

ANALISI DEL PROBLEMA

Osservazione dei cambiamenti del sistema

Da requisito è necessario costruire un'interfaccia web per l'utente Manager. Ciò rende obbligatorio condividere lo stato del sistema con l'interfaccia web.

Sono stati valutati due approcci.

Il primo, più semplice, prevede che la parte web lanci delle richieste verso il sistema ad intervalli regolari, per richiedere lo stato.

Questo approccio di polling è stato scartato perché poco efficiente in quanto effettua delle richieste anche quando non c'è un effettivo cambiamento di stato, e potrebbe portare in casi limite alla congestione del protocollo di rete.

Il secondo approccio è tramite interrupt.

In questo caso, la parte web osserva i cambiamenti delle componenti del sistema utilizzando il pattern observer.

In tal modo riceve delle notifiche soltanto quando c'è un cambiamento di stato.

È stato adottato quindi quest'ultimo utilizzando come supporto software il protocollo COAP che offre le API adatte al paradigma sopra citato.

Navigazione

Dall'analisi dei requisiti emerge un primo problema, ovvero la gestione della navigazione del waiter/robot.

Il primo punto da chiarire è quindi la localizzazione del waiter all'interno della tearoom.

Per questo motivo è necessario costruire una rappresentazione della tearoom come mappa, che contenga gli elementi chiave del sistema, cioè: tables, home, entrancedoor, exitdoor, barman ed il waiter stesso.

Nello specifico la mappa è costituita da una griglia dove ogni elemento sopra elencato occupa una cella. In questo modo, ognuno di essi può essere individuato con delle coordinate bidimensionali.

Per quanto riguarda il coverage, non è necessaria una scansione dell'ambiente a tempo di esecuzione in quanto la mappa e quindi la copertura dell'area viene fornita al waiter già pronta, in quanto è già nota la posizione di ciascun elemento.

Il secondo punto è quello della ricerca degli obiettivi da raggiungere ed il planning del percorso.

Per questo diventa indispensabile utilizzare un algoritmo capace di individuare tutti i passi per portare il waiter a destinazione.

L'algoritmo allo stato dell'arte in ambito path finding è l'A*, che permette di trovare la soluzione ottima al problema.

Per questo motivo sarà utilizzata l'implementazione del planner già presente *in house*.