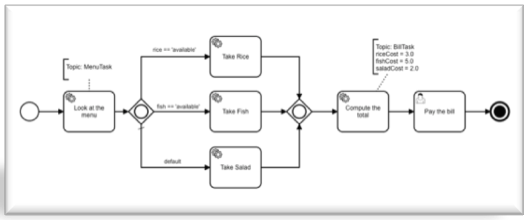
12/11/ 2018 - Giacomo De Liberali - 580595

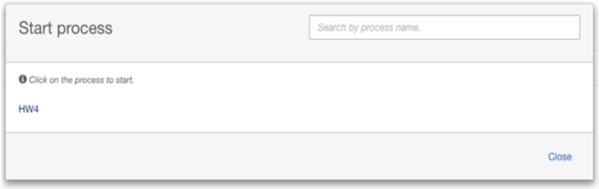
HOMEWORK 4: CAMUNDA & WOPED

1. IMPLEMENTATION



The exercise consists in the modeling of a business process with Camunda and its implementation. Once the model is ready and successfully deployed, it could be stared from the Camunda Web interface.

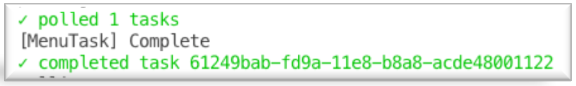
The new process can be started with the "Start process" button in Task Lists Web interface. Before starting the process, we should also start node workers, in order to keep them in polling on their channels allowing them to complete service tasks.



A. WORKFLOW

The workflow is composed by different tasks, for instance:

a) Look at the menu: Camunda External Task associated to the "*MenuTask*" channel. There is a node.js worker polling in this channel waiting for a new task to appear. Once the process is started from the Camunda tasks list, it gets pushed in a queue and picked from the external worker. This worker will:



```
✓ polled 1 tasks
[MenuTask] Complete
✓ completed task 61249bab-fd9a-11e8-b8a8-acde48001122
```

   a. set the process variable "*fish*" to "*available*" in the 40% and "not_*available*" otherwise;

   b. set "*rice*" process variable to "*available*" in the 60%, "not_*available*" otherwise.

Once variables are set, the worker will complete its task and the flux proceeds. At this point the (XOR) gateway activate one of following transition:

- $P(TakeFish) = 0.4\ \cdot (1 - 0.6) = 0.16$
- $P(TakeRice) = 0.6\ \cdot (1 - 0.4) = 0.36$
- $P(TakeRice\ \wedge TakeFish) = (1 - 0.4) \cdot (1 - 0.6) = 0.24$
- $P(TakeSalad) = 1 - (0.16 + 0.36 + 0.24)\ = 0.24$

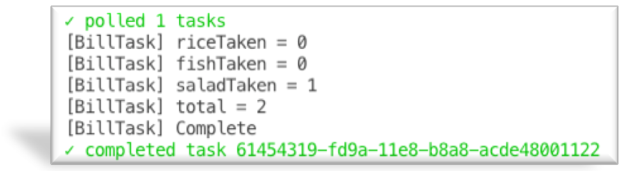
based on the process variables currently set. The probability of each transition is expressed above.

b) Take Rice: Comunda Service Task that will set the result of an expression in a process variable if the flow is passing through its transition. The affected variable could be “*riceTaken*”, in which the value “1” would appear. Executed if “*rice*” == “*available*”. It could possibly run in parallel with the task C.

c) Take Fish: Comunda Service Task. The affected variable could be “*fishTaken*”, in which the value “1” would appear. Executed if “*fish*” == “*available*”. It could possibly run in parallel with the task B.

d) Take Salad: Comunda Service Task. The affected variable could be “*saladTaken*”, in which the value “1” would appear. This is the default value of the inclusive gateway, in fact its branch is executed if and only if nor “*rice*” nor “*fish*” are “*available*” (They are both “*not_available*”).

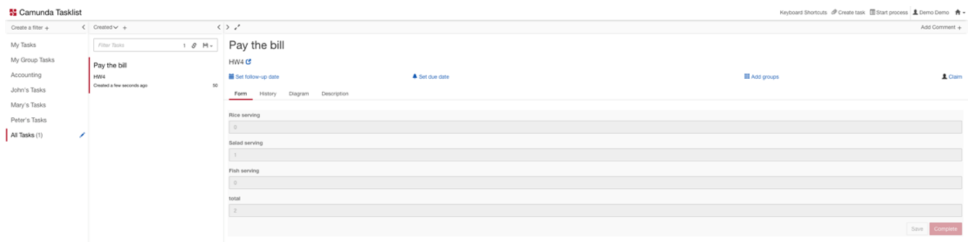
e) Compute the total: Camunda External Task associated to the “*BillTask*” channel. There is a node.js worker polling in this channel waiting for a new task to appear.



```
✓ polled 1 tasks
[BillTask] riceTaken = 0
[BillTask] fishTaken = 0
[BillTask] saladTaken = 1
[BillTask] total = 2
[BillTask] Complete
✓ completed task 61454319-fd9a-11e8-b8a8-acde48001122
```

Once a new task is picked from the queue by the worker, the “*riceTaken*”, “*fishTaken*” and “*saladTaken*” process variables are read to allow the worker to compute the total bill for the user. The products’ cost are, in this moment, hardcoded inside the worker, but they are needed to calculate the total, which is latter set as a process variable.

f) Pay the bill: User Task. This task must be claimed by the current user and must



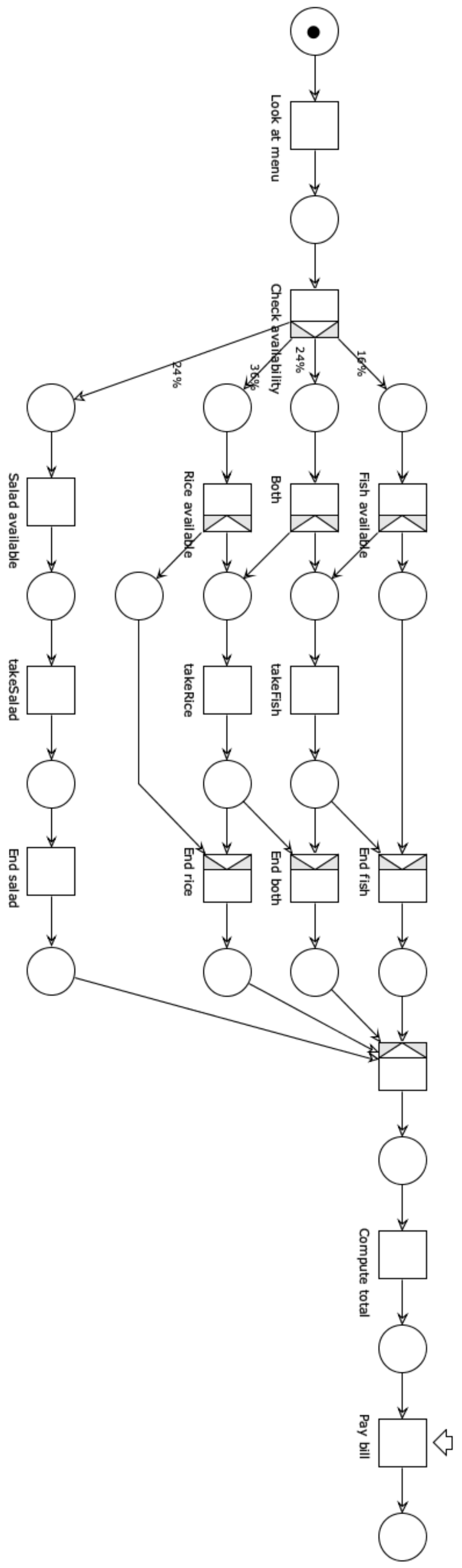
Pay the bill
HW4
Created a few seconds ago

Pay the bill
HW4

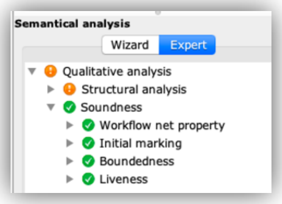
Rice serving
0
Salad serving
1
Fish serving
0
total
2

also be completed by him. Once the user claims the task, a new form is presented with the result of the current process instance. The form fields are defined in this task in the Camunda Modeler, and their read-only value are process variables created by previous task E (“*fishTaken”, “riceTaken”, “saladTaken“ and “total”*).

2. WORKFLOW NET WITH WOPED



The workflow net is sound, as WoPeD states, since it is live and bound.


Semantical analysis
Wizard
Expert
Qualitative analysis
Structural analysis
Soundness
Workflow net property
Initial marking
Boundedness
Liveness