Lecture 5: Modeling Software Behaviour

- → UML sequence Diagrams
- → Comparing Designs
- → Explaining Design Patterns
- → Style tips

© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

Things to Model

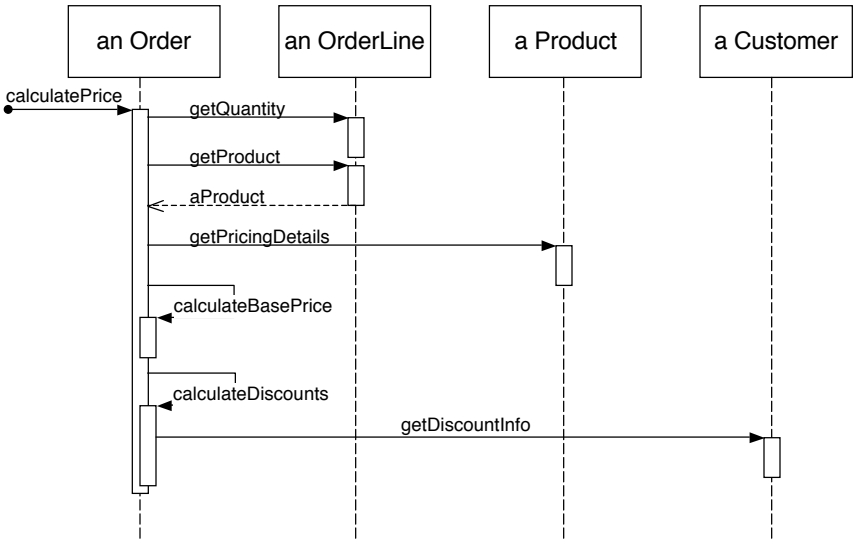
E.g. Structure of the code
- Code Dependencies
- Components and couplings

E.g. Behaviour of the code
- Execution traces
- State machines models of complex objects

E.g. Function of the code
- What functions does it provide to the user?

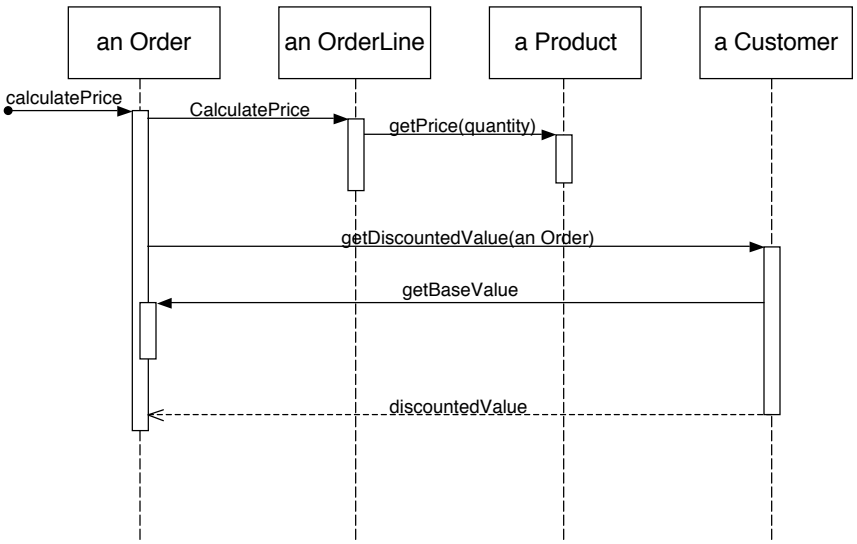
© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

