

INTERACTIONS

- An interaction is a behavior that is composed of a set of messages exchanged among a set of objects within a context (it can be class, an operation, classifier, system or subsystem) to accomplish a purpose.
- A message specifies the communication between objects for an activity to happen. It has following parts: its name, parameters (if any), and sequence number. Figure 1: Messages, Links, and Sequencing.
- objects in an interaction can be concrete things (represents something in the real world.) or prototypical things (any instance of Person- like a professional)

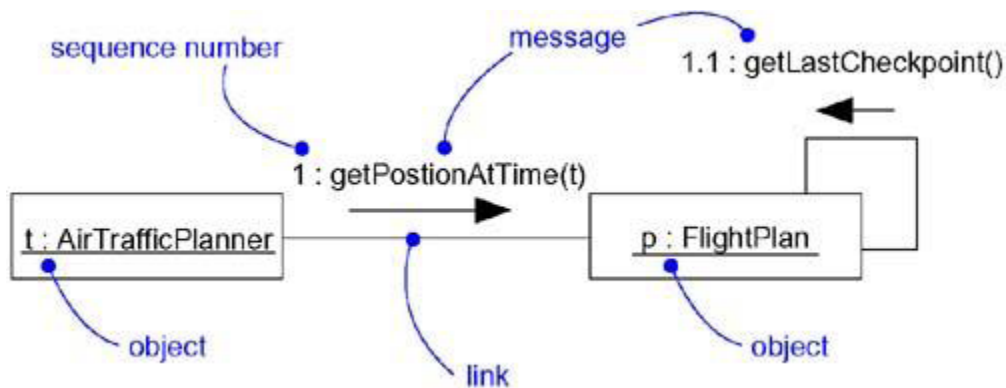


Figure 1 Messages, Links, and Sequencing

A link is a semantic connection (path) among objects through which a message/s can be sent. In general, a link is an instance of an association. Figure 2: Links and Associations. The semantics of link can be enhanced by using following prototypes as adornments

- `<<association>>` – Specifies that the corresponding object is visible by association
- `<<self>>` – Specifies that the corresponding object is visible because it is the dispatcher of the operation
- `<<global>>` – Specifies that the corresponding object is visible because it is in an enclosing scope
- `<<local>>` – Specifies that the corresponding object is visible because it is in a local scope

- <<parameter>> – Specifies that the corresponding object is visible because it is a parameter

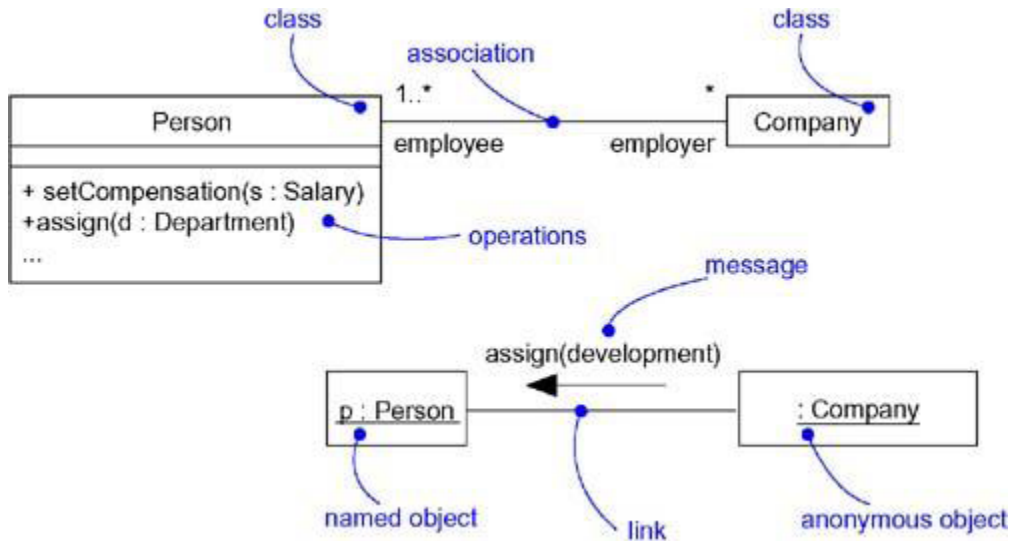


Figure 2: Links and Associations

message – indicates an action to be done. Complex expressions can be written on arbitrary string of message. Different types of messages are,

- Call -Invokes an operation on an object; an object may send a message to itself, resulting in the local invocation of an operation
- Return – Returns a value to the caller
- Send – Sends a signal to an object
- Create – Creates an object
- Destroy – Destroys an object; an object may commit suicide by destroying itself

