

Introduction

This edition of the newsletter identifies some problems with capturing Use Case scenarios using text, and offers a potential solution to this problem.

Capturing Use Case Scenarios Using Text

A common method of modelling the steps within a Use Case is to define a series of scenarios each of which is described as a number of steps. These steps model the “conversation” between Actor(s) and the System and are typically captured using numbered lists.

These scenarios fall into two categories:

- a) Basic Path – models the normal expected behaviour through a Use Case and where no errors would occur.
- b) Alternate – models a variation from the Basic Path, for example when an error occurs. All error conditions would be modelled using separate Alternate paths.

A typical textual scenario is listed below, this example is for a use case “Make a Reservation” for a simple Hotel Management system.

Basic Flow

- 1) Use case begins when a Customer submits a request to make a reservation
 - 2) Displays "Reservation Room, Date, Nights" Form
- 3) Customer completes the "Reservation Room, Date, Nights" Form
 - 4) Searches for available rooms that match the details entered on the submitted form
 - 5) Calculates price for the reservation
 - 6) Displays the "Terms and Conditions" form which also includes the total cost
- 7) Customer agrees to the Terms and Conditions
- 8) Customer submits the "Terms and Conditions" form
 - 9) Displays the "Customer Details Form"
- 10) Customer completes the "Customer Details Form"
- 11) Customer submits the completed form
 - 12) Displays the "Credit Card Details Form"
- 13) Customer completes the "Credit Card Details Form"
- 14) Customer submits the completed form
 - 15) Assigns unique reservation number
 - 16) Records the reservation
 - 17) Displays "Reservation Confirmed" message
 - 18) Sends e-mail confirmation to Customer
 - 19) Use Case ends

Alternate – No Rooms Available

- 1) At Basic Flow Step 4) no rooms available that match reservation request
- 2) Displays "No Rooms Available Message"
- 3) Resumes at Basic Flow step 2)

Alternate – Refuses Terms and Conditions

- 1) At Basic Flow Step 7) Customer declines "Terms and Conditions"
- 2) Use case ends

This textual description can be modelled very easily within Enterprise Architect using the Scenario tab on the Use Case Element properties dialog.

The Problems

Capturing Use Case scenarios textually using the method above is the most common in systems modelling but does suffer from problems especially when maintaining the scenarios or trying to view the scenarios as a whole.

It is usual to model Alternate paths as being branches from the Basic Flow (as shown in the example above) and this relates the flows together. Because of this relationship a common alternate to the example above is to use the Basic Flow branch step as a prefix for each step of the Alternate flow as shown below:

Alternate – No Rooms Available

- 4.1) No rooms available that match reservation request
- 4.2) Displays "No Rooms Available Message"
- 4.3) Resumes at Basic Flow step 2)

This raises a small problem within Enterprise Architect. Although a simple auto numbered list is provided for entering the textual steps in the scenarios tab of the Use Case element properties dialog, this does not cater for prefixes as shown above, so this will have to be entered manually, and any additional steps within the Alternate flow will result in a manual re-ordering of the steps.

This will also be true of course for all scenario paths if the auto numbered list has not been used to capture the steps (as in the example above).

On a more serious level, if the steps in the Basic Flow are changed, then this could affect all Alternate Flows, since the branch and resume steps in the Alternate Flow will now be incorrect. Consider for example, if we add some steps in the Basic Flow after Step 3) which captures the validation process. Then the two Alternate flows are no incorrect because they refer to out-of-date steps in the Basic Flow.

This manual updating of scenarios in order to maintain consistency is both labour intensive and error prone. But, using Enterprise Architect there is no solution to this problem.

Even if the Scenarios are captured external using Word for example, and then the Word document linked to the use case as an external file, the problem of maintaining the scenarios is still relevant. It appears that there is no way this problem can be overcome.

A Possible Solution

The possible solution to the scenario maintenance problem is to use a dedicated external editor designed specifically for such a task, and then to integrate this tool with Enterprise Architect. Does such a tool exist? The answer is yes and it's free.

