Validation (3): Sequence Diagrams

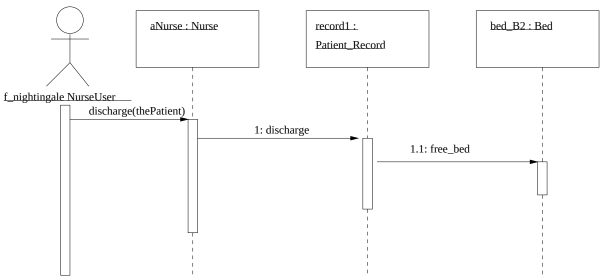
CS3: SEOC1

Note 8

Sequence Diagrams

These provide a more detailed view of an interaction than the collaboration diagram. In many cases the information represented in the two kinds of diagram is the same. For simple situations it seems (to me) that sequence diagrams are easier to use and comprehend

Simple Sequence Diagram



Sequence Diagram Layout

- Each object in the interaction is represented by its named icon along the top of the diagram.
- Under each icon is the lifeline for that object. Activations are marked by a widened section in the lifeline. Time progresses downwards.
- Directed links between lifelines labelled with messages represent messages exchanged between objects.
- For a link to be present there must be an association between the corresponding classes in the class diagram.
- Return links can also be inserted

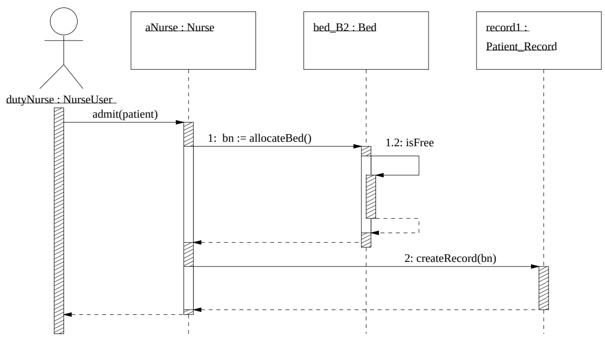
More Advanced Features

Sending messages to self Means an object has two activations simultaneously. Represent by another displaced activation line on top of the first activation

Suppressing Detail Use *packages* to suppress detailed interaction and structure a collection of objects

Return Values: Often worthwhile to label the return value because it may be used later in the interaction

HIS Sequence Diagram



Creation and Deletion

Collaboration Diagram: Here the objects in the collaboration are labelled:

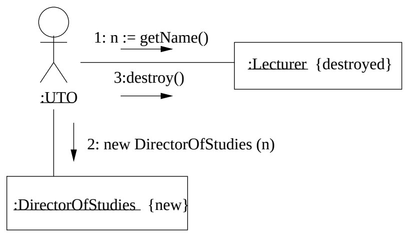
- `new` for objects created in the collaboration.
- `destroyed` for objects destroyed during the collaboration.
- `transient` means `new` and `destroyed`.

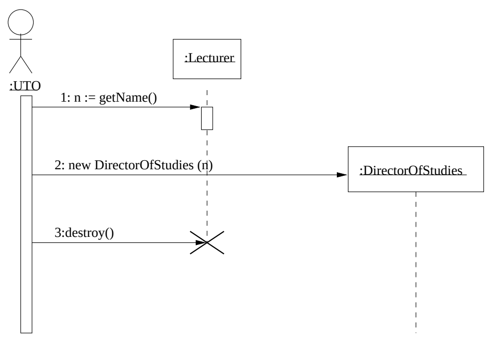
Sequence Diagram: Here we can use the lifeline.

new objects: have their icon inserted when they are created.

destroyed objects: have their lifeline terminated: × marks the spot!

