Flow Inheritence

Summary

Flow inheritance allows one flow to inherit the configuration of another flow. Inheritance can occur at both the flow and state levels. A common use case is for a parent flow to define global transitions and exception handlers, then each child flow can inherit those settings.

In order for a parent flow to be found, it must be added to the flow-registry just like any other flow.

Description

Is flow inheritance like Java inheritance?

Flow inheritance is similar to Java inheritance in that elements defined in a parent are exposed via the child, however, there are key differences.

A child flow cannot override an element from a parent flow. Similar elements between the parent and child flows will be merged. Unique elements in the parent flow will be added to the child.

A child flow can inherit from multiple parent flows. Java inheritance is limited to a single class.

Types of flow inheritance

Flow level inheritance

Flow level inheritance is defined by the parent attribute on the flow element. This property expresses flow to inherit by separating with comma.
Lower flow inherits the upper flows respectively in the order indicated in the list.
When you add the contents and element in the upper flow as the first inheritance, it is regarded as lower flow again, and receives the next upper flow.
Following example shows that common-transitions are inherited first and then common-states are inherited.

```
<flow parent="common-transitions, common-states">
```

State level inheritance

State level inheritance is similar to flow level inheritance, except only one state inherits from the parent, instead of the entire flow. Different from Flow inheritance, only one upper level is allowed. In addition, identifier of Flow State to inherit should be defined.
Flow and State identifier are separated by **#**.
Upper and lower State should be the same type. For example, view-state cannot inherit ent-state. Only view-state can be inherited.

```
<view-state id="child-state" parent="parent-flow#parent-view-state">
```

Abstract Flow

Often parent flows are not designed to be executed directly. Can set to abstract not to execute such flow.
FlowBuilderException occurs if you try to perform abstract flow.

```
<flow abstract="true">
```

Inheritance Algorithm

When a child flow inherits from it's parent, essentially what happens is that the parent and child are merged together to create a new flow. There are rules for every element in the Web Flow definition language that govern how that particular element is merged.

There are 2 types of elements: mergeable and non-mergeable.
mergeable element tries to merge the element if they are same. non-mergeable element is not directly included in final Flow.
Is not modified during merge process.

Caution

- *Path of external resource in the upper Flow should be absolute path.** Absolute path may be broken if the directories where upper flow is located and where lower flow is located are different when merging two flows.

If it is merged generally, all relative path located at the upper flow is changed to lower flow standard.

Mergeable Element

If the elements are of the same type and their keyed attribute are identical, the content of the parent element will be merged with the child element.
Merge algorithm merges sub elements on the upper and lower that continues to merge.
If not, add to elements of upper flow to the lower flow as a new element.

In most cases, elements in upper flow are added to lower flow elements.
Exception to this rule includes the action element to add at the time of start(evaluate, render, set).
Use the results of upper action as the results of lower action.

Mergeable element is as follows:

- action-state: id
- attribute: name
- decision-state: id
- end-state: id
- flow: merge always
- if: test
- on-end: merge always
- on-entry: merge always
- on-exit: merge always
- on-render: merge always
- on-start: merge always
- input: name
- output: name
- secured: attributes
- subflow-state: id
- transition: on and on-exception
- view-state: id

Non mergeable elements

Non-mergeable elements are:

- bean-import
- evaluate
- exception-handler
- persistence-context
- render
- set
- var

Reference

- Spring Web Flow reference 2.0.x
- Spring Web-Flow Framework Reference beta with Korean (by Park Chan Wook)
- Whiteship's Note