

Use Cases

UC1: Publishing an event

This use case pertains to a publisher posting (i.e. publishing) an event **on one or more channels**.

The **publisher** is associated with publishing strategies, which encapsulate(压缩) the logic that determines the channel or channels to which an event will be posted.

Note that events can be of different types. They have a header and a body (i.e. payload), and are generated by an **event factory**.

The publication of an event (发送 event) is performed via the **publishing strategy**, associated with a publisher, and can occur in two variations

a) a publisher specifying 说明 which event to publish and consequently **relying on the strategy** to determine to which channel to post it to 发起人说明白 event 的内容是什么然后 strategy 自己制定要 po 的 channel

b) a publisher **relying on the strategy** to both generate the appropriate event and decide to which channel to post it to. 发起人直接 让 strategy 来弄

For this project, we have a configuration file named **strategies.str** that denotes which publisher is associated with which publishing strategy. 有很多的发起人 每个人发送的 strategy 都不一样

The file is organized in tuples of the form <publisher-ID, strategy-ID>. Each line of this configuration file is one such tuple. **The operation of publishing an event** on a specific channel is handled by a Channel Event Dispatcher module which finally posts the event using the channel's interface. Note that *publisher-ID* and *strategy-ID* have implemented in the sample code as of type *int* (see Strategies.str file).

The **precondition** of this use case is that **the publisher and the event must exist**. If the channel the event is about to be published does not exist then it is created. The publishing occurs via a strategy each publisher is associated with. The strategy refers to channels by name. 假如不存在就创建

The **postcondition** of this use case is that there should be **an event in the queue of the specified channel**, and **the channel exists** (either it is created as part of this event publication, or existed before).