specification goal corresponding to any active instance goal.

3.1.1.2.2.3 SR.GTD-1.2.3 (Critical Requirement)
The GMoDS Test Driver Event Script Language (GTD-ESL) shall include the following XML elements and attributes as shown in Figure 5 below and defined in the following requirements.
Figure 5 GMoDS Test Driver Event Script Language

3.1.1.2.2.3.1 SR.GTD-1.2.3.1 (Critical Requirement)

The GTD-ESL shall have a top-level “Script” element containing one or more “Event” element children.

3.1.1.2.2.3.2 SR.GTD-1.2.3.2 (Critical Requirement)

The GTD-ESL “Event” element shall represent a single goal model event and have the following attributes: “type” whose possible values are ACHIEVED, FAILED, POSITIVE_TRIGGER, or NEGATIVE_TRIGGER, “specGoal” whose value is the unique identifier of the specification goal whose instance goal is achieved, failed, or from which the positive or negative trigger
emanates, “instGoal” whose value is the unique integer identifier of a particular instance of the specification goal, and “name” whose value is the user-defined event name if the “type” is POSITIVE_TRIGGER or NEGATIVE_TRIGGER.

3.1.1.2.3.3 SR.GTD-1.2.3.3 (Critical Requirement)
The GTD-ESL “Event” element attributes “type”, “specGoal”, and “instGoal” must be provided.

3.1.1.2.3.4 SR.GTD-1.2.3.4 (Critical Requirement)
The GTD-ESL “Event” element attribute “name” must be provided if “type” has value POSITIVE_TRIGGER or NEGATIVE_TRIGGER.

3.1.1.2.3.5 SR.GTD-1.2.3.5 (Critical Requirement)
The GTD-ESL “Event” element shall contain 0 or more “Parameter” element children.

3.1.1.2.3.6 SR.GTD-1.2.3.6 (Critical Requirement)
The GTD-ESL “Parameter” element shall represent the parameters of a positive or negative trigger and have the following attributes: “name” whose value is the parameter name and “value” whose value is the parameter value.

3.1.1.2.3.7 SR.GTD-1.2.3.7 (Critical Requirement)
The GTD-ESL “Parameter” element “value” attribute shall be treated as if the parameter’s data type is a Java String.

3.1.1.2.3 SR.GTD-1.3 (Critical Requirement)
The GMoDS Test Driver shall cause GMoDS to populate its specification goal tree.

3.1.1.2.4 SR.GTD-1.4 (Critical Requirement)
The GMoDS Test Driver shall cause GMoDS to initialize its instance goal tree.

3.1.1.2.5 SR.GTD-1.5 (Critical Requirement)
The GMoDS Test Driver shall issue each goal event defined in the event script to GMoDS after the specified delay time (milliseconds) relative to the previously issued goal event.
3.1.1.2.6 SR.GTD-1.6 (Critical Requirement)
Upon initialization of the GMoDS Test Driver in this use case, the GMoDS Test Driver shall enter manual mode and await user interaction.

3.1.1.2.6.1 SR.GTD-1.6.1 (Critical Requirement)
If the user clicks “Play” in manual mode, the GMoDS Test Driver enters automatic mode and begins to execute each event as defined in 3.1.1.2.5 above. If GMoDS Test Driver is in random event operation, GMoDS Test Driver automatically generates a new random event and executes it after the random delay time until the specified number of random events has been issued.

3.1.1.2.6.2 SR.GTD-1.6.2 (Critical Requirement)
If the user clicks “Next” in manual mode, the GMoDS Test Driver issues the next unexecuted goal event and waits for the next user interaction. If GMoDS Test Driver is in random event operation and manual mode, the GMoDS Test Driver generates the next random event, appends it to the currently executing script, issues the event to GMoDS, and waits for the next user interaction.

3.1.1.2.6.3 SR.GTD-1.6.3 (Critical Requirement)
If the user clicks “Pause” in automatic mode, the GMoDS Test Driver enters manual mode and waits for the next user interaction.

3.1.1.2.6.4 SR.GTD-1.6.4 (Critical Requirement)
If there are no more pre-defined events remaining or the specified number of random events have been issued, the GMoDS Test Driver disables the “Play” and “Next” controls.
### 3.1.2 Use Case GTD-2 Issue Random Goal Events

Table 1 below shows GMoDS Test Driver options to configure random events.

