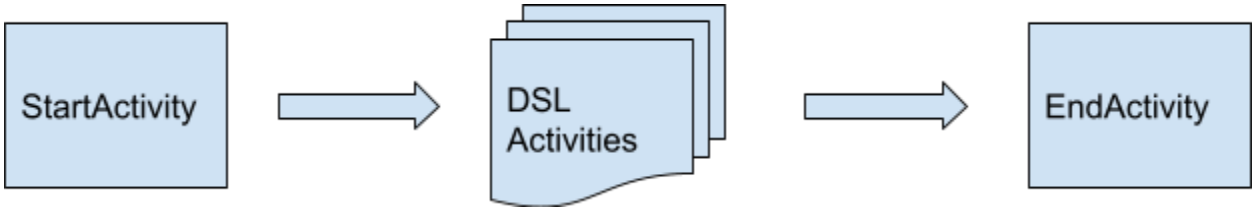
Workflow DSL Mapping

The Workflow DSL specification (https://docs.google.com/document/d/1QAMUTea_0jG6CyrJqDppHlQecL6OJ348__9iaBzcLn8/edit#heading=h.9dis3mo62rpn) is complete for a Workflow DSL format. This document will describe how the specification maps onto our Workflow implementation on top of Temporal.io.

Workflow

The main Workflow DSL will be read by a Workflow class. Two maintenance Activities will be prepended to the existing Activities specified in the DSL.

- An initial 'StartActivity' will be prepended to the Activities list which will initialize the workflow and read/validate the DSL.
- A final 'EndActivity' will be appended to the Activities which will handle cleaning up the Workflow, returning expected results data, and archiving any data.



Each Workflow DSL that gets initialized and run will have a unique Workflow ID in temporal. This will be a combination of the unique Workflow name, as specified in the DSL, and version property.

Activites

DSL will list 'activities' which are separated by the 'type' property e.g. 'operation', 'inject', 'message', etc. Each type of Activity will be implemented by a separate Activity class and interface, for example 'MessageActivity'.

Each Activity can contain a list of one or more Actions, which can be calls to REST api services, or generating a Message. These actions will also be handled in the respective Activity implementation class in Temporal. Common features for activities like 'sleep', `timeout`, `retry` will be consolidated in a parent Activity class.

Messages

The `message' section will include a list of declared Messages (Events or Signals). These messages should follow the CloudEvents specification (github.com/cloudevents). Messages can be referenced inside of the OnMessage property inside of a Workflow Activity definition. For example:

```
"onMessages": [{
  "messageRefs": ["NewApplicationEvent"],
  ...
}]
```

It is possible for a Workflow to be both a 'consumer' and 'producer' of messages in the DSL.

Example Workflow

```
{
    "id": "helloStooges",
    "version": "1.0",
    "specVersion": "0.8",
    "Hash": "2ed01372d1149baa5a79d25e5eda3372",
    "name": "Hello Stooges",
    "description": "Hello Stooges Workflow",
    "start": "Hello Larry",
    "isChild": "true",
    "activities":[
        {
            "name":"Hello Larry",
            "version": "1.0",
            "type": "message",
            "onMessages": [{
                 "messageRefs": ["SayHello"],
                 "actions": [{
                     "type": "rest",
                     "operation": "http://127.0.0.1/hello/larry",
                     "method": "GET",
                     "arguments": {
                        "id": "${ .currentLight.id }",
                        "lumens": "${ .currentLight.lumens }"
                     }
                 }]
             }],
            "transition": "Hello Moe"
        },
        {
            "name":"Hello Moe",
            "version": "1.0",
            "type": "operation",
```

```
            "actions": [
                {
                    "name": "Say Hello Moe",
                    "type": "rest",
                    "operation": "http://127.0.0.1/hello/moe",
                    "sleep": {
                        "after": "PT1S"
                    }
                }
            ],
            "transition": "Hello Curly"
        },
        {
            "name":"Hello Curly",
            "version": "1.0",
            "type": "inject",
            "data": {
                "result": "Hello Curly"
            },
            "end": true
        }
    ],
    "messages": [
        {
            "name": "SayHello",
            "messageType": "Signal"
            "source": "127.0.0.1",
            "type": "com.boommtown.workflow-poc.signals",
            "kind": "consumed",
        }
    ]
}
```