Collaboration and Deployment Diagrams

The last of our UML.
Collaboration Diagrams

Show **same** info as Sequence Diagrams
- Both show order of events.
  - Collaboration focuses on object “roles”
  - Sequence focuses on timing of messages

Objects which interact together to form some task, along with the links between them.

Collaboration looks more like class/object diagrams, but focuses on one particular chain of events.
Collaboration Example
(notice similarities to sequence example!)

1.1.1.*: roomAvailable():boolean

:Window

1: makeReservation():RoomNumber

:HotelChain

1.1: makeReservation():RoomNumber

iteration

aHotel:Hotel

1.1.1.*: roomAvailable():boolean

1.1.1: getRoom():Number

:Reservation

1.1.1.2: notify():void

:Confirm

1.1.1.2.1: tellHotel():void

1.1.1.2.2: tellOnlineCust():void

sequence number (see next page!)
Flow in a Collaboration Diagram

- Flow follows the arrows – imagine a token moving from object to object.
- If something is returned, it happens in the reverse order of the arrows (never drawn).
- Look at hotel example. Returns a room number. ALSO sends a confirmation.
Collaboration: Sequence Numbers

- Highest level message starts with number 1.
- Each call made from that level is numbered in order 1, 2, 3, etc.
- Then each call made from the next level adds a .1, .2, etc.
- Use .* for an iteration.
More on Sequence Numbers

Another way to think of it. When an object sends a message, it’s first message is numbered 1.

AFTER it receives a response, the object can send a second message with number 2.

Unless doesn’t wait for response, so is asynchronous. Remember how that would look?

- In that case, just number in the order that the messages were sent.
Another Collaboration Example

The Robot Pizza Company

:OnlineCustomer

1: orderPizza(): Pizza

1.1: turnOnAppliances()

1.2: cook()

:PizzaPlace

1.2.*: loadIngredients()

:D:Delivery

1.2.2: send()

:AutoPizzaCooker
Packages

Just like class diagrams. But hide related groups of **objects**.

1: checkCreditFunds()

2: placeOrder()
   {if funds}

:OrderOnline

:OnlineBank

RobotPizza Objects

package

constraint too!
Actors in Interaction Diagrams

Sometimes clearer if an actor starts an interaction.

RobotPizza Objects

1.1: checkCreditFunds()
1.2: placeOrder() {if funds}

1.2.1: deliver()

customer

:OrderOnline

:OnlineBank

package

constraint too!
Deployment Diagram (very quickly!)

- Used to show how software will be physically deployed on servers, databases, PCs, printers, etc.
  1. shows physical connections between hardware
  2. shows what software runs on each component
Deployment Example

astro: Server
  :GameEngine

(ClientPC)
  p1:PlayerInterface

:LAN >>

:Database Server
  :SQLScoreKeeper

<<ethernet>>

daisy: Printer
  :OurDriver
Other Types of UML Diagrams

Nine basic types.
- In approx. order of importance for a software engineer

- Class Diagram
- Use Case diagram
- Sequence Diagram
- Collaboration diagram
- Object Diagram
- Deployment Diagram

- Activity Diagram
- Statechart Diagram
- Component Diagram

 discussed elsewhere
(you know enough UML now to pick this up if you need it!)