LabQakExamples2021 - Case Study - es0

Introduction

Requirements

es0: Design and build a software system (named ***workshift***) that allows to manage a machine according to three turns:

1. in the first turn (in the morning) the machine is able to handle messages (dispatches) of type m1:m1(V)
2. in the second turn (in the afternoon) the machine is able to handle messages (dispatches) of type m2:m2(V)
3. in the third turn (in the nigth) the machine 'sleeps'

Messages of types m1 and m2 can be sent by external entities at every time.

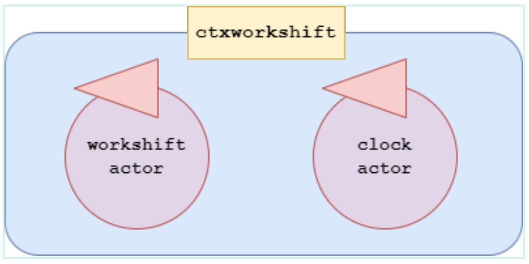
Requirement analysis

The consumer requires to design a simple system (using the ***QA-System*** given by the same consumer) that manages a machine in order to make it able to handle two different types of message in two dedicated time periods. The system must reproduce three different period (morning, afternoon and night) and the requirements say that:

- **m1** messages must be handled only in the **morning**;
- **m2** messages must be handled only in the **afternoon**;
- the machine must be **inactive** during the **night** (it sleeps);

Then, the system must reproduce the *flow of the days* of the reality but **nothing is specified about the duration** of the periods by the consumer.
Finally, the consumer specify that m1 and m2 messages can be sent by external entities at every time:

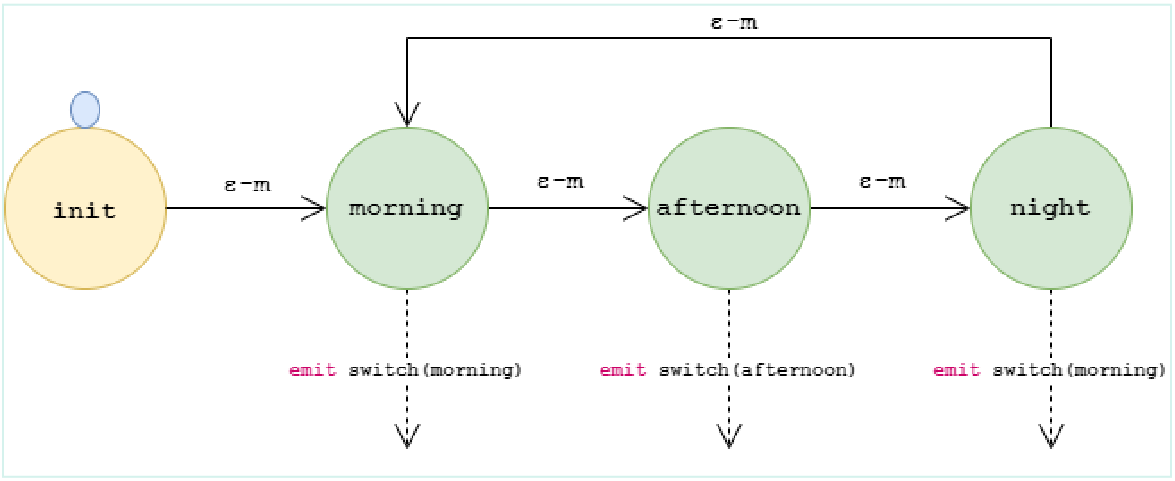
- the ***QA-System*** implicitly let to receive message over the network, then the requirement of the **external entities** is already satisfied by using a ***context*** appropriately set (see QAK21Intro);
- messages can be sent *at every time* so it means that a certain type of message can be received even if the machine isn't in the period dedicated to handle it; also this requirement is directly satisfied by the ***QA-System*** that provided a mechanism to enqueue messages.

According to the tools provided by the ***QA-System***, in order to meet the requirements and realize the new system, we need to introduce two actors: • ***workshiftactor*** that is the machine handling the two type of messages in their dedicated period; • ***clockactor*** that represents a sort of clock that *regulates* the flow of the periods by emitting **events.**	

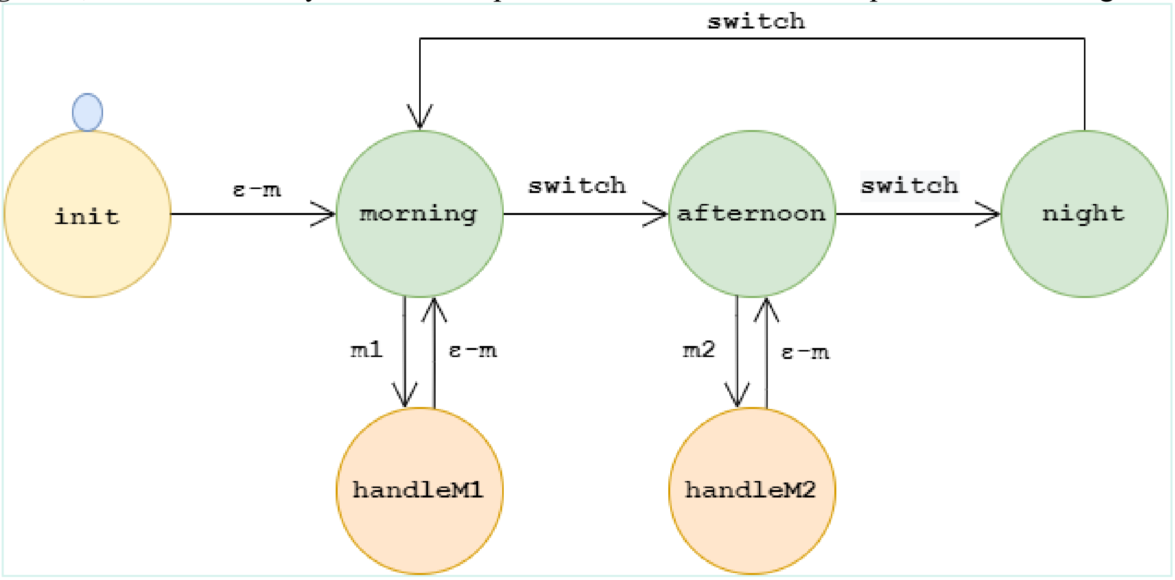
Problem analysis

workshift.qak
Zooming into the actors, we can say that:

- the **clockactor** simulate the flow of the time transiting into the states releated to the *three periods* (morning, afternoon and night), **emitting an event when entering in a state** and wait a certain time before passing to the next.



- the **workshiftactor** also transits in three state (one for each period) and wait for messages with the type described by the requirements; when an handable message was received, the actor transists to the relative *handling state*, then immediately return to the previous state when finish to process the message.



In the workshift.qak file, we also added two sender in order to make a more complete example.

By student
Name: Luca Marchegiani
Email: luca.marchegiani3@studio.unibo.it
Git Repo: https://github.com/LM-96/MarchegianiLuca