

USE/OCL Modeling of the Formal Specification

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
2	References.....	3
3	USE Modeling	3
3.1	Limitations of USE 2.6.2.....	3
3.2	Modeling EventScriptImpl::addEvent in USE.....	3
3.2.1	Modeling a POSITIVE_TRIGGER event	4
3.2.2	Modeling an ACHIEVED event	9
3.2.3	Modeling a MODIFIED event.....	10

1 Introduction

This documents the validation of the formal specification of the method `EventScriptImpl::addEvent` with USE version 2.6.2.

2 References

1. “System Architecture Document 2.0” available at <http://people.cis.ksu.edu/~mfraka/FrakaMSE.html>.
2. “USEOCLmodeling.zip” available at <http://people.cis.ksu.edu/~mfraka/FrakaMSE.html>. This file contains the USE model and command scripts used to model the pre and post conditions of the method `EventScriptImpl::addEvent`.
3. USE - A UML-based Specification Environment – Documentation, 17 March 2011, available at <http://www.db.informatik.uni-bremen.de/projects/USE/#doc>.

3 USE Modeling

An action item from MSE presentation 2 was:

- Perform USE/OCL modeling of state snapshots to validate the pre and post conditions of the `EventScriptImpl::next` method in the formal specification.

I performed this modeling using USE 2.6.2.

3.1 Limitations of USE 2.6.2

USE 2.6.2 does not support the OCL “isSent” operator (denoted ‘^’) necessary for the most important post conditions of the `EventScriptImpl::next` method. A MODIFIED event type causes the next method to send the message “modifyInstanceGoal” to GMoDS, and all other event types cause the next method to send the message “event”. In addition, USE 2.6.2 does not support the “init” constraint on a class attribute. Finally, I was unable to get USE 2.6.2 to allow more than 1 local variable to be defined in a “let” expression.

As a result of these limitations, I requested and was granted permission to model the `EventScriptImpl::addEvent` method.

3.2 Modeling `EventScriptImpl::addEvent` in USE

Table 1 below lists the scripts contained in [2] that I used to model the formal specification of the method `EventScriptImpl::addEvent`.

Table 1 USE scripts modeling `EventScriptImpl::addEvent`

Script	Comment	Figure
GTD.use	OSE class, association, and constraint model	See [1] section 9 p. 28.
gtd-valid-pt.cmd	Snapshot of pre state adding a valid #POSITIVE_TRIGGER	Figure 1

Script	Comment	Figure
gtd-valid-post.cmd	Script to invoke pre/post conditions (valid post conditions)	Figure 2
gtd-invalid-post.cmd	Script that invokes pre/post conditions (invalid post conditions)	Figure 3
gtd-invalid-specevt.cmd	Snapshot of pre state adding an invalid #POSITIVE_TRIGGER due to an invalid SpecificationEvent ID.	Figure 5
gtd-invalid-specgoal.cmd	Snapshot of pre state adding an invalid #POSTIVE_TRIGGER due to an invalid ParamterizedSpecificationGoal ID.	Figure 6
gtd-invalid-pt-paramnames.cmd	Snapshot of pre state adding an invalid #POSITIVE_TRIGGER due to an invalid parameter name.	Figure 7
gtd-invalid-achieved.cmd	Snapshot of pre state adding an invalid #ACHIEVED event due to referencing a non-leaf goal.	Figure 8
gtd-invalid-modified.cmd	Snapshot of pre state adding an invalid #MODIFIED event due to no parameters specified.	Figure 9 Figure 10
gtd-valid-modified.cmd	Snapshot of pre state adding a valid #MODIFIED event.	Figure 11 Figure 12
gtd-invalid-modified-paramnames.cmd	Snapshot of pre state adding an invalid #MODIFIED event due to mismatch on parameter names.	Figure 13

3.2.1 Modeling a POSITIVE_TRIGGER event

Figure 1 below shows an object diagram of a pre state when adding a valid #POSITIVE_TRIGGER event.

