

ColdStorageSprint0V1

Introduction

Requirements

I requisiti sono scritti dal committente TemaFinale23.

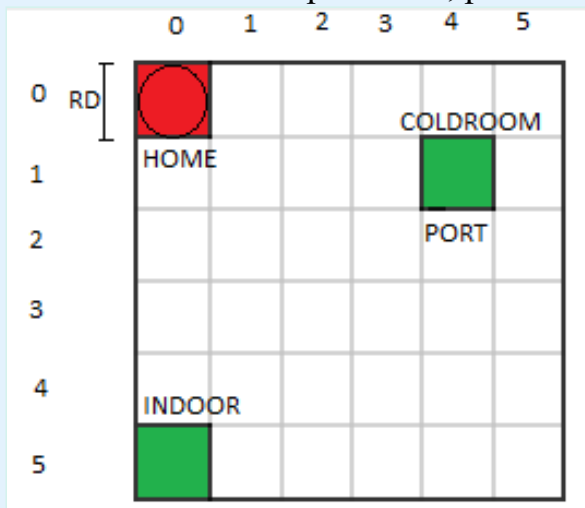
Goal di questo sprint:

- individuare un architettura logica iniziale che definisca le macro-entità del sistema e le loro interazioni
- definire un piano di lavoro iniziale

Requirement analysis

Dai requisiti possiamo evincere che:

- **ColdStorageService** è il sistema che si intende sviluppare composto dagli elementi sotto elencati.
- **Service area** è l'area di lavoro del sistema e si assume sia rettangolare e in piano. L'area viene modellata come suddivisa in una griglia di celle quadrate di dimensione costante pari a RD, pari alla dimensione del DDR robot.



È composta da:

- INDOOR: rappresenta il punto di arrivo dei **Fridge truck** per la consegna del carico di peso **FW**.
- ColdRoom: container dove viene depositato il carico dal **trasport trolley** ed ha una capacità massima pari a **MAXW**.
- Home: indica la posizione di partenza del **trasport trolley** e il punto di attesa

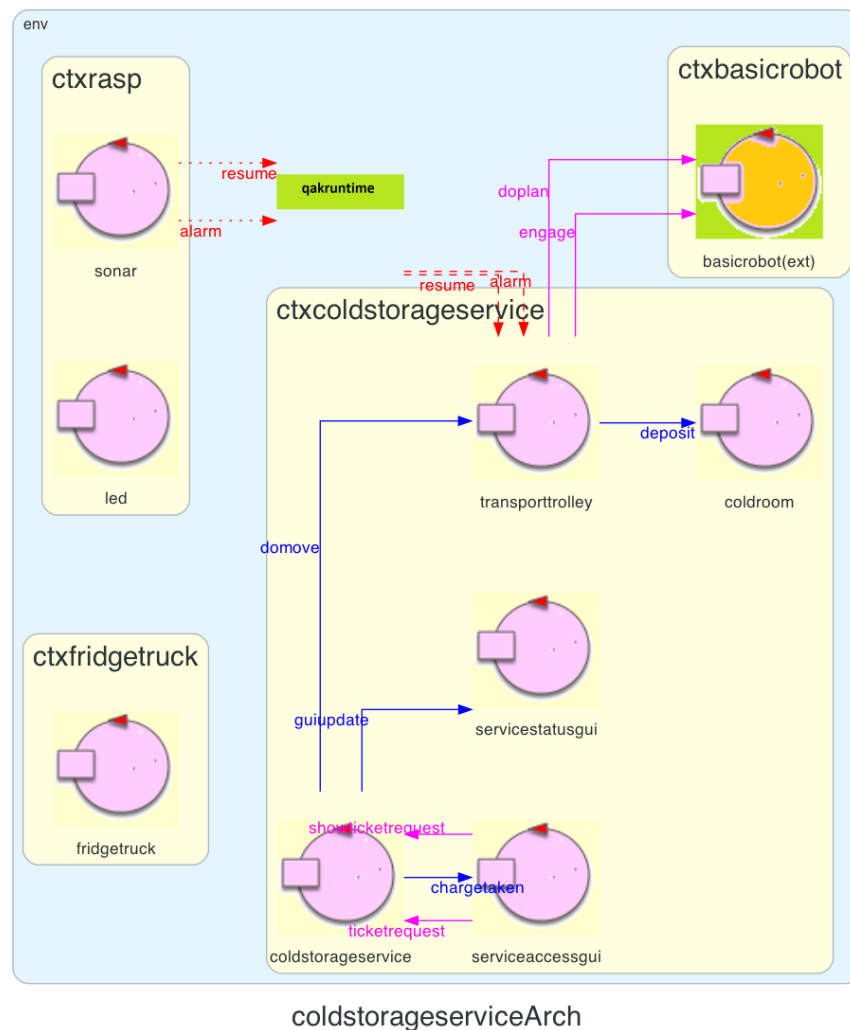
dello stesso in caso non ci siano richieste da elaborare.