<table>
<thead>
<tr>
<th>Option</th>
<th>Definition</th>
<th>Default</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min String Length</td>
<td>Minimum length for a String representing a parameter value.</td>
<td>1</td>
<td>While paused or in manual mode.</td>
</tr>
<tr>
<td>Max String Length</td>
<td>Maximum length for a String representing a parameter value.</td>
<td>10</td>
<td>While paused or in manual mode.</td>
</tr>
<tr>
<td>Min Delay Time</td>
<td>Minimum delay time between events.</td>
<td>100 milliseconds</td>
<td>While paused or in manual mode.</td>
</tr>
<tr>
<td>Max Delay Time</td>
<td>Maximum delay time between events.</td>
<td>5000 milliseconds</td>
<td>While paused or in manual mode.</td>
</tr>
<tr>
<td>Number of Random Events</td>
<td>The total number of random events to issue.</td>
<td>25</td>
<td>While paused or in manual mode.</td>
</tr>
</tbody>
</table>

#### 3.1.2.1 Description

The user selects “Issue Random Events”. The GMoDS Test Driver replaces the currently loaded event script with an empty script and begins random goal event generation. The GMoDS Test Driver relies on the GMoDS component to discover the possible goal events based on the events that have been issued to GMoDS. The “init” event provides the initial set of active instance goals. The GMoDS Test Driver generates a random event as specified in the random event configuration based on the current active instance goals and appends this event to the currently executing event script. The GMoDS Test Driver issues the last-generated event to GMoDS, according to “3.1.1.2.5 and 3.1.1.2.6” described above, after a random delay time, if GMoDS Test Driver is in automatic mode or immediately if in manual mode. Upon execution of that event, the “Use Case GTD-2 Issue Random Goal Events” resumes event generation until the final random event is issued. The GMoDS Test Driver keeps the currently executing goal event script which can be saved using the “Use Case GTD-3 Save Goal Event Script” described below. Each active instance goal’s specification goal defines the positive and negative trigger events that can be executed from that goal. In addition, the GMoDS Test Driver can issue an ACHIEVED or FAILED event for each active instance goal. The union of all positive and negative trigger, ACHIEVED, and FAILED events defined by all active instance goals and their corresponding specification goals defines the possible random events at any time.

#### 3.1.2.2 Associated Functional Requirements

**3.1.2.2.1 SR.GTD-1.1**

See 3.1.1.2.1 above.
3.1.2.2.1  SR.GTD.2.1.1 (Non-critical Requirement)
The GMoDS Test Driver shall treat all parameter types as if they were String. That is, the system shall make no attempt to generate a value of the type specified for the parameter in the goal diagram. Instead, the type of the java object will be String for all parameter values. The system will generate a random String for each parameter value.

3.1.2.2.1.2 SR.GTD.2.1.2 (Non-critical Requirement)
The GMoDS Test Driver may be configured with the minimum and maximum string lengths for randomly generated strings. The system shall default to a minimum string length of 1 and a maximum string length of 10.

3.1.2.2.1.3 SR.GTD.2.1.3 (Non-critical Requirement)
The GMoDS Test Driver may be configured with the minimum and maximum delay time in milliseconds between randomly issued goal events. The system shall default to a minimum delay time of 100 milliseconds and maximum delay time of 5000 milliseconds. The system shall not accept a minimum delay time of less than 1 millisecond.

3.1.2.2.1.4 SR.GTD.2.1.4 (Non-critical Requirement)
The GMoDS Test Driver may be configured with the number of random goal events to issue. The system will default to 25 random goal events to issue.

3.1.2.2.2 SR.GTD-1.3
See 3.1.1.2.3 above.

3.1.2.2.3 SR.GTD-1.4
See 3.1.1.2.4 above.
3.1.2.2.4 **SR.GTD-2.2 (Critical Requirement)**

The GMoDS Test Driver shall incrementally issue random goal events based on the current active instance goals. The union of all positive and negative trigger, ACHIEVED, and FAILED events defined by all active instance goals and their corresponding specification goals defines the possible random events at any time.

3.1.2.2.5 **SR.GTD-2.3 (Critical Requirement)**

The GMoDS Test Driver shall keep a history of randomly-generated goal events to form the current event script being executed.