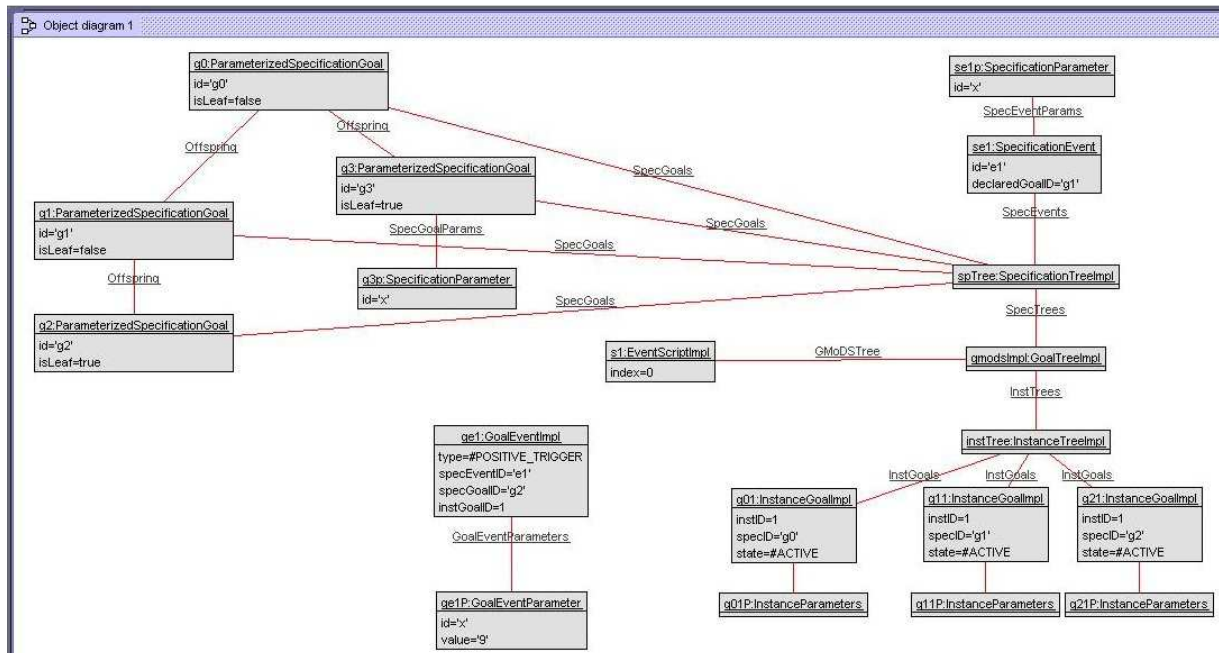


Figure 1 Valid snapshot prior to adding a POSITIVE_TRIGGER event

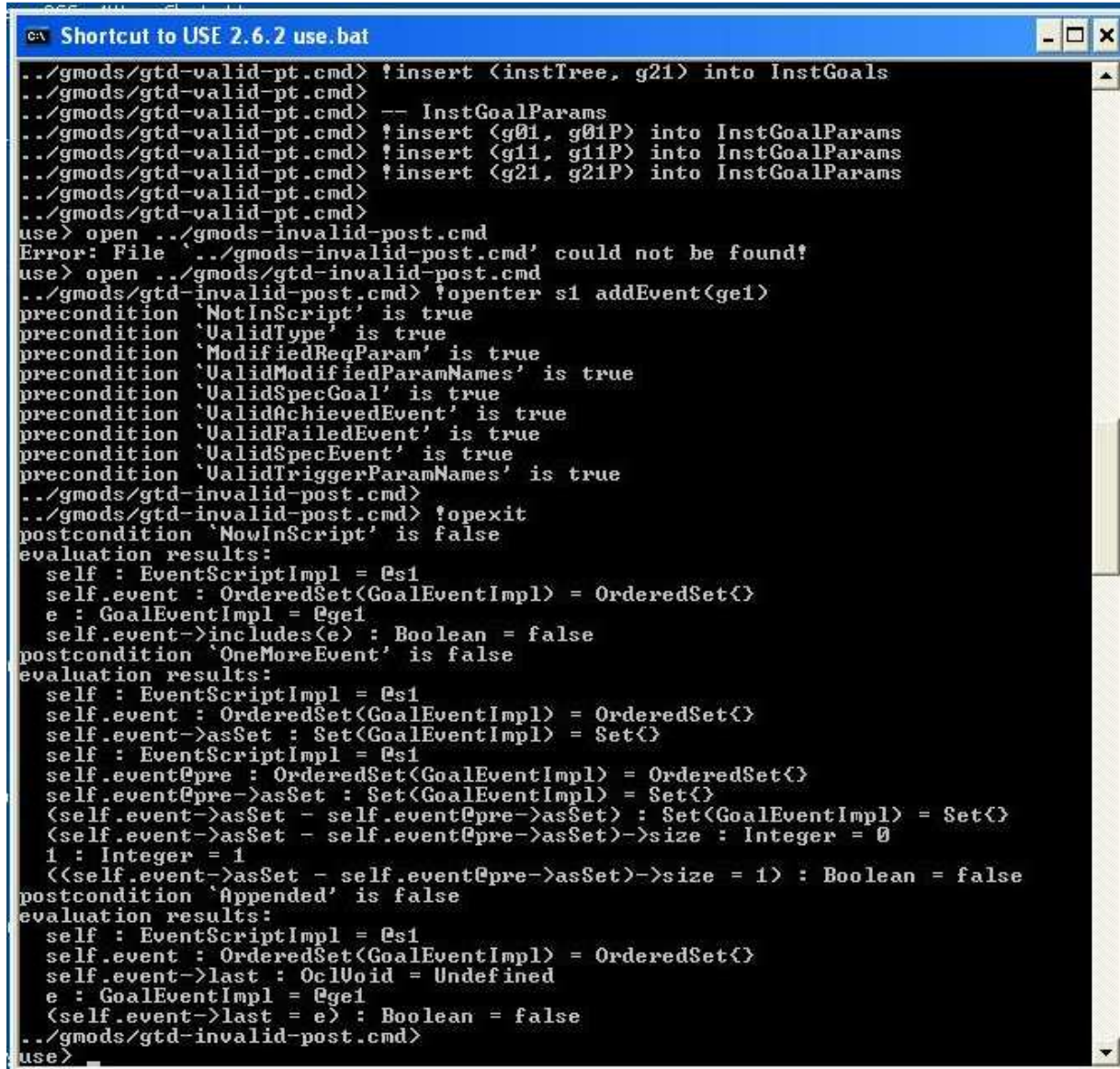
Figure 2 below shows that the pre conditions and post conditions are valid for the above snapshot when executing the script gtd-valid-post.cmd manually.

```

C:\ Shortcut to USE 2.6.2 use.bat
..../gmods/gtd-valid-pt.cmd> ?insert <g1I, g1IP> into InstGoalParams
..../gmods/gtd-valid-pt.cmd> ?insert <g2I, g2IP> into InstGoalParams
..../gmods/gtd-valid-pt.cmd>
..../gmods/gtd-valid-pt.cmd>
use> open ..../gmods/gtd-valid-post.cmd
..../gmods/gtd-valid-post.cmd> ?openter s1 addEvent<ge1>
precondition 'NotInScript' is true
precondition 'ValidType' is true
precondition 'ModifiedReqParam' is true
precondition 'ValidModifiedParamNames' is true
precondition 'ValidSpecGoal' is true
precondition 'ValidAchievedEvent' is true
precondition 'ValidFailedEvent' is true
precondition 'ValidSpecEvent' is true
precondition 'ValidTriggerParamNames' is true
..../gmods/gtd-valid-post.cmd>
..../gmods/gtd-valid-post.cmd> -- Enforce post conditions
..../gmods/gtd-valid-post.cmd> ?insert <s1,ge1> into Events
..../gmods/gtd-valid-post.cmd>
..../gmods/gtd-valid-post.cmd> ?opexit
postcondition 'NowInScript' is true
postcondition 'OneMoreEvent' is true
postcondition 'Appended' is true
..../gmods/gtd-valid-post.cmd>
use>

```

Figure 2 Valid pre/post conditions when adding a POSITIVE_TRIGGER event



```

C:\> Shortcut to USE 2.6.2 use.bat
../gmods/gtd-valid-pt.cmd> ?insert <instTree, g21> into InstGoals
../gmods/gtd-valid-pt.cmd>
../gmods/gtd-valid-pt.cmd> -- InstGoalParams
../gmods/gtd-valid-pt.cmd> ?insert <g01, g01P> into InstGoalParams
../gmods/gtd-valid-pt.cmd> ?insert <g11, g11P> into InstGoalParams
../gmods/gtd-valid-pt.cmd> ?insert <g21, g21P> into InstGoalParams
../gmods/gtd-valid-pt.cmd>
../gmods/gtd-valid-pt.cmd>
use> open ../gmods-invalid-post.cmd
Error: File '../gmods-invalid-post.cmd' could not be found!
use> open ../gmods/gtd-invalid-post.cmd
../gmods/gtd-invalid-post.cmd> ?openter s1 addEvent(gel)
precondition 'NotInScript' is true
precondition 'ValidType' is true
precondition 'ModifiedReqParam' is true
precondition 'ValidModifiedParamNames' is true
precondition 'ValidSpecGoal' is true
precondition 'ValidAchievedEvent' is true
precondition 'ValidFailedEvent' is true
precondition 'ValidSpecEvent' is true
precondition 'ValidTriggerParamNames' is true
../gmods/gtd-invalid-post.cmd>
../gmods/gtd-invalid-post.cmd> ?opexit
postcondition 'NowInScript' is false
evaluation results:
  self : EventScriptImpl = @s1
  self.event : OrderedSet<GoalEventImpl> = OrderedSet{}
  e : GoalEventImpl = @ge1
  self.event->includes(e) : Boolean = false
postcondition 'OneMoreEvent' is false
evaluation results:
  self : EventScriptImpl = @s1
  self.event : OrderedSet<GoalEventImpl> = OrderedSet{}
  self.event->asSet : Set<GoalEventImpl> = Set{}
  self : EventScriptImpl = @s1
  self.event@pre : OrderedSet<GoalEventImpl> = OrderedSet{}
  self.event@pre->asSet : Set<GoalEventImpl> = Set{}
  (self.event->asSet - self.event@pre->asSet) : Set<GoalEventImpl> = Set{}
  (self.event->asSet - self.event@pre->asSet)->size : Integer = 0
  1 : Integer = 1
  ((self.event->asSet - self.event@pre->asSet)->size = 1) : Boolean = false
postcondition 'Appended' is false
evaluation results:
  self : EventScriptImpl = @s1
  self.event : OrderedSet<GoalEventImpl> = OrderedSet{}
  self.event->last : OclVoid = Undefined
  e : GoalEventImpl = @ge1
  (self.event->last = e) : Boolean = false
../gmods/gtd-invalid-post.cmd>
use>

```

Figure 3 Invalid post conditions

Figure 3 above shows that the post conditions are violated in the above snapshot if the script gtd-invalid-post.cmd is executed.


```

C:\> Shortcut to USE 2.6.2 use.bat
../gmods/gtd-valid-pt.cmd>
use> !openter s1 addEvent(ge1)
<input>:line 0:-1 no viable alternative at input '<EOF>'
use> !openter s1 addEvent(ge1)
precondition 'NotInScript' is true
precondition 'ValidType' is true
precondition 'ModifiedReqParam' is true
precondition 'ValidModifiedParamNames' is true
precondition 'ValidSpecGoal' is true
precondition 'ValidAchievedEvent' is true
precondition 'ValidFailedEvent' is true
precondition 'ValidSpecEvent' is true
precondition 'ValidTriggerParamNames' is true
use> !insert (s1, ge1) into Events
use> !opexit
postcondition 'NowInScript' is true
postcondition 'OneMoreEvent' is true
postcondition 'Appended' is true
use> !openter s1 addEvent(ge1)
precondition 'NotInScript' is false
precondition 'ValidType' is true
precondition 'ModifiedReqParam' is true
precondition 'ValidModifiedParamNames' is true
precondition 'ValidSpecGoal' is true
precondition 'ValidAchievedEvent' is true
precondition 'ValidFailedEvent' is true
precondition 'ValidSpecEvent' is true
precondition 'ValidTriggerParamNames' is true
use>

```

Figure 4 Invalid already in script

Figure 4 above shows that executing addEvent twice for the same event violates the “NotInScript” precondition.

```

C:\> Shortcut to USE 2.6.2 use.bat
../gmods/gtd-invalid-specevt.cmd> -- InstTrees
../gmods/gtd-invalid-specevt.cmd> !insert (gmodsImpl, instTree) into InstTrees
../gmods/gtd-invalid-specevt.cmd>
../gmods/gtd-invalid-specevt.cmd> -- InstGoals
../gmods/gtd-invalid-specevt.cmd> !insert (instTree, g01) into InstGoals
../gmods/gtd-invalid-specevt.cmd> !insert (instTree, g11) into InstGoals
../gmods/gtd-invalid-specevt.cmd> !insert (instTree, g21) into InstGoals
../gmods/gtd-invalid-specevt.cmd>
../gmods/gtd-invalid-specevt.cmd> -- InstGoalParams
../gmods/gtd-invalid-specevt.cmd> !insert (g01, g01P) into InstGoalParams
../gmods/gtd-invalid-specevt.cmd> !insert (g11, g11P) into InstGoalParams
../gmods/gtd-invalid-specevt.cmd> !insert (g21, g21P) into InstGoalParams
../gmods/gtd-invalid-specevt.cmd>
../gmods/gtd-invalid-specevt.cmd>
use> !openter s1 addEvent(ge1)
precondition 'NotInScript' is true
precondition 'ValidType' is true
precondition 'ModifiedReqParam' is true
precondition 'ValidModifiedParamNames' is true
precondition 'ValidSpecGoal' is true
precondition 'ValidAchievedEvent' is true
precondition 'ValidFailedEvent' is true
precondition 'ValidSpecEvent' is false
precondition 'ValidTriggerParamNames' is false
use>