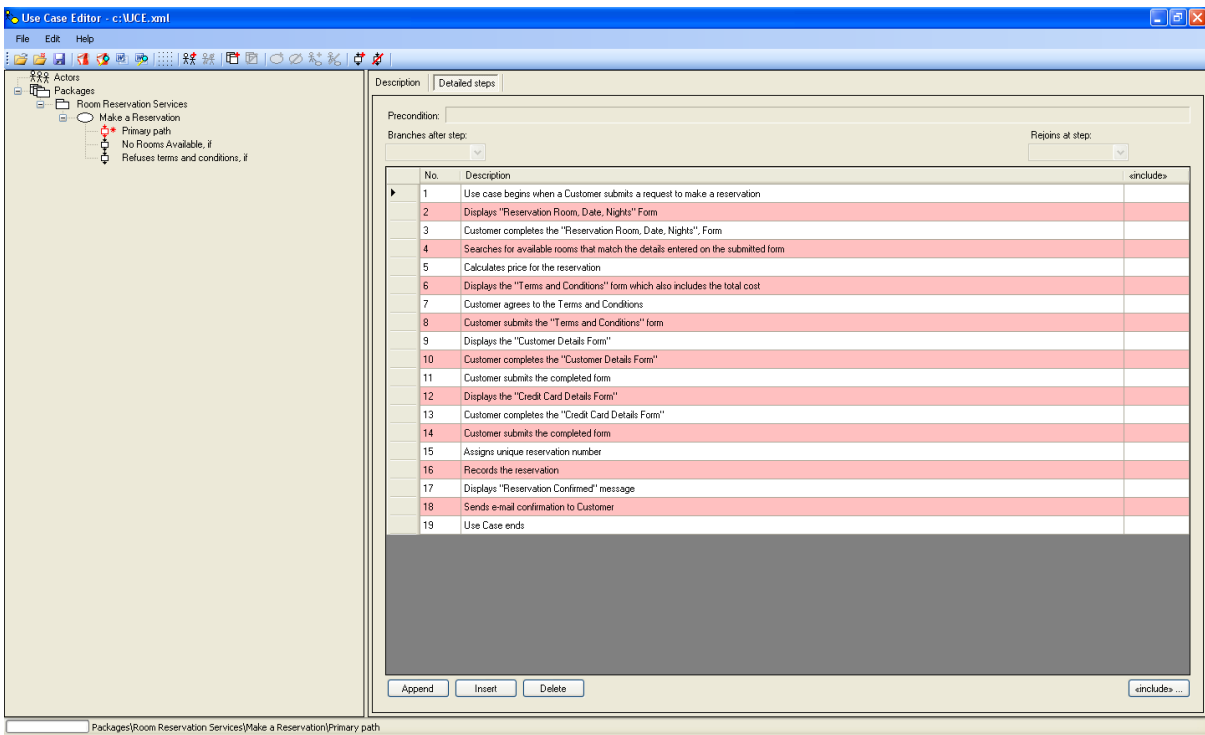
The tool is called the "Use Case Editor" and is available for free download from www.ropley.com. It should be noted that this tool is supplied without any maintenance agreement, however any bug reports, feature enhancement request can be directed to the author. At this point it must be stressed that neither the author of the newsletter nor Dunstan Thomas has any connection with the author of the Use Case Editor other than as a user of the tool.

The Use Case Editor is more than just an editor for scenarios, it does allow the complete Use Case Model to be captured and documented, but it is the scenario editor aspect that is the focus of this newsletter.

It is not the intention to detail a tutorial on the Use Case Editor, rather just to highlight its use in maintaining integrated scenarios.