Creation and deletion
Using UML: Figures 9.7 and 9.8

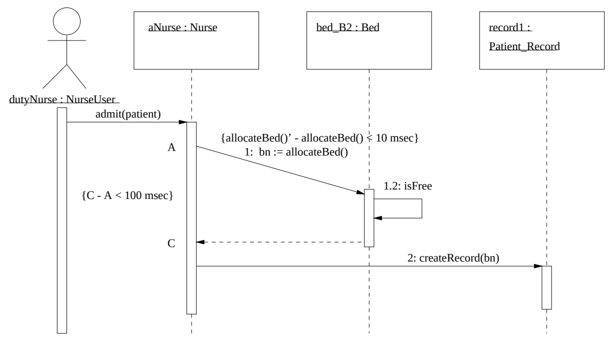




Timing

- This is most easily dealt with in sequence diagrams
- Label the points of issue and return for a message. Use these labels in expressing timing constraints.
- This technique also works for message sending that takes time (so arrows are sloping down).
- We could also make metric information in the diagram contribute to representing timing, but this is not recommended (why not?)

HIS Sequence Diagram, with Timing



How do interaction diagrams help?

Check use cases: This is the main focus of these lectures

Check class can provide an operation: shows how a class realises some operation by interacting with other objects

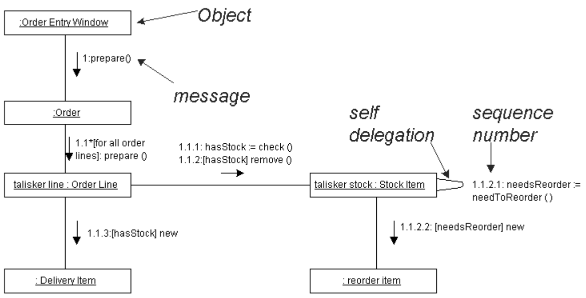
Describe design pattern: Parameterising by class provides a scheme for a generic interaction (part of *Software Architecture*).

Describe how to use a component: captures how some component can interact.

Summary

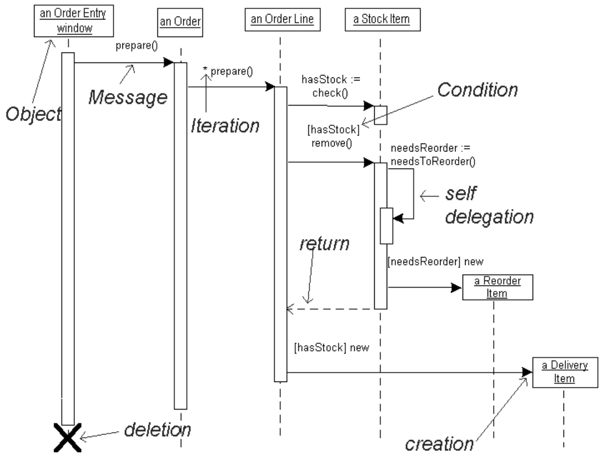
- Captures some elements of the dynamics of systems.
- Support a number of different activities.
- Describe interaction in some detail, including timing.
- Detail can be hidden using packages.

Annotated Example Collaboration Diagram



12

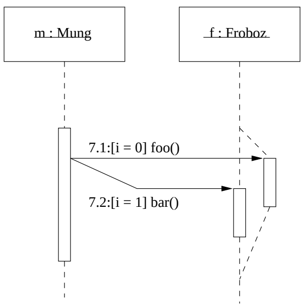
Annotated Example Sequence Diagram



13

Fragment of sequence diagram with conditional behaviour and branching lifeline

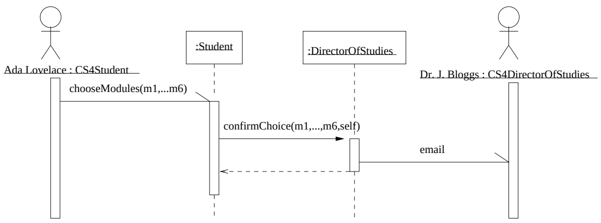
Using UML: Figure 10.2



14

Asynchronous message passing in a sequence diagram

Using UML: Figure 10.6



15