- **Transport trolley** è l'attore che si occupa di impartire comandi al basicrobot. L'implementazione di quest'ultimo è fornita dal committente e modella il concetto di DDR robot. Le interazioni utili per la comunicazione tra **transport trolley** e basicrobot sono:
 - Request moverobot : moverobot(TARGETX, TARGETY)
 - Request doplan : doplan(PATH, OWNER, STEPTIME)
- **Fridge truck** sono soggetti esterni all'applicazione in grado di interagire con la **ServiceAccessGUI**.
- **ServiceAccessGUI** è rappresentata nell'architettura come un attore che si interpone tra l'utente (**fridge truck**) e il **ColdStorageService**.
- **ServiceStatusGUI** è anch'essa modellata come attore ed ha la responsabilità di mostrare un riepilogo dello stato del sistema al **Service-manager**.
- **Sonar** e **Led** sono componenti realizzati come attori appartenenti ad un diverso contesto (RaspberryPi).

Dialogo col committente

- Il processo di scarico della **ColdRoom** sarebbe da realizzare mediante un attore apposito. Tale realizzazione non è d'interesse in questo processo ma può essere implementato in uno sviluppo futuro.

Architettura logica



Dall'architettura logica possiamo evidenziare i seguenti punti d'interesse:

- Il sistema è distribuito su almeno 3 nodi computazionali diversi: BasicRobot, RaspberryPi e ColdStorageService.
- Saremo chiamati a sviluppare i macro-componenti `coldstorageService`, `coldroom`, `transporttrolley`, `sonar`, `led`, `serviceaccessgui`, `servicestatusgui`.
- Le interazioni fra `serviceaccessgui` e `coldstorageService` sono di tipo request-response ed sono implicate dal testo dei requisiti.
- L'interazione fra `sonar` e `transporttrolley` è modellata come evento.
- Si può notare che il `coldstorageService` non interagisce direttamente col `basicrobot` ma solamente mediante il `transporttrolley`.

- Il Fridge truck è un operatore umano che non fa parte del sistema. Fanno invece parte del sistema i messaggi che questo genera interagendo con la ServiceAccessGUI.

Piano di lavoro

Sprint	Goal	Tempo stimato	Note
Sprint 1	realizzazione dei prototipi dei componenti ColdRoom e ColdStorageService	3 giorni	
	testing	4 ore	il testing non prevederà l'interazione diretta con il basicrobot in quanto il TransportTrolley non sarà ancora interamente realizzato
Sprint 2	realizzazione del TrasportTrolley	2 giorni	
	testing	3 ore	il testing verrà effettuato utilizzando il simulatore fridge truck in quanto le GUI non saranno ancora sviluppate
Sprint 3	realizzazione delle GUI	1 giorno	lo sviluppo delle due GUI può essere eseguito in parallelo in quanto non sono interagenti tra loro
	testing	2 ore	
Sprint 4	realizzazione degli Alarm Requirements (componenti Sonar e Led)	1 giorno	
	testing	1 ora	il testing verterà su tutti i componenti del sistema

Lo sprint 3 e lo sprint 4 possono essere eseguiti in parallelo.



By Dominici Luca email: luca.dominici3@studio.unibo.it,
GIT repo: GIT repo:
<https://github.com/lucaDomo/Iss23DominiciLuca>



By Zacchiroli Enrico email:
enrico.zacchiroli@studio.unibo.it, GIT repo: GIT repo:
<https://github.com/zack-99/IssLab2023EnricoZacchiroli>



Tema finale GIT repo: <https://github.com/role-nzo/ISSTemaFinale/>