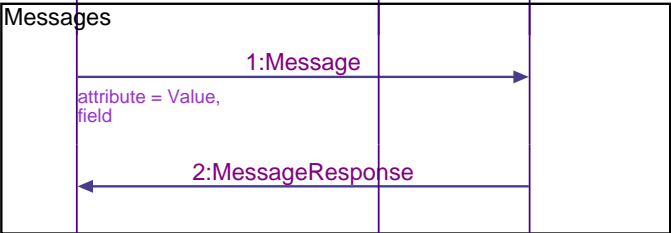




FDL Basics Tutorial

A block remark is shown across the full sequence diagram. A block remark may be shown across multiple lines.

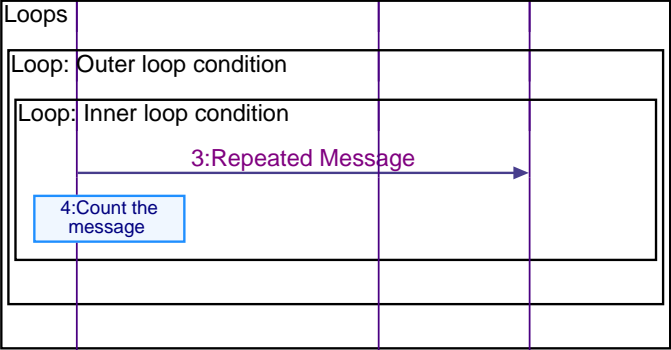
Messages



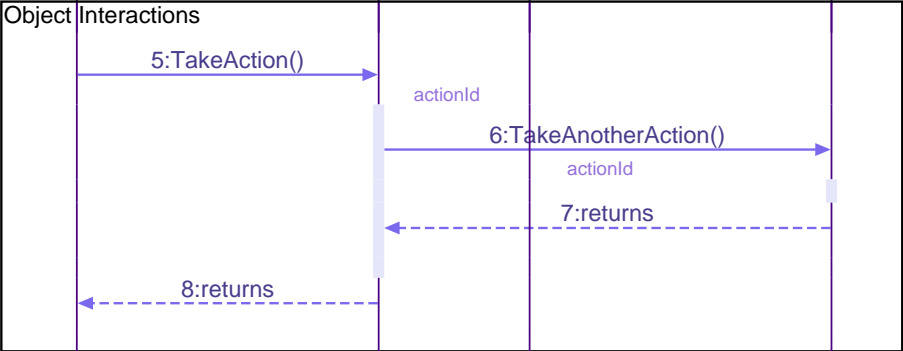
Message Statement: Model messages interactions with parameters.

Message Statement: Model messages interactions (no parameters specified in this case).

Loops

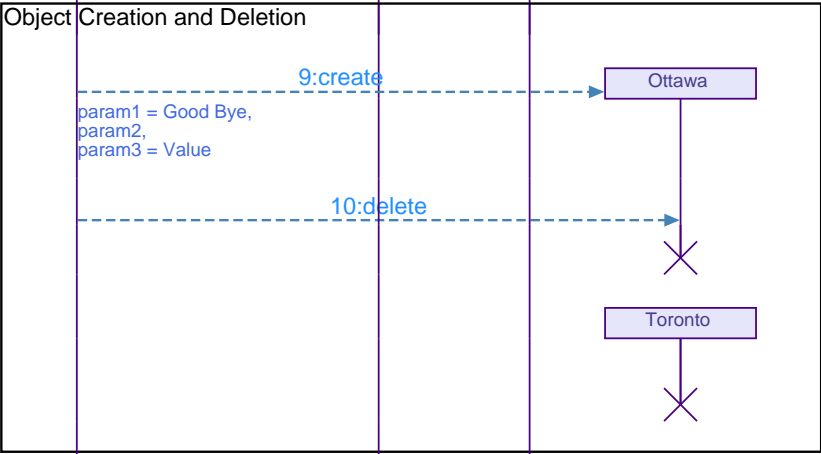


Object Interactions



Model method invocation and return.

Object Creation and Deletion



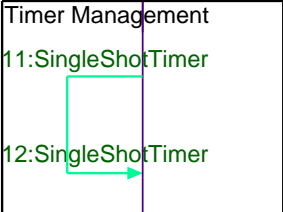
One object creates another.

One object deletes another.

Create object without specifying the creator. Use in sharing the same axis spot between multiple entries.

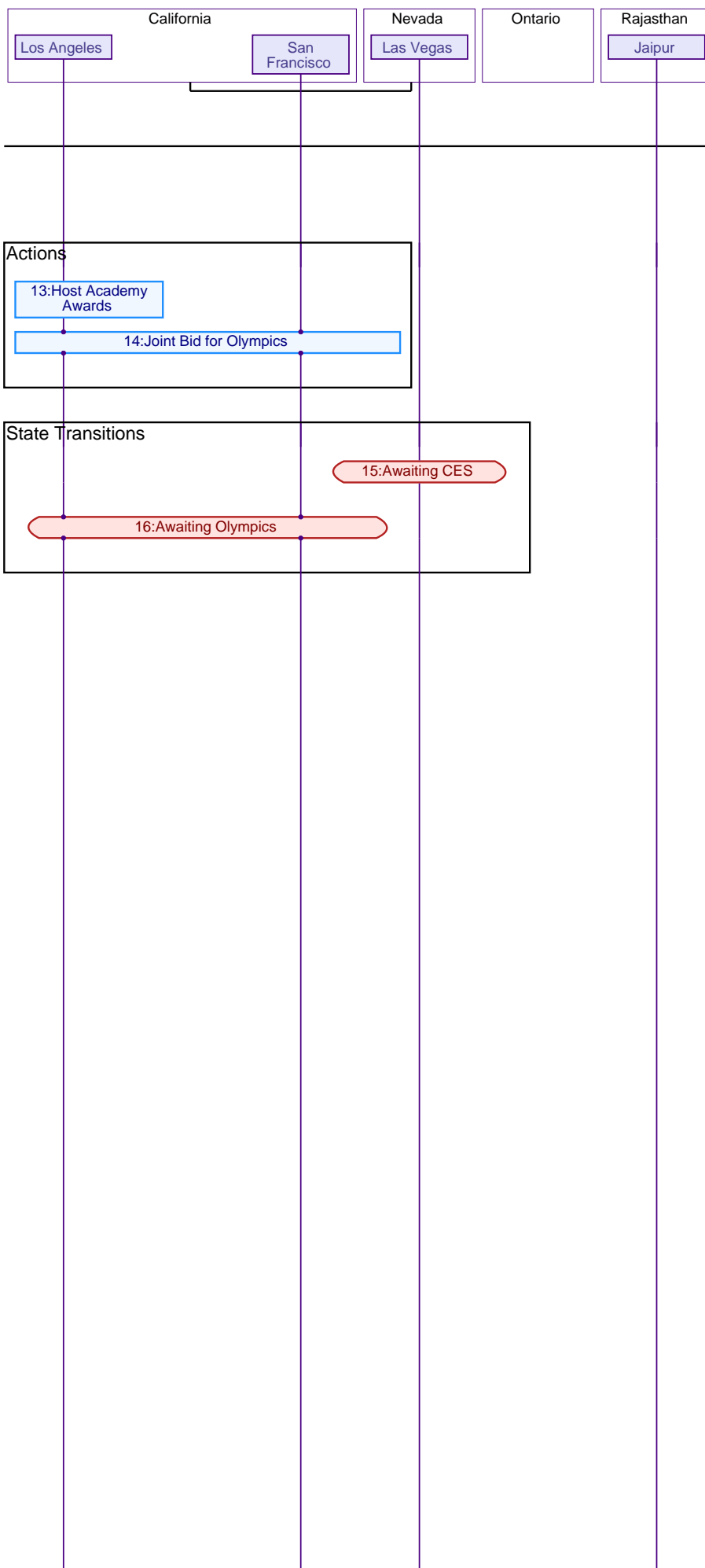
Delete the object. The axis may be reused by another object.