```

Figure 5 Invalid SpecificationEvent

Figure 5 above shows that the script gtd-invalid-specevt.cmd violates the “ValidSpecEvent” precondition.

```

C:\ Shortcut to USE 2.6.2 use.bat
../gmods/gtd-invalid-specgoal.cmd> -- InstTrees
../gmods/gtd-invalid-specgoal.cmd> !insert <gmodsImpl, instTree> into InstTrees
../gmods/gtd-invalid-specgoal.cmd>
../gmods/gtd-invalid-specgoal.cmd> -- InstGoals
../gmods/gtd-invalid-specgoal.cmd> !insert <instTree, g01> into InstGoals
../gmods/gtd-invalid-specgoal.cmd> !insert <instTree, g11> into InstGoals
../gmods/gtd-invalid-specgoal.cmd> !insert <instTree, g21> into InstGoals
../gmods/gtd-invalid-specgoal.cmd>
../gmods/gtd-invalid-specgoal.cmd> -- InstGoalParams
../gmods/gtd-invalid-specgoal.cmd> !insert <g01, g01P> into InstGoalParams
../gmods/gtd-invalid-specgoal.cmd> !insert <g11, g11P> into InstGoalParams
../gmods/gtd-invalid-specgoal.cmd> !insert <g21, g21P> into InstGoalParams
../gmods/gtd-invalid-specgoal.cmd>
use> !openter s1 addEvent<ge1>
precondition 'NotInScript' is true
precondition 'ValidType' is true
precondition 'ModifiedReqParam' is true
precondition 'ValidModifiedParamNames' is true
precondition 'ValidSpecGoal' is false
precondition 'ValidAchievedEvent' is true
precondition 'ValidFailedEvent' is true
precondition 'ValidSpecEvent' is false
precondition 'ValidTriggerParamNames' is true
use>

```

Figure 6 Invalid SpecificationGoal

Figure 6 above shows the script gtd-invalid-specgoal.cmd violates the “ValidSpecGoal” and “ValidSpecEvent” pre conditions. Figure 7 below shows the script gtd-invalid-pt-paramnames.cmd violates the “ValidTriggerParamNames” pre condition.

```

C:\ Shortcut to USE 2.6.2 use.bat
../gmods/gtd-invalid-pt-paramnames.cmd> !insert <instTree, g01> into InstGoals
../gmods/gtd-invalid-pt-paramnames.cmd> !insert <instTree, g11> into InstGoals
../gmods/gtd-invalid-pt-paramnames.cmd> !insert <instTree, g21> into InstGoals
../gmods/gtd-invalid-pt-paramnames.cmd> !insert <instTree, g31> into InstGoals
../gmods/gtd-invalid-pt-paramnames.cmd>
../gmods/gtd-invalid-pt-paramnames.cmd> -- InstGoalParams
../gmods/gtd-invalid-pt-paramnames.cmd> !insert <g01, g01P> into InstGoalParams
../gmods/gtd-invalid-pt-paramnames.cmd> !insert <g11, g11P> into InstGoalParams
../gmods/gtd-invalid-pt-paramnames.cmd> !insert <g21, g21P> into InstGoalParams
../gmods/gtd-invalid-pt-paramnames.cmd> !insert <g31, g31P> into InstGoalParams
../gmods/gtd-invalid-pt-paramnames.cmd>
../gmods/gtd-invalid-pt-paramnames.cmd> -- InstParams
../gmods/gtd-invalid-pt-paramnames.cmd> !insert <g31P, g31p1> into InstParams
../gmods/gtd-invalid-pt-paramnames.cmd>
use> !openter s1 addEvent<ge1>
precondition 'NotInScript' is true
precondition 'ValidType' is true
precondition 'ModifiedReqParam' is true
precondition 'ValidModifiedParamNames' is true
precondition 'ValidSpecGoal' is true
precondition 'ValidAchievedEvent' is true
precondition 'ValidFailedEvent' is true
precondition 'ValidSpecEvent' is true
precondition 'ValidTriggerParamNames' is false
use>

```

Figure 7 Invalid Parameter Names for a Positive Trigger Event

3.2.2 Modeling an ACHIEVED event

```

C:\ Shortcut to USE 2.6.2 use.bat
./gmods/gtd-invalid-achieved.cmd> -- InstTrees
./gmods/gtd-invalid-achieved.cmd> !insert (gmodsImpl, instTree) into InstTrees
./gmods/gtd-invalid-achieved.cmd>
./gmods/gtd-invalid-achieved.cmd> -- InstGoals
./gmods/gtd-invalid-achieved.cmd> !insert (instTree, g01) into InstGoals
./gmods/gtd-invalid-achieved.cmd> !insert (instTree, g11) into InstGoals
./gmods/gtd-invalid-achieved.cmd> !insert (instTree, g21) into InstGoals
./gmods/gtd-invalid-achieved.cmd>
./gmods/gtd-invalid-achieved.cmd> -- InstGoalParams
./gmods/gtd-invalid-achieved.cmd> !insert (g01, g01P) into InstGoalParams
./gmods/gtd-invalid-achieved.cmd> !insert (g11, g11P) into InstGoalParams
./gmods/gtd-invalid-achieved.cmd> !insert (g21, g21P) into InstGoalParams
./gmods/gtd-invalid-achieved.cmd>
use> !openter s1 addEvent(gel)
precondition 'NotInScript' is true
precondition 'ValidType' is true
precondition 'ModifiedReqParam' is true
precondition 'ValidModifiedParamNames' is true
precondition 'ValidSpecGoal' is true
precondition 'ValidAchievedEvent' is false
precondition 'ValidFailedEvent' is true
precondition 'ValidSpecEvent' is true
precondition 'ValidTriggerParamNames' is true
use>

```

Figure 8 Invalid ACHIEVED event

Figure 8 above shows that the script gtd-invalid-achieved.cmd violates the “ValidAchievedEvent” pre condition. A slight modification of this script would violate the “ValidFailedEvent” pre condition.

