

CS 494



Object-Oriented Analysis & Design

Interaction Diagrams

- Examples of Collaboration and Sequence Diagrams

© 2001 T. Horton

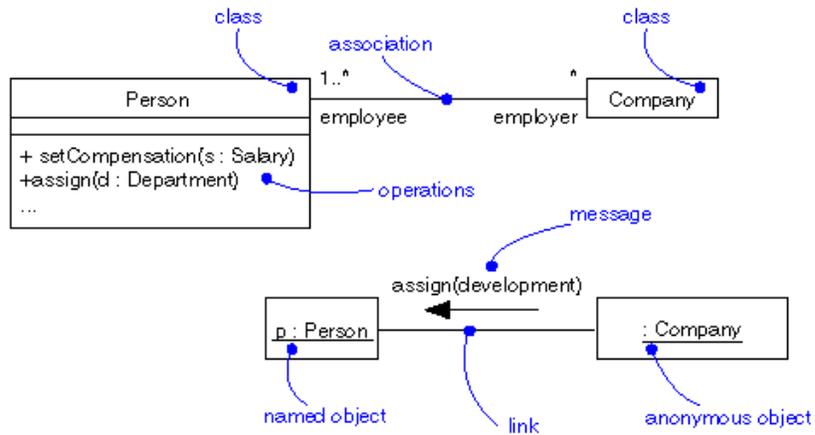
10/17/01 H-1

Dynamic Views in UML

- Class diagrams are models of data types
 - What non-fundamental types are you using? How are they related?
 - Sometimes referred to as the *static model*
- **Running** programs define objects of various types that interact
 - Control is passed between objects' methods
 - Information is passed and returned
- UML has two almost identical diagrams for this:
 - Collaboration diagram: object-centered
 - Sequence diagram: time-oriented
- Important: Each diagram models a particular scenario
 - Often we just model important or interesting scenarios

