LABORATORIO DI INGEGNERIA DEI SISTEMI SOFTWARE

Introduction

Requirements

Descrizione requirements a questa pagina

Goals dello sprint 0:

1. Identificare i componenti principali del servizio **ColdStorageService**
2. Formulare un modello di interazione dei componenti individuati
3. Modellare una architettura logica del sistema che faccia da punto di riferimento per gli sviluppi successivi

Requirement analysis

Domain analysis

Termine	Descrizione
ColdStorageService	Servizio composto da un insieme di elementi, quali: • Service area; • DDR robot; • ServiceAccessGUI; • ServiceStatusGUI; • Alarm system.
Service area	Stanza piatta a pianta rettangolare nella quale il transport trolley può navigare, contiene diversi punti di interesse: • INDOOR; • ColdRoom; • HOME; e un ostacolo di posizione e dimensione predefinita. Viene modellata come un rettangolo.
INDOOR	Porta di accesso adibita al trasferimento di un carico dal camion refrigerato al transport trollery, è identificata dalle coordinate in cui si deve posizionare il transport trolley per accedervi nel modello della service area
ColdRoom	Contenitore refrigerato adibito al deposito di cibo ritirato dal camion refrigerato da parte del transport trolley, identificata dalle coordinate della porta di accesso PORT nel modello della service area
MAXW	Limite massimo di chilogrammi che la ColdRoom può contenere in un dato momento.
Transport trolley	Rappresenta un robot virtuale di forma quadrata e lato RD, è inizialmente posizionato in HOME e può muoversi avanti e indietro, ruotare di 90° e fermarsi; sfrutta il DDR robot per muoversi nella Service area.
HOME	Posizione di partenza del transport trolley, identificata dalle coordinate nel modello della service area.
RD	Lunghezza del lato del transport trolley.

