25/06/2020

**PIAZZA\_SPRINT 1**

## Sprint Goal

Essere in grado di muovere il robot fisico all’interno della TeaRoom.

## Requisiti

-Vogliamo essere in grado di muovere il ddr virtuale del progetto [it.unibo.virtualRobot2020](../../it.unibo.virtualRobot2020/userDocs/LabVirtualRobot.html).

-Vogliamo che, impostando una posizione all’interno della tea room come goal, il robot sia in grado di raggiungerla in autonomia.

Assunzione 1: il robot conosce già la mappa della Tea Room.

Assunzione 2: non possono essere impostate come goal delle posizioni contenenti un ostacolo. Di conseguenza, non essendoci ostacoli mobili, il robot non colliderà mai.

## Analisi dei Requisiti

Per poter formalizzare i requisiti e poter impostare sin da ora dei Test Plan è necessario **formalizzare il concetto di mappa della TeaRoom.**

Da requisiti iniziali si tratta di una stanza rettangolare. Possiamo pensare di rappresentarla come una matrice di NxM celle quadrate, ciascuna di lato uguale al diametro della circonferenza circoscrivente il robot waiter.

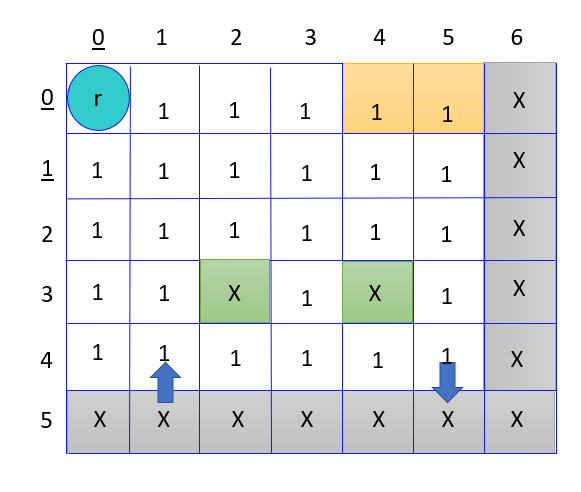
Possiamo anche pensare di dare un significato al contenuto delle celle. In particolare:

1: cella esplorata.

0: cella non esplorata.

X: cella sopra la quale si trova un ostacolo (un oggetto o una parete).

r: cella in cui si trova il robot.

La mappa della TeaRoom risulta essere una matrice 6x7 in cui la riga 5 e la colonna 6 non sono percorribili dal robot.

Individuando ciascuna cella con due coordinate (X,Y) le celle significative sono:

home in (0,0)

teatable1 in (3,2)

teatable2 in (3,4)

entrancedoor in (4,1)

exitdoor in (4,5)

servicedesk in (0,4) e (0,5)

### **Bozza di Test Plan**

1. Se (X,Y) = r e inviamo un comando (diversi da l e r) al robot dobbiamo ottenere (X,Y) = 1 poiché il robot si sarà spostato.
2. Se diamo come goal una cella (X,Y) = 1 ci aspettiamo che, al termine dello spostamento, valga (X,Y) = r.

## Analisi del Problema

Quali problematiche sorgono?

Controlla sempre l’overview iniziale perché alcune potrebbe essere necessario riprenderle taggandole.

### **Problematiche:**

Affinché il robot possa muoversi consapevolmente all’interno della Tea Room, cella dopo cella, dovrà essere in grado di muoversi facendo degli step. Ogni step verrà fatto secondo una direzione (down | left | right | up) e porterà il robot nella cella successiva lungo quella direzione (se è libera).

Il problema poi richiede che il robot sia in grado di pianificare un serie di azioni che gli permettano di raggiungere una posizione goal a partire dalla posizione in cui si trova.

Dal momento che pianificare una sequenza di azioni ed eseguirle sono due concetti logicamente distinti possiamo pensare di modellare il sistema in due entità distinte seguendo il paradigma mente-corpo.

Così facendo otterremo due componenti che, avendo una singola responsabilità, saranno dei ‘servizi riusabili’ anche per altri scopi.

### **Architettura Logica**

Si delinea quindi la seguente architettura logica del sistema:

### **Modello eseguibile del sistema**

Parlare del planner che usa AIMA 3.0. Cos’è e perché serve introdurlo.

Introdurre una base di conoscenza in Prolog.

COME MOTIVARE il DIVIDE ET IMPEREA da analisti:

As analysts we can motivate the introduction of the stepper actor as the opportunity to define a 'reusable service'.

-*waiter.qak*: è la mente. E’ un cameriere che conosce la mappa della stanza importandola da un file .bin, conosce le informazioni riguardo al suo dominio (cioè riguardo alla teaRoom e al suo stato) e ha la possibilità di richiedere ad un attore subordinato, l’attore waiterwalker.qak, di volersi spostare in una determinata cella.

-*waiterwalker.qak*: è il pianificatore. Riceve le richieste *movetoCell(X,Y),* pianifica una sequenza di mosse per raggiungere la posizione goal a partire dalla posizione corrente, e le comanda al *basicrobot* con delle richieste step by step. Il *basicrobot* si interfaccierà con la parte technology dependent (il robotSupport) in modo da comandare il movimento fisico del robot.

**Il waiter, o qualcuno per lui, deve conoscere la** **pianta della tea-room:**

Dai TestPlan elencati nell’overview iniziale è emersa la necessità di tenere traccia degli spostamenti del robot e della sua posizione attuale.

Quindi aggiungiamo 2 nuovi stati logici al waiter.

-movingTo(cell(X,Y) )

-at(cell(X,Y) )