Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

Prof. Elisabetta Di Nitto, Raffaela Mirandola

20133 Milano (Italia)
Piazza Leonardo da Vinci, 32
Tel. (39) 02-2399.3400
Fax (39) 02-2399.3411

Software Engineering II

February 15 2016

Last Name
First Name
Id number (Matricola)

Note

1. The exam is not valid if you don't fill in the above data.
2. Write your answers on these pages. Extra sheets will be ignored. You may use a pencil.
3. The use of any electronic apparatus (computer, cell phone, camera, etc.) is strictly forbidden.
4. You cannot keep a copy of the exam when you leave the room.

Question 1 Alloy (7 points)
A software company is contracted to develop a controller for a microwave oven. The oven must respect a specific safety requirement: microwaves can be emitted only when the door of the oven is closed. By construction the oven ensures that the cooking timer can be activated only when the door is closed. The controller has to ensure that microwaves are emitted only when this is not forbidden by the safety requirement.
Applying the Jackson & Zave approach to this problem, the machine is our controller and the followings are a representation of the relevant phenomena:

- *doorOpen*: it represents the situation of the door open.
- *doorSensor*: it represents a sensor to check if the door is open. It is On when the door is open.
- timer: it represents the status (On/Off) of the timer.
- *microwavesAct*: it represents the activation (On/Off) of microwaves.

You are asked to answer to the following questions.

A. Identify from the list above the phenomena that are shared between the world and the machine and, among these, those that are controlled by the machine and those that are controlled by the world. Provide a justification for your answer.
B. Define an Alloy model for the described system. You can skip modeling of operations. Focus instead on defining signatures and facts.

Solution

Part A
doorOpen: This is a world phenomenon as the machine cannot see if the door is open. doorOpen is True when the door is open, False otherwise.
doorSensor: This is a shared phenomenon and is controlled by the world. More specifically, its value is determined by the status of the door. When doorSensor is On, the door is assumed to be closed (Not doorOpen).
timer: This is a shared phenomenon and is controlled by the world. In fact, the user can start the timer, provided that the status of the door does not prevent its activation.
microwavesAct: This is a shared phenomenon controlled by the machine. In fact, it is the task of the controller to activate the microwaves when the timer is started (and there are no safety issue) and to deactivate it when the timer is off.

Part B

```
abstract sig Status {}
one sig On extends Status {}
one sig Off extends Status {}

abstract sig BoolValue {}
one sig True extends BoolValue {}
one sig False extends BoolValue {}

sig Oven {
  doorOpen: one BoolValue,
  timer: one Status,
  doorSensor: one Status,
  microwavesAct: one Status
}{ (doorSensor = On iff doorOpen = False) and
   (microwavesAct = On implies doorSensor = On) and
   (timer = On iff microwavesAct = On) }

sig Controller {
```

```
  controlledOven: one Oven
}

fact noTwoControllerOnTheSameOven {
  no disj c1, c2: Controller | c1.controlledOven & c2.controlledOven != none
}

fact allOvenControlled {
  all o: Oven | o in Controller.controlledOven
}
```

Questions 2 Planning (6 points) and Design (5 points)

A public transportation company of a large urban area aims at opening a service for enabling sharing of electric vehicles. The vehicles can be cars, scooters and electric bicycles. The company provides its users with the charging stations for vehicles, and with a number of vehicles. Each user can either rent a vehicle (only one at a time), or use the charging stations provided by the company to recharge his own vehicle. The company handles bookings for both vehicles and charging stations.
The software system supporting the operation of the service should allow i) the customers to place/modify/remove their reservations, to return a vehicle or a charging station plug, ii) the company to enter information about vehicles and charging stations, to monitor the actual situation of its vehicles and charging stations, to charge users for the usage of their service. From the side of customers, the service should be accessible through a mobile device.

You have been contacted by the company to develop the software.
Planning

- Determine the size of the project by applying the Function Points approach. Provide a motivation for each choice you make.
- Assuming that we have estimated an effort for the project equal to 20 PM and that the project team is composed of 2 full time skilled engineers and 2 full time juniors, define a plan for the implementation and unit/integration/system testing of the software system. More specifically, identify tasks, their scheduling (symbolically indicate with M1, … Mn the n months of the project) and dependencies. Allocate team members to tasks.

Design

- Define the high level architecture of the software system. Describe each component and the way it is connected to the others.
- Define the UML sequence diagrams to describe the behaviour of the system in the following case:
    - A customer is using one of the vehicles of the company and the time slot he has declared at the reservation is expiring. Thus, the system sends to the customer information about the closest charging stations where he can leave the vehicle. The customer replies asking for charging stations available in a different area. The system provides them together with the information on the extra-time needed to reach these other charging stations (this extra-time is calculated by relying on an external map service).

Sketch of solution concerning function points calculation and first part of design

Function points calculation
ILF, all simple (weight 7)
Reservation
Vehicle

Charging station
User
Total 28

ELF, all medium (weight 7)
Vehicle
Charging station
PaymentGateway
Total 21

External Inputs, all simple (weight 3)
Login/logout
Registration
Update profile
Remove a reservation
Modify a reservation
Return vehicle
Return charging station plug
Enter information about vehicles and charging stations
Total 24

External Inquiries
Place a reservation, complex (weight 6)
Check the status of vehicles and charging stations, medium (weight 4)
Total 10

External output, all medium (weight 5)
Penalize defaulting customers
Monitor vehicles and charging stations
Charge users
Total 15

Grand total 98

Design
The system can be seen as composed of the following parts:
Front-end part
- GUI at the company site
- End user app

Application logic
- Profile manager, it manages login/logout, registration and provide update by end users
- Reservation manager
- Vehicle and Charging stations manager
- Connector to the payment gateway
- Vehicle and charging stations monitor
- Reservation monitor

While the first four components are typical server-side components that are activated upon the arrival of a request from the front-end part, the last two components receive a stream of data from

Data layer includes all entities to be persisted in the database (see the list included in the ILF)

Question 3 testing (4 points)

A. Consider the following statement: "The number of errors in a program strictly decreases with

successive versions. When errors are discovered, they are in fact removed by the next version of the application". Is this statement right? Give a short motivation for your answer (otherwise your answer, even if correct, does not count).

B. Consider the following test cases and indicate to which test phase each of them belongs to. Provide an explanation for your answers:

Case 1
Input: Call the main function of the system every 0.2 secs
Expected output: On average, the system answers within 0.1 secs

Case 2:
Input: Call method X of class A
Expected output: Check that the return value is the expected one.

Caso 3:
Input: Call method Y exposed by component K through its public interface
Expected output: Check if component L connected to K has modified its state as expected

Solution