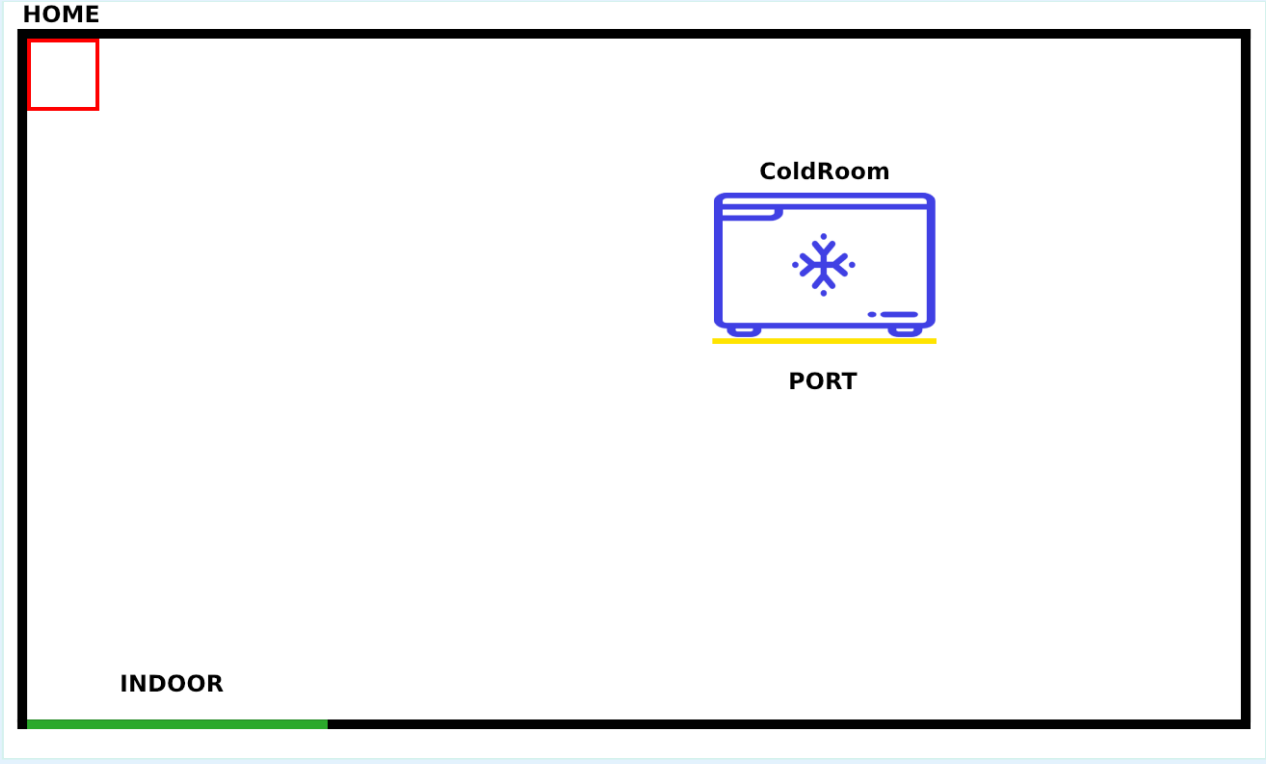
Azione di deposito	Operazione di deposito che coinvolge il camion refrigerato e il transport trolley: 1. il transport trolley preleva un FoodLoad dal camion refrigerato posizionandosi alla porta INDOOR; 2. il transport trolley si sposta da INDOOR a PORT della ColdRoom; 3. il carico viene depositato nella ColdRoom. Quando l'azione termina, il transport trolley prende in carico una nuova azione di deposito (se è disponibile un FoodLoad) o torna in HOME.
FoodLoad	Carico di cibo (frutta, vegetali...) che necessita di essere refrigerato, sia durante il trasporto che nella fase di accumulazione.
Camion refrigerato	Entità esterna al sistema che deposita FW kg di prodotti alimentari alla porta INDOOR.
ServiceAccessGUI	Interfaccia grafica che permette a un utente di: • visualizzare il peso corrente del materiale immagazzinato nella Cold Room; • inviare la richiesta di deposito di una **FoodLoad** FW kg di prodotti alimentari, comunicando un ticket in caso di richiesta accettata; • immettere il codice del ticket quando il camion refrigerato si trova alla porta INDOOR, all'esterno della service area; • visualizzare il messaggio di charge taken.
ServiceStatusGUI	Interfaccia grafica che permette a un **service manager** di visualizzare: • lo stato del transport trolley; • il **peso corrente** del cibo nella ColdRoom; • il **numero di richieste rifiutate** dall'inizio del servizio, considerando solo quelle rifiutate per mancanza di spazio nella ColdRoom.
DDR robot	Differential Drive Robot fornito dal committente insieme al software BasicRobot per controllarlo. Mette in atto i comandi inviati dal transport trolley.
Service manager	Utente umano che può interagire con la ServiceStatusGUI per monitorare lo stato della service area
Sonar	Dispositivo collegato a un RaspberryPi in grado di misurare la distanza di oggetti in fronte a esso.
Alarm device	Componente astratto che, sfruttando un sonar, comunica al transport trolley di fermarsi quando quando la distanza rilevata è inferiore a DLIMT.
Led	Dispositivo collegato a un RaspberryPi usato come dispositivo di avviso: • è **spento** se il transport trolley è situato in HOME; • **lampeggia** se il transport trolley è in movimento; • è **acceso** se il transport trolley non è in movimento.
Warning device	Componente astratto in grado di mostrare tre stati distinti: • transport trolley situato in HOME • transport trolley in movimento • transport trolley non in movimento (e non in HOME) Comanda al led di assumere un certo stato a seconda dello stato del transport trolley.
DLIMT	Distanza massima dal sonar per la quale si attivano i relativi alarm requirements: se viene rilevata una distanza minore l'alarm system comunica al transport trolley di fermarsi.
TICKETTIME	Quantità di secondi che esprime la durata della validità di un ticket.
FW	Quantità di carico che un camion refrigerato deve richiede di depositare espressa in chilogrammi.
MINT	Tempo minimo che deve trascorrere prima che il transport trolley possa nuovamente gestire un messaggio di stop inviato dall'alarm system,

	espresso in millisecondi.
Ticket	Rappresenta una prenotazione da parte dell'operatore di un camion refrigerato che identifica univocamente una azione di deposito in attesa di charge taken, è identificato da un **numero univoco**.
Stato service area	Informazioni a riguardo alla service area in un dato momento, comprende: • stato del transport trolley • peso di cibo nella ColdRoom • numero di richieste rifiutate
Charge taken	Messaggio che rappresenta la presa in carico di un FoodLoad da parte del transport trolley. Questo messaggio è relativo ad una singola richiesta di deposito da parte di un camion refrigerato

Service area

In base alla descrizione fornita dal cliente, abbiamo modellato la service area come un rettangolo privo di ostacoli e circondato da 4 pareti, all'interno del quale sono presenti le 4 aree rilevanti descritte in precedenza:

- **HOME**
- **ColdRoom**
- **INDOOR**
- **PORT**



In un colloquio con il committente è stata introdotta anche la presenza di **ostacoli** all'interno della service area, non meglio definiti.
Si rimanda ad un successivo incontro con il committente la conferma della dimensione della service area e la revisione dello specifico posizionamento di ogni punto di interesse, compreso degli eventuali ostacoli.

Transport trolley

In questa analisi abbiamo identificato lo stato corrