Proposal for an application

Lisa Cattalani and Andrea Pari
Alma Mater Studiorum - University of Bologna, via Venezia 52
47023 Cesena, Italy
lisa.cattalani@studio.unibo.it, andrea.pari6@studio.unibo.it

April 5, 2017

1 Introduction

The current report describes the entire software development process adopted to analyze, design and implement a specific software system. The process has been divided into two phases: the first phase concerns the definition of a model of the software system, while the second phase concerns the implementation of the software system carried out after that the product owner has viewed the model.

2 Vision

We want to show how to manage the development of a distributed, heterogeneous and embedded software system by focusing on the analysis and design of the system. The software system at issue will be designed and implemented as an application in the field of *Internet of Things*.

3 Goals

The goal is to build a software system able to evolve from an initial prototype (defined as result of a problem analysis phase) to a final and testable product. This goal will be achieved by working in a team and by "mixing" in a proper (pragmatically useful) way an agile (SCRUM) software development with modeling.

4 Requirements

A differential drive robot (called from now on robot) must reach an area (B) starting from a given point A. To reach the area B, the robot must cross an area equipped with N (N>=1) distance sensors (sonars). The signal emitted by each sonar is reflected by a wall put in front of it at a distance of approximately 90 cm. Moreover:

- The section of the wall in front of each sonar is painted with a different illustration.
- The robot is equipped with a distance sensor (sonar) and (optionally) with a Web Cam both positioned in its front. It owns also a Led
- The robot should move from A to B by travelling along a straight line, at a distance of approximately 40-50 cm from the base-line of the sonars.

Design and build a (prototype of a) software system such that:

- shows the sonar data on the GUI associated to a console running on a conventional PC. For example (see the project it.unibo.qactor.radar):
- evaluates the expression: (sk + sk+1 + ... sN) / (N-k+1) where k is the number of the first sensor not reached by the robot and sk is the value of the distance currently measured by that sensor. If the value of the expression is less than a prefixed value DMIN(e.g. DMIN=70), play an alarm sound.

- when the robot reaches the area in front of a sonar, it
  - first stops
  - then rotates to its left of approximately 90 degrees
  - starts blinking a led put on the robot
  - takes a photo of the wall (in a simulated way only, if no WebCam is available) and sends the photo to console by using the MQTT protocol
  - rotates to its right of approximately 90 degrees to compensate the previous rotation
  - stops the blinking of the led and continues its movement towards the area B
- when the robot leaves the area in front to the last sonar, it continues until it arrives at the area B
- stops the robot movement as soon as possible:
  - when an obstacle is detected by the sonar in front of the robot
  - when an alarm sound is played
  - the user sends to the robot a proper command (e.g. STOP)
- makes it possible to restart the system (by manually repositioning the robot at point A) without restarting the software

5 Requirements analysis

Shown below are the terms that are present in the text and describes the requirement for what we suppose to be necessary specify the meaning to understand the application domain of the system and avoid eventually ambiguity, misunderstanding, or omissions. (the terms are order compared in the text). Di seguito vengono riportati quei termini presenti nel testo che descrivono i requisiti per cui si ritiene necessario specificare il significato al fine di comprendere il dominio applicativo del sistema ed evitare eventuali ambiguità, incomprensioni od omissioni. (I termini sono posti in ordine di comparsa nel testo).

Terms	Meaning	Synonyms
Differential drive robot	The differential drive robot is a two-wheeled drive robot system with independent actuators for each wheel. The name refers to the fact that the motion vector of the robot is sum of the independent wheel motions	Robot
Distance sensors	Device that allow to locate the presence of an object in front of it.The sensor sends waves, behind the time of the response of the waves it understands the distance to the object.	Sonar, sensor
Prototype	A prototipe is a version of a system software used to demostrate concepts, to try project option and, to find out more about problems and possible solution.	
GUI	Graphics User Interface associated to the console.Thanks to this interface the user must visualize the data trasmitted by the sonar.(del robot o dei raspberry sul muro?)	
Console	Is the means by which the user and the robot interact each other. The console receiv the pictures from the robot send by the mqtt protocol.	
Alarm sound	Is the sound emitted by the console when the calculated value of the expression is less than the predetermined threshold.	
MQTT protocol	Comunication protocol through which the robot sends pictures to the console.	
Obstacle	Any kind of object locate by the sonar of the robot that prevent to reach the area B.	
User	Final adopter of the system which is able to give commands to robot.	
Commands	Commands give from the user to the robot by the console.There are two possible commands: STOP(from the console) to stop the movement of the robot, RESTART (manually) to reposition the robot at point A.	
Restart the system	Processo attraverso il quale il robot viene riposizionato nel punto A senza dover fare il restart del software.	

