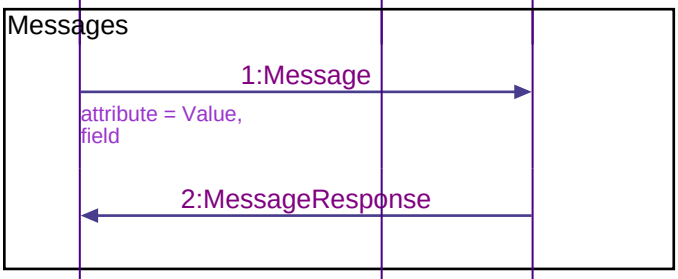
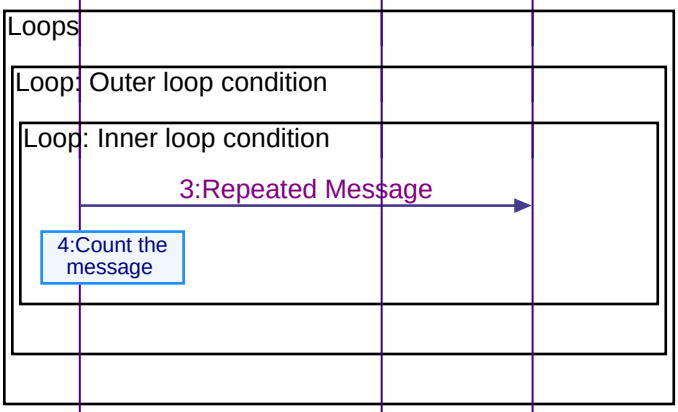
FDL Basics Tutorial

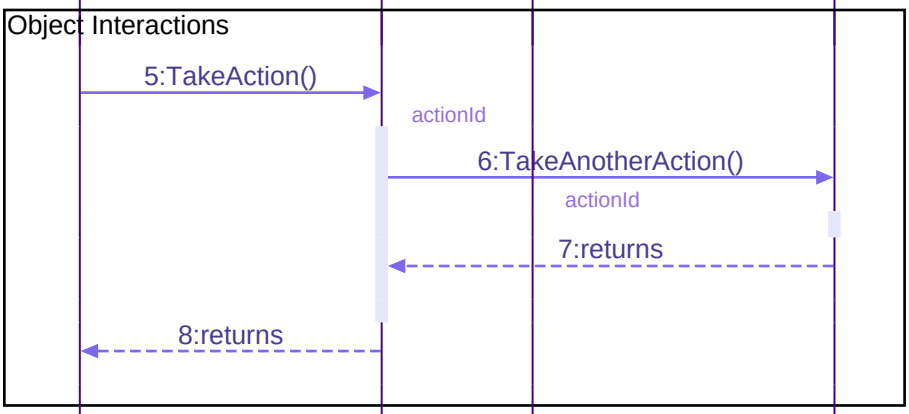
A block remark is shown across the full sequence diagram. A block remark may be shown across multiple lines.



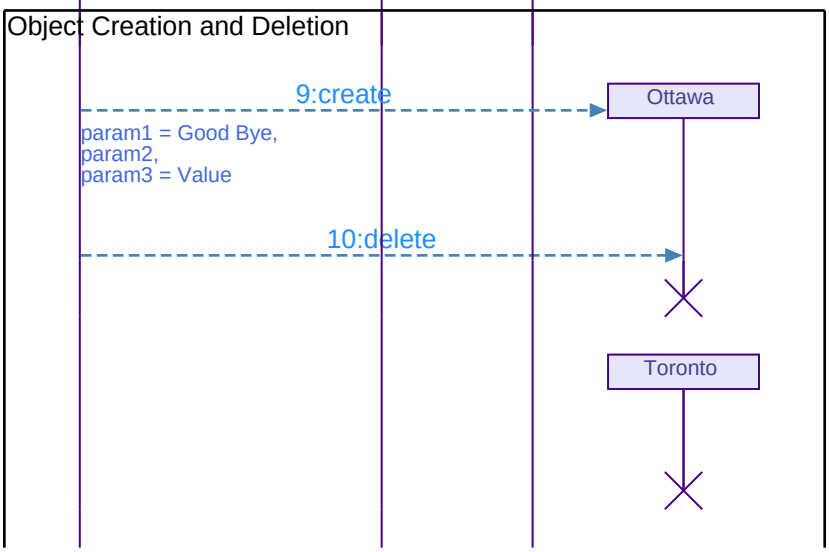
Message Statement: Model messages interactions with parameters.