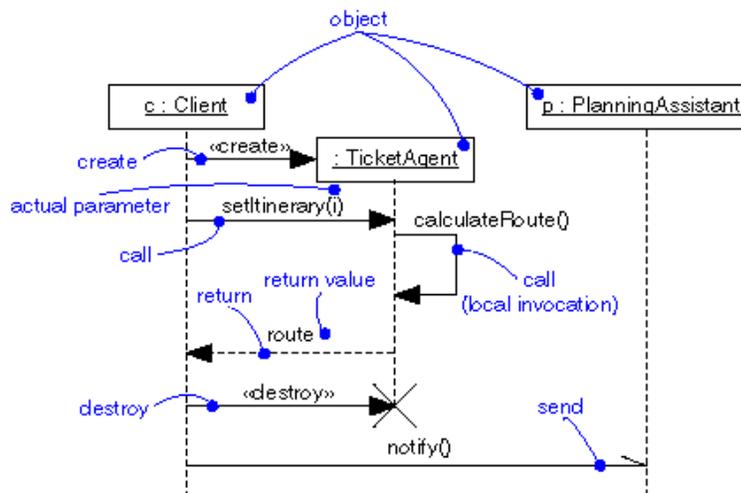
10/17/01 H-2

Class Diagram with a Related Collaboration



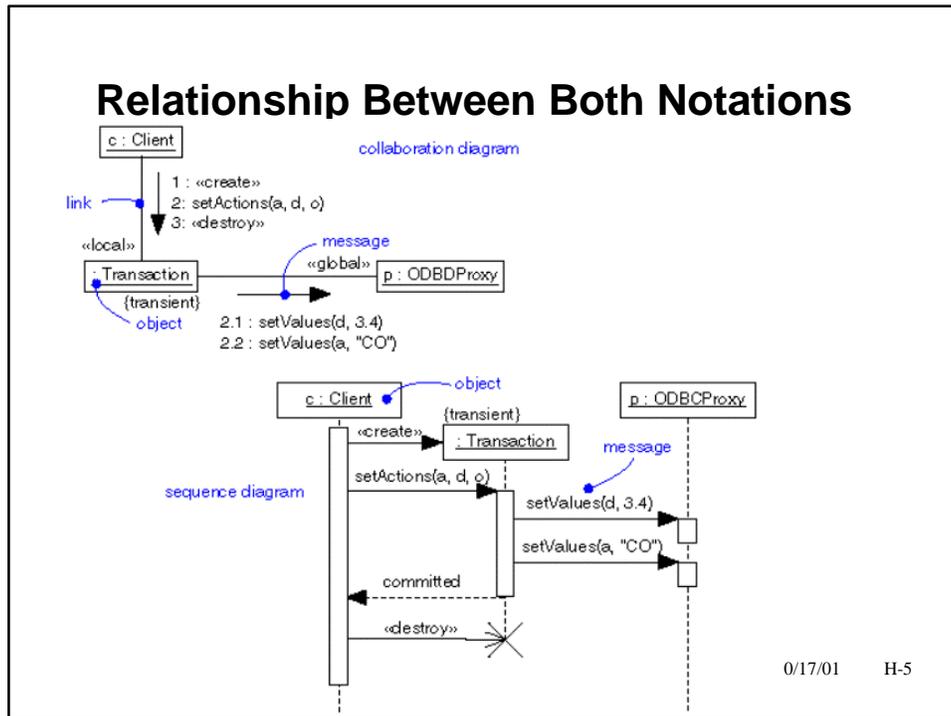
10/17/01 H-3

Sequence Diagram: Ex. 1

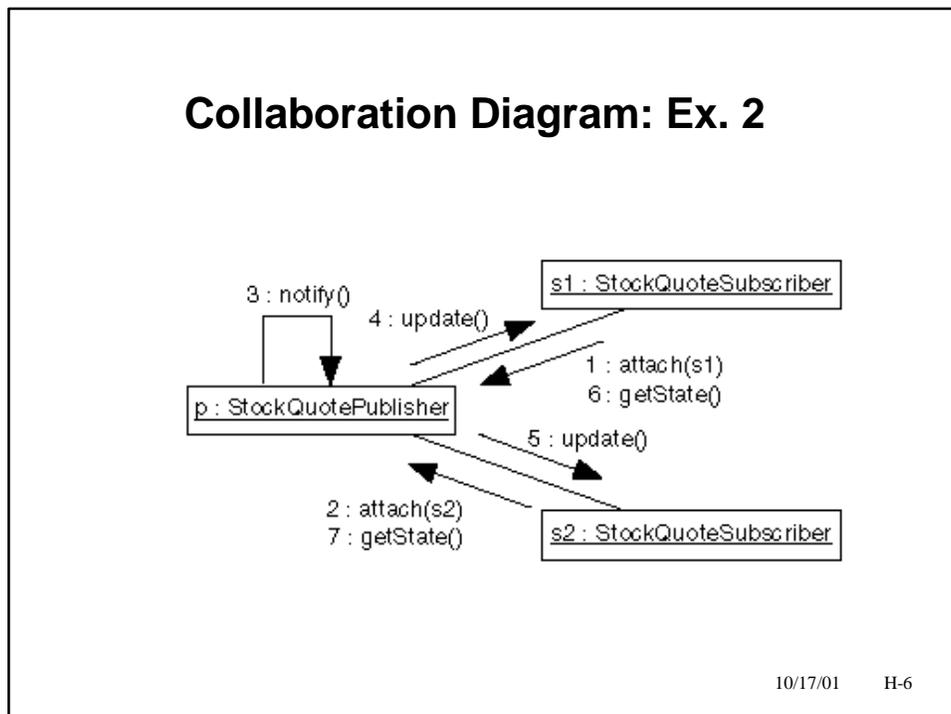


10/17/01 H-4

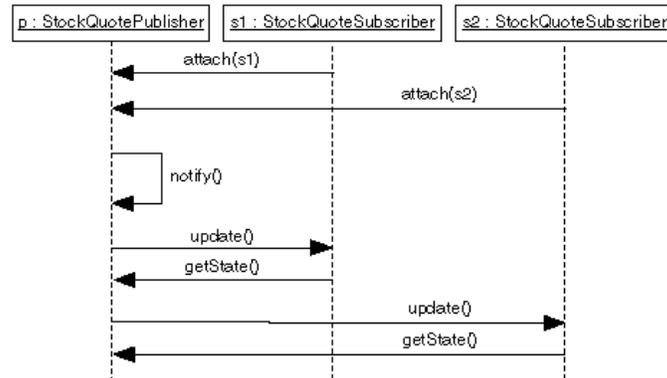
Relationship Between Both Notations



Collaboration Diagram: Ex. 2



Sequence Diagram: Ex. 2



10/17/01 H-7

Use Cases and Sequence Diagrams

- So far sequence diagrams model interactions between implementation-level objects
- Also, can be used to model use cases
 - Actor(s) interacting with the System
 - Most useful if more than one Actor
 - Messages are used informally here
 - Example: ATM customer tries to overdraw
- Also, sequence diagrams sometimes used in high-level design
 - model interactions between major subsystems

10/17/01 H-8

Iteration, Control, Return Values

- Messages can be labeled with a condition:
[hasStock] sellWidget()
- Messages can be repeated:
* msg() or *[k=1..2] msg()
- Return values:
 - Maybe on dashed “return” arrow, or
 - On message call: n = getName()
