submission by

Shahaf Shabo 318803590

Tomer Klein 208642736

Instructions

-In every moment in the User interface, you can press 0 to go back to the main menu in the UI.

-In order to load xml file, you will need to press 1 and paste you path to the xml, then all the validation for the stepper will happen.

You can load it again as much as you can, but it will delete (from choice) every other flow Execution or xml that happened before.

-For general flow definition press 2 from the main menu, there you will be able to check every single detail about each flow Individually.

-In order to execute flow press 3 from the main menu, there you will be able to choose from each flow in the system that loaded from the last xml you provided.

-The program will present a list of mandatory inputs and a list of optional inputs and will ask the user to choose what input he would like to enter

-After all mandatory inputs are provided the program will ask whether you would like to continue entering inputs or start executing the flow

In the end all steps execution details and formal outputs will be shown.

-In order to get statistic about all previous executions happened in the stepper press 4 from the main menu.

There you will see each flow execution by name, ID and date.

From each execution you will see general details about the execution includes flow name, id, result, time, user's inputs, outputs and all these details for each step in the flow.

For general stepper statistic press 5 from the main menu there you will find a system statistic like average timing of each flow or each step in flow, how many times the step used etc.

Make sure you enjoy 😊

Significant classes:

1)FlowExecutor:

takes inputs from user and puts those inputs values inside a StepExecutionContext variable that will be passed on to each step the function is going to invoke, after execution of the flow returns statistics of execution

2)FlowLoader:

loads the stepper from xml file the user gives using jaxb then loads the jaxb generated onto our implemented StepperDefinition class

3)FlowDefinition:

contains data about a single flow inside a step

4)FlowValidator:

given a flow, validates the flow to make sure he is up to standards

5)AbstractStepDefinition:

defines a step based on its name,readOnly,inputs and outputs, this class is implemented on all steps in the program

6)AbstractDataDefinition:

defines a data definition based on its name,userFriendlyness and type this class is implemented on all data definitions in the program

7)StepDefinitionRegistry:

given a step name creates the single and only instance of a step (since there is no need to create a new one ever) and returns it