Worksheet 4

Spring Semester, 2010-2011

Creating the Sequence Diagram

J.J. Collins, Howell Jordan, and Anne Meade

1. Introduction

Sequencing, selection/branching and iteration are the basic building blocks used to model and implement many software engineering artifacts. These could be design time or runtime concepts.

A scenario is an instance of a use case, and is modelled as an interaction sequence diagram. This requires sequencing, branching/selection, and iteration.

Figure 1 depicts a sequence diagram for a use case "Postpone On-Site Activity" from the Select "Examples" datastore.

- The sequence diagram is documented with structured English on the right.
- Branching/selection is indicated by the IF-ELSE statement.
- Note the use of Indentation. For example
- When executing the method *Postpone Activity*, *If Current Resource is Unavailable Then Notify De-allocation* ……".
- The white circle symbol indicates that the sequence of messages being passed has been modelled in the sequence diagram for the Use Case Allocate Resource to Activity

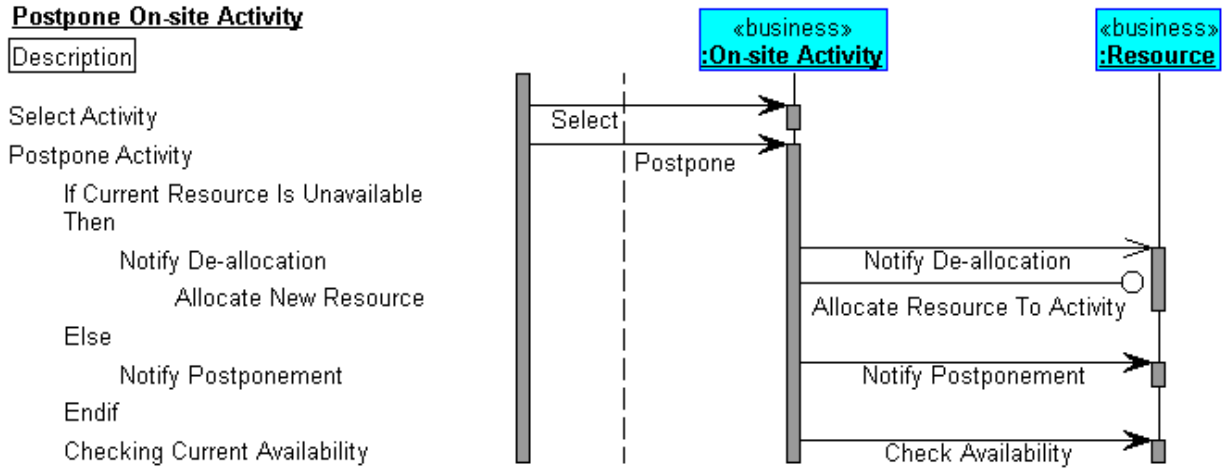


Figure 1

2. Create a new Sequence Diagram.

Before one can create a sequence diagram, the classes and their operations, and any other artifacts such as actors and interfaces required for the sequence diagram must first be specified in the class diagram. Then

1. Open your project (a model) in Select Component Architect.
2. Check the dictionary to see if you have the actors and use cases already created, if not create the Actors and Classes that appear in the diagram.
3. Open a new Sequence Diagram, by right clicking on the Sequence Diagrams folder.

3. Adding Actors and Classes to a Sequence Diagram.

1. Open the dictionary view and drag the actors and classes necessary for the diagram onto the new sequence diagram.
2. Arrange the classes in the order they appear in the diagram.
    a. Notice that the actors and classes are prefixed with an instance label.

4. Creating Sequences.

1. Choose the sequence button on the diagram toolbar, and then right click on the diagram window. Describe the operation in the text box that appears on the left side of the page.
2. Next, choose the operation tool  from the diagram toolbar, and click it from the message sender to recipient, and then select an operation from the operations of that class that appear in the popup window.

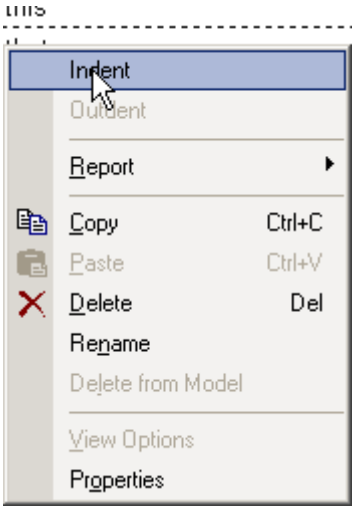
5. Creating Selection.

1. Choose the selection button  on the diagram toolbar.
2. Next, click on Sequence in the diagram toolbar, and place the sequence within the selection. Describe the guard condition and operation in the text box that appears on the left side of the page.
3. Next, select the operation tool from the toolbar and click it from the message sender to recipient, and then select an operation from the operations of that class that appear in the popup window.

6. Specifying Reflexive Calls Using the Indent Option

For example: *Object O* receives a message to execute *methodA()*. During execution of *methodA()*, *Object O* invokes *methodB()* upon itself – a reflexive call.

1. Draw a the fragment to illustrate the execution of *methodA()* by *Object O* (sequence and operation).
2. Create another sequence (description) for *methodB()*
3. Right click on the description for *methodB()* on the left side of the diagram and select indent.



4. Then draw the operation representing the invocation of *methodB*, where the sender and receiver are the same object – *Object O*.

7. Notes

1. Drawing Iteration is similar in manner to Selection, except that one chooses the Iteration tool  on the diagram toolbar.
2. Selection and Iteration automatically indent the sequence description.
3. Use indent when modelling nested calls that may not be reflexive.
4. Select Architect does not support some of the Sequence diagram concepts in UML 2.x. I suggest that you do not use frame boxes to model interaction frames, as they add confusion to the visual depiction.