# Review Facility

As mentioned in the SRS covering UML Notes, the capability can be used for a number of purposes including review comments. This document describes an approach for using UML Notes to capture review comments.

# Overview

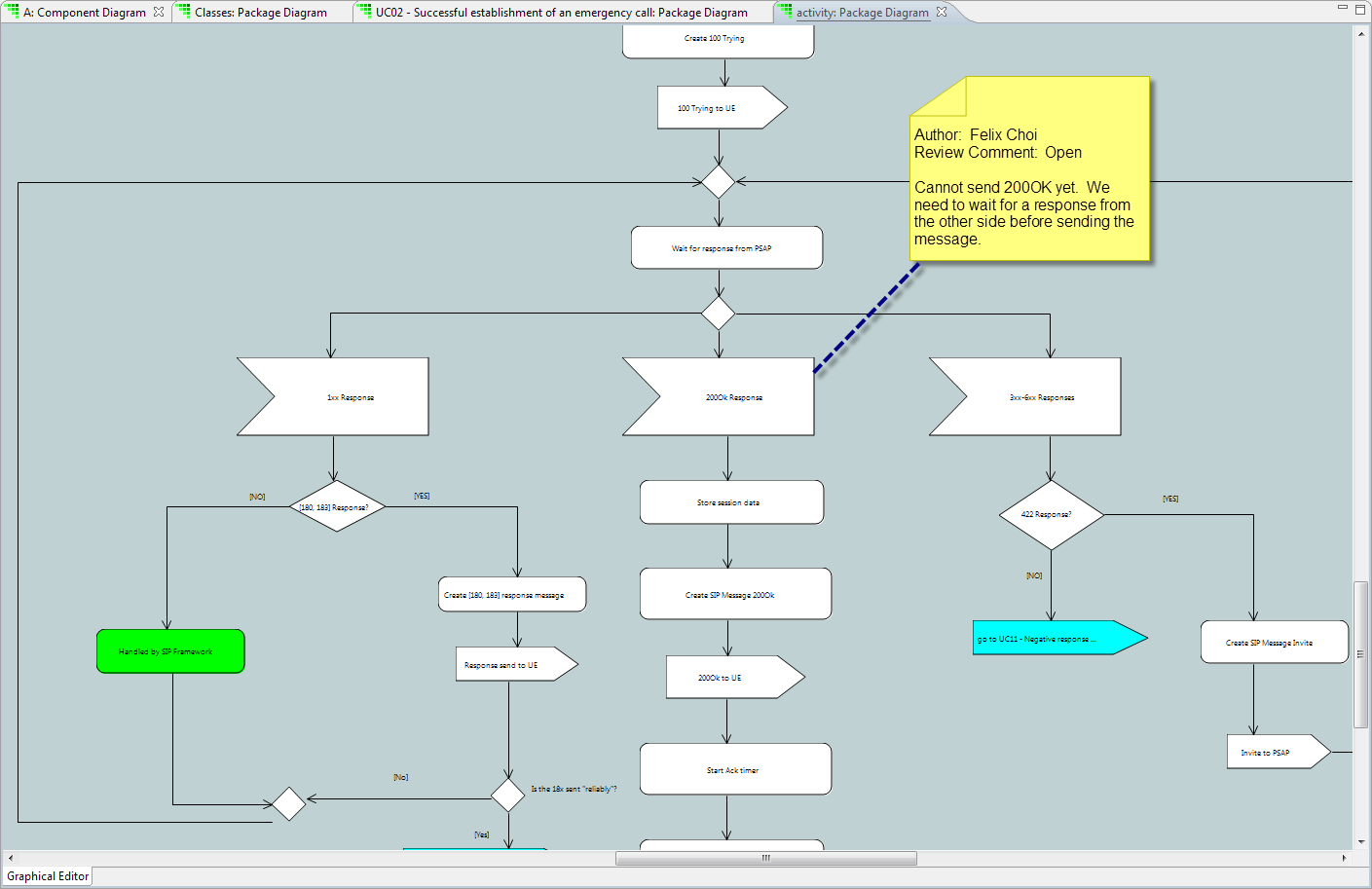
This approach utilizes the following capabilities:

* Layers. Support for graphical layers exists in BridgePoint. Briefly, any number of layers can be created and named by the user. Any graphical model element can be added to any number of existing layers. Model elements can also be removed from a layer. Layers can then be selectively hidden or shown, causing all model elements associated with the layer to be either hidden or shown. If a model element is associated with more than one layer, it is visible whenever one or more of its associated layers is shown.
* UML Notes. Support for Notes is planned for a future BridgePoint release and is described in an SRS.
* Content Attribute List. Presently, this support has not been specified and is therefore not yet planned for any BridgePoint release. This conceptual facility provides a list of all content attributes (ref. SRS for UML Notes). The list can be filtered and sorted, and it provides simple navigation from an item in the list to the location of that item within the model.

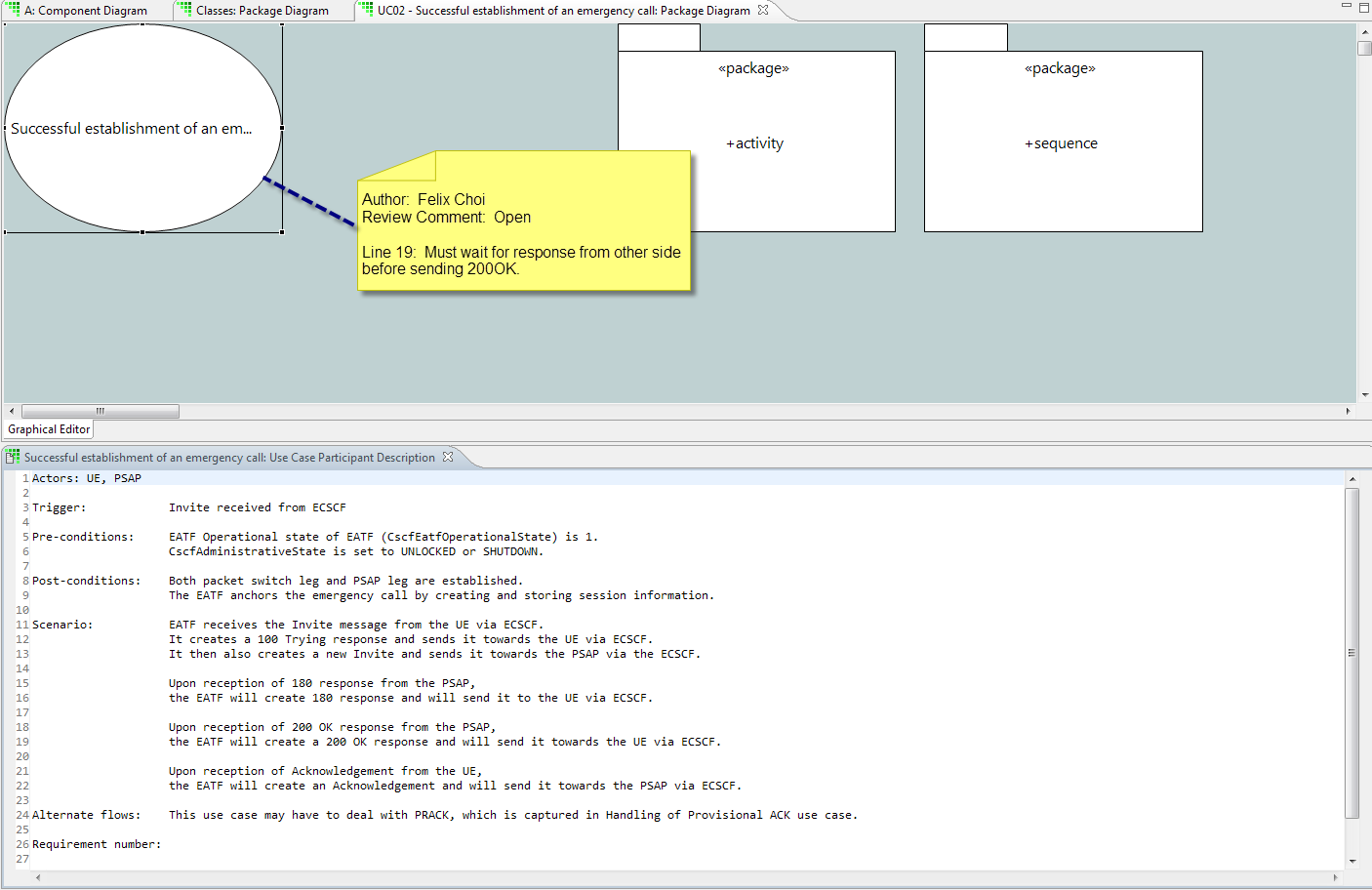
An alternative to using multiple layers would employ keywords within the review comment text such as “Review Comment: [Open|Accepted|Rejected|Closed]”. This approach would likely make it easy to use the xtUML search facility to locate all review comments of a particular disposition. The relative advantage of the layer-based approach is the ability to hide from view all review comments of a particular disposition.

# Examples

Two examples are shown as mocked-up screen shots below.



A review comment placed on an activity diagram, associated with a particular activity step.



A review comment attached to a use case, referring to a particular line number within the use case specification. The screen shot above illustrates the inherent ability of Eclipse to display multiple (canvas or text) editors on the same screen. In this case the use case specification is shown in a text editor below the canvas editor on which the use case oval is defined. This approach allows anyone involved in reviewing models and addressing review comments to see the contents of a review comment along with the textual specification to which it applies.

# Use Cases

### UC01 – Create review comment

#### Pre-conditions

1. A model ready for review exists.

#### Post-conditions

1. One or more review comments exist.

#### Scenario

1. User creates a UML Note.
2. User enters review comments into the Note.
3. User attaches the Note to one or more model elements to which the Note applies.
4. User adds the Note to the layer designated for open review comments.

### UC02 – Accept and reject review comments

#### Pre-conditions

1. A number of review comments exist, each one captured in a UML Note which is associated with a layer designated for open review comments.

#### Post-conditions

1. Each review comment is associated with one of two layers, one designated for accepted review comments to be addressed for the designers and another designated for rejected review comments which can be ignored by the designers.

#### Scenario

1. User displays the content attribute list.
2. User filters the content attribute list to include only those attached to the layer designated for open review comments.
3. For each review comment in the filtered list,
   1. User employs automatic navigation from content attribute list to the content attribute (UML Note containing a review comment).
   2. Editor displays the review comment along with the model element(s) associated with it.
   3. User determines disposition of the comment (either accepted or rejected).
   4. User removes the review comment from the layer designated for the open review comments.
   5. User adds the review comment to the layer designated for the appropriate disposition (accepted or rejected).

### UC03 – Address review comments

#### Pre-conditions

1. A number of review comments exist, each one captured in a UML Note which is associated with a layer designated for review comments to be addressed.

#### Post-conditions

1. Models have been modified as specified in review comments.
2. All addressed review comments are associated with a layer designated for comments that are closed.

#### Scenario

1. User displays the content attribute list.
2. User filters the content attribute list to include only those attached to the layer designated for review comments to be addressed.
3. For each review comment in the filtered list,
   1. User employs automatic navigation from content attribute list to the content attribute (UML Note containing a review comment).
   2. Editor displays the review comment along with the model element(s) associated with it.
   3. User modifies the model elements in question to address the review comment.
   4. User removes the review comment from the layer designated for the review comments to be addressed.
   5. User adds the review comment to the layer designated for review comments that are closed.

### UC04 – Clarify a review comment

#### Pre-conditions

1. A number of review comments exist, each one entered as a UML Note and associated with a layer designated for comments to be addressed

#### Post-conditions

1. Review comment has been clarified, allowing user to modify models to address it.

#### Scenario

Duplicate steps 1 and 2 from UC03.

1. User employs automatic navigation from content attribute list to the content attribute.
2. Editor displays the content attribute along with the model element(s) associated with the content attribute.
3. User determines that this review comment requires clarification before it can be addressed.
4. User notices the identity of the author as displayed on the note.
5. User contacts author of the note who clarifies the review comment.