3.1.3 **Use Case GTD-3 Save Goal Event Script**

3.1.3.1 **Description**

The user selects “Save Script” and is prompted for a file in which to save the current goal event script. If the user selects a file that exists, the GMoDS Test driver asks for confirmation that it should overwrite that file. If the user selects a file name that does not exist or confirms the overwrite operation, the GMoDS Test Driver saves the current goal event script to the file.

3.1.3.2 **Associated Functional Requirements**

3.1.3.2.1 **SR.GTD-3.1 (Non-critical Requirement)**

The GMoDS Test Driver shall provide a “Save Script” menu item that will cause the GMoDS Test Driver to save the currently executing goal event script to a file.

3.1.3.2.2 **SR.GTD-3.2 (Non-critical Requirement)**

The GMoDS Test Driver shall allow the user to specify the file to contain the saved script.

3.1.3.2.2.1 **SR.GTD-3.2.1 (Non-critical Requirement)**

If the user selects a file that exists, the GMoDS Test driver shall ask for confirmation that it should overwrite that file.

3.1.3.2.2.2 **SR.GTD-3.2.2 (Non-critical Requirement)**

If the user selects a file name that does not exist or confirms the overwrite operation, the GMoDS Test Driver shall save the current goal event script to the file.
3.2 GMoDS Visualizer

Figure 6 GMoDS Visualizer Use Cases

Figure 6 above shows use cases for the GMoDS Visualizer. The client component (simulated agent component or GMoDS Test Driver) instantiates the GMoDS component, populates the specification tree, and initializes the instance goal tree. The client component then constructs the GMoDS Visualizer passing in a reference to the GMoDS component. The GMoDS Visualizer registers itself with GMoDS’ EventRegistry as the ChangeManager. The GMoDS Visualizer displays the specification goal tree pulled from the GMoDS component. The GMoDS Visualizer displays the initial instance goal tree provided by GMoDS and changes to the instance goal tree pushed to it by GMoDS. The User can alter the appearance of the specification goal tree or instance goal tree as a whole, change the appearance of a specific instance goal, or collapse/expand a sub-tree of instance goals. Table 2 below shows the GMoDS Visualizer options.
Table 2 GMoDS Visualizer Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Definition</th>
<th>Default</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash Period</td>
<td>Total time a changed instance goal is flashed.</td>
<td>1 second</td>
<td>In manual mode.</td>
</tr>
<tr>
<td>Flash Cycle</td>
<td>Time between instance goal color inversions in a single flash.</td>
<td>0.1 second</td>
<td>In manual mode.</td>
</tr>
<tr>
<td>Specification Tree Show/Hide Parameters</td>
<td>Show or hide parameter names and types throughout the specification goal tree.</td>
<td>Show</td>
<td>At run-time.</td>
</tr>
<tr>
<td>Instance Tree Show/Hide Parameters</td>
<td>Show or hide parameter names, values, and origin throughout the entire instance goal tree.</td>
<td>Show</td>
<td>At run-time.</td>
</tr>
<tr>
<td>Show/Hide Instance Goals of Particular Specification Goals</td>
<td>Show or hide particular specification goals’ derived instance goals (a check list of goal types to show is presented to the user).</td>
<td>Show</td>
<td>At run-time.</td>
</tr>
<tr>
<td>Triggered Goal Background</td>
<td>A triggered goal’s background color.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Triggered Goal Foreground</td>
<td>A triggered goal’s foreground color.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Triggered Goal Flash Background</td>
<td>A triggered goal’s background color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Triggered Goal Flash Foreground</td>
<td>A triggered goal’s foreground color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Active Goal Background</td>
<td>An active goal’s background color.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Active Goal Foreground</td>
<td>An active goal’s foreground color.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Active Goal Flash Background</td>
<td>An active goal’s background color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Active Goal Flash Foreground</td>
<td>An active goal’s foreground color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Achieved Goal Background</td>
<td>An achieved goal’s background color.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Achieved Goal Foreground</td>
<td>An achieved goal’s foreground color.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Achieved Goal Flash Background</td>
<td>An achieved goal’s background color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Achieved Goal Flash Foreground</td>
<td>An achieved goal’s foreground color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Failed Goal Background</td>
<td>A failed goal’s background color.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Failed Goal Foreground</td>
<td>A failed goal’s foreground color.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Failed Goal Flash Background</td>
<td>A failed goal’s background color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Failed Goal Flash Foreground</td>
<td>A failed goal’s foreground color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Removed Goal Background</td>
<td>A removed goal’s background color.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Removed Goal Foreground</td>
<td>A removed goal’s foreground color.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Option Definition