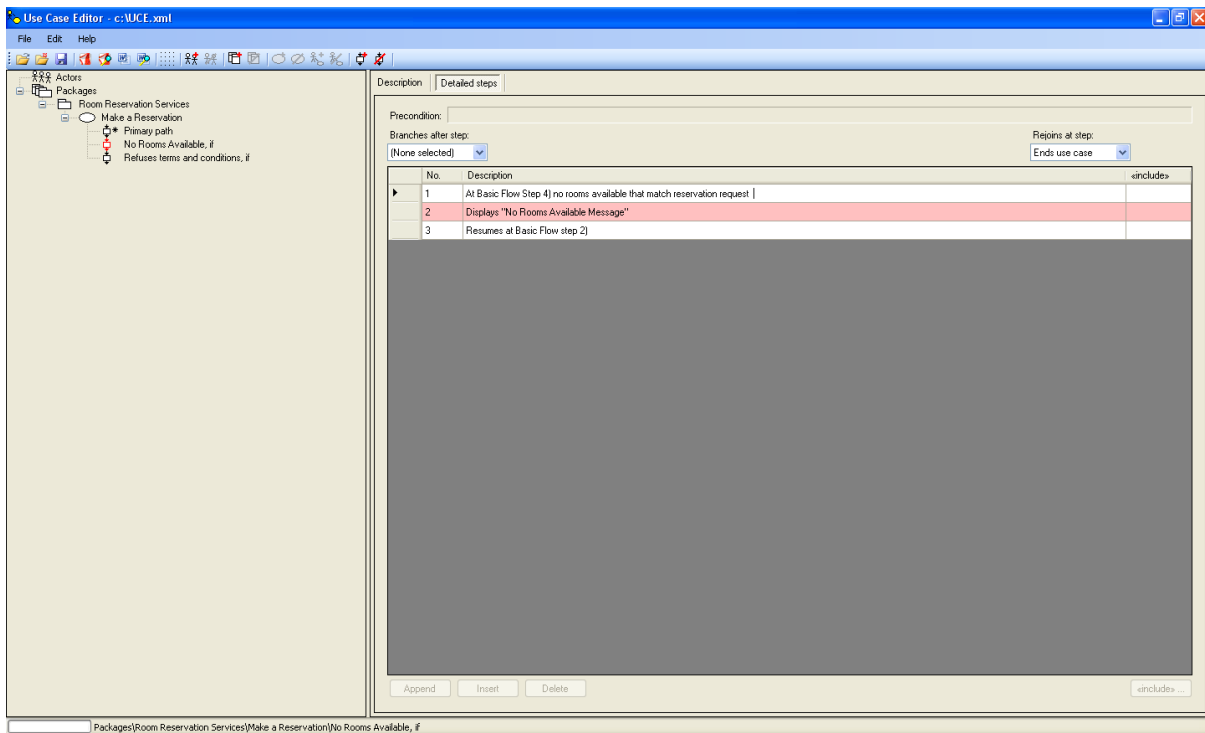
Using the Use Case Editor to Capture Scenario Steps

Using the example above, the Basic Flow of the Make Reservation Use Case would be captured using the Use Case Editor as shown in the screen shot below:



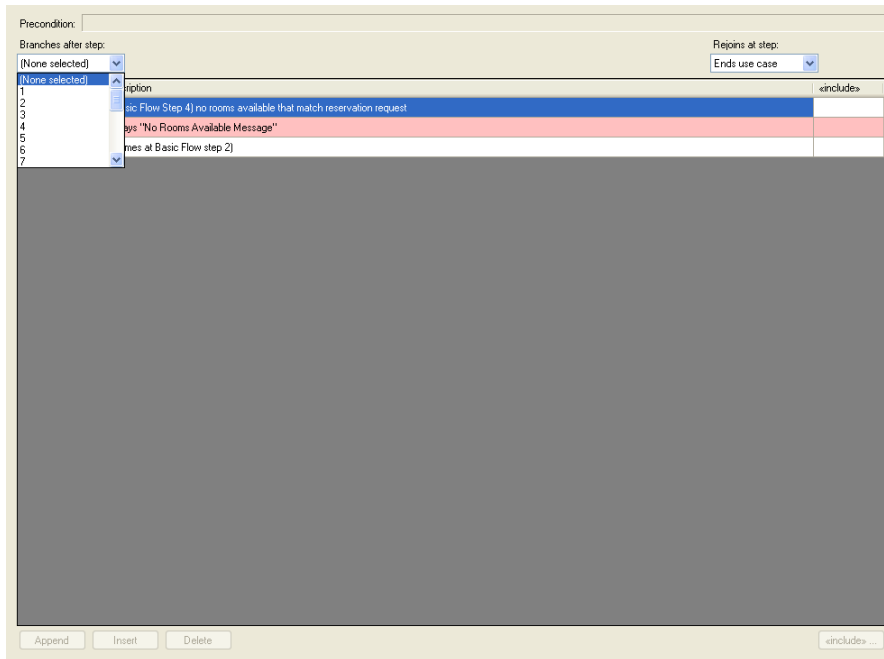
As can be seen each step of the Basic Flow is captured as a single line and there are functions available to Append, Insert and Delete lines as appropriate.

