Workplan Automated Car-Parking

Requirements

A company intends to build an *automating parking service* composed of a set of elements:

- A software system, named ***ParkManagerService***, that implements the required automation functions.
- A DDR robot working as a ***transport trolley***, that is intially situated in its ***home*** location. The transport trolley has the form of a square of side length ***RD***.
- A ***parking-area*** is an empty room that includes;
  - an ***INDOOR*** to enter the car in the area. Facing the INDOOR, there is a ***INDOOR-area*** equipped with a ***weigthsensor*** that measures the ***weigth*** of the car;
  - an ***OUTDOOR*** to exit from the parking-area. Just after the OUTDOOR, there is ***OUTDOOR-area*** equipped with a ***outsonar***, used to detect the presence of a car. The OUTDOOR-area, once engaded by a car, should be freed within a prefixed interval of time ***DTFREE***;
  - a number N (N=6) of ***parking-slots***;
  - a ***thermometer*** that measures the temperature ***TA*** of the area;
  - a ***fan*** that should be activated when **TA > TMAX**, where ***TMAX*** is a prefixed value (e.g. 35)

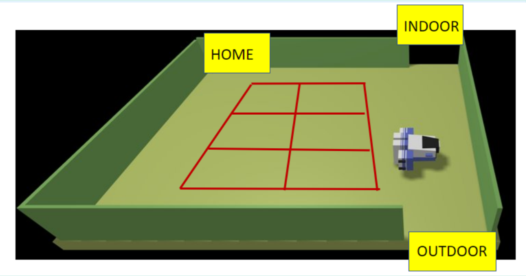
  A ***map*** of the parking area, represented as a grid of squares of side length ***RD***, is available in the file parkingMap.txt:

  ```
  |r, 0, 0, 0, 0, 0, 0, X,
  |0, 0, X, X,  0, 0, 0, X,
  |0, 0, X, X,  0, 0, 0, X,
  |0, 0, X, X,  0, 0, 0, X,
  |0, 0, 0, 0, 0, 0, 0, X,
  |X, X, X, X, X, X, X, X,
  ```

  The map includes the positions of the parking-slots (marked above with the symbol **X**) and of the ***fixed obstacles*** in the area (the walls marked with the symbol **X**).

  The area marked with **X** is a sort of 'equipped area' upon which the transport trolley cannot walk. Thus, to get the car in the parking-slot **(2,2)**, the transport trolley must go in cell **(1,2)**.

  The proper scene for the WEnv is reported in: parkingAreaConfig.js



- a ***parking-manager*** (an human being) which supervises the state of the parking-area and handles critical situations.

The job of our company is to design, build and deploy the ***ParkManagerService***.

User stories

As a **client - parking phase** :

- I intend to use a ***ParkServiceGUI*** provided by the ParkManagerService to notify my interest in *entering* my auto in the parking-area and to receive as answer the number ***SLOTNUM*** of a free parking-slot (1<=SLOTNUM<=6). SLOTNUM**==0** means that no free slot is available.
- If SLOTNUM **>0**, I move my car in front to the INDOOR, get out of the car and afterwards press a ***CARENTER*** button on the ***ParkServiceGUI***. Afterwards, the transport trolley takes over my car and moves it from the INDOOR to the selected parking-slot. The ParkServiceGUI will show to me a receipt that includes a (unique) ***TOKENID***, to be used in the *car pick up* phase.

As a **client - car pick up phase** :

- I intend to use the ParkServiceGUI to submit the request to pick up my car, by sending the TOKENID previously received.
- Afterwards, the transport trolley takes over my car and moves it from its parking-slot to the OUTDOOR-area.
- I move the car, so to free the OUTDOOR-area.

As a **parking-manager**:

- I intend to use the ***ParkServiceStatusGUI*** provided by the ParkManagerService to observe the ***current state*** of the parking area, including the value TA of the temperature, the state of the fan and the state of the transport trolley (**idle, working or stopped**).
- I intend to ***stop*** the transport trolley when **TA > TMAX**, activate the fan and wait until **TA < TMAX**. At this time, I stop the fan and resume the behavior of the transport trolley. Hopefully, the **start/stop of the fan** could also be automated by the ParkManagerService, while the **start/stop of the transport trolley** is always up to me.
- I expect that the ParkManagerService sends to me an ***alarm*** if it detectes that the OUTDOOR-area has not been cleaned within the DTFREE interval of time.