<table>
<thead>
<tr>
<th>Option</th>
<th>Definition</th>
<th>Default</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Removed Goal Flash Background</td>
<td>A removed goal’s background color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Removed Goal Flash Foreground</td>
<td>A removed goal’s foreground color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Obviated Goal Background</td>
<td>An obviated goal’s background color.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Obviated Goal Foreground</td>
<td>A removed goal’s foreground color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Obviated Goal Flash Background</td>
<td>A removed goal’s background color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
<tr>
<td>Obviated Goal Flash Foreground</td>
<td>A removed goal’s foreground color during a flash.</td>
<td></td>
<td>At run-time.</td>
</tr>
</tbody>
</table>

### 3.2.1 Use Case GV-1 Display Specification Tree

#### 3.2.1.1 Description
The GMoDS Visualizer displays the specification tree including all goals, parent/child relations, positive trigger relations, negative trigger relations, and precedes relations with their string identifiers. The system uses the current setting for “Specification Tree Show/Hide Parameters” to decide whether goal, positive trigger, and negative trigger parameter names and types are shown. The default setting causes the system to display these parameters and types. See Figure 7 below.

![Figure 7 Use Case GV-1](image-url)
3.2.1.2 Associated Functional Requirements

3.2.1.2.1 SR.GV-1.1 (Critical Requirement)
The system shall display the specification goal tree as a graphical tree using minimum white space padding between adjacent tree elements. Each goal will have a white background and black foreground lines and characters.

3.2.1.2.2 SR.GV-1.2 (Critical Requirement)
The system shall display the string name of all specification goals, parent/child connectives («and» and «or»), trigger events, negative trigger events, and precedes relations («precedes»).

3.2.1.2.3 SR.GV-1.3 (Non-critical Requirement)
The system shall use the current “Specification Tree Show/Hide Parameters” setting to decide whether to display parameter name and type for goals or events.

3.2.1.2.4 SR.GV-1.4 (Critical Requirement)
The system shall show all parent/child, precedes, positive trigger, and negative trigger relations as lines connecting two specification goals.

3.2.1.2.5 SR.GV-1.5 (Critical Requirement)
The lines connecting the source specification goal to the destination specification goal for trigger and negative trigger relations shall have an arrow head pointing to the destination goal.

3.2.1.2.6 SR.GV-1.6 (Critical Requirement)
Parent/child, precedes, and trigger relation lines shall be solid.

3.2.1.2.7 SR.GV-1.7 (Critical Requirement)
Negative trigger relation lines shall be dashed.

3.2.1.2.8 SR.GV-1.8 (Non-critical Requirement)
The system shall separate specification goal names from parameters using a horizontal line if parameters are displayed. If parameters are not displayed no such horizontal line shall be shown.
3.2.1.2.9 SR.GV-1.9(Non-critical Requirement)
The system shall show for each specification goal each combination of parameter name and parameter type on its own single separate line, with the parameter name and type separated by a space, a semi-colon, and another space.

3.2.1.2.10 SR.GV-1.10(Non-critical Requirement)
The system shall show all event parameters on a single line between the opening parenthesis and closing parenthesis separated by a comma. The final parameter shall be followed by the closing parenthesis and no comma.

3.2.1.2.11 SR.GV-1.11(Critical Requirement)
Parent/child relation lines shall not intersect with each other.

3.2.1.2.12 SR.GV-1.12(Non-critical Requirement)
The system shall minimize the number of intersections between precedes, positive trigger, negative trigger, and parent/child relation lines.

3.2.1.2.13 SR.GV-1.13(Critical Requirement)
The system shall not allow any lines to intersect goal rectangles.

3.2.1.2.14 SR.GV-1.14(Critical Requirement)
The system shall provide scrolling and zooming of the specification goal tree view.