3.2.3 Modeling a MODIFIED event

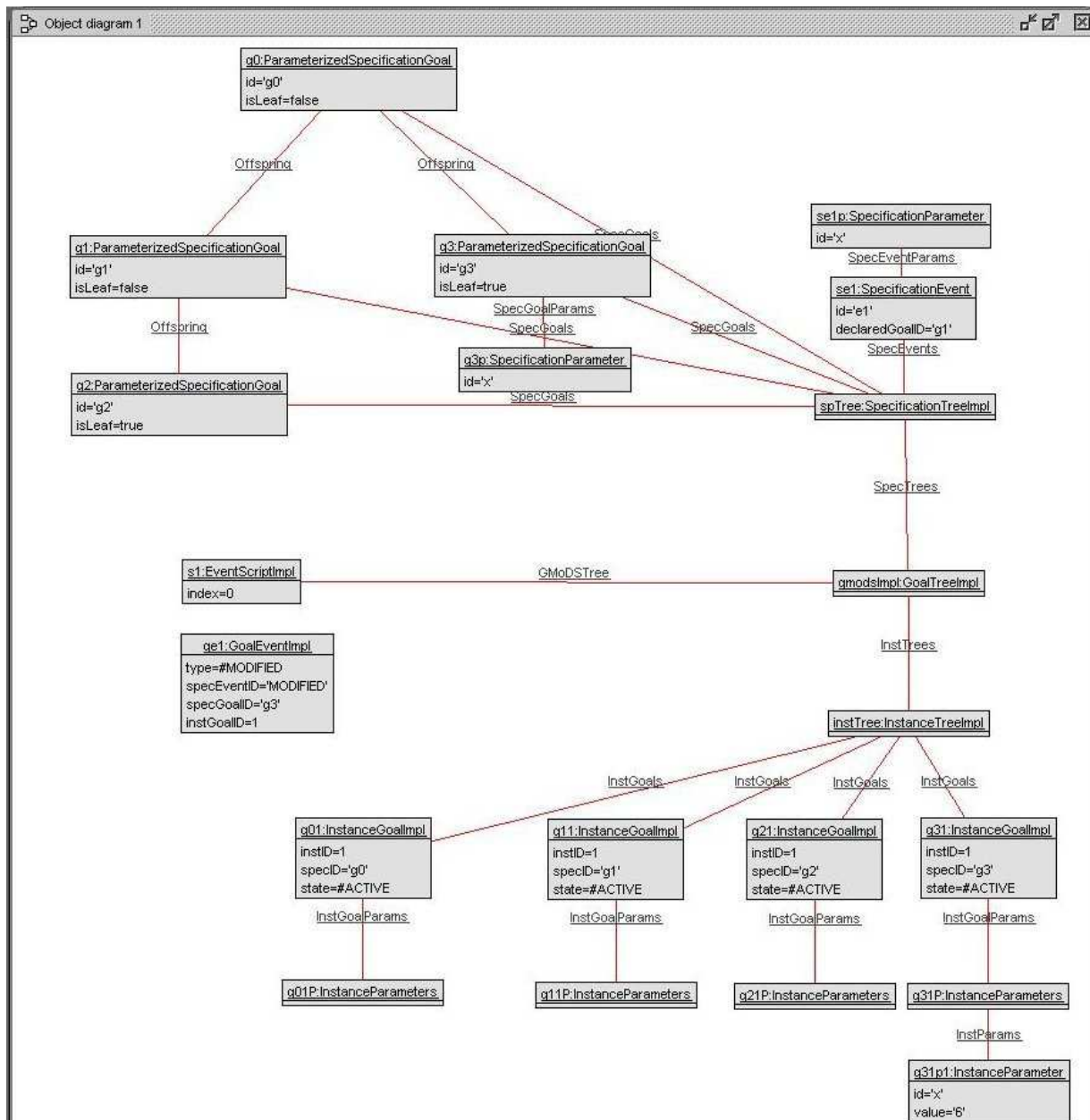


Figure 9 Valid snapshot prior to adding an invalid MODIFIED event with no parameters