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





Model method invocation and return.

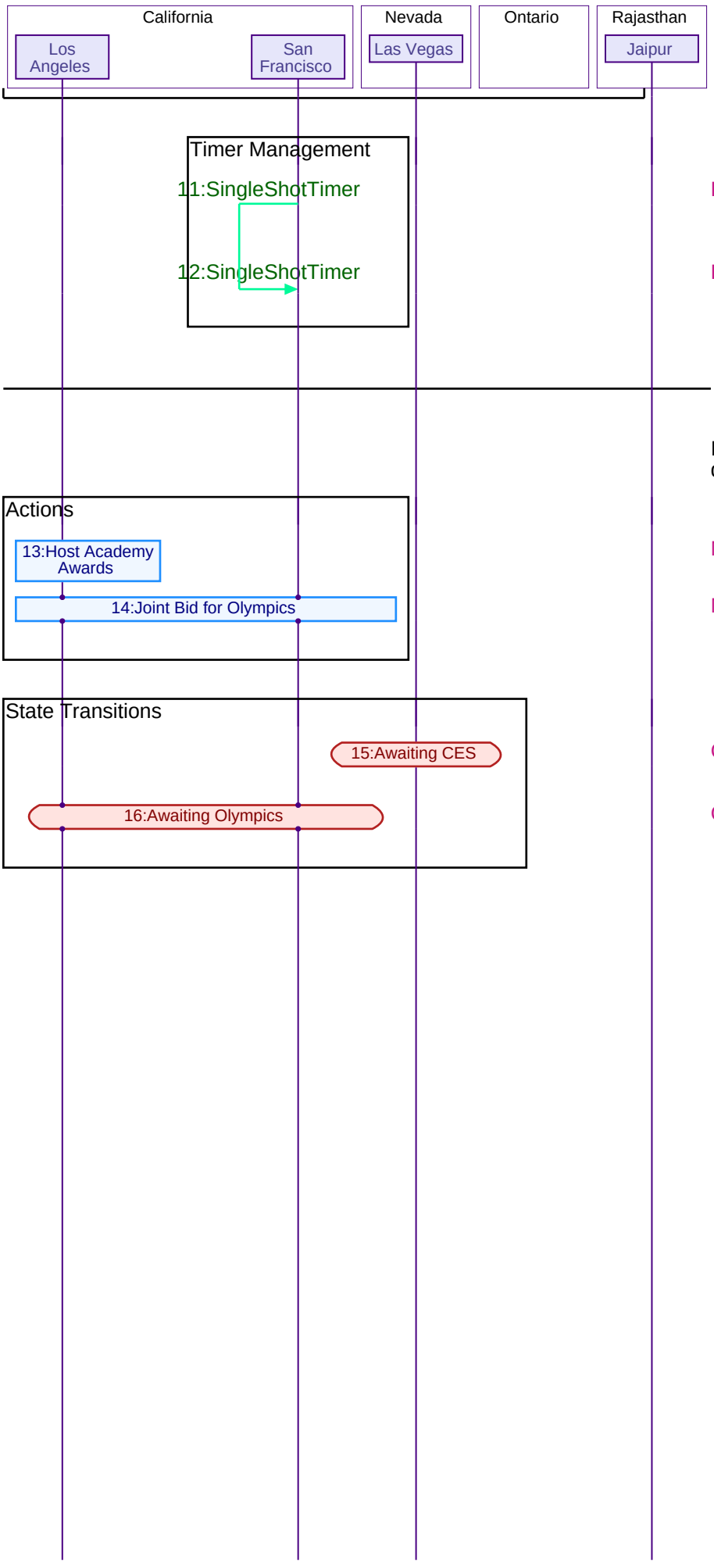


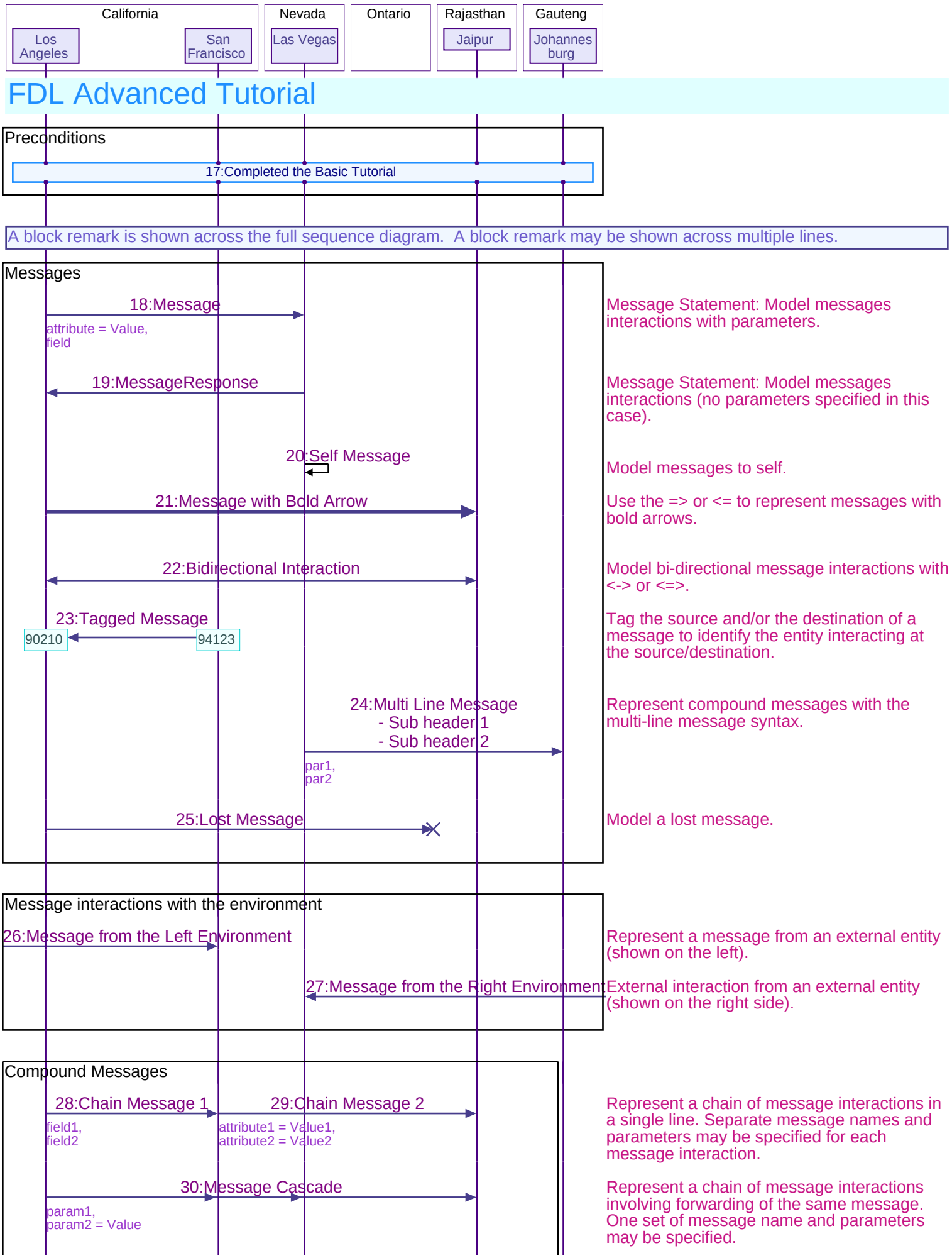
One object creates another.