The Alternate Flows are captured in a similar manner:

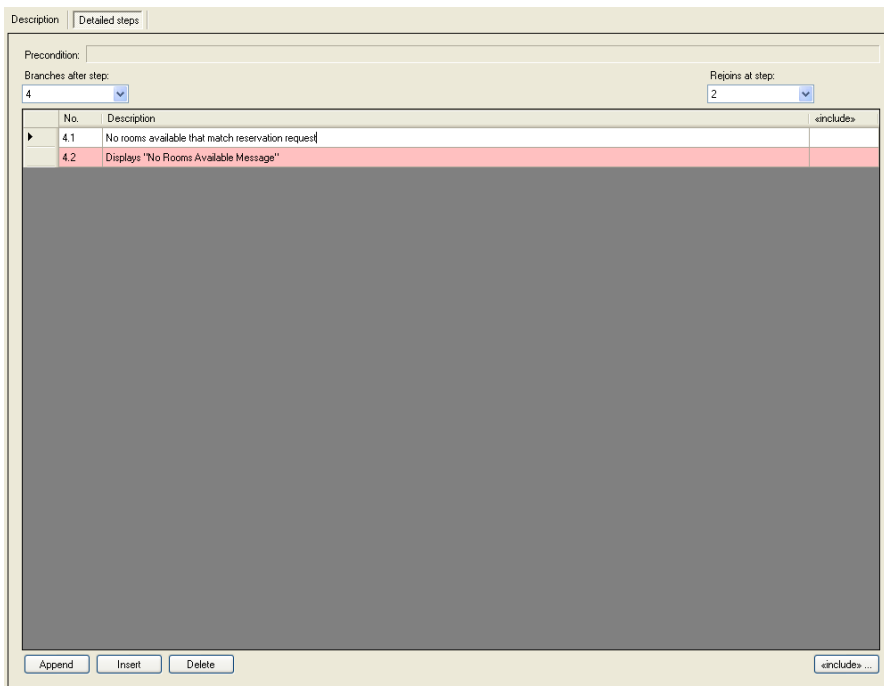


However, by using the drop down lists “Branches after step” and “Rejoins at step” it is possible to link this Alternate flow with the Steps of the Basic Flow.

The contents of these drop down lists are populated from the step numbers in the Basic Flow as shown below:

