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 2 – Written Exam 2 (WE2)

July 26th, 2021

Notes

1. Remember to write your name and Id number (matricola) on the pieces of paper that you hand in.
2. You may use a pencil.
3. Incomprehensible handwriting is equivalent to not providing an answer.
4. The use of any electronic apparatus (computer, cell phone, camera, etc.) is strictly forbidden.
5. The exam is composed of 2 parts, one focusing on requirements, and one focusing on design. Read carefully all points in the text!
6. **Total available time: 1h and 30 mins**

System Description: SmartAgriculture

A high-tech farmer has deployed in her fields a number of sensors measuring the humidity of soil and leaves and a number of cameras that continuously stream images of the plants, with the possibility to zoom in to analyse the status of their leaves and fruits. The sensors and cameras are connected to a central server in the farm to which they transfer data periodically. Moreover, they can be controlled through some programmatic interfaces through which it is possible to change the following parameters:

- For sensors: the frequency of data collection.
- For cameras: the frequency of image collection, the zoom percentage, the angle of the camera with respect to the nominal position. By changing the zoom percentage and the angle it is possible to collect images from different positions.

Starting from this situation, the farmer asks you to develop a software system that will be called *SmartAgriculture*. The system must analyse the humidity data and, based on this, automatically activate/deactivate the irrigation system. Moreover, it must oversee the field using the cameras and alert the farmer when 1) someone or a large animal is in the field, or 2) parasites are present on the leaves. The system must be configurable so that it can work properly depending on the specific types of plants, in particular their watering requirements and the type of parasites that attack them.

Part 1 Requirements (6 points)

Q1 (1 points)

With reference to the Jackson-Zave distinction between the world and the machine, identify the relevant world and machine phenomena for *SmartAgriculture*, including the shared ones, providing a short description if necessary. For shared phenomena specify whether they are controlled by the world or the machine. Focus in particular on phenomena that are relevant to describe the requirements of the system.

Q2 (2 points)

Referring to the phenomena identified above, define in natural language one specific goal for *SmartAgriculture*, together with the associated domain assumptions and requirements. Explain why the identified requirements and assumptions are relevant to the fulfillment of the goal.

Q3 (3 points)

Describe the relevant elements of the domain of the *SmartAgriculture* system. You can use a UML Class Diagram for this purpose.

Solution

Q1

World phenomena:

The farmer plants certain species
Some parasites attack the plants
The temperature of the air changes
Someone enters the field

Shared phenomena, world-controlled

The farmer enters in the system the information about the plants currently in the field
The sensors transmit their data to the central server
The cameras transmit their data to the central server

Shared phenomena machine-controlled
The system configures the cameras and sensors based on the information inserted by the farmer
The system controls the movement of cameras to inspect specific details
The system alerts the farmer when a human being or an animal is in the field
The system alerts the farmer when it recognizes the presence of some parasites on the plants
The system activates/deactivates the irrigation system

Q2

Goal
G1: The farmer wants to be aware of the presence of human beings, animals or parasites with a high level of precision and recall. This means that the number of real events that are not identified is very limited (precision) and that the number of false alarms is very limited as well (recall).

Requirements
R1: The system must allow users to insert the information about the plants in the field (including watering requirements and type of parasites that may attach them)
R2: The system must be able to collect data coming from sensors and camera with a percentage of information loss that is below a certain threshold
R3: The system must be able to recognize the presence of a moving element in the field and to understand whether it is a human being or an animal
R4: The system must be able to define the frequency of data collection on sensors
R5: The system must be able to define the frequency of data collection and to change the zoom ratio and the angle of cameras.
R6: The system must be able to analyze the images containing leaves details and recognize the presence of parasites, if any
R7: The system must be able to alert the farmer via an SMS on her cellular phone and a bell that can be heard from any place in the farm
Additional requirements concerning the control of watering are not listed here, because they are not necessary to meet the specific stated goal, though they are relevant for the application.

