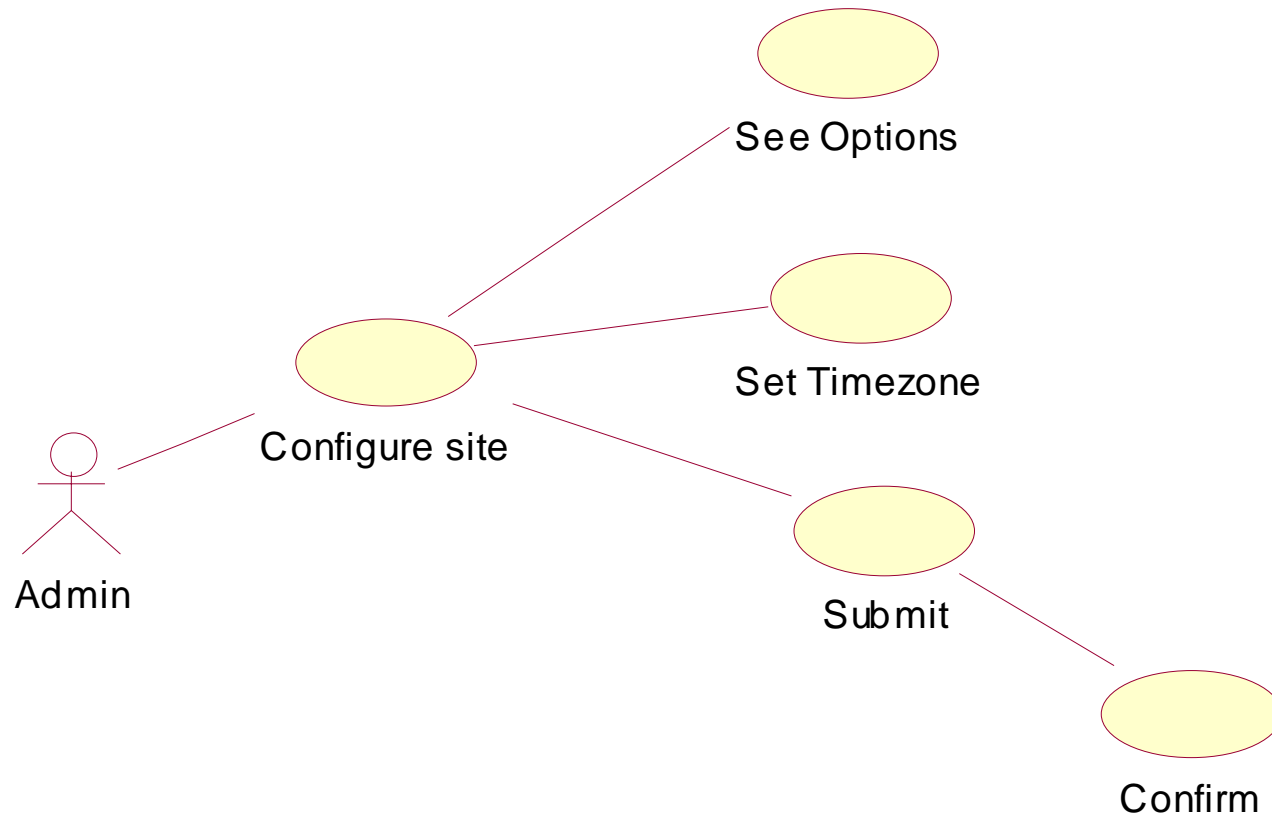


Overview

- Last week
 - Object diagrams
 - Package diagrams
 - Use cases and use case diagrams
- This week
 - Use case diagrams
 - Sequence diagrams