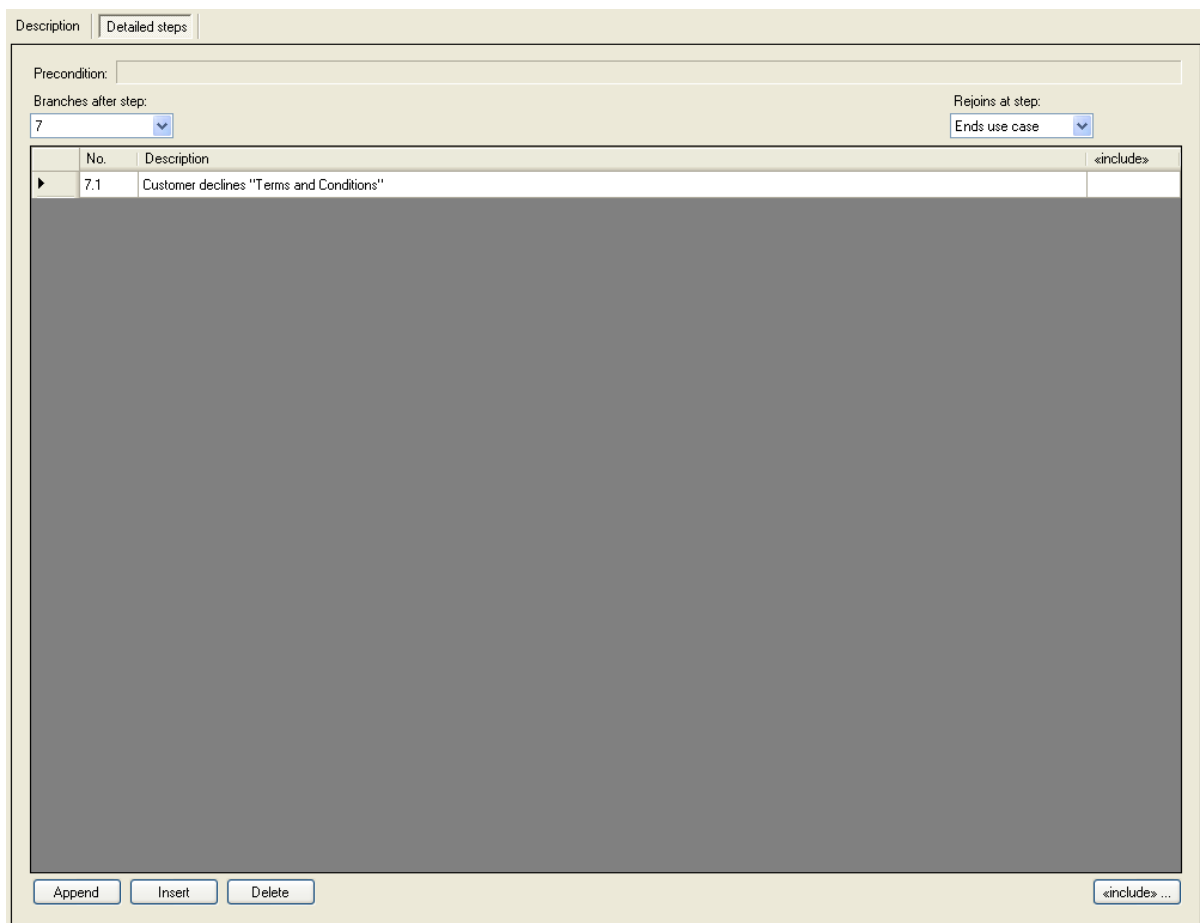


Using this we can modify the Alternate Flow steps and the result is as shown below:



Note that each step is no prefixed with the corresponding step in the Basic Flow from which this Alternate flow branches

Similarly, the other Alternate Flow would be captured in the Use Case Editor as:



The screenshot shows the 'Detailed steps' tab of the Use Case Editor. It features a 'Precondition' field, a 'Branches after step' dropdown set to '7', and a 'Rejoins at step' dropdown set to 'Ends use case'. Below these is a table with one row:

No.	Description	<include>
7.1	Customer declines "Terms and Conditions"	

At the bottom, there are buttons for 'Append', 'Insert', 'Delete', and '<include> ...'.

Should the Basic Flow be updated, then both these Alternate Flows would be updated simultaneously. This is illustrated in the sequence of screen shots below:

Three new steps have been added to the Basic Flow, these appear below as 4, 13, and 17