3.2.2 Use Case GV-2 Display Instance Tree

3.2.2.1 Description
The GMoDS Visualizer displays the initial instance goal tree in a tree-like structure including all instance goals, and parent/child relations with their string identifiers. By default, goal parameter names, values, and the values’ origin (I – inherited, T – trigger, M – modification) are shown. The system shows each instance goal state visually. See Figure 8 below.
3.2.2.2 Associated Functional Requirements

3.2.2.2.1 SR.GV-2.1(Critical Requirement)

The system shall display the instance goal tree as a graphical tree using minimum white space padding between adjacent tree elements. Each instance goal will have a background color that indicates the current state of the instance goal and black foreground lines and characters. The state colors shall be as indicated in the “Legend” in Figure 8 above by default but shall be editable at run time.

3.2.2.2.2 SR.GV-2.2(Critical Requirement)

The system shall display the instance goal name for each instance goal.

3.2.2.2.3 SR.GV-2.3(Non-critical Requirement)

The system shall display a collapse/expand toggle rectangle, if the instance goal has children, centered on the lower edge of the instance goal. An instance goal displaying its children will display the character “-“ in the collapse/expand toggle. An instance goal hiding its children will display “+“ in the collapse/expand toggle.

3.2.2.2.4 SR.GV-2.4(Non-critical Requirement)

The system shall display a show/hide parameter toggle rectangle, if the instance goal has parameters, centered on the left edge of the instance goal. An instance goal showing its
parameters will display the character “H” in the show/hide parameter toggle. An instance goal hiding its parameters will display the character “S” in the show/hide parameter toggle.

3.2.2.2.5 SR.GV-2.5(Critical Requirement)
The system shall connect each parent instance goal to one of its child instance goals using a line with an arrow pointing to the child, whose source is the collapse/expand toggle control on the parent instance goal. The arrow head shall be centered on the top edge of the child instance goal.

3.2.2.2.6 SR.GV-2.6(Non-critical Requirement)
The system shall separate instance goal names from parameters using a horizontal line if parameters are displayed. If parameters are not displayed no such horizontal line shall be shown.

3.2.2.2.7 SR.GV-2.7 (Non-critical Requirement)
The system shall show each instance goal parameter, parameter value, and parameter value origin combination on a single line separated by a space, a semi-colon, and another space. One line will be used for each combination of instance goal parameter, parameter value, and parameters value origin.

3.2.2.2.8 SR.GV-2.8(Non-critical Requirement)
The system shall abbreviate the parameter value origin values as I (inherited), T (trigger), and M (modification).

3.2.2.2.9 SR.GV-2.9(Critical Requirement)
The system shall provide scrolling and zooming of the instance goal tree view.

3.2.2.2.10 SR.GV-2.10 (Non-critical Requirement)
The system shall allow the user to specify that instance goals of particular specification goals be shown or hidden.

3.2.2.2.11 SR.GV-1.11
See 3.2.1.2.11 above.
3.2.2.12  \textit{SR.GV-1.13}

See 3.2.1.2.13 above

3.2.3  Use Case GV-3 Update Instance Tree

3.2.3.1 Description
The system receives notification from GMoDS that some aspect of the instance tree has changed. The system modifies the display to reflect the changed information and flashes the affected instance goals for a pre-determined period.

3.2.3.2 Associated Functional Requirements

3.2.3.2.1  \textit{SR.GV-3.1 (Critical Requirement)}

The system shall flash all instance goals for which it has received a change for a pre-defined period.

3.2.3.2.1.1  \textit{SR.GV-3.2 (Non-critical Requirement)}

The default flashing period shall be 1 second. The default flashing cycle shall be 0.1 second. Both the flashing period and flashing cycle shall be editable in manual mode.

3.2.3.2.1.2  \textit{SR.GV-3.3 (Critical Requirement)}

The system shall flash an instance goal by changing its background and foreground from its state color to its defined flash color and back once every flashing cycle. The user may avoid the flashing effect by making the state and flash background and foreground colors match.

3.2.4  Use Case GV-4 Change Specification Tree View

3.2.4.1 Description
The user changes the display of parameters throughout the specification goal tree by selecting hide or show parameters from a menu bar menu item. All specification goal and event parameters are hidden or shown as specified by the user. The system minimizes the display area consumed by the tree at all times. The system reduces the size of elements that include parameters when the parameters are hidden and expands the elements when parameters are shown. See Figure 9 below and compare with Figure 7 above.
3.2.4.2 Associated Functional Requirements

3.2.4.2.1 SR.GV-4.1 (Non-critical Requirement)

The system shall show or hide all specification goal and event parameters as specified by the user.