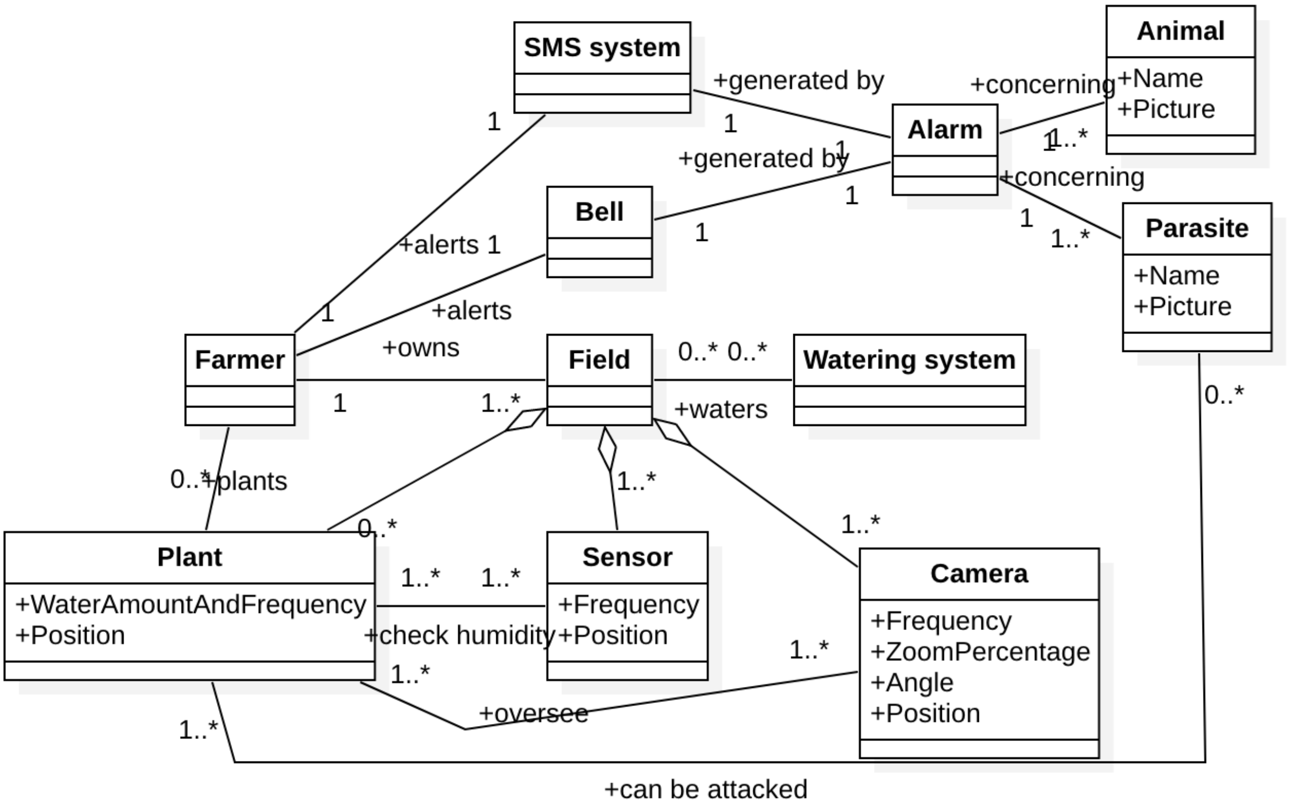
Domain assumptions
DA1: The network connecting sensors and cameras to the *SmartAgriculture* system works properly.
DA2: Sensors and cameras work properly.
DA3: The information inserted by the farmer and concerning the plants on the field are correct and precise
DA4: The external SMS transmission system works properly
DA5: The bell works properly
Additional domain assumptions concerning the control of watering are not listed here, because they are not necessary to meet the specific stated goal, though they are relevant for the application.

- DA1, DA2, R2, R4 and R5 guarantee that the system is able to acquire the information needed for the analysis.
- DA3, R1, R3 and R6 guarantee that the system is aware of the possible types of parasites that can attack the plants and it is, therefore, able to recognize them with the required level of precision and recall when receiving the data.
- DA4, DA5 and R7 guarantee that the farmer can be alerted when a problem occurs.

Q3

The following is a possible Class Diagram describing the domain of the application. We assume that the farmer is alerted through a bell installed in the farm and an SMS. Notice that the cardinalities 1-1 concerning the associations between the Alarm, SMS system, Bell and Farmer may be critical in a real case as they suggest the presence of some single points of failure (is the farmer the only one who can receive the SMS? Could the farmer involve also others? Could we replicate the alerting with a new SMS

/ a new bell ring in order to ensure that the right person is certainly alerted?). The aforementioned issues should trigger a new interaction with our farmer to make sure that the point is properly addressed.