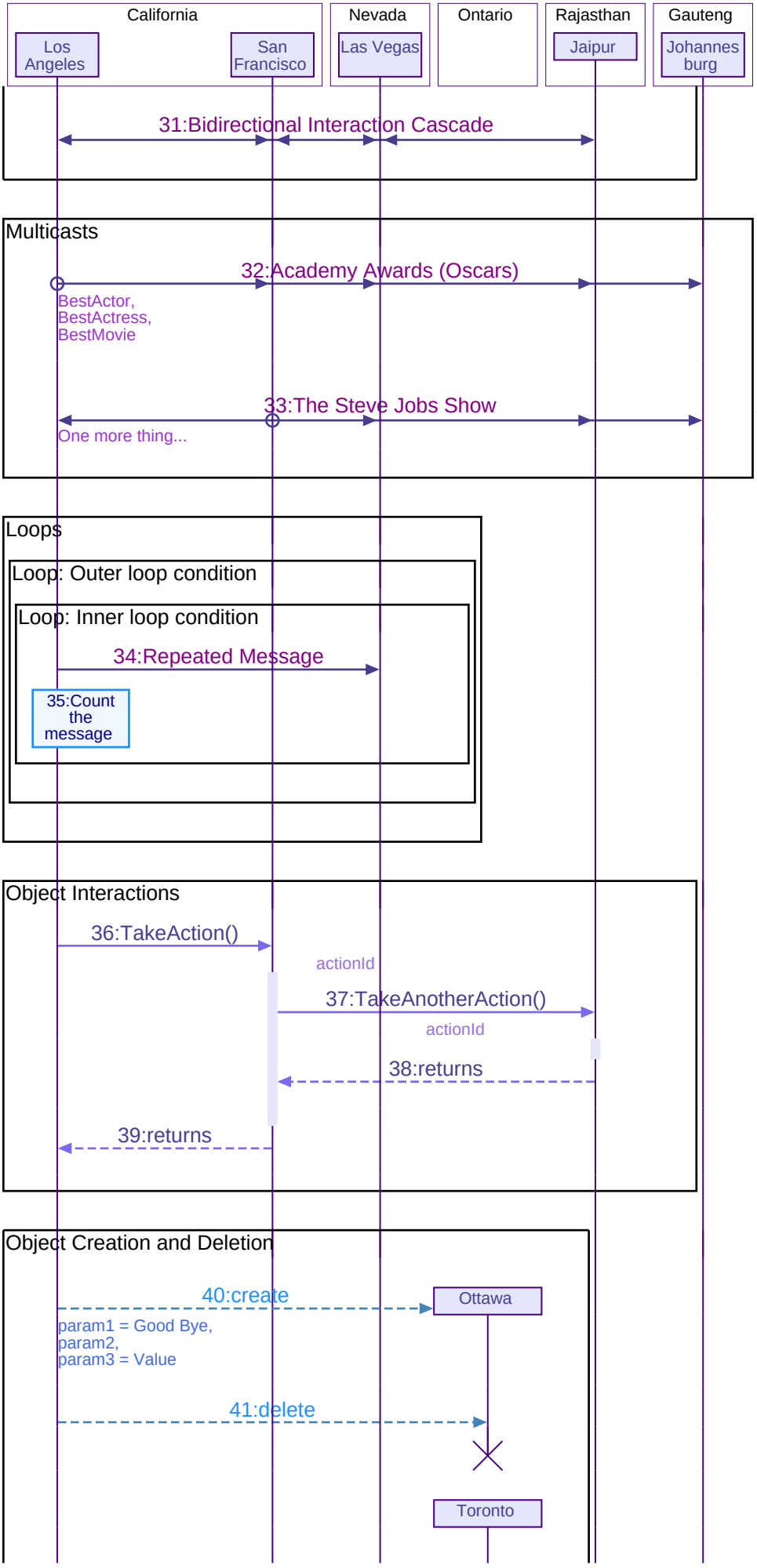
One object deletes another.

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.


California
Los Angeles
San Francisco
Nevada
Las Vegas
Ontario
Rajasthan
Jaipur
Timer Management
11:SingleShotTimer
12:SingleShotTimer
Represent a timer start.
Represent a timeout.
ISSUE: Why is 42 the answer to the ultimate question of life the universe and everything?
Actions
13:Host Academy Awards
14:Joint Bid for Olympics
Model actions taken by an object.
Model joint actions taken by multiple objects.
State Transitions
15:Awaiting CES
16:Awaiting Olympics
Change the state of an object.
Change the joint state of two objects.