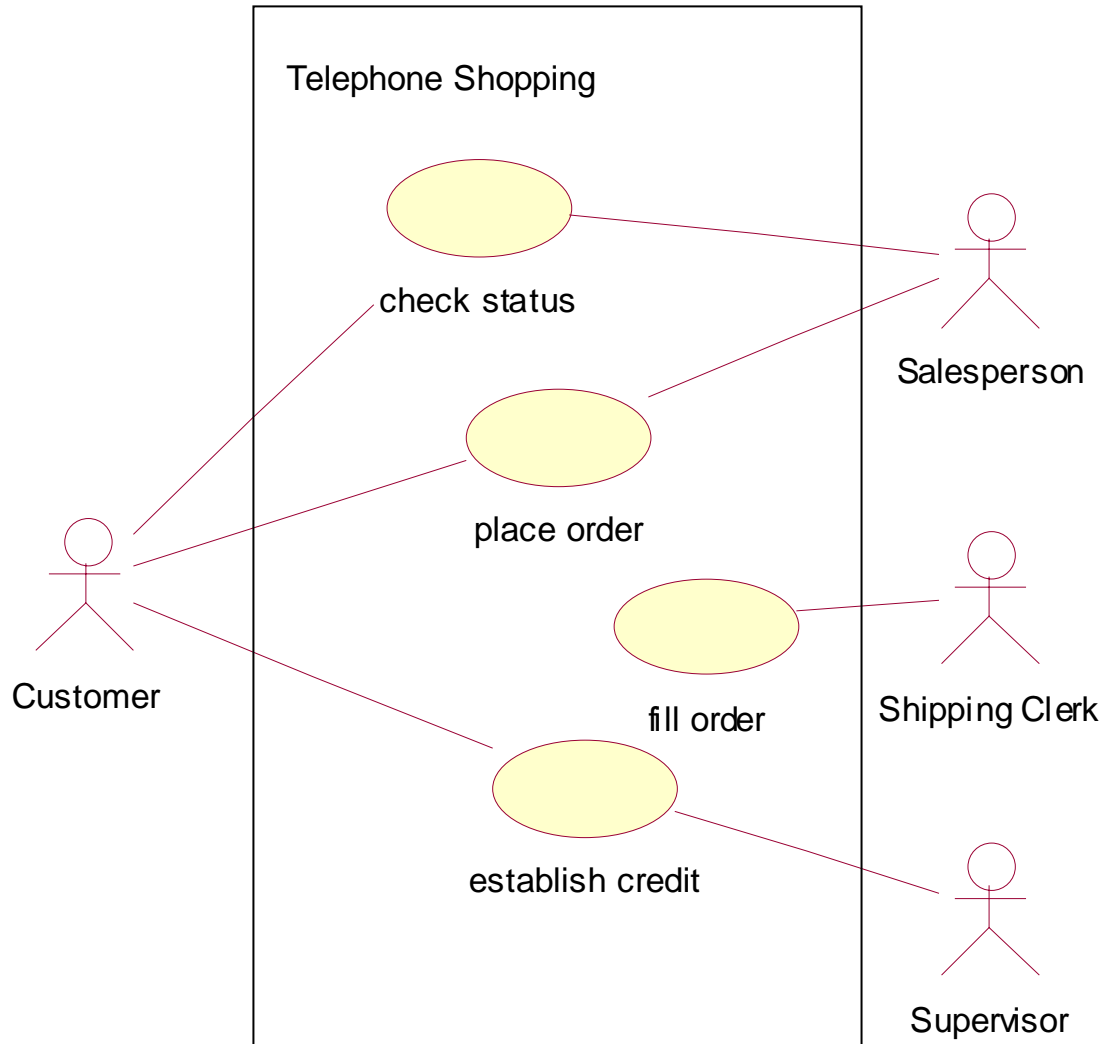
Use Case Diagram



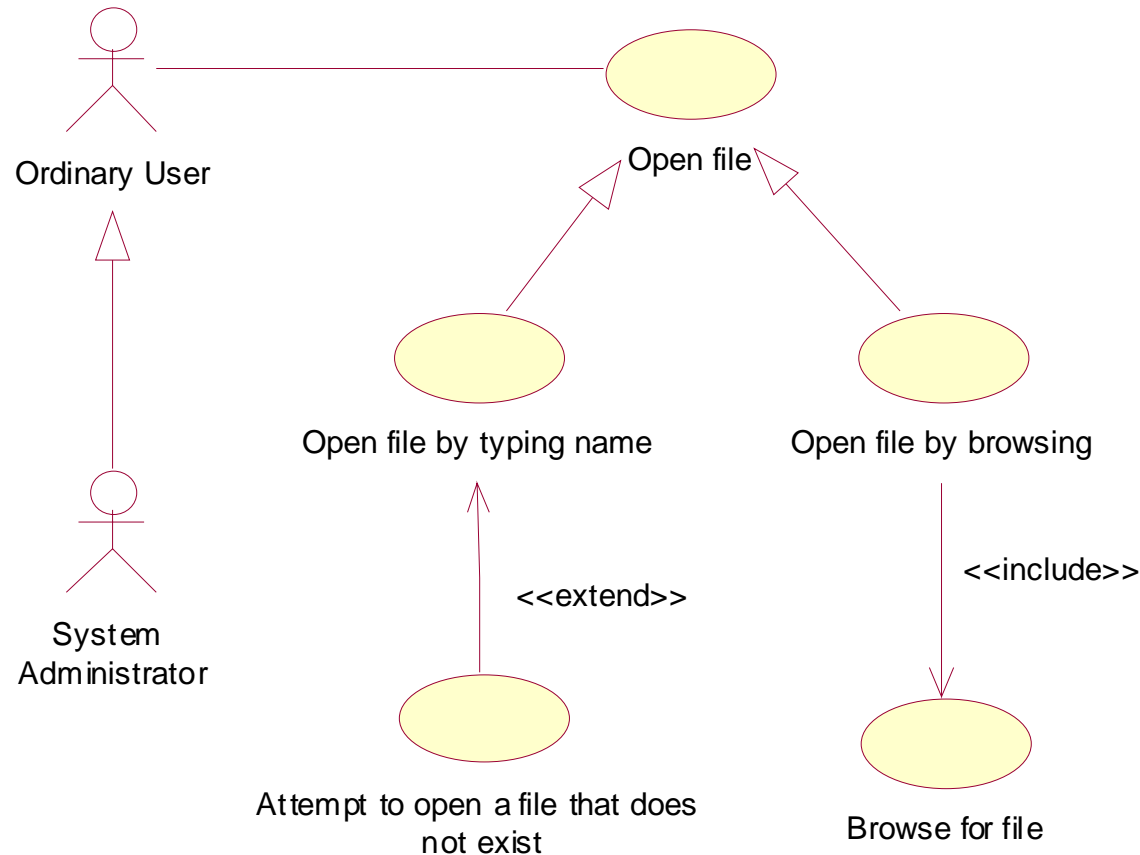
Elements of Use Case Diagrams

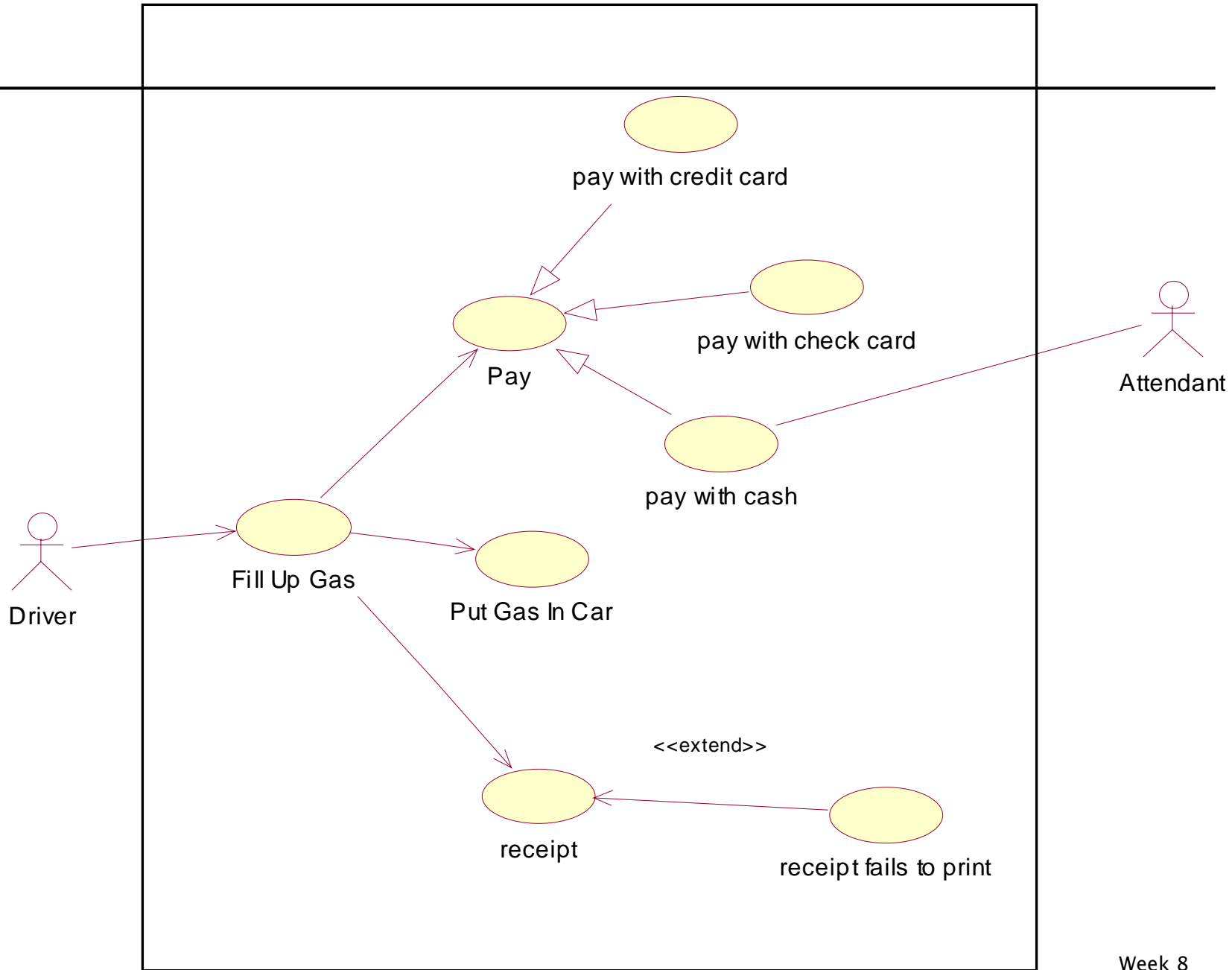
- The main purposes of a use case diagram:
 - Show all the names of the use cases, like a table of contents
 - Show relationships between actors and use cases
 - Show relationships between use cases
- Stick figure: Actor
- Oval: Use case
- Extension
 - Optional interactions to cover exceptions
- Inclusion
 - For common substeps; can be re-used in diagram
- Generalization
 - Like superclasses; for representing several similar use cases