Requirements

The ParkManagerService should create the ParkServiceGUI (for the client) and the ParkServiceStatusGUI (for the manager) and then perform the following tasks:

- ***acceptIN***: accept the request of a client to park the car if there is at least one parking-slot available, select a free slot identified with a unique SLOTNUM.
  A request of this type can be elaborated only when the **INDOOR-area is free**, and the transport trolley is at home or working (**not stopped** by the manager). If the INDOOR-area is already engaged by a car, the request is not immediately processed (the client could simply wait or could - optionally - receive a proper notice).
- ***informIN***: inform the client about the value of the SLOTNUM.

If **SLOTNUM>0**:

1. ***moveToIn***: move the transport trolley from its current localtion to the INDOOR ;
2. ***receipt***: send to the client a receipt including the value of the TOKENID ;
3. ***moveToSlotIn***: move the transport trolley from the INDOOR to the selected parking-slot;
4. ***backToHome***: if no other request is present, move the transport trolley to its home location, else ***acceptIN*** or ***acceptOUT***.

If **SLOTNUM==0**:

- ***moveToHome***: if not already at home, move the transport trolley to its home location.

- ***acceptOUT***: accept the request of a client to get out the car with TOKENID. A request of this type can be elaborated only when the **OUTDOOR-area is free** and the transport trolley is at home or working (**not stopped** by the manager). If the OUTDOOR-area is still engaged by a car, the request is not immediately processed (the client could simply wait or could - optionally - receive a proper notice).

1. ***findSlot***: deduce the number of the parking slot (***CARSLOTNUM***) from the TOKENID;
2. ***moveToSlotOut***: move the transport trolley from its current localtion to the CARSLOTNUM/parking-slot ;
3. ***moveToOut***: move the transport trolley to the OUTDOOR ;
4. ***moveToHome***: if no other request is present move the transport trolley to its home location;
   else ***acceptIN*** or ***acceptOUT***

- ***monitor***: update the ParkServiceStatusGUI with the required information about the state of the system.

- ***manage***: accept the request of the manager to stop/resume the behavior of the transport trolley.

About the devices

All the sensors (weigthsensor, outsonar, thermometer) and the fan should be properly simulated by mock-objects or mock-actors.

When using a real robot

No further requirement.

When available a Raspberry and a sonar

The outsonar could be a real device. We can simulate the presence/absence of a car.

When using only the virtual robot or no real sonar available

Consider the new requirement:

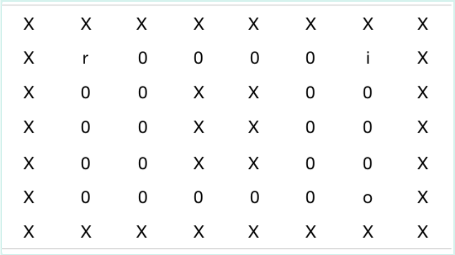
- **authorize**: allow a manager to use the ParkServiceStatusGUI only if she/he owns **proper permissions**.

Requirement analysis

Our interaction with the customer has made it clear what he means for:

- DDR robot: an device capable of moving from the commands received via the network,as described in the document VirtualRobot2021.html provided by the customer;;
- transport-trolley: a DDR robot capable of loading a car and transporting it within the parking-area. The seguent link is a software offered by the costumer: it.unibo.qak21.basicrobot
- parking-area: an empty room showed in the map, where:

• "o" is the OUTDOOR-area; • "i" is the INDOOR-area ; • "r" is the home ; • "X" are the parking-slots.	

```
X X X X X X X X
X r 0 0 0 0 i X
X 0 0 X X 0 0 X
X 0 0 X X 0 0 X
X 0 0 X X 0 0 X
X 0 0 0 0 0 o X
X X X X X X X X
```

