## FIRAS USE CASE 3

## Use Case Descriptions

This section documents the complete business scenarios within the scope of this project.

### Subscriber Handling Events

In this project scenario, one subscriber will be notified by the system (channel) about an event that has been posted. The channel takes into account the subscriber’s state to determine how to the subscriber deals with the event.

**Description:**

The scenario deals with how a subscriber handles an event once an event is posted by the publisher. The subscriber, who must be interconnected with the said channel through subscription prior to receiving the notification, will have a state associated with them which consists of a specific ID. By using this unique ID, the subscriber will handle the event according to the logic that is mandated by the state.

**Actors:**

The follows actors are associated with this scenario:

1. Subscriber: a member (entity) of a channel who has access to the events of such channel and can handle them accordingly. The subscriber is identified by their unique ID, as well as a state ID.
2. Publisher: An entity that publishes/posts events to one or more channels. A publisher publishes an event to a channel based on a strategy, or to a default channel(s) if the no specific strategy is selected.

**Preconditions:**

Prior to the scenario being performed:

1. The subscriber must exist as an entity. If they do not exist, then the channel cannot reach said entity.
2. The subscriber must be accountable for a state ID. This state ID determines how an event can be handled.
3. The subscriber must be subscribed to the channel which the event is posted in. If they are not, the channel will not notify them of the posted event.

**Post Conditions:**

After the preconditions are met, the following will occur:

1. The subscriber handles the event according to the logic dictated by its state.

**Scenario Text:**

1. Use *Publishing an Event*
   1. Perform publication via publishing strategy
   2. Generate appropriate event
   3. Queue event to correct channel
2. Notify Subscribers of Event
   1. Check to see if subscriber is blocked on channel
   2. Visit specific topic of channel and notify all subscribers of event
3. Determine State of Subscriber
   1. Call configuration file *states.sts* to learn subscriber’s state
4. Handle Event
   1. Subscriber handles event according to their state

**Alternative Courses:**

None.

**Extends:**

None.

**User Interfaces:**

The file *states.sts,* which consists of a compilation of tuples of the form <subscriber-ID, state-ID>, is implemented of type *int* which determines the subscriber’s unique personal ID as well as state ID. This allows the channel to recognize how a subscriber will handle a particular event.

**Constraints:**

None.

**Questions:**

None.

**Notes:**

None.