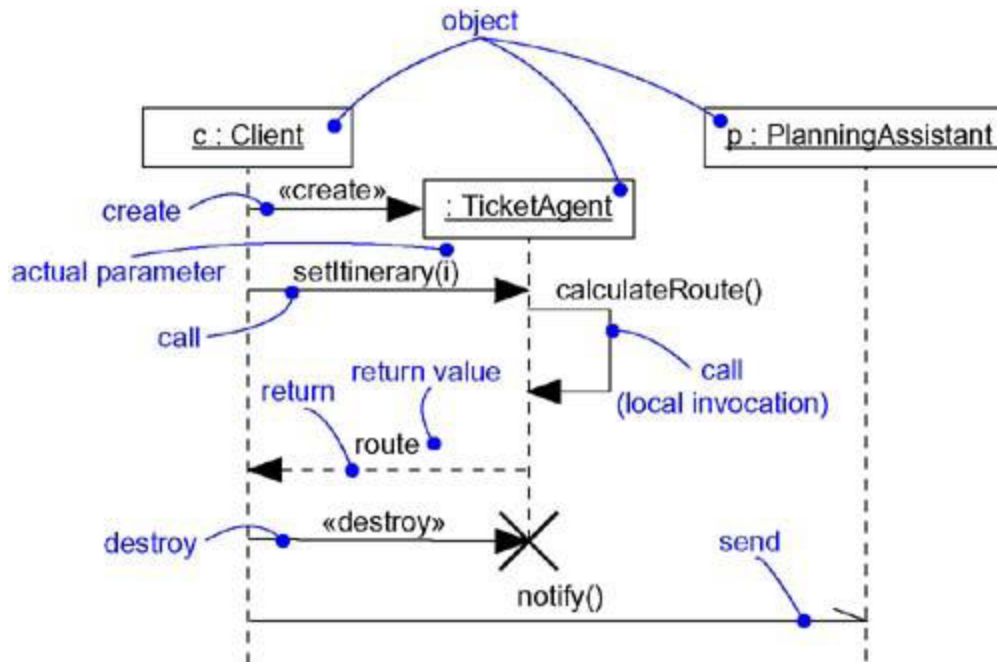


Figure 3 Messages

Sequencing

- a sequence is a stream of messages exchange between objects
- sequence must have a beginning and is rooted in some process or thread
- sequence will continue as long as the process or thread that owns it lives

Flow of control (2 types)

In each flow of control, messages are ordered in sequence by time and are visualized by prefixing the message with a sequence number set apart by a colon separator

A procedural or nested flow of control is rendered by using a filled solid arrowhead, as in Figure 4: shows Procedural Sequence

A flat flow of control is rendered by using a stick arrowhead as in Figure 5: Flat Sequence

Distinguishing one flow of control from another by prefixing a message's sequence number with the name of the process or thread that sits at the root of the sequence more-complex forms of sequencing, such as iteration, branching, and guarded messages can be modeled in UML.

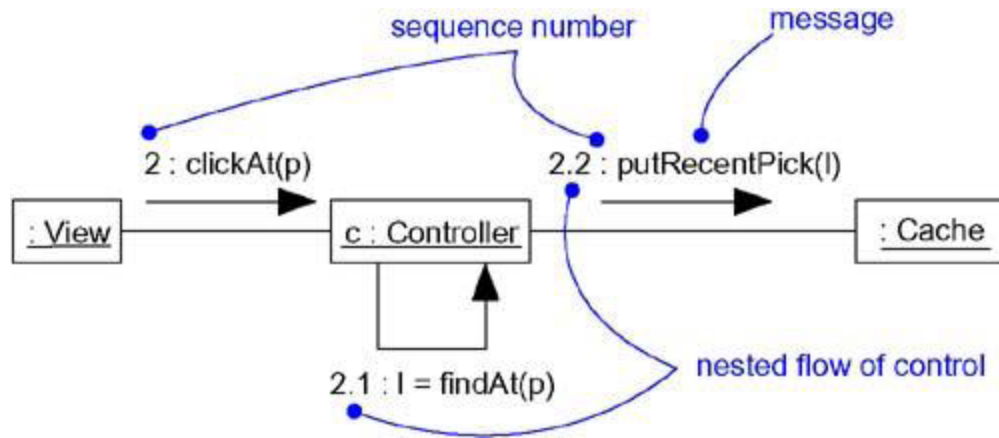


Figure 4: Procedural Sequence

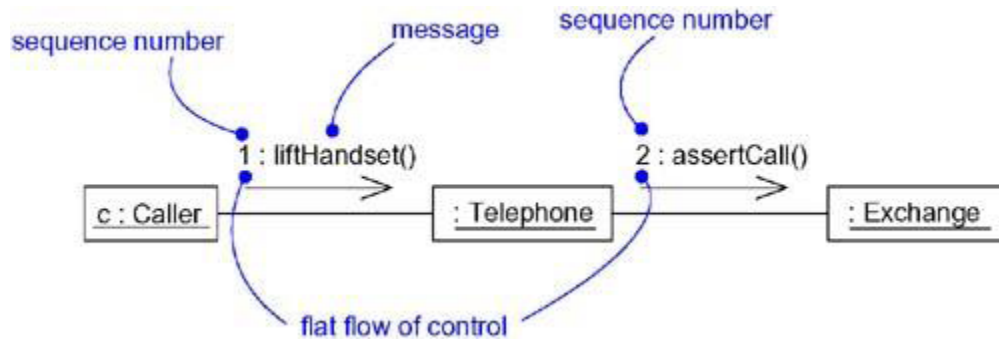


Figure 5: Flat Sequence

Creation, Modification, and Destruction of links

Enabled by *adding the following constraints to the element*

- new – Specifies that the instance or link is created during execution of the enclosing interaction
- destroyed – Specifies that the instance or link is destroyed prior to completion of execution of the enclosing interaction
- transient – Specifies that the instance or link is created during execution of the enclosing interaction but is destroyed before completion of execution