California
Los Angeles
San Francisco
Nevada
Las Vegas
Ontario
Rajasthan
Jaipur
Gauteng
Johannes burg
FDL Advanced Tutorial
Preconditions
17:Completed the Basic Tutorial
A block remark is shown across the full sequence diagram. A block remark may be shown across multiple lines.
Messages
18:Message
attribute = Value,
field
Message Statement: Model messages interactions with parameters.
19:MessageResponse
Message Statement: Model messages interactions (no parameters specified in this case).
20:Self Message
Model messages to self.
21:Message with Bold Arrow
Use the => or <= to represent messages with bold arrows.
22:Bidirectional Interaction
Model bi-directional message interactions with <-> or <=>.
23:Tagged Message
90210
94123
Tag the source and/or the destination of a message to identify the entity interacting at the source/destination.
24:Multi Line Message
- Sub header 1
- Sub header 2
par1,
par2
Represent compound messages with the multi-line message syntax.
25:Lost Message
Model a lost message.
Message interactions with the environment
26:Message from the Left Environment
Represent a message from an external entity (shown on the left).
27:Message from the Right Environment
External interaction from an external entity (shown on the right side).
Compound Messages
28:Chain Message 1
field1,
field2
29:Chain Message 2
attribute1 = Value1,
attribute2 = Value2
Represent a chain of message interactions in a single line. Separate message names and parameters may be specified for each message interaction.
30:Message Cascade
param1,
param2 = Value
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.

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

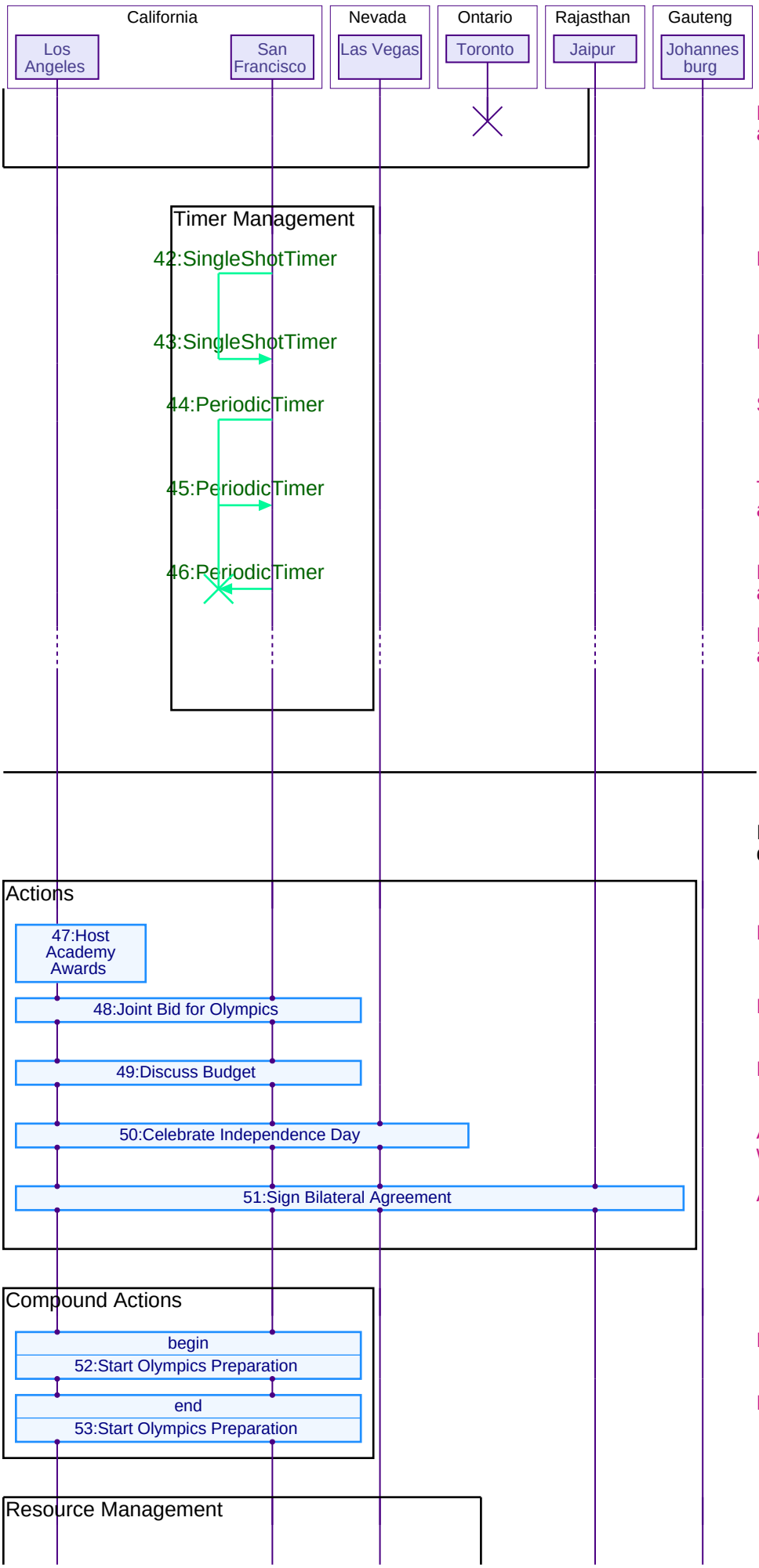
Another multicast. This time the multicast source is not at the edge.