- room : a conventional room of a house;
- car : a conventional car;
- home: an area where the robot is at the beginning and where it goes when it hasn't any requests;
- INDOOR-area: the area in front of the parking INDOOR where is the weightsensor and the trasport trolley loads the car;
- OUTDOOR-area: the area in front of the parking OUTDOOR where is the OUTsonar and the trasport trolley releases the car;
- parking-slots: number of slots in the parking where the robot parks the car;
- weightsensor: an simulate device capable of detecting the presence of a car and measuring its weight. It is in the INDOOR-area and sends the information taken via the network as event emitter;
- outsonar: an simulate or (optionally) a real device capable of detecting the presence of a car. It is in the OUTDOOR-area and sends the information taken via the network as event emitter. In the case it is a real device the costumer gives us the seguent software: SonarAlone.c;
- thermometer: an simulate device capable of detects the temperature in the parking-area. It sends the information taken via the network the network as event emitter;
- fan: a device apable to lower the temperature of the parking. It could be in on/off state and it is possible switch the state with dispatch messages via the network;
- SLOTNUM: unique identifier of a parking-slot;
- TOKENID : unique string assigned to a client when his entrance request was accepted;
- client: an entity who sends meemessages to the ParkerServiceManager via the network;
- ParkServiceGUI : an user interface that allows a client to comunicate with the parkManagerService;
- CARENTER: a ParkServiceGui graphic element that allows an user to notify the intention to let the car in;
- ParkServiceStatusGUI: an user interface that allows a parking-manager to comunicate with parkManagerService using graphic representations and that shows the parking-area current state;
- current state: TA temperature, state of the fan (on/off), state of the trasport trolley (idle, working or stopped);
- stop: when the robot isn't in a idle or working state;
- alarm: a message that has been sent by the ParkManagerService to the ParkServiceStatusGUI if the OUTDOOR-area has not been vacated within an interval of time DTFREE;
- fixed obstacles : any fixed element in the parking-aerea which the robot could collide with (e.g. a wall);
- request : a message which was sent by the client to the ParkerServiceManager in order to receive a service.

Problem analysis

To destroy the abstraction gap is possible to use QACTOR which is a modelling language and defines a work model of the system based on actors behavior.

Components

ParkManagerService

ParkManagerService application server that interacts via network with OUTSonar, weightsensor, DDR robot, thermometer and fan. It should build ParkServiceGUI and ParkServiceStatusGUI. The ParkManagerService could be modelled as a QActor.

The transport-trolley

The robot, which has the trasport trolley behavior, was given by the costumerand it is described in the file: VirtualRobot2021.html. The comunication via the network should happened in two different ways:

- sending messages to the port 8090 using HTTP POST protocol
- sending messages to the port 8091 using websocket

The cominication with the the virtual robot will be managed by a QActor called trasport trolley which will be developped starting from the "basicrobot". The documentation of the basic robot is avaiable in the repository it.unibo.qak21.basicrobot.
It is preferable to use an asynchronous comunications to reduce the number of messages exchanged between the componends and to avoid that the application waits any answers. The transport trolley will be modeled as a QActor.

ParkServiceGUI and ParkServiceStatusGUI

The system should present two GUI: ParkServiceStatusGUI and ParkServiceGUI. Because it is preferable that the grafic interfaces will be portable, that are indipendent from hardware components and from the operative system of the user device, the choise will be to realize the Web-application in the following way:

- they could be two HTML pages both with INPUT sections (e.g. CARENTER, or STOP) and OUTPUT sections (e.g. TOKENID, CURRENT STATE).
- The ParkServiceStatusGUI should need an authentication/identification mechanism. To beeing authorized the parking-manager will provide an identification test: for example a pawssword choosen by the parking-manager.

The techology should be SpringBoot to reduce development times.
It is preferable use a request/response model for the ParkServiceStatusGUI interface because when the client send a command the application must answer with helpful informations (e.g. CARENTER wait TOKENID)
The ParkServiceStatusGUI should monitor the the system current state. It should ask to the application the information needed. The implementation could be done with a polling mechanism, that is a request/response interaction, but this is only one possible solution. Another one could be a dispatch interaction, or using an observer.

OUTsonar, weightsensor and thermometer

In our analysis, this element is an autonmous active component that does not 'known' any other component. Thus, it is actually modelled as an emitter of events.

- Let us name sonar the component chosen by the user to generates sonar-data.
- Let us name weightsensor the component chosen by the user to generates weightsensorr-data.
- Let us name thermometer the component chosen by the user to generates thermometer-data.

The costumer gives us the software: SonarAlone.c. We should build a first executable model with QActors and then use them to developped the interaction with a real device.
All others sensors will be modeled like QActors.

Fan

In our analysis the component will be activated by the ParkManagerService or the parking-manager.
The costumer didn't given any software or device about the component but the costumer asked us to use mock-actor or mock-objects to simulate the behavior.
The ParkManagerService and the parking-manager will communicate with the fan with dispatches because it isn't necessary a response to continue the esecution.
The fan will be modeled as a QActor.

A first executable model

Client- parking and pickup phase

Assumptions:

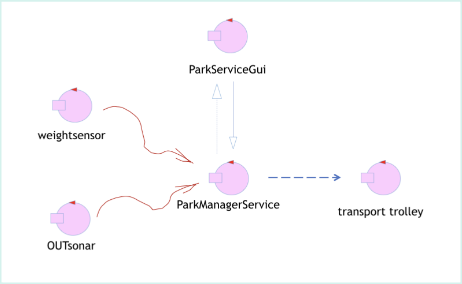
1. one client simulation;
2. no map contraints ;
3. temperature under TMAX -> fan off;
4. free sensors;
5. free slots;
6. trasport trolley states: idle or working. Never stopped.