5.1 *Modello dei casi d'uso*

Un user story rappresenta un bisogno dell'utente finale espresso in linguaggio naturale e comprensibile da chiunque. Essa permette di descrivere con sufficiente precisione il contenuto di una funzionalità da sviluppare. La frase contiene generalmente tre elementi descrittivi della funzionalità: CHI, COSA, PERCHE'.

Per modellare i casi d'uso si utilizzerà il seguente formato: "As a <ROLE>, I want <GOAL> so that <BENEFIT>"

- as a **user**, I want that the robot moves from point A to area B
- as a **user** I would the robot to stop when being in front of a sonar and turn of about 90° so that it can take a photo and send it to the PC, and after keep moving forward.
- as a **user** I want that an alarm sound is played if the distance calculated by the expression is under a certain threshold.
- as a **user**, I want to show the sonar data on the GUI
- as a **user**, I want to use the console to send commands to the robot
- as a **user**, I want to see on the console the photos sent by the robot
- as a **user**, I want to restart the system (by manually repositioning the robot at point A) without restarting the software.

6 User Interaction

Per rappresentare l'interazione tra agenti esterni e il sistema utilizzeremo diagrammi uml. L'unico attore è l'utente che interagisce con il sistema attraverso una serie d'azioni/comandi che rappresentano un caso d'uso nel diagramma. Il sistema è composto principalmente dalla console e dal robot.

NB: caso d'uso solo quello che può far direttamente l'utente, quindi mettere il robot, inviargli i comandi e guardare sul radar la posizione del robot. Scenari: Console , il robot,

Name	Place Robot
Descrizione	Piazzare il robot
Attori	Utente
Precondizioni	La scelta del punto determinerà la traiettoria che poi avrà il robot.
Scenario principale	L'utente metterà il robot in un punto A. A partire da quel punto il robot potrà iniziare a muoversi.
Scenari alternativi	—
Postcondizioni	Il robot deve riavviarsi ogni volta che viene posizionato.

Name	Send Command
Descrizione	Invio di un comando al sistema da parte dell'utente
Attori	Utente
Precondizioni	La console è attiva e funzionante sul computer e il sistema è pronto a ricevere comandi.
Scenario principale	1. L'utente invia un comando al robot tramite il pannello dei comandi presente sul PC. I comandi inviati sono START per azionare il robot e STOP per fermarlo. 2. Il server invia il comando al robot. 3. Il robot valuta il comando e lo esegue.
Scenari alternativi	—
Postcondizioni	Il robot si aziona se riceve il comando START, si ferma se riceve il comando STOP.

Name	View Sonar
Descrizione	Visualizzare la posizione del robot sul sonar
Attori	Utente
Precondizioni	La console dovrà funzionare sul pc.
Scenario principale	L'utente potrà vedere il robot nella GUI e si potranno seguire i movimenti.
Scenari alternativi	—
Postcondizioni	Il robot si aziona se riceve il comando START, si ferma se riceve il comando STOP.

7 Problem analysis

8 Problem analysis

I thought that my two entities have to speack each other, so I need a third entity, a generic Control. A part of this, the fact that the DevicePhoto have to stop after take a photo and found a marker, I considered that as a asyncAction, infact if the marker is not found, the DevicePhoto continued after PT times to takes a photo.

9 Project

I thought that my two entities have to speack each other, so I need a third entity, a generic Control. A part of this, the fact that the DevicePhoto have to stop after take a photo and found a marker, I considered that as a asyncAction, infact if the marker is not found, the DevicePhoto continued after PT times to takes a photo.

10 Implementation

I thought that my two entities have to speack each other, so I need a third entity, a generic Control. A part of this, the fact that the DevicePhoto have to stop after take a photo and found a marker, I considered that as a asyncAction, infact if the marker is not found, the DevicePhoto continued after PT times to takes a photo.

11 Deployment

I thought that my two entities have to speack each other, so I need a third entity, a generic Control. A part of this, the fact that the DevicePhoto have to stop after take a photo and found a marker, I considered that as a asyncAction, infact if the marker is not found, the DevicePhoto continued after PT times to takes a photo.