BON

Dynamic Model

Based on slides
by Prof. Paige

Purpose of Dynamic Model

- Analysis and design should **not focus** on implementation
  - » **Static relationships & contracts do not change, minimize implementation dependence**

- Reasonable to want to ensure implementation is possible
  - » **For other considerations see ...**
    - – **Swartout, W., Balzer, R., *On the Inevitable Intertwining of Specification and Implementation*, Communications of the ACM, July 1982, Vol 25, No 7, pp. 438-440**

- Need a specification of **how** the classes **can fulfill** their specifications by calling routines of other classes

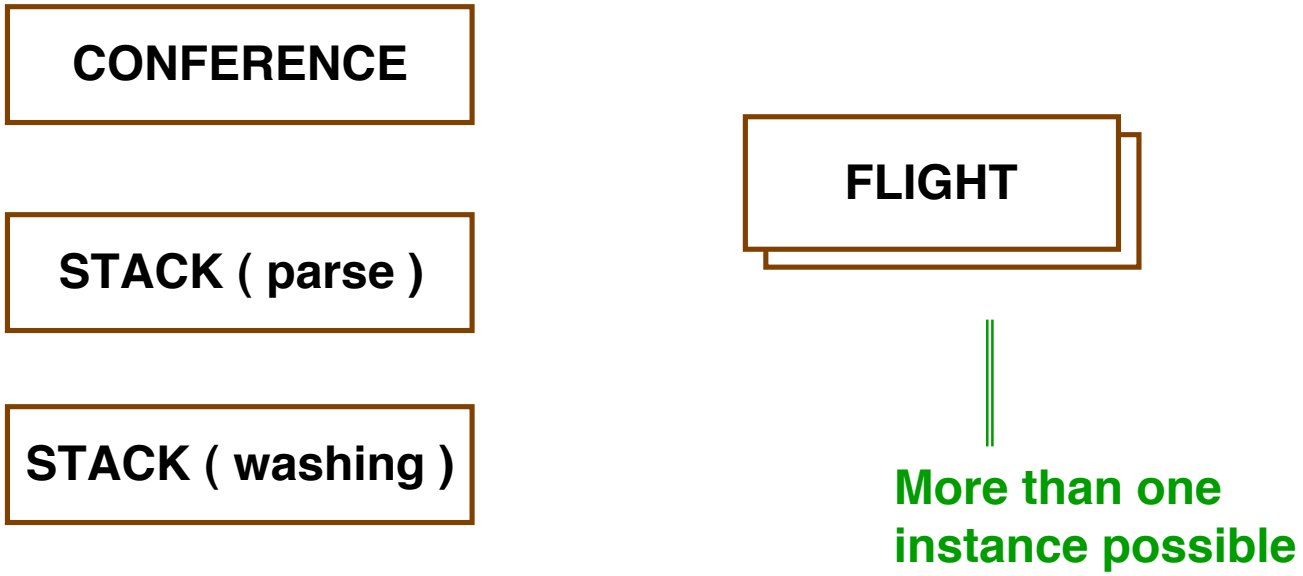
Dynamic Model

- What makes up a dynamic model in BON?
  - » **Feature calls – object communication**
    - » **Also known as message passing or object communication**
- Using feature calls in a dynamic model supports seamlessness
  - » **Feature calls map directly to a programming language**
- Some design methods use finite state machines to specify what an object does in reaction to a message
  - » **Difficult to translate, in all but simple machines, into programs – lack of seamlessness**

Object Representation

How do we describe objects in BON?

Use rectangles containing their class name with an optional qualifier (e.g. a name)



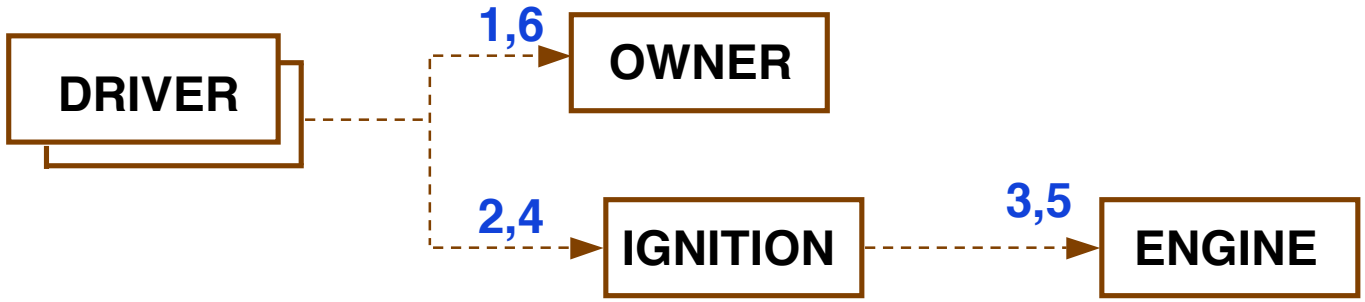
Communication Between Objects

- Pass or send a message, call a feature, invoke an operation – are all synonymous

- A message is indicated by a dashed arrow from the calling to the receiving object

Scenario with Object Communication

- Message links may be annotated with sequence numbers representing order of calls.
  - » **Cross reference to entries in a scenario box**



Scenario: Borrow car and go for a drive

1 Driver gets keys from owner
2 Driver turns ignition
3 Engine starts
4 Driver removes key
5 Engine stops
6 Driver returns keys to owner

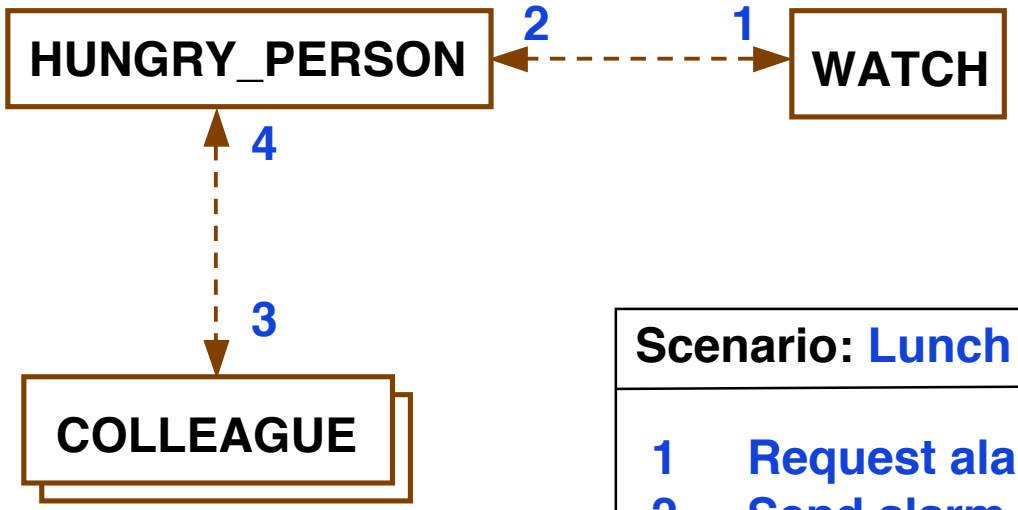
Communication Properties

- Message are always **potential**
  - » **They do not have to occur – Flat battery**