Figure 9 above shows a snapshot of an invalid #MODIFIED event which is invalid because it specifies no parameters.

```

C:\> Shortcut to USE 2.6.2 use.bat

..\gmods/gtd-invalid-modified.cmd> ?insert (instTree, g01) into InstGoals
..\gmods/gtd-invalid-modified.cmd> ?insert (instTree, g11) into InstGoals
..\gmods/gtd-invalid-modified.cmd> ?insert (instTree, g21) into InstGoals
..\gmods/gtd-invalid-modified.cmd> ?insert (instTree, g31) into InstGoals
..\gmods/gtd-invalid-modified.cmd>
..\gmods/gtd-invalid-modified.cmd> -- InstGoalParams
..\gmods/gtd-invalid-modified.cmd> ?insert (g01, g01P) into InstGoalParams
..\gmods/gtd-invalid-modified.cmd> ?insert (g11, g11P) into InstGoalParams
..\gmods/gtd-invalid-modified.cmd> ?insert (g21, g21P) into InstGoalParams
..\gmods/gtd-invalid-modified.cmd> ?insert (g31, g31P) into InstGoalParams
..\gmods/gtd-invalid-modified.cmd>
..\gmods/gtd-invalid-modified.cmd> -- InstParams
..\gmods/gtd-invalid-modified.cmd> ?insert (g31P, g31p1) into InstParams
..\gmods/gtd-invalid-modified.cmd>
use> ?openter s1 addEvent(gel)
precondition 'NotInScript' is true
precondition 'ValidType' is true
precondition 'ModifiedReqParam' is false
precondition 'ValidModifiedParamNames' is true
precondition 'ValidSpecGoal' is true
precondition 'ValidAchievedEvent' is true
precondition 'ValidFailedEvent' is true
precondition 'ValidSpecEvent' is true
precondition 'ValidTriggerParamNames' is true
use>

```

Figure 10 Invalid MODIFIED event with no parameters

Figure 10 above shows that invoking the `addEvent` violates the “ModifiedReqParam” precondition for the above snapshot.