3.2.4.2.2 SR.GV-1.8

See 3.2.1.2.8 above.

3.2.4.2.3 SR.GV-1.9

See 3.2.1.2.9 above.

3.2.5 Use Case GV-5 Change Instance Tree View

3.2.5.1 Description

The user changes the display of parameters, their values, and the values’ origin throughout the instance goal tree by selecting hide or show parameters from a menu bar menu item. All instance goal parameters are hidden or shown as specified by the user. The system minimizes the display area consumed by the tree at all times. The system reduces the size of elements that include parameters when the parameters are hidden and expands the elements when parameters are shown. See Figure 10 below and compare with Figure 8 above.
3.2.5.2 Associated Functional Requirements

3.2.5.2.1 SR.GV-5.1 (Non-critical Requirement)

The system shall show or hide all instance goal parameters as specified by the user.

3.2.5.2.2 SR.GV-2.6

See 3.2.2.2.6 above.

3.2.5.2.3 SR.GV-2.7

See 3.2.2.2.7 above.

3.2.6 Use Case GV-6 Change Instance Goal View

3.2.6.1 Description

The user toggles the display of parameters, their values, and the values’ origin for a specific instance goal. The user clicks the view toggle control (a rectangle enclosing the letter S or H) on a specific instance goal to toggle the display of parameters, their values, and the values’ origin. The system minimizes the display area consumed by the tree at all times. The system reduces the size of elements that include parameters when the parameters are hidden and expands the elements when parameters are shown. See Figure 11 below where goal g5<2>’s parameters have been hidden.
3.2.6.2 Associated Functional Requirements

3.2.6.2.1 SR.GV-6.1 (Non-critical Requirement)

The system shall toggle the display of parameter names, value, and value origins for the specific instance goal whose parameter display toggle control has been clicked.

3.2.7 Use Case GV-7 Change Instance Sub-tree View

3.2.7.1 Description

The user collapses or expands a specific instance goal sub-tree by clicking on its expand/collapse toggle. When the system collapses a sub-tree it consumes less space in the display area. The system redraws the instance tree with child elements of the specified goal removed if the instance goal sub-tree is collapsed. A collapsed sub-tree draws its’ expand/collapse toggle as “+”. An expanded sub-tree draws its’ expand/collapse toggle as “-“. A minimum white space pad surrounds each visible instance goal, placing collapsed instance goal sub-trees as near as possible to their peer instance goals. See Figure 11 below, where goal g5<2>’s sub-tree is collapsed.

3.2.7.2 Associated Functional Requirements

3.2.7.2.1 SR.GV-7.1 (Non-critical Requirement)

The system shall collapse the specific instance goal sub-tree hiding all descendant goals if the user clicks on the collapse toggle control of that instance goal.
3.2.7.2.2 *SR.GV-7.2 (Non-critical Requirement)*
The system shall expand the specific instance goal sub-tree showing all descendant goals whose parent goal has not been collapsed, if the user clicks on the expand toggle control of that instance goal.

3.2.7.2.3 *SR.GV-7.3 (Non-critical Requirement)*
The system shall not change the expand/collapse state of any instance goal whose expand/collapse control was not directly clicked.

### 4 Assumptions
- There is no need to stack collapsed instance goals under instance goals from the same specification goal. The view space savings from hiding the descendents will shrink the displayed tree sufficiently to allow simultaneous viewing of the desired number of instance goals.
- Java JRE 1.6 or above will be available on platforms using the GMoDS Test Driver or Visualizer.

### 5 Constraints
- Applications using GMoDS shall not be forced to include the GMoDS Test Driver or GMoDS Visualizer components in their projects but may optionally do so.

### 6 Environment
- Application environment
  - JDK 1.6 or higher available at [http://www.sun.com/java](http://www.sun.com/java).
- Development environment
  - Eclipse IDE for Java Developers 1.2.1.20090918-0703
- GMoDS Version 2
  - The GMoDS component is the GoalModel2 module in the CVS repository cvs.projects.cis.ksu.edu at the repository path /cvsroot/gmods.