- Group as for clusters

Bidirectional Communication

- A set of message relations in each direction between two objects



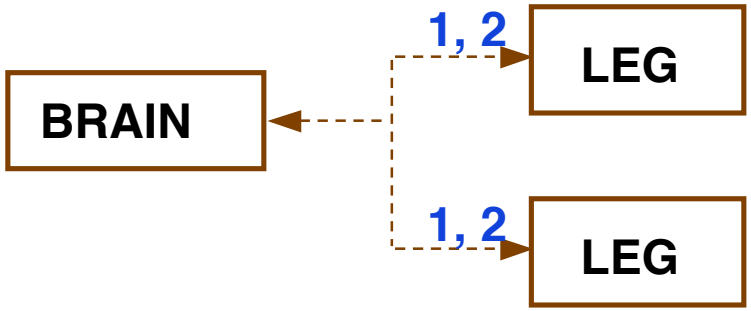
Scenario: Lunch time

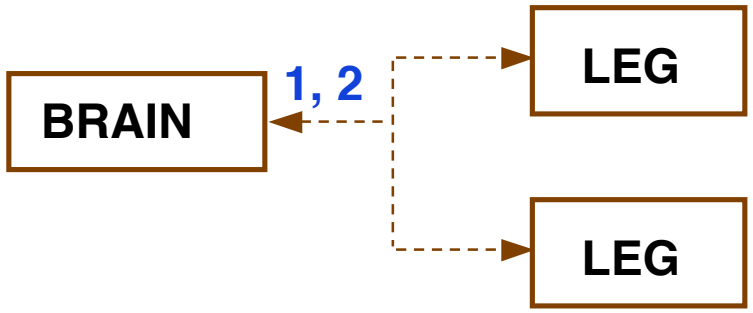
1 Request alarm at noon
2 Send alarm, time for lunch
3 Person tells colleagues
4 Colleagues respond

Send / Receive to Many Objects

» Broadcast
Label close to sender means send to all receivers

» Send to one instance only
Label close to receiver means receive from all senders



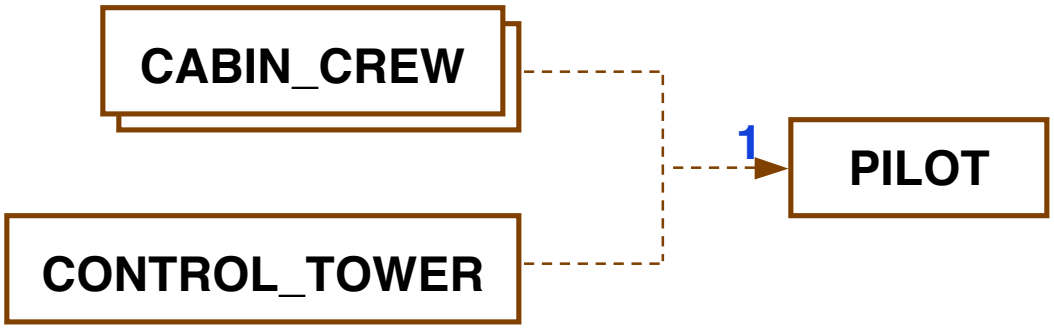


Scenario: Get coffee	
1	Brain – go forward alternate legs
2	Leg – bump door

Scenario: Jump	
1	Brain – go forward both legs
2	Leg – land

Joining Concurrent Messages

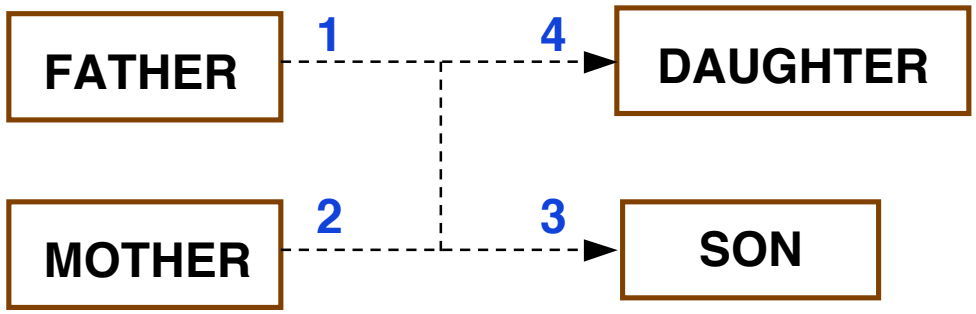
- Here the pilot receives messages from the cabin crew and the control tower



Scenario: Get ready for takeoff
1 Pilot receives "seat belts fastened" from cabin crew, "cabin doors secured" from crew and "runway clear" from control tower

Family Communications

- Messages 1 and 2 go to both receivers
  Messages 3 and 4 come from both senders



Scenario: Farewell at train station

1 Father gives children an extra $50 each
2 Mother gives children boxed lunches
3 Parents tell son never to ski out of bounds
4 Parents tell daughter what men are really after