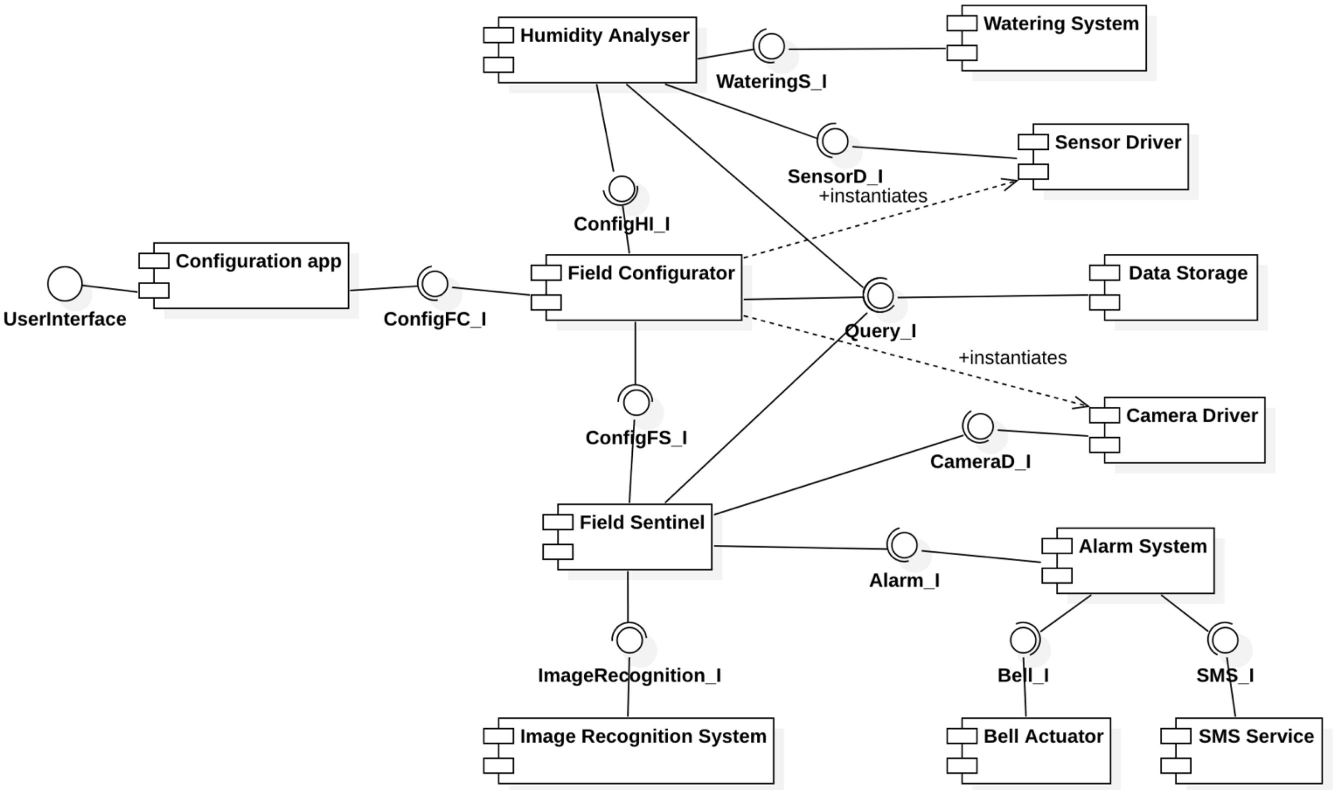
Part 2 Design (8 points)

Q4 (4 points)
Assuming you need to implement system *SmartAgriculture* analyzed above, identify the most relevant components and interfaces and describe them through a UML Component or design-level Class Diagram.

Q5 (4 points)
Provide a brief description of each component identified in Q4 and list the operations provided by the corresponding interfaces.
For each operation, you do not need to precisely specify its parameters; however, you should give each operation a meaningful enough name to understand what it does; you can also briefly describe what information the operations use/produce.

Solution

Q4



Q5

Configuration app

This is the front-end that allows the farmer to introduce in the system information concerning the plants and to configure the system itself. Through the interface *UserInterface*, it offers the following operations:

- AddNewPlant with parameters name, watering requirements, list of common parasites.
- ModifyPlant
- DeletePlant
- InsertSensorsConfiguration with parameter a list of data about sensors (position, frequency). This operation results in the sensor data to be stored in the Data Storage.
- InsertCameraConfiguration with parameter a list of data about cameras (position, frequency, zoom percentage, angle)

In the execution of all the above operations, the Configuration app will interact with the Field Configurator through its ConfigFC_I interface.