Sequence Diagrams



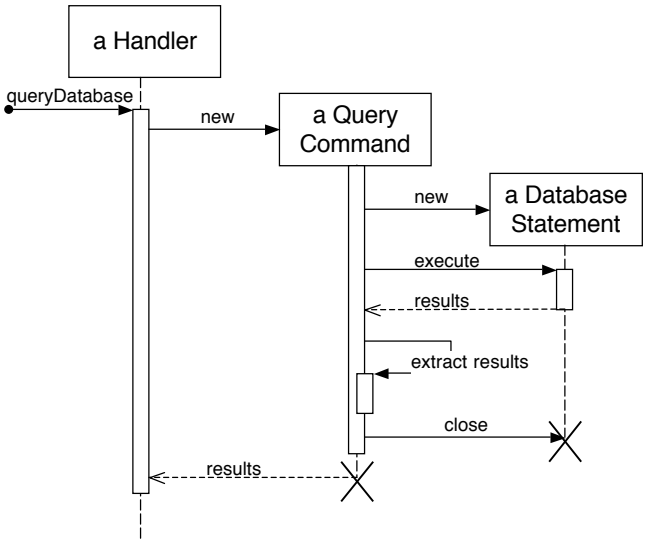
© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

Design Choices…



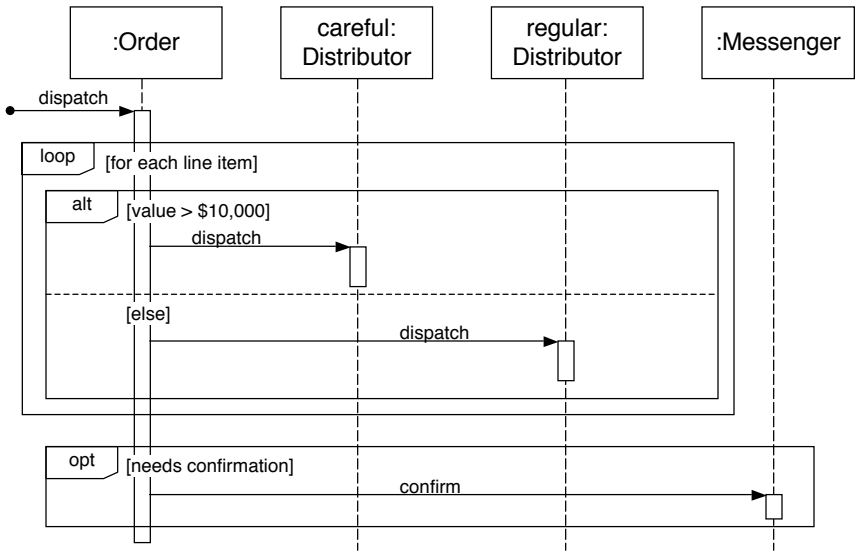
© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

Creating and Deleting Objects



© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

Interaction Frames



© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

Interaction Frame Operators

Operator	Meaning
alt	Alternative; only the frame whose guard is true will execute
opt	Optional; only executes if the guard is true
par	Parallel; frames execute in parallel
loop	Frame executes multiple times, guard indicates how many
region	Critical region; only one thread can execute this frame at a time
neg	Negative; frame shows an invalid interaction
ref	Reference; refers to a sequence shown on another diagram
sd	Sequence Diagram; used to surround the whole diagram (optional)

© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

When to use Sequence Diagrams

Comparing Design Options

- Shows how objects collaborate to carry out a task
- Graphical form shows alternative behaviours

