IN2013, Week 4 – State Machines

Model Answers

Dr Peter T. Popov

18th October 2018

Scenario

Question 1 (state machine). Consider the BAPPERS case study and the design class diagram shown in Appendix 1.

Create a behavioral state machine, which shows the lifecycle of an instance of the class Job.

Broadly the Job goes through two stages: "Being Created" and "Being Processed". "Being Created" represents the transformations of the Job state due to adding/removing tasks, i.e. before the Job is stored and placed for processing.

"Being Processed", on the other hand, represents the transformation of the Job state as a result of tasks being processed – from the start of the first task in the list of tasks to completing the last task, at which time the job itself becomes completed.

Consider also the case of changing the Job priority from "normal" when the job is created to "urgent".

Most of the state changes are caused by different call events, i.e. invocations of the methods defined for the class Job. Therefore, analyse carefully the methods defined for the class job in Appendix 1.

Model answer

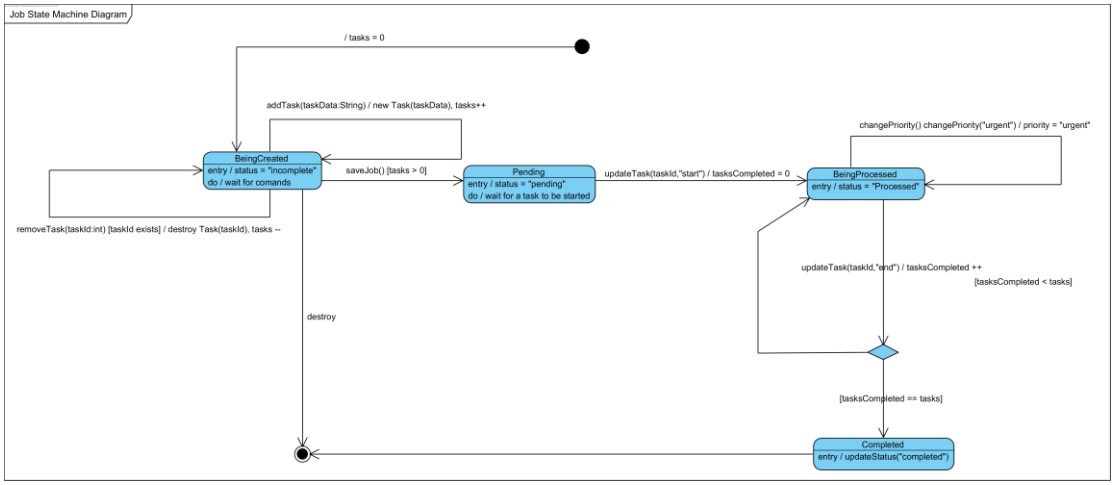
A state machine diagram that models Job and uses the call events as defined in the class diagram is shown below.

The basic flow is that we have a state "BeingProcessed" which reacts to 3 different call events – addTask(), removeTask() and saveJob(). The first two – as the names suggest - increase or decrease the number of tasks included in the Job with some trivial checks put in place. The 3rd method leads to saving the Job and placing it the queue of jobs for processing. In this model answer we assume that once the job is created (i.e. saved) the number of tasks will not change, which is clearly a simplification of what might be needed. In real operation.

The other 3 states "Pending", "BeingProcessed" and "Completed" capture three distinct states. The first is the state of the job waiting to be taken for processing, i.e. until the first task of the Job is started. The trigger here is the event updateTask("start"), which is meant to represent starting a task and will change the state of the Job to "BeingProcessed". Note that in this new state Job reacts to call events updateTask("end") and changePriority(): the former indicates that another task is completed and the latter – a change of job priority.

Once the final task in the job is completed, the Job itself becomes completed and moves from the "BeingProcessed" to "Completed", where the lifecycle terminates.