Other features could be provided to the end users through the Configuration app, for instance, to allow the user to acquire information about the latest operations activated by the system as well as to monitor the status of sensors and cameras. For the sake of simplicity, given that these operations were not required in the system description, we have decided not to include them. They could be the subject of a new version of the system.

Data Storage

This is the database storing information about the plants, sensors and cameras. It offers an interface (Query_I) for the typical CRUD (Create, Read, Update, Delete) operations. Such interface is used by the three main components of the system (Field Configurator, Humidity Analyzer and Field Sentinel).

Field Configurator

This component is the one that realizes the configuration operations performed by the Configuration app. As such its operations are the same as the ones defined for the Configuration app:

- AddNewPlant/ModifyPlant/DeletePlant: the relevant information is stored in the Data Storage.
- InsertSensorsConfiguration with parameter a list of data about sensors (position, frequency). This operation results in the sensor data to be stored in the Data Storage. Moreover, all needed Sensor

Drivers are instantiated and their reference is passed to the Humidity Analyser through the ConfigHA_I interface.
- InsertCameraConfiguration with parameter a list of data about cameras (position, frequency, zoom percentage, angle). This operation results in the camera data to be stored in the Data Storage. Moreover, all needed Camera Drivers are instantiated and their reference is passed to the Field Sentinel through the ConfigFS_I interface.

Watering System
This is the external component actuating the irrigation. Its interface offers the operations to switch on and off the water flow and to regulate such flow.

Sensor Driver
This component interacts with a Sensor to configure it. Moreover, it collects the data it produces. We assume we have one Sensor Driver per sensor. Therefore, we can have multiple instances of this component in the system. As stated in the text, sensor data is stored in the central server. So, we assume that Sensor Driver finds such data in a file with a proper name. The interface of this component (SensorD_I) offers the following operations:

- ConfigureSensor with parameter frequency of data collection
- AcquireNewlyProducedData. This operation returns the new data present in the central server.
- GetSensorConfiguration. This operation returns the data collection frequency and the position of the sensor.

Camera Driver
This component interacts with a Camera to configure it. Moreover, it collects the images produced by the camera. As stated in the text, images are stored in the central server. So, we assume that this component finds images in a specific directory in the file system. The interface of this component (CameraD_I) offers the following operation:

- ConfigureCollectionFrequency with the new frequency as parameter.
- AcquireNewlyProducedImages This operation returns the new images present in the central server.
- ModifyCameraZoomRatio with the new zoom ratio as parameter.
- ModifyCameraAngle with the new angle as parameter.
- GetCameraConfiguration. This operation returns the entire configuration of the camera.

Alarm System
This component activates the Bell Actuator and the SMS Service. Its main purpose is to decouple the system from these two different alerting mechanisms required by our customer. Its interface offers a single operation that is SendAlert. The phone number to which the SMS must be sent is a parameter of this operation.

Bell Actuator
This is the external system that activates the bell.

SMS Service
This is the external system that sends the SMS to the phone number passed as parameter.

Humidity Analyser
This component activates periodically. It interacts with the Sensor Drivers to acquire the newly generated data by the sensors and controls the watering system accordingly. It accesses the Data Storage to acquire the information about the watering requirements. Through the interface ConfigHA_I it offers the following operation:
ConnectToSensorDriver with parameter a Sensor Driver to interact with.

Image Recognition System

This component performs image recognition. Through its interface it offers three operations:

- RecognizeHumanBeing: this receives an image as a parameter and returns true if a human being is present in the image
- RecognizeAnimal: this receives an image as parameter and returns true if an animal is present in the image
- RecognizeParasite: this receives images with leaves and images with parasites and returns true if the parasites appear on the leaves.

Field Sentinel

This component activates periodically and oversees the cameras operation through the Camera Drivers. It performs the following steps:

- Collect the newly generated images concerning the field overview.
- Sends such images to the Image Recognition System calling the RecognizeHumanBeing and RecognizeAnimalOperations.
- If human beings or animals are found, then it interacts with the Alarm System.
- Interacts with the Data Storage to know which plants are on the field.
- In a loop, until all possible camera positions at the maximum zoom ratio are used:
    - Interacts with the Camera Drivers to zoom in or change the angle
    - Collects images of the leaves from the Camera Drivers
    - Calls the Image Recognition System to recognize parasites. We can assume that the images of parasites are stored with the information about the plants.
- If at least a parasite is found, calls the Alarm System.

Through the interface ConfigFS_I the offers to the Field Configurator the following operation:

- ConnectToCameraDriver with parameter a Camera Driver to interact with.