Use Case Diagram

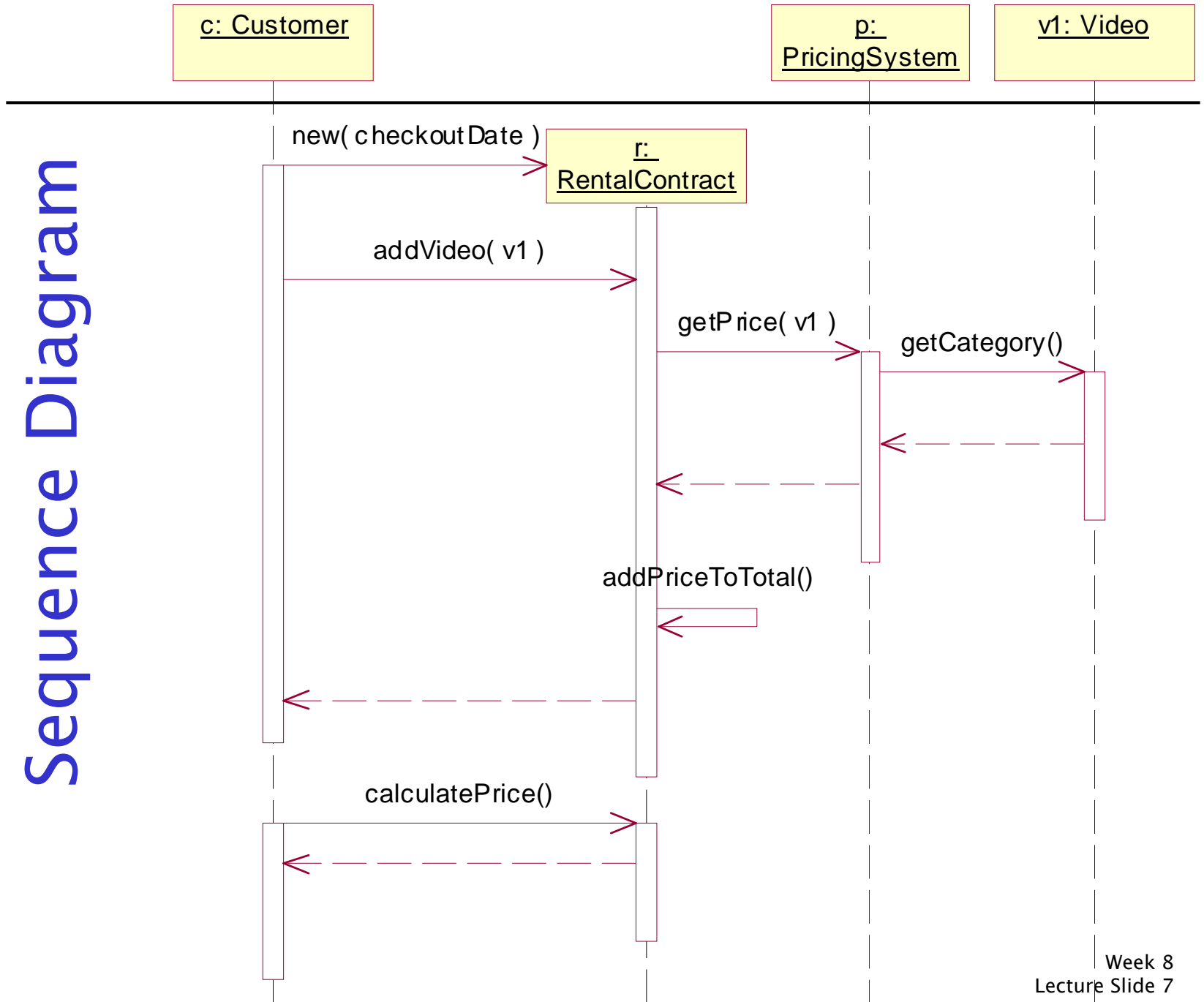


Example of generalization, extension and inclusion





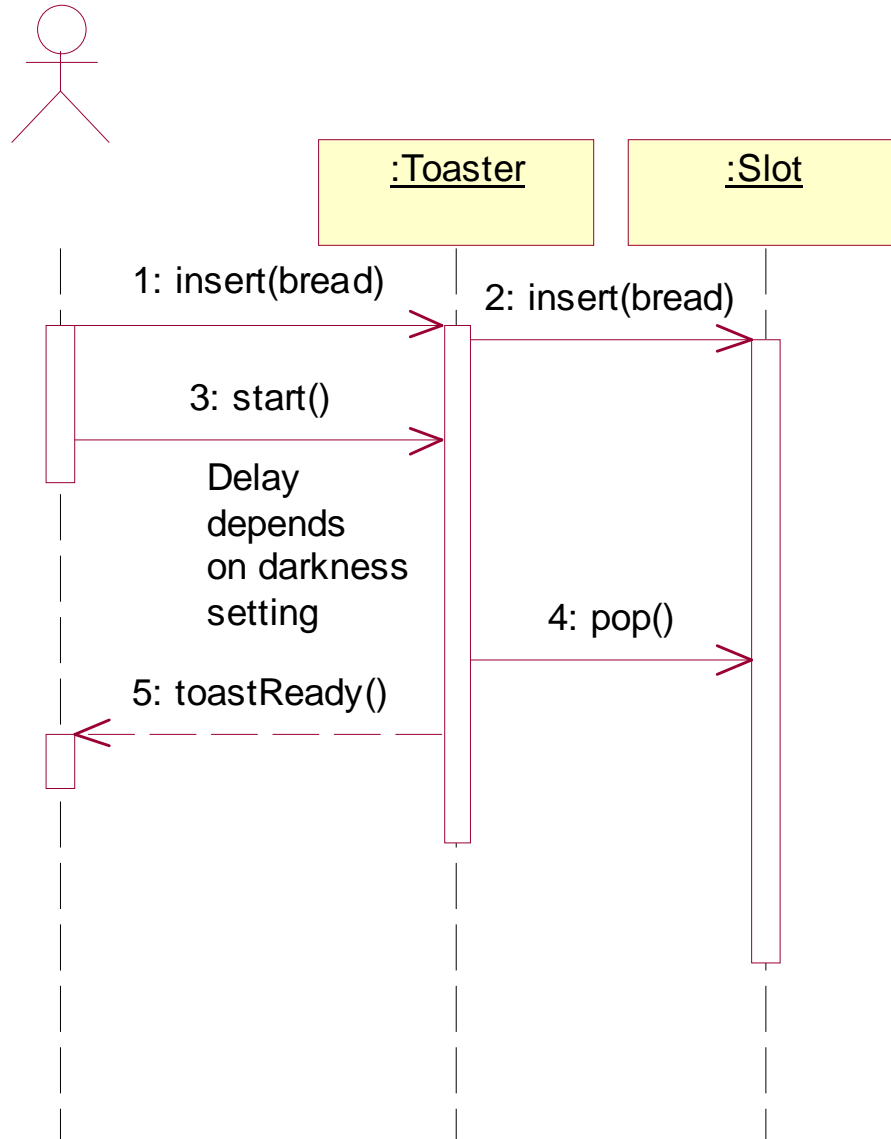
Sequence Diagram



Elements of Sequence Diagrams

- **Objects**
 - Life lines: time goes from top to bottom
 - May be several instances of one class
 - Boxes on life lines show if object is active
- **Messages**
 - Analogous to method calls in a program
 - Can have parameters
 - Special messages
 - new — shown by position of object
 - delete — shown through big X
 - Return messages
 - Self-calls
- **Additional syntax for asynchronous calls (concurrent)**

Making Toast



Canceling a Flight