Timer Management



Represent a timer start.

Represent a timeout.



ISSUE: Why is 42 the answer to the ultimate question of life the universe and everything?

Model actions taken by an object.

Model joint actions taken by multiple objects.

Change the state of an object.

Change the joint state of two objects.



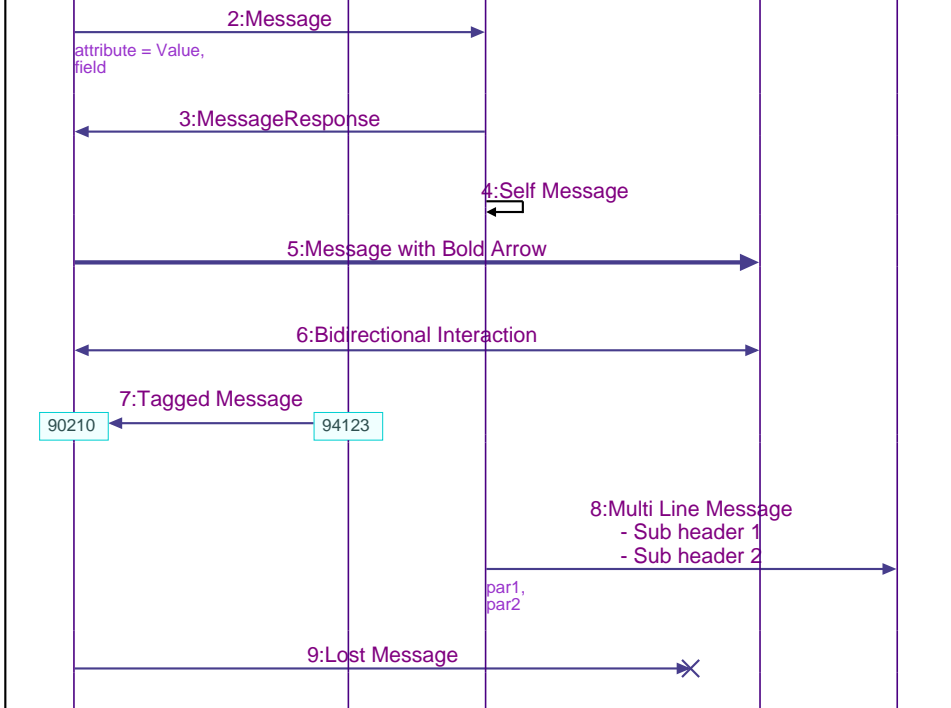
FDL Advanced Tutorial

Preconditions



A block remark is shown across the full sequence diagram. A block remark may be shown across multiple lines.

Messages



Message Statement: Model messages interactions with parameters.

Message Statement: Model messages interactions (no parameters specified in this case).

Model messages to self.

Use the **=>** or **<=** to represent messages with bold arrows.

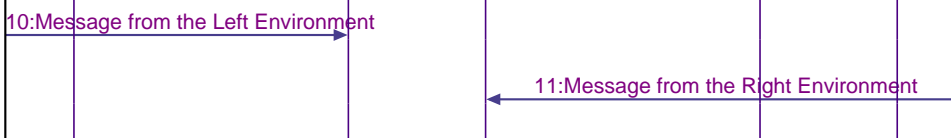
Model bi-directional message interactions with **<->** or **<=>**.

Tag the source and/or the destination of a message to identify the entity interacting at the source/destination.

Represent compound messages with the multi-line message syntax.

Model a lost message.

