

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)

CONFERENCE

STACK (parse)

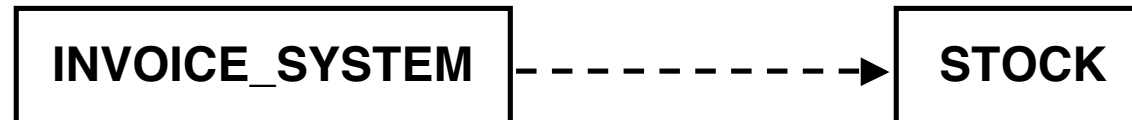
STACK (washing)

FLIGHT

More than one
instance possible

Communication Between Objects

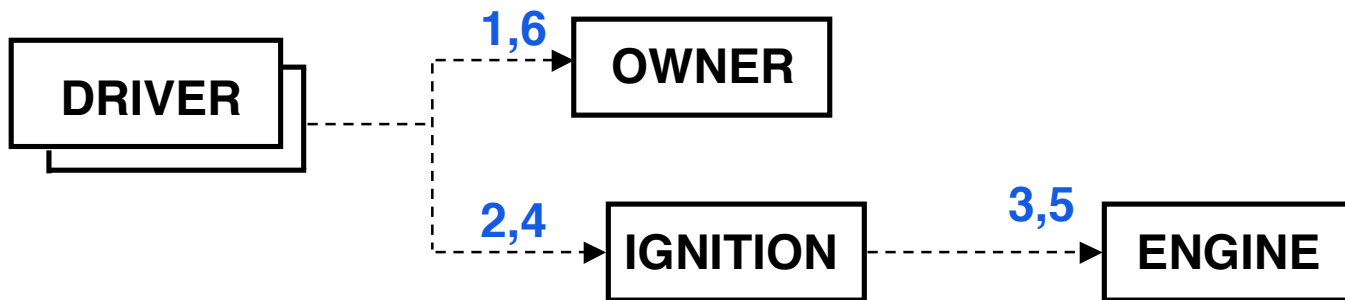
- **Pass or send a message, call a feature, invoke an operation – are all synonymous**
- **A message is indicated by a dashed arrow from the calling to the receiving object**



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

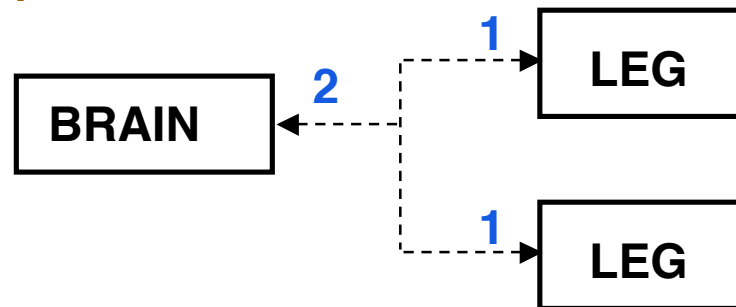
- 1 Driver gets keys from owner
- 2 Driver turns ignition
- 3 Engine starts
- 4 Driver removes key
- 5 Engine stops
- 6 Driver returns keys to owner

Communication Properties – 1

- Message are always **potential**
 - » **They do not have to occur – Flat battery**
- Bidirectional messages are possible
- Group as for clusters

Communication Properties – 2

- Can specify concurrency
 - » **send/receive to many objects – broadcast, or to one instance only (differentiate with note, if not obvious)**



Scenario: Get coffee	
1	Step forward – alternate legs
2	Bump door

Scenario: Jump	
1	Step forward – both legs
2	Land