In the following executable model there is the representation of the parking phase as described in the users stories given by the costumer.

In this first phase the trasport trolley was modeled just like an actor that recived the information by the ParkManagerService actor. Therefore the map wasn't taken into consideration. Furthermore because it is one client simulation the TOKENID wasn't taken into consideration too.

To move the trolley (to the indoor area) the ParkManagerService wait an event sent by the weightsensor.

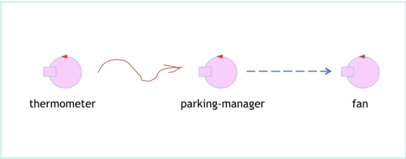
When the trasport trolley leaves the car in the OUTDOOR area the ParkManagerService recived an event sent by the outsonar. It inform the ParkManagerService that the OUTDOOR area is occupied by a car. The executable model is present in the following file: clientPhase.qak.



Parking-manager

Assunzioni:

1. no allarm simulation;
2. no comunication with ParkManagerService and transport trolley;
3. initially temperature under TMAX -> fan off;

In the following executable model there is the representation of the parking manager that turn on the fan when T is higher than TMAX. To to this the manager check the informations of the temperature recived as events by the thermometer. When is higher than TMAX it sent a dispatch fanstart to the fan to turn on it. The executable model is present in the following file: manager.qak.	

It should be necessary using a data structure where we should insert the persistent data of the system, for example occupied parking-slots and the TOKENID assigned to the client.
We stimate that the system should be developped in 45 days.

Product Backlog

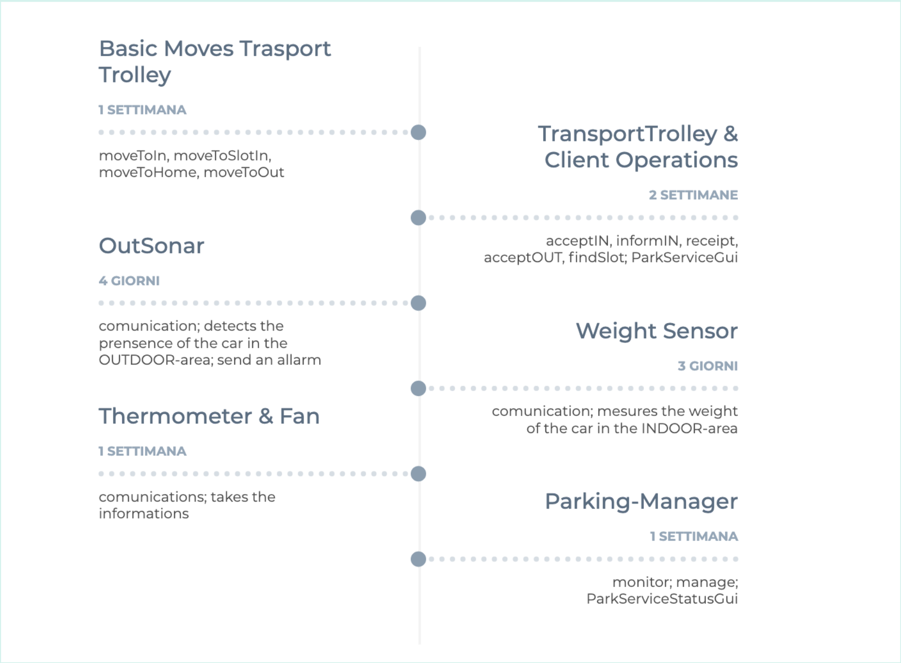
After an initial analysis of the requirements proposed by the costumer, we propose the following list of priorities of requirements (Max priority = 1):