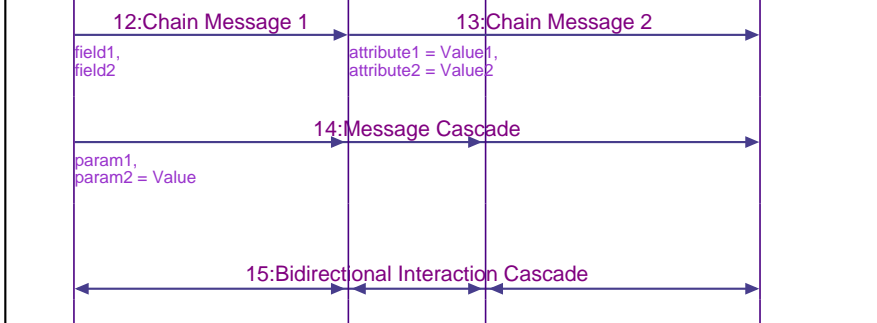
Message interactions with the environment



Represent a message from an external entity (shown on the left).

External interaction from an external entity (shown on the right side).

Compound Messages

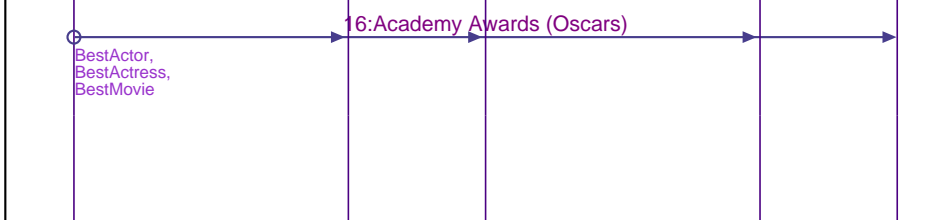


Represent a chain of message interactions in a single line. Separate message names and parameters may be specified for each message interaction.

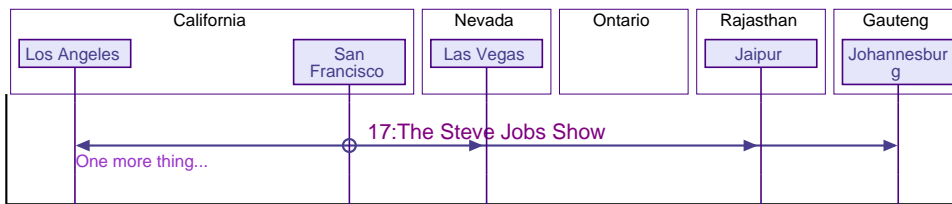
Represent a chain of message interactions involving forwarding of the same message. One set of message name and parameters may be specified.

Cascades work for bidirectional interactions as well.

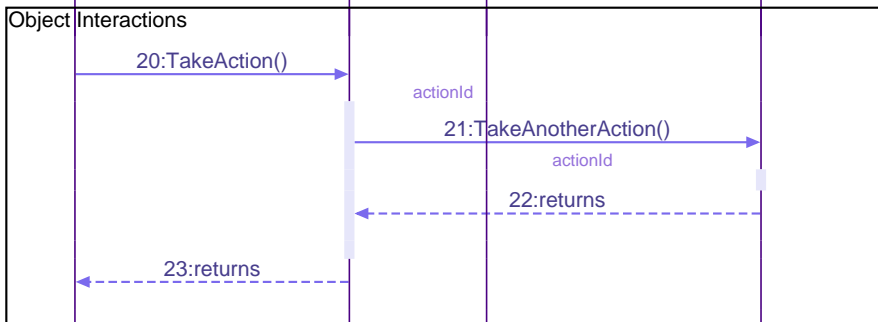
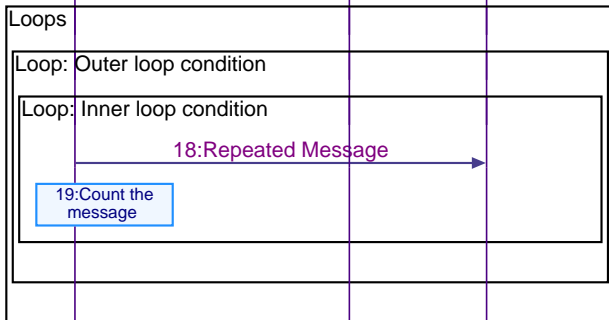
Multicasts