 - Again, note returns not always explicitly drawn

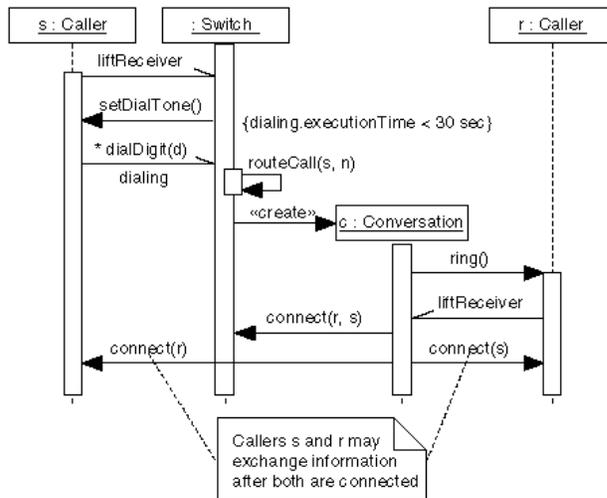
10/17/01 H-9

Notes on Messages

- Various types of messages supported
 - Filled solid arrow head:
 - like procedure calls, nested flow of control. Caller waits for action to complete.
 - “Half” arrow head:
 - Asynchronous flow of control. Caller is not blocked, continues to do something.
 - Dashed arrow:
 - Return from procedure call. May be omitted if it’s implicit at end of an activation box
 - Use stereotypes to define anything else.

10/17/01 H-10

Iterations, Constraints, Asynch. Messages



10/17/01 H-11

Phone Switch Example

- Optional for class lecture...

10/17/01 H-12

Phone Switch Example

- Classes
 - Phone: a person's telephone
 - Line: a "number" associated with a phone
 - Lines are busy, calls are made on lines, etc.
 - Connection: dynamically created, represents an active call between lines
 - Switch: a phone switch is a computer system that manages phones, lines and connections
 - Console: a terminal attached to the switch
 - Simulator: we're writing a simulation!

