TOSCA Imperative workflows

Priya TG (Netcracker)

June 19, 2019

# Problem Statement

TOSCA Imperative workflows in v1.3 of TOSCA Simple Profile YAML introduces support of custom workflow language. The workflow grammar supports referencing artifacts as illustrated in the example of v1.3 of the specification (see below).





**The following gap needs to be addressed in v1.3 grammar:**

There is NO DEFINED INTERFACE regarding how information from the service template (e.g. node template properties) will be passed to external workflow. Thus, the external workflow remains disconnected from the service template.

To illustrate this, let us consider an example of Imperative workflows using native TOSCA grammar that allows for such input assignments. The highlighted text of the example below allows assignments from properties of node templates to workflow inputs.

|  |
| --- |
| tosca\_definitions\_version: tosca\_simple\_yaml\_1\_3description: sample workflow snippet node\_types: tosca.example.WebApplication: derived\_from: tosca.nodes.WebApplication properties: flavor\_id: type: string required: true image\_id: type: string required: true . . .  interfaces: Standard: create: . . .topology\_template: substitution\_mappings: node\_type: tosca.example.WebApplication . . .  node\_templates: web\_db: type: tosca.nodes.DBMS.MySQL properties: db\_name: . . . db\_user: . . . db\_password: . . . db\_port: . . . interfaces: Standard: create:  inputs:  db\_user: { get\_property: [ web\_db, db\_user] } db\_password: { get\_property: [ web\_db, db\_password ] }  web\_server: type: tosca.nodes.WebServer.Apache properties: user\_name: . . . interfaces: standard: create:  inputs: web\_user: { get\_property [ web\_server, user\_name] } web\_port: { get\_input: web\_port }   workflows: instantiate: inputs: db\_user: type: string db\_password: type: string web\_user: type: string web\_port: type: string  steps: step\_1: target: [web\_db] activities: - call\_operation:  operation: tosca.interface.standard.create on\_success: . . .  on\_error: . . . step\_2: target: [web\_server] activities: - call\_operation:  operation: tosca.interface.standard.create on\_success: . . .  on\_error: . . . |

When scripts are referenced from workflows, such input assignments are not possible because workflow grammar allows input definition , but not input assignment.

|  |
| --- |
| tosca\_definitions\_version: tosca\_simple\_yaml\_1\_3description: sample workflow snippet node\_types: tosca.example.WebApplication: derived\_from: tosca.nodes.WebApplication properties: flavor\_id: type: string required: true image\_id: type: string required: true . . .  interfaces: Standard: create: . . .topology\_template: substitution\_mappings: node\_type: tosca.example.WebApplication . . .  node\_templates: web\_db: type: tosca.nodes.DBMS.MySQL properties: db\_name: . . . db\_user: . . . db\_password: . . . db\_port: . . . . . .   web\_server: type: tosca.nodes.WebServer.Apache properties: user\_name: . . . . . .  workflows: instantiate: inputs: db\_user: type: string db\_password: type: string web\_user: type: string web\_port: type: string  implementation: primary: file: artifacts/instantiate.workflow.yaml type: tosca.artifacts.Implementation.nfv.Mistral # outputs: # <attribute\_mappings> |

## Resolution

We discussed possible ways to address this:

1. **Allow intrinsic functions with default values of workflow inputs**. The current v1.3 grammar is not very clear about the use of intrinsic functions for default values in workflow input definition.

|  |
| --- |
|  workflows: instantiate:  inputs: db\_user: type: string default: get\_property: { [ web\_db, db\_user] } db\_password: type: string default: { get\_property: [ web\_db, db\_password ] } web\_user: type: string default: { get\_property [ web\_server, user\_name] } web\_port: type: stringdefault: { get\_input: web\_port } |

1. **Allow input assignments apart from input definitions for workflow inputs**. However, this could be confusing since this is not consistent with the current workflow grammar.

|  |
| --- |
|  workflows: instantiate:  inputs: db\_user: type: string value: { get\_property: [ web\_db, db\_user] } db\_password: type: string value: { get\_property: [ web\_db, db\_password ] } web\_user: type: string value: { get\_property [ web\_server, user\_name] } web\_port: type: stringvalue : { get\_input: web\_port }   |

1. Allow the workflow referencing external workflows to be described inline?

 ***The second issue is: Such requirements of passing service template properties to external workflows need replication of almost the entire service template in workflow inputs. Can this be addressed in TOSCA grammar?***