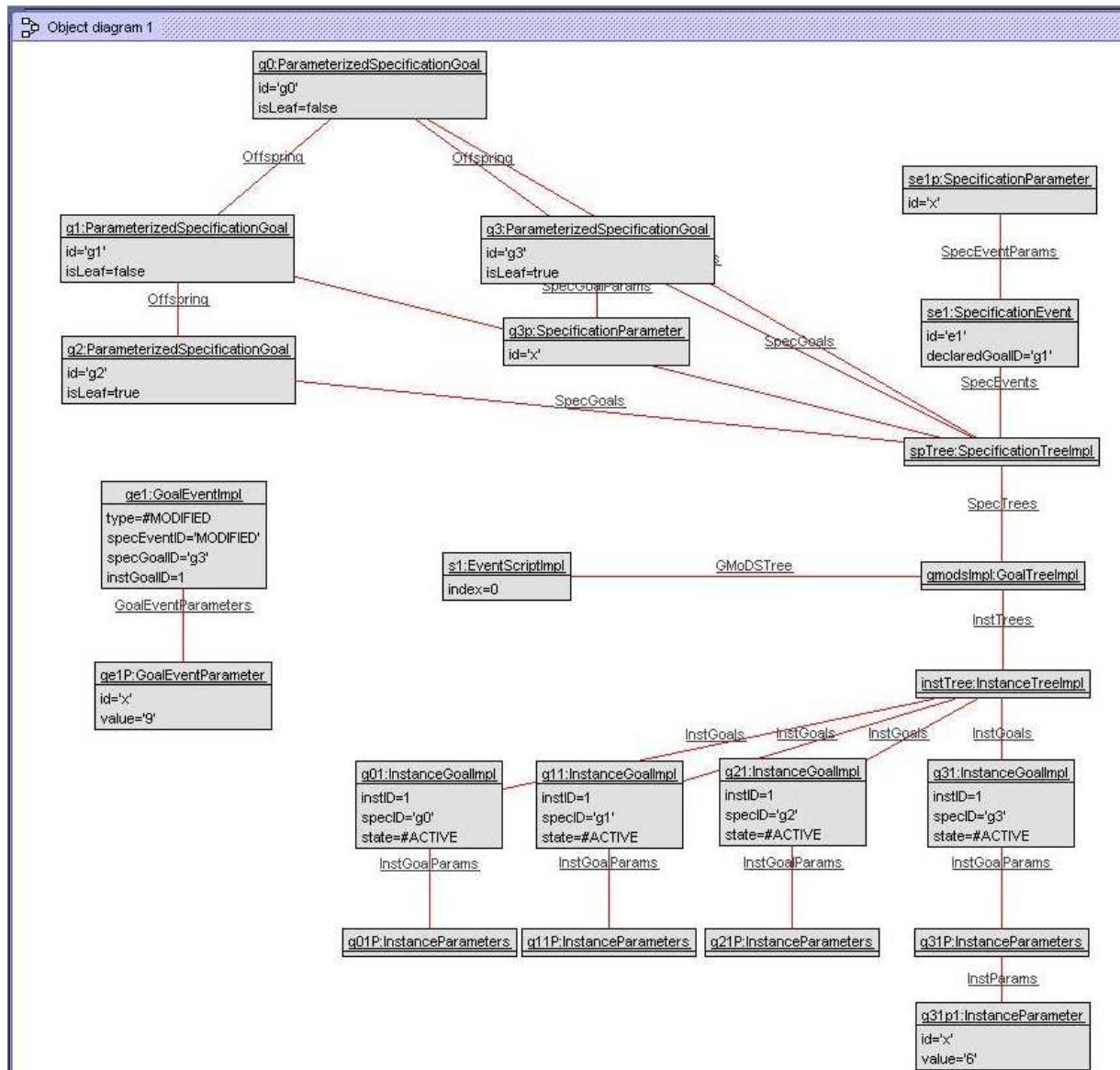


Figure 11 Valid snapshot prior to adding a valid **MODIFIED** event with parameters

Figure 11 above shows a snapshot of the pre state when adding a valid **#MODIFIED** event. Figure 12 below shows that invoking `addEvent` on this snapshot produces valid pre conditions.


```

C:\> Shortcut to USE 2.6.2 use.bat

..../gmods/gtd-valid-modified.cmd> ?insert <instTree, g01> into InstGoals
..../gmods/gtd-valid-modified.cmd> ?insert <instTree, g11> into InstGoals
..../gmods/gtd-valid-modified.cmd> ?insert <instTree, g21> into InstGoals
..../gmods/gtd-valid-modified.cmd> ?insert <instTree, g31> into InstGoals
..../gmods/gtd-valid-modified.cmd>
..../gmods/gtd-valid-modified.cmd> -- InstGoalParams
..../gmods/gtd-valid-modified.cmd> ?insert <g01, g01P> into InstGoalParams
..../gmods/gtd-valid-modified.cmd> ?insert <g11, g11P> into InstGoalParams
..../gmods/gtd-valid-modified.cmd> ?insert <g21, g21P> into InstGoalParams
..../gmods/gtd-valid-modified.cmd> ?insert <g31, g31P> into InstGoalParams
..../gmods/gtd-valid-modified.cmd>
..../gmods/gtd-valid-modified.cmd> -- InstParams
..../gmods/gtd-valid-modified.cmd> ?insert <g31P, g31p1> into InstParams
..../gmods/gtd-valid-modified.cmd>
use> ?openter s1 addEvent(gel)
precondition 'NotInScript' is true
precondition 'ValidType' is true
precondition 'ModifiedReqParam' is true
precondition 'ValidModifiedParamNames' is true
precondition 'ValidSpecGoal' is true
precondition 'ValidAchievedEvent' is true
precondition 'ValidFailedEvent' is true
precondition 'ValidSpecEvent' is true
precondition 'ValidTriggerParamNames' is true
use>

```

Figure 12 Valid pre conditions adding a MODIFIED event

Figure 13 below shows that the script `gtd-invalid-modified-paramnames.cmd` violates the “ValidModifiedParamNames” pre condition.

```

C:\> Shortcut to USE 2.6.2 use.bat

../gmods/gtd-invalid-modified-paramnames.cmd> -- InstGoalParams
../gmods/gtd-invalid-modified-paramnames.cmd> ?insert (g01, g01P) into InstGoalP
arams
../gmods/gtd-invalid-modified-paramnames.cmd> ?insert (g11, g11P) into InstGoalP
arams
../gmods/gtd-invalid-modified-paramnames.cmd> ?insert (g21, g21P) into InstGoalP
arams
../gmods/gtd-invalid-modified-paramnames.cmd> ?insert (g31, g31P) into InstGoalP
arams
../gmods/gtd-invalid-modified-paramnames.cmd>
../gmods/gtd-invalid-modified-paramnames.cmd> -- InstParams
../gmods/gtd-invalid-modified-paramnames.cmd> ?insert (g31P, g31p1) into InstPar
ams
../gmods/gtd-invalid-modified-paramnames.cmd>
use> ?openter s1 addEvent(gel)
precondition 'NotInScript' is true
precondition 'ValidType' is true
precondition 'ModifiedReqParam' is true
precondition 'ValidModifiedParamNames' is false
precondition 'ValidSpecGoal' is true
precondition 'ValidAchievedEvent' is true
precondition 'ValidFailedEvent' is true
precondition 'ValidSpecEvent' is true
precondition 'ValidTriggerParamNames' is true
use>

```

Figure 13 Invalid parameter name in a MODIFIED event