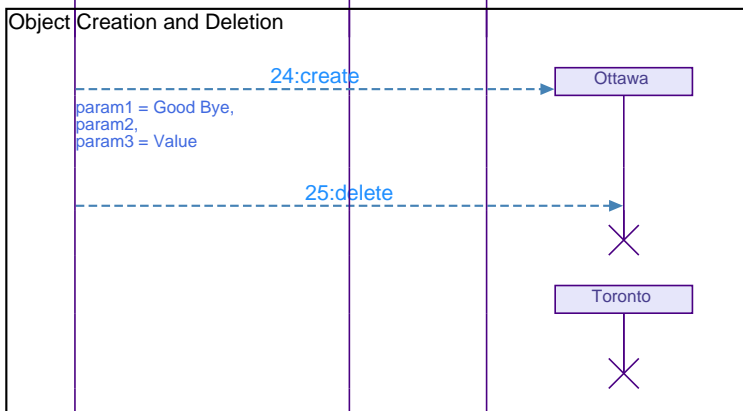
Model multicasts using this statement. The multicast sources is shown with a circle.



Another multicast. This time the multicast source is not at the edge.



Model method invocation and return.

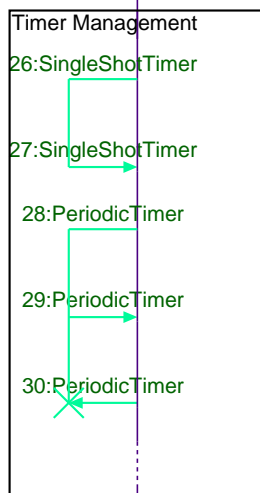


One object creates another.

One object deletes another.

Create object without specifying the creator. Use in sharing the same axis spot between multiple entries.

Delete the object. The axis may be reused by another object.



Represent a timer start.