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.

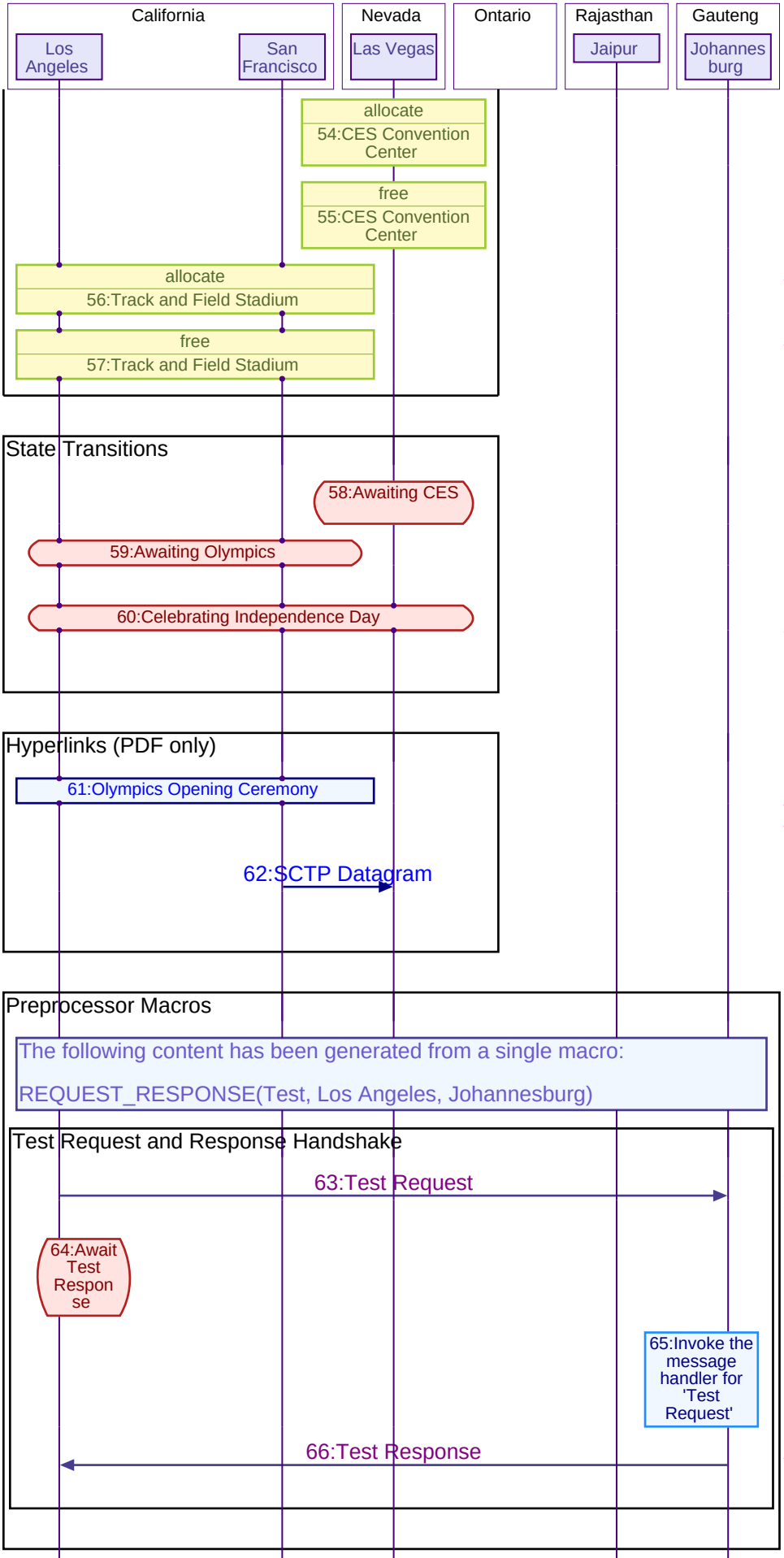
Model actions at component level.