Representation of interactions

interaction goes together with objects and messages

represented by time ordering of its messages (sequence diagram), and by emphasizing the structural organization of these objects that send and receive messages (collaboration diagram)

Modeling a Flow of Control

To model a flow of control,

- Set the context for the interaction, whether it is the system as a whole, a class, or an individual operation
- Set the stage for the interaction by identifying which objects play a role; set their initial properties, including their attribute values, state, and role
- If your model emphasizes the structural organization of these objects, identify the links that connect them, relevant to the paths of communication that take place in this interaction Specify the nature of the links using the UML's standard stereotypes and constraints, as necessary
- In time order, specify the messages that pass from object to object As necessary, distinguish the different kinds of messages; include parameters and return values to convey the necessary detail of this interaction
- Also to convey the necessary detail of this interaction, adorn each object at every moment in time with its state and role

For example, Figure 6: shows Flow of Control by Time which is a set of objects that interact in the context of a publish and subscribe mechanism

Figure 7: Flow of Control by Organization is shown below for the same context given above.

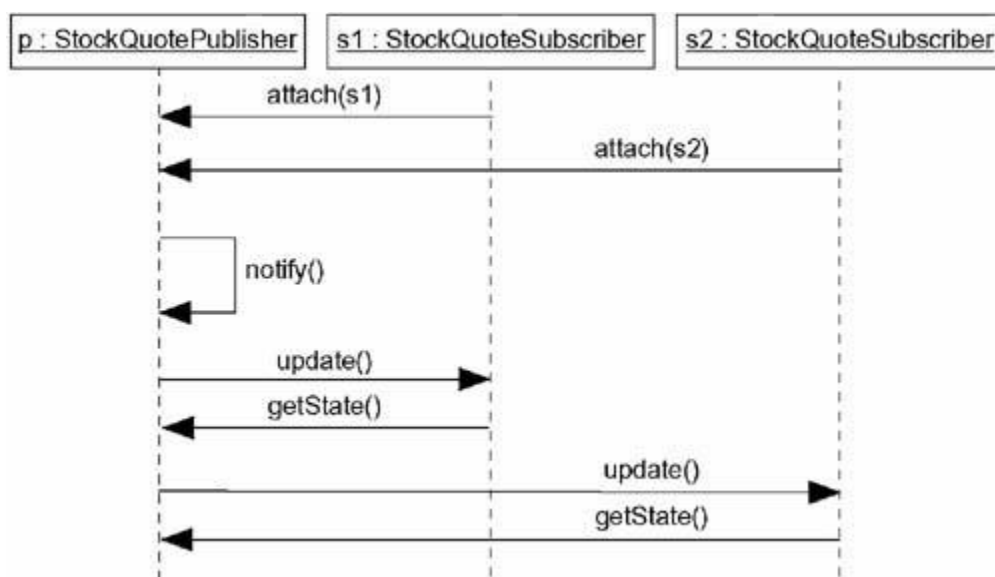


Figure 6: Flow of Control by time

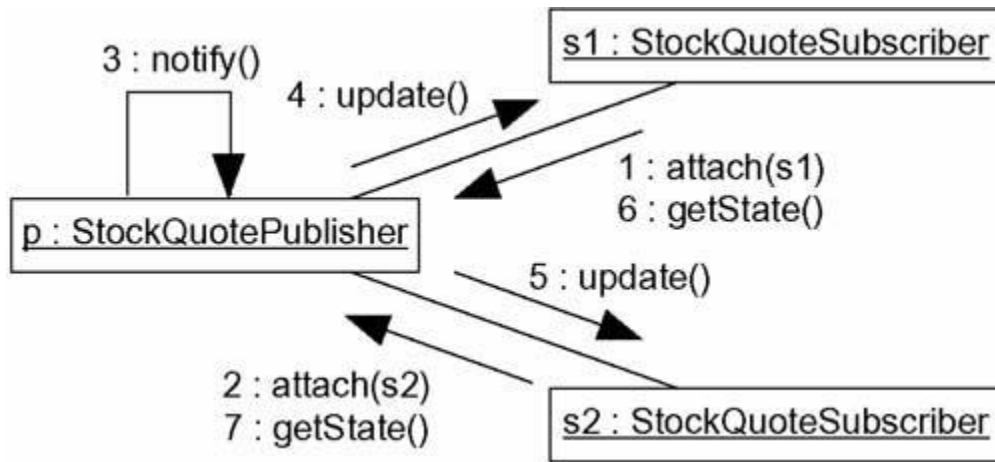


Figure 7: Flow of Control by Organization

Source : <http://praveenthomasln.wordpress.com/2012/04/05/interactions-s8-cs/>