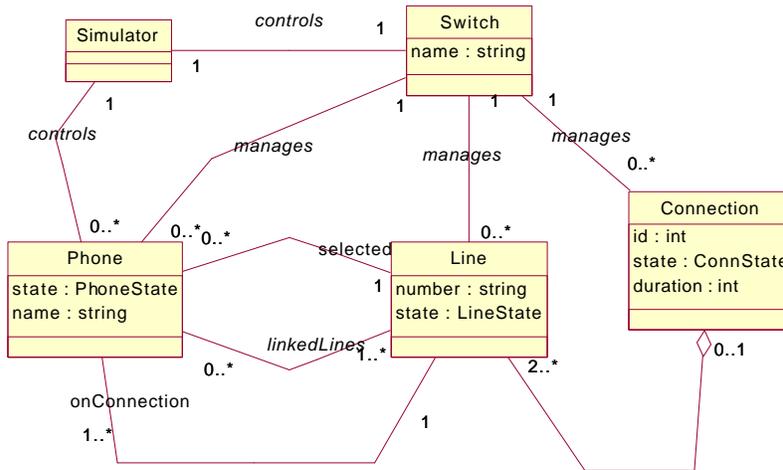
10/17/01 H-13

Associations in Our Model

- Switch manages other objects
- More than one Phone may be "linked" to one line
 - Like for office secretary, boss
- A Phone may have more than one Line
 - But, a Phone has only one Line "selected" at one time
 - Must choose a Line to call or answer
 - Also, when a Line is connected, not all Phones that can possibly use that Line may be participating in the call.
- A Connection requires at least two Lines to exist

10/17/01 H-14

Conceptual Class Diagram



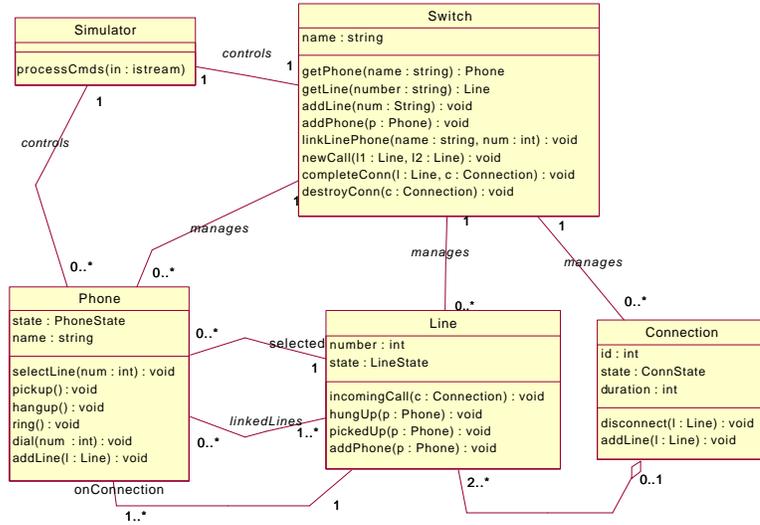
10/17/01 H-15

Scenario

- A normal call
 - Phone1 chooses Line 1 and picks up
 - Phone 1 dials number for Phone2
 - Phone 2 rings
 - Phone 2 picks up
 - Call completed and the two people talk
 - Phone 2 hangs up
 - Phone 1 is disconnected
 - Phone 1 hangs up

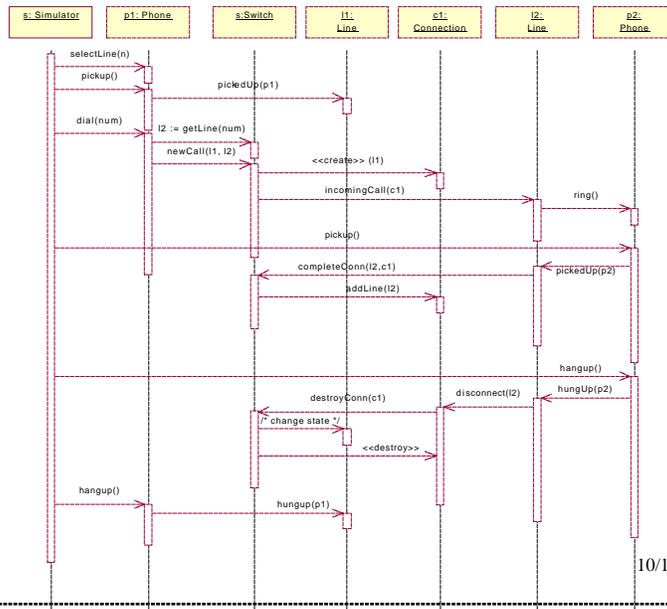
10/17/01 H-16

Class Diagram with Operations



10/17/01 H-17

Sequence Diagram for a Normal Phone Call



10/17/01 H-18