- the transport-trolley carries the car to the parking space assigned by the ParkerManagerService (1);
- the transport-trolley carries the car to the the exit after having picked it up from the parking-slots (1);
- the transport-trolley mustn't pass over the parking-slots (2);
- the transport-trolley go back to its house if it hasn't requests (2);
- the client request to enter in the parking-area using the ParkServiceGui (2);
- the client request to exit from the parking-area using the ParkServiceGui (2);
- theOUTsonar detects the presence of a car in the OUTDOOR-area(3);
- the weigthsensor detects and measures the weight of the car in the INDOOR-area(3);
- the ParkManagerService turns on the fan if the parking area temperature is higher than TMAX (4);
- the ParkManagerService turns off the fan if the parking-area temperature is lower than TMAX (4);
- the transport-trolley will be stopped by the parking-manager if the thermometer mesures a temperature higher than TMAX (4);
- the transport-trolley will be resume by the parking-manager if the thermometer mesures a temperature lower than TMAX and if the robot was stopped (4);
- the parking-manager turns off the fan if the parking area temperature is lower than TMAX (5);
- the parking-manager turns on the fan if the parking area temperature is higher than TMAX (5);
- the ParkManagerService sends an alarm to the ParkServiceStatusGUI if the OUTSonar detects that the OUTDOOR-area has not been vacated within the DTFREE time interval (5);
- the parking-manager views the parking-area current status using the ParkServiceStatusGUI (5);
- the parking-manager must authenticate himself to access the ParkServiceStatusGUI (6).

Sprints

Sprint	Sprint Backlog
Realize basic movements of the	To realize basic movements of the trasport trolley (moveToIn, moveToSlotIn, moveToHome, moveToOut), the application should be able to:

transport trolley	• generate a sequence of elementary actions in order to reach the assigned places; • provide a map of the environment to verified the transport-trolley's position; • handle the transport-trolley positioning constraints inside the parking area; • planning the transport-trolley's movements to move it in an organized way; • communicate with the transport-trolley.
Realize Client operations	To carry out the operations requested by the client (acceptIN, informIN, receipt, acceptOUT, findSlot) it is necessary that the application, in addition to being able to perform the basic transport-trolley moves, is capable of: • handle the TOKENID, SLOTNUM, CARSLOTNUM; • the client send/receive information or command to/from client; • check the availability of resources, INDOOR-area, OUTDOOR-area, parking-slot, verification of the presence of client requests; • tell to the robot how it should behave given the customer's requests; • realize the ParkServiceGui.
Realize the management of information sent by the OUTSonar	To handle the information sent by the OUTsonar, the application must be able to: • communicate with the l'OUTSonar; • manage the information received from OUTSonar relating to the status of the OUTDOOR-area; • send an alarm after a DTFREE time if the OUTDOOR-area is not yet vacated.
Realize the management of information sent by the weigthsensor	To handle the information sent by the weightsensor, the application must be able to: • communicate with the weightsensor; • manage the information received from weightsensor relating to the status of the OUTDOOR-area.
Realize the management of information sent by the thermometer and fan	To handle the information sent by the thermometer and fan, the application must be able to: • communicate with the thermometer; • communicate with the fan; • manage the information received from thermometer relating to the temperature of the parking-area; • send the stop / start commands to the fan.
Realize parking-manager operations	To carry out the operations requested by the parking-manager (manage, monitor), the application must be able to: • retrieve all the information needed by the parking-manager; • communicate with the parking-manager; • forward the stop / resume commands received from parking-manager to the transport-trolley; • forward the stop / start commands received from al parking-manager to the fan; • realize the ParkServiceStatusGUI; • update the ParkServiceStatusGUI with information regarding the system status: TA, state fan, state transport-trolley; • authorize the parking-manager to use ParkServiceStatusGUI only after he was being authenticated.



Basic Moves Trasport Trolley

1 SETTIMANA

moveToIn, moveToSlotIn, moveToHome, moveToOut

TransportTrolley & Client Operations

2 SETTIMANE

acceptIN, informIN, receipt, acceptOUT, findSlot; ParkServiceGui

OutSonar

4 GIORNI

comunication; detects the prensence of the car in the OUTDOOR-area; send an allarm

Weight Sensor

3 GIORNI

comunication; mesures the weight of the car in the INDOOR-area

Thermometer & Fan

1 SETTIMANA

comunications; takes the informations

Parking-Manager

1 SETTIMANA

monitor; manage; ParkServiceStatusGui

Testplan

Testplan 1: we should check if the application return the correct number or if all the slot are occupied the SLOTNUM is 0. The code is present in the followiong file ParkmanagerserviceTest.kt.

Testplan 2: we should simulate and check if the parking-manager turn on/off the fan if the temperature is higher/lowre than TMAX. The code is present in the followiong file ThermometerTest.kt.

Deployment

The project is located in the repository:
https://github.com/noemival/ParkManagerService_2021/tree/main/it.unibo.parkManagerService

By studentName email: antonio.iacobelli@studio.unibo.it

By studentName email: noemi.valentini5@studio.unibo.it