Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

Prof. Elisabetta Di Nitto and Matteo Rossi

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

Software Engineering II

July 13th, 2020

Last Name
First Name
Id number (Matricola)

Notes

This exam is handled online. Rules

1. Only use a computer, NOT a tablet, NOR a smartphone
2. Activate the feed of your webcam
3. Share the screen of your computer
4. Keep the microphone on
5. No dual screens
6. No virtual machines
7. When you upload a file through the form, make sure to include your "person id" (the 8-digit number that starts with "10") in the name of the file, AT THE BEGINNING OF THE NAME (so the name of the file should be, say, "10143828_etc.", if your person id is "10143828").
8. The exam is open book, so you can check the course materials (notes, slides, books, past exams, etc.), which can be in paper form, or in electronic form. If the materials are in electronic form, you MUST use the same computer on which you are taking the exam to display them.
9. You cannot interact with other people during the exam.
10. The exam is composed of three exercises. Read carefully all points in the text!
11. **Total available time: 1h and 30 mins**

Scores of each question:

Question 1 (MAX 8) _____

Question 2 (MAX 5) _____

Question 3 (MAX 3) _____

Question 1 Alloy (8 points)
Consider a system that counts the number of passengers on public transports and at their respective stops by scanning what Bluetooth devices are in the vicinity of the vehicles and of the stops.
The system considers 2 types of public transportation means: buses and trams. Vehicles of different types are differently equipped, in the sense that buses are equipped with 2 Bluetooth devices (one at the front, one at the back), whereas trams, which are smaller, are equipped with only one Bluetooth device in the center of the vehicle. Bus/tram stops are also equipped with one Bluetooth device each. On the passenger side, each passenger has a mobile device, which also has a Bluetooth device. Each Bluetooth device has a unique identifier, and it can be used to scan what other Bluetooth devices are in its vicinity (the result of the scan is the list of ids of detected Bluetooth devices).
Each bus/tram and each stop has its own id. Vehicles and stops keep track of the Bluetooth ids that they detect through Bluetooth scans. More precisely, each vehicle/stop has an internal function that periodically retrieves and updates the list of Bluetooth ids that are detected in its vicinity through scans. Each vehicle/stop offers a function that allows clients to retrieve the current list of stored Bluetooth ids.

Point A (3 points). Define suitable signatures capturing the elements of the system described above, and in particular buses, trams, stops, Bluetooth devices, passengers, mobile devices.

Point B (2 points). Define a signature capturing the configuration of the system at some point in time, and a constraint stating the uniqueness of the elements' identifiers in the system (you do not need to express constraints other than the requested one about the uniqueness of the identifiers).

Point C (3 points). Define a predicate `updateDetectedDevices` formalizing the operation that, given a system configuration `conf` and given an element `elem` that is a stop/bus/tram of the system, updates both `conf` and `elem` by updating the list of detected Bluetooth devices that is stored with `elem` with all those that are currently sensed by its Bluetooth devices.

Solution

Point A

```
sig ID{}
sig WithID{
  id : ID
}
sig BluetoothDevice extends WithID {
  sensedIDs : set ID
}
sig DetectingElement extends WithID {
  detectedIDs : set ID
  btdevices : some BluetoothDevice
}
sig Vehicle extends DetectingElement {}
sig Stop extends DetectingElement {}{
  #btdevices = 1
}
sig Bus extends Vehicle {}{
  #btdevices = 2
}
sig Tram extends Vehicle {}{
```

```
  #btdevices = 1
}
sig MobileDevice{
  btdevice : BluetoothDevice
}
sig Passenger{
  mobile : MobileDevice
}
```

Point B.

```
Sig SystemStatus{
  vehicles : set Vehicle
  stops : set Stop
  passengers : set Passengers
}{
  let haveids = vehicles + stops +
              (vehicles+stops).btdevices + passengers.mobile.btdevice |
      all distinct wid1, wid2 : haveids | not wid1.id = wid2.id
}
```

Point C.

```
pred updateDetectedDevices( ss, ss’ : SystemStatus,
                            de, de’ : DetectingElement){
  // pre-condition
  de in ss.vehicles + ss.stops

  // post-condition
  ss’.passengers = ss.passengers
  de’.id = de.id
  de’.btdevices = de.btdevices
  de’.detectedIDs = de.btdevices.sensedIDs
  de in Stop implies ss’.vehicles = ss.vehicles and
                     ss’.stops = ss.stops – de + de’
             else ss’.stops = ss.stops and
                  ss’.vehicles = ss.vehicles – de + de’
}
```

Question 2 Requirement engineering (5 points)
Consider the system described in the previous exercise.

a.
Referring to the Jackson-Zave distinction between the world and the machine, identify:

- at least 2 world phenomena that are not shared with the machine;
- at least 2 machine phenomena that are not shared with the world;

- at least 1 shared phenomenon controlled by the world;
- at least 1 shared phenomenon controlled by the machine.

b.
Define in natural language one specific goal for the system, one domain assumption and one requirement referring to the phenomena identified above.

Solution
a.
World-only Phenomena
A passenger takes the bus
A passenger gets out of a bus
A passenger activates the Bluetooth on their mobile device

Machine-only Phenomena
The system stores the information about a sensed ID
The stop subsystem retrieves from its Bluetooth device the list of sensed IDs
The subsystem on a bus transfers the set of currently sensed IDs to the central subsystem

World-controlled shared Phenomena
A passenger's mobile device Bluetooth device broadcasts its ID to neighboring devices

Machine-controlled shared Phenomena
A detecting element on a bus receives the ID from a Bluetooth device

b.
Goal:
the organization wants to count the number of passengers entering in a bus/tram or waiting at a stop

Requirements:
the bus/tram/stop subsystem should periodically search for Bluetooth devices
the bus/tram/stop subsystem saves the passengers mobile devices IDs when received

Domain assumptions:
all passengers own a mobile device
all passengers keep their Bluetooth device on while traveling on public transports

Question 3 Project Management (3 points)

Consider again the passenger counting system.

In addition to the part counting people described in the previous sections, the system offers also to operators the possibility to look for specific buses/trams/stops and retrieve the number of passengers currently in the system. In addition, it allows operators to set, for each bus/tram/stop, the threshold of number of passengers over which the bus/tram/stop should be considered overcrowded. Finally, it periodically provides a report with the highest number of passengers reached in each bus/tram/stop, highlighting the overcrowded situations.

Calculate the function points for the part of the system targeted to operators. Motivate your choices.

Refer to the following table to associate weights to the function types:

	Weights		
Function types	Simple	Medium	Complex
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiry	3	4	6
Internal Logic Files	7	10	15
External Interface Files	5	7	10

Solution

ILF
Vehicles (vehicle ID, line number, capacity, Bluetooth device ID) → simple (weight 3)
Stops (stop ID, address) → simple (weight 3)
CollectedDeviceIDs (vehicle ID or stop ID, time of collection, deviceID) → medium as typically this is a table with a large number of elements (weight 4)

ILF tot = 10

EIF none

E Inputs
Set overcrowded threshold for nus/tram/stop → simple (weight 3)

E Outputs
Produce a periodic report with all deviceIDs per vehicle/stop → medium, as the number of elements to be considered and the visualization of the situation might not be trivial (weight 5)

E Inquiry
getDeviceID currently in a certain vehicle or stop → simple 3

Total number of FP = 10 + 3 + 5 + 3 = 21