**Authors:**

Firas Aboushamalah

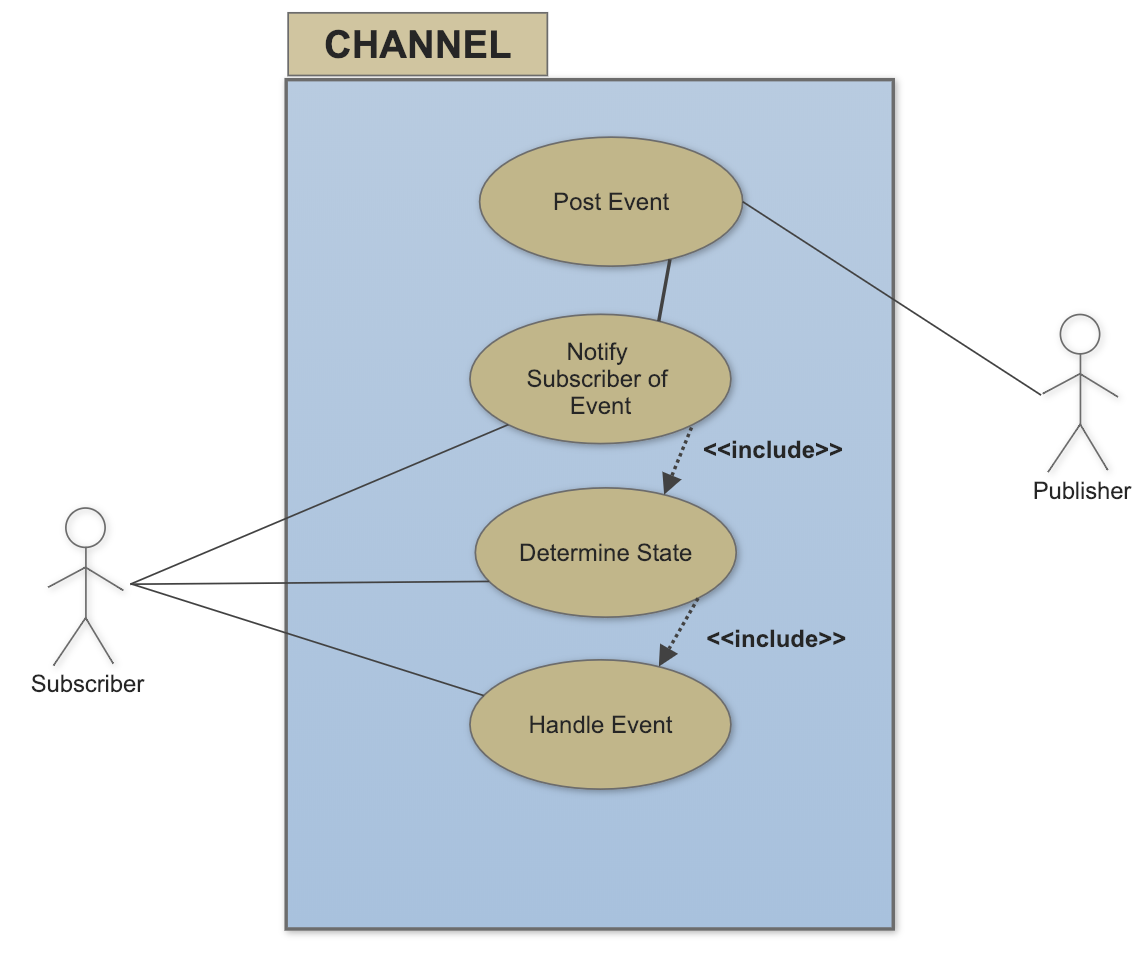
**Source Documents:**

<http://agilemodeling.com/essays/umlDiagrams.htm>

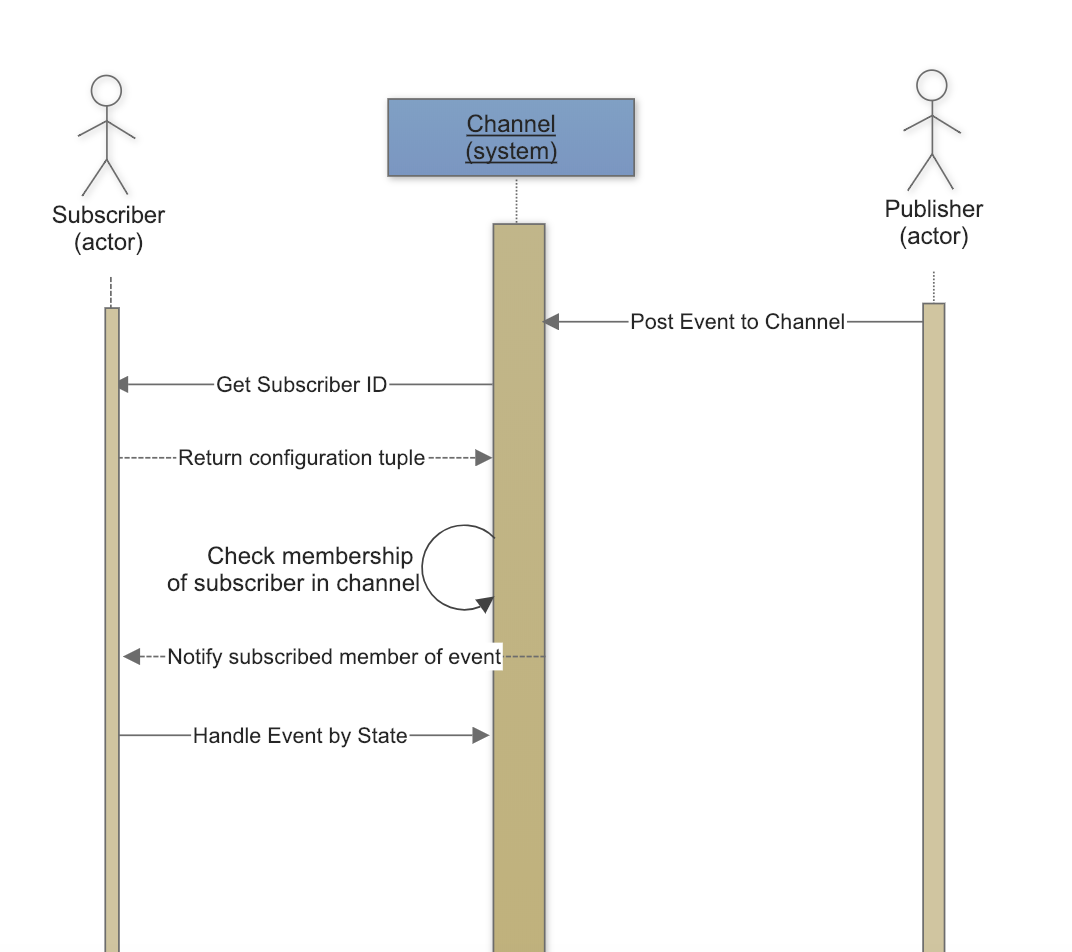
http://www.math-cs.gordon.edu/courses/cs211/ATMExample/

Use Case Diagrams

Model of how a subscriber handles an event in the Sub-Pub system.



## Sequencing Diagrams

**

## Collaboration Diagrams