Description | Detailed steps

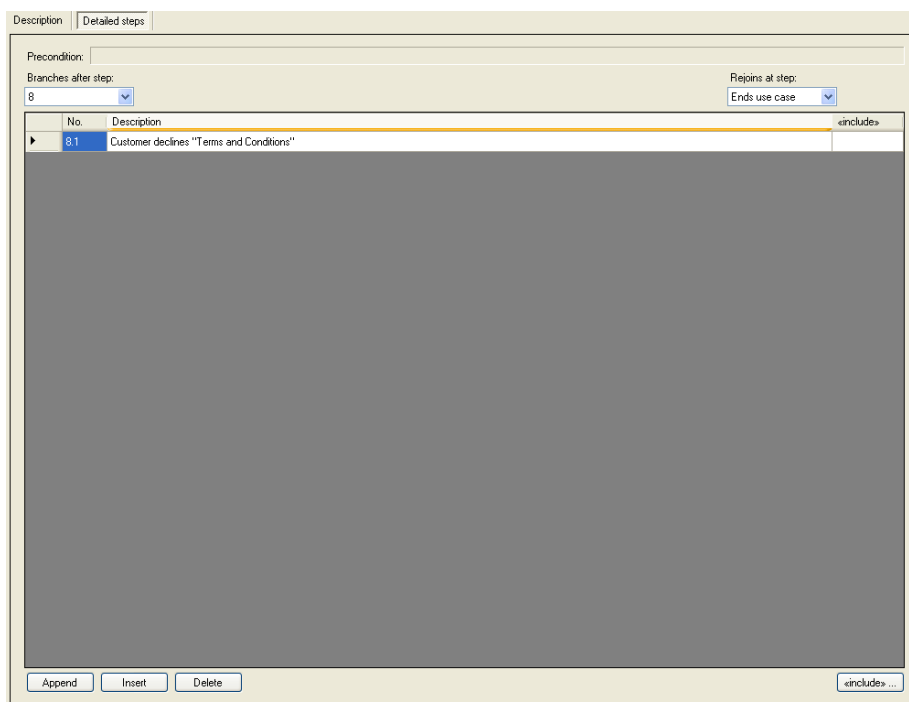
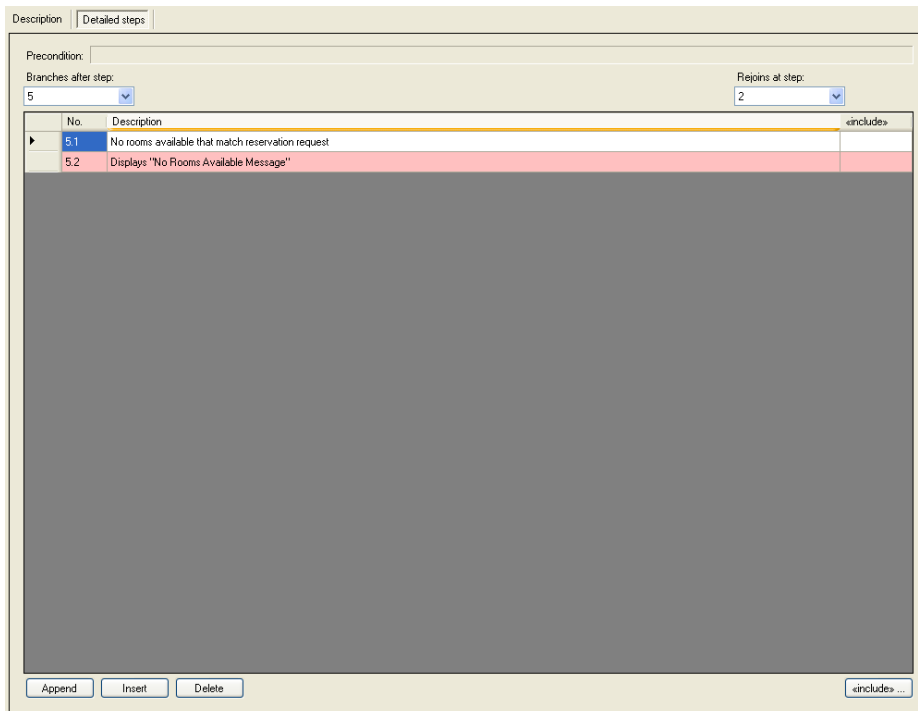
Precondition:

Branches after step: Rejoins at step:

No.	Description	«include»
1	Use case begins when a Customer submits a request to make a reservation	
2	Displays "Reservation Room, Date, Nights" Form	
3	Customer completes the "Reservation Room, Date, Nights", Form	
4	Validates the data on the Form	
5	Searches for available rooms that match the details entered on the submitted form	
6	Calculates price for the reservation	
7	Displays the "Terms and Conditions" form which also includes the total cost	
8	Customer agrees to the Terms and Conditions	
9	Customer submits the "Terms and Conditions" form	
10	Displays the "Customer Details Form"	
11	Customer completes the "Customer Details Form"	
12	Customer submits the completed form	
13	Validates the data on the Form	
14	Displays the "Credit Card Details Form"	
15	Customer completes the "Credit Card Details Form"	
16	Customer submits the completed form	
17	Validates the data on the Form	
18	Assigns unique reservation number	
19	Records the reservation	
20	Displays "Reservation Confirmed" message	
21	Sends e-mail confirmation to Customer	
22	Use Case ends	

Append Insert Delete «include» ...

The Alternate Flows have been updated as appropriate



Integrating the Use Case Editor with Enterprise Architect

Now that we can capture and maintain our scenario steps, we have introduced a new problem; the scenario steps are external to the Enterprise Architect model.

The solution to this is to write an Add-In for Enterprise Architect which launches the Use Case Editor and passes to it any scenario text that has been captured in Enterprise Architecture. After editing the use case steps, the data is extracted from the Use Case editor and stored back into the Use Case element within Enterprise Architect.

The data transfer between the Use Case Editor and Enterprise Architect is XML.

Such an Add-In has been developed at Dunstan Thomas, however it is beyond the scope of this newsletter to detail the construction of such an Add-In, but if you would like any assistance in integrating the Use Case Editor with Enterprise Architect, then please do not hesitate to contact us.