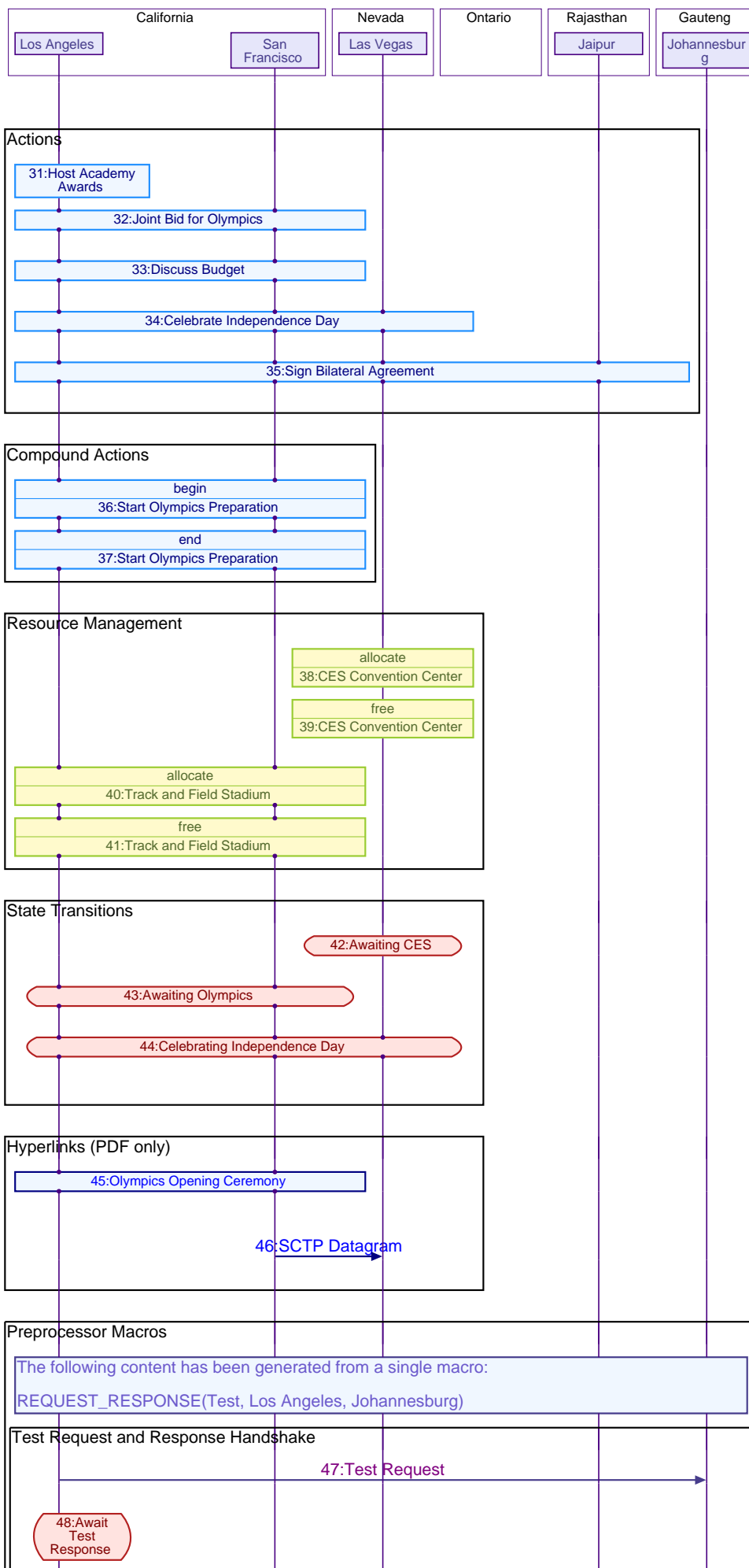
Represent a timeout.

Start a periodic timer.

Timeout for a periodic timer. The timer stays active even after a timeout.

Model a timer stop. Works for periodic as well a single shot timers.

Represent the passage of time with a dotted axis.



ISSUE: Why is 42 the answer to the ultimate question of life the universe and everything?

Model actions taken by an object.

Model joint actions taken by multiple objects.

Model actions at component level.

A module level action. In this case, all entities within USA are included.

Action involving multiple modules.

Model an action start.

Model an action end.

Model resource allocation.

Free the resource. EventStudio will warn you if a allocated resource is not freed.

A joint resource allocation.

A joint resource free.

Change the state of an object.

Change the joint state of two objects.

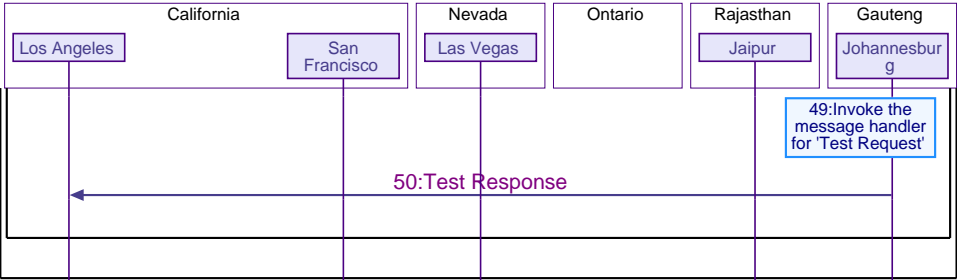
State specified at module level. The state transition will be included in module level and component level diagrams as well.

Click on action box to visit the Olympics website. A different style has been applied with the style prefix `[_hyperlink]`.

Click on the message name to get details about the message. `[_hyperlink]` style prefix has also been applied.

Test Request sent from Los Angeles to Johannesburg.

Los Angeles awaits Test Response.



Invoke the message handler Test Request.

Acknowledge message.