A module level action. It 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].

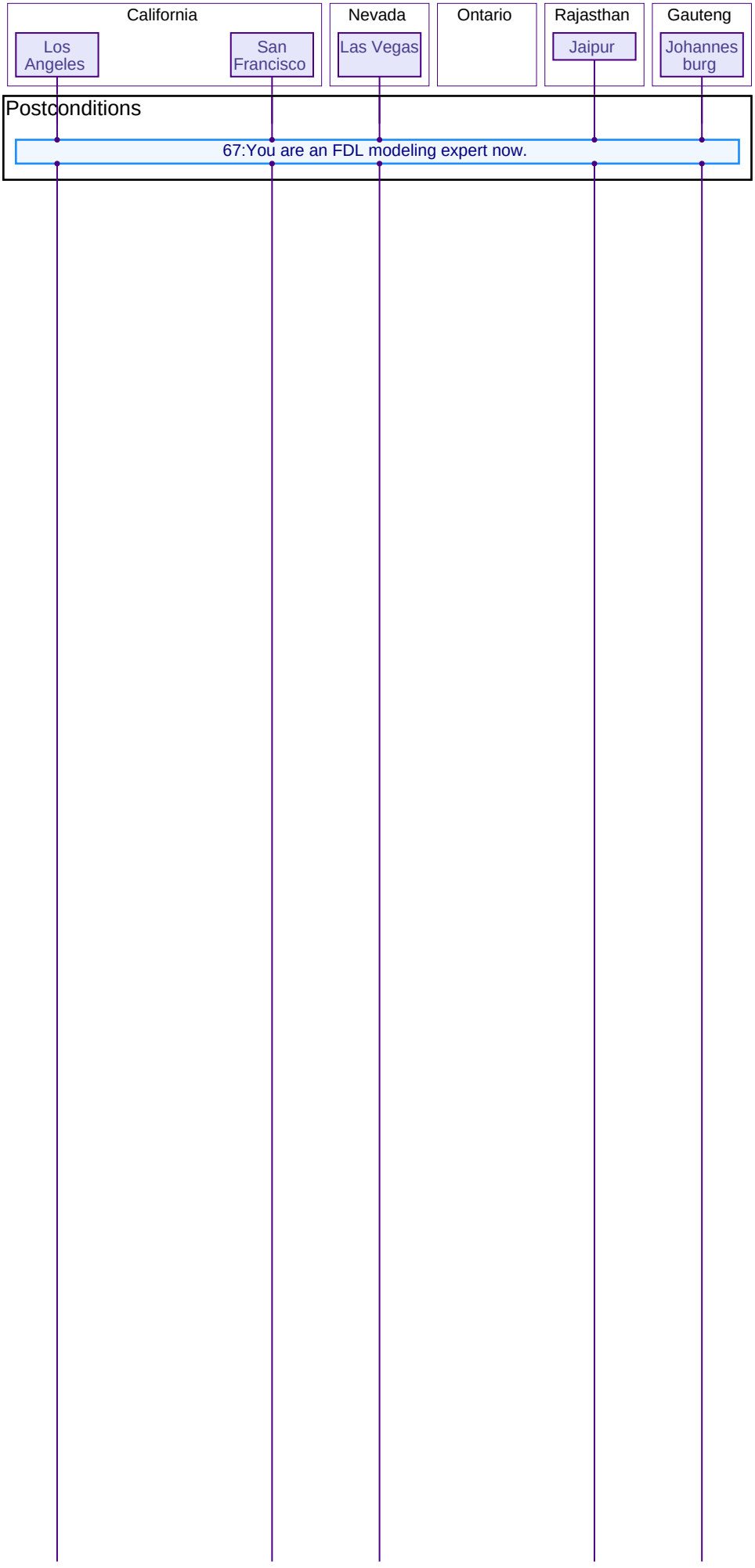
Click on the message name to get details about the message. [_hyperlink] style prefix has also been applied.

Test Request sent from Los Angeles to Johannesburg.

Los Angeles awaits Test Response.

Invoke the message handler Test Request.

Acknowledge message.


California
Los Angeles
San Francisco
Nevada
Las Vegas
Ontario
Rajasthan
Jaipur
Gauteng
Johannesburg
Postconditions
67:You are an FDL modeling expert now.