Assessing Bottlenecks

- E.g. an object through which many messages pass

Explaining Design Patterns

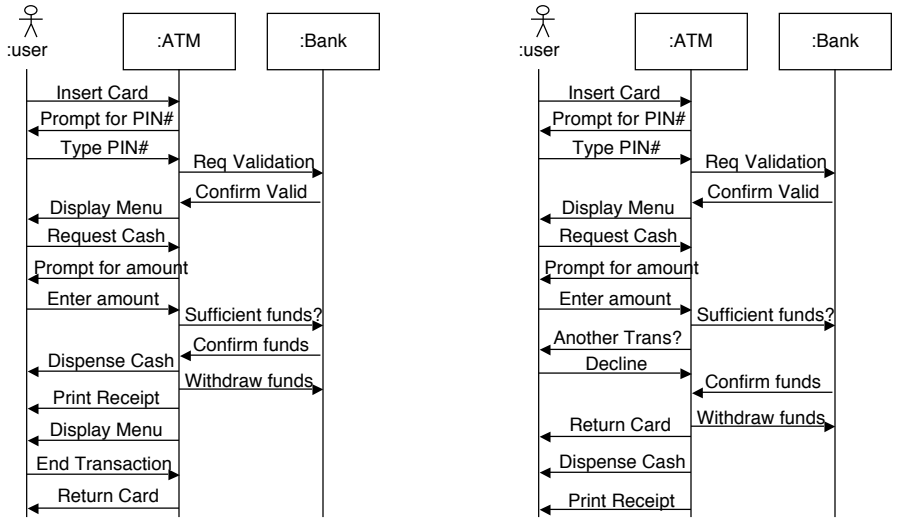
- Enhances structural models
- Good for documenting behaviour of design features

Elaborating Use Cases

- Shows how the user expects to interact with the system
- Shows how the user interface operates

© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

Comparing Designs



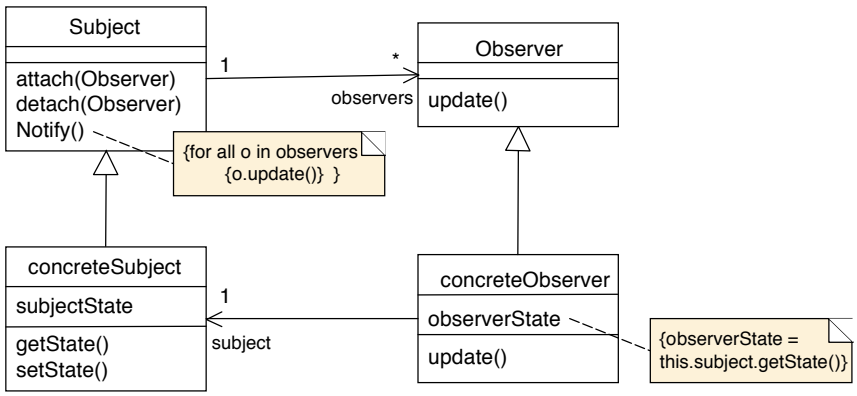
© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

Modeling a Design Pattern

E.g. Observer Pattern

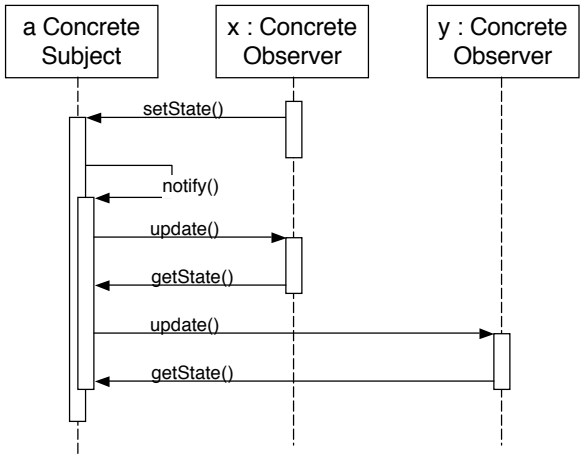
For a one-to-many dependency, when you need to maintain consistency

The subject pushes updates to all the observers



© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

Sequence Diagram for Observer



© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.

Style Guide

Spatial Layout

- Strive for left-to-right ordering of messages
- Put proactive actors on the left
- Put reactive actors on the right

Readability

- Keep diagrams simple
- Don't show obvious return values
- Don't show object destruction

Usage

- Focus on critical interactions only

Consistency

- Class names must be consistent with class diagram
- Message routes must be consistent with (navigable) class associations

© 2008 Steve Easterbrook. This presentation is available free for non-commercial use with attribution under a creative commons license.