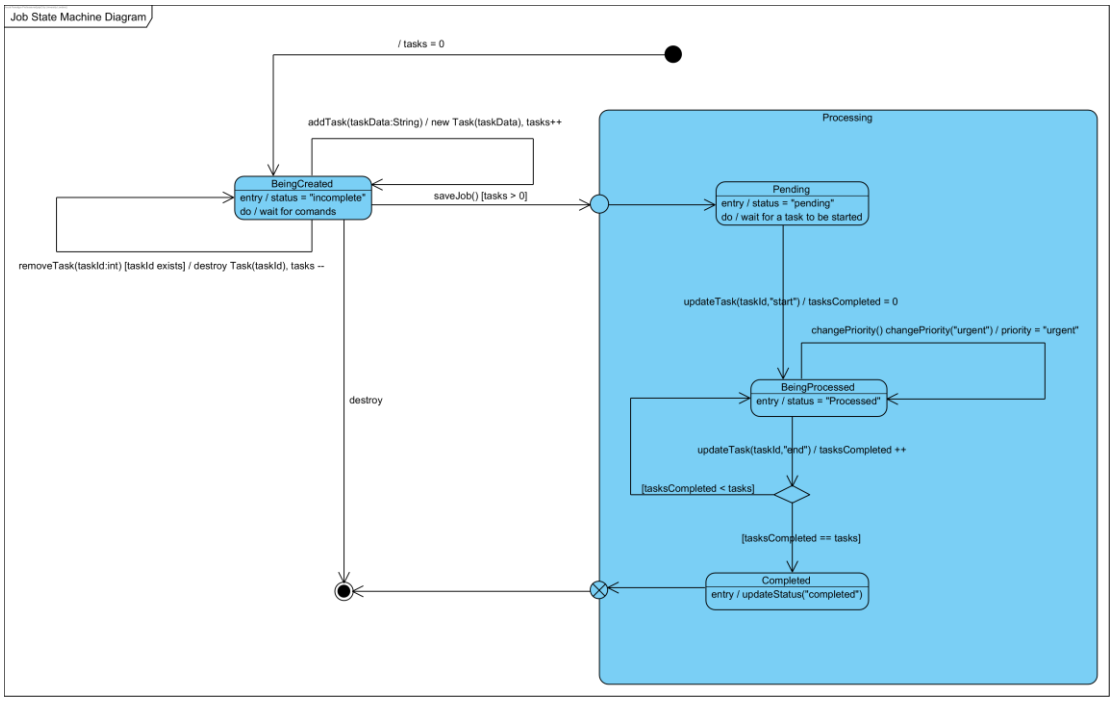
There is a video on Moodle (Media Resource), in which I developed the state machine from scratch.



Question 2: Refine your answer by introducing a composite state "Operational" which represent all states included in the "Being Processed" stage of the Job lifecycle.

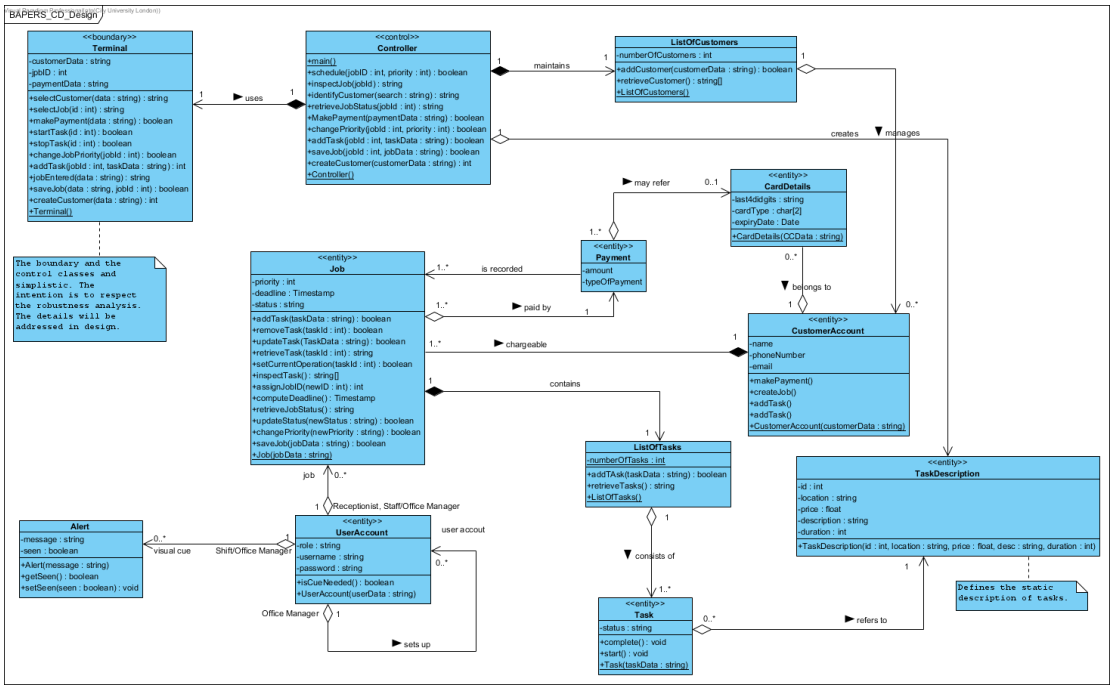
Model answer

A refined state machine diagram in which the "Operational" state is a composite state that includes 3 operational states is shown below.



The two diagrams are clearly identical: they just demonstrate the flexibility of the syntax to model state machines.

Appendix 1. Design class diagram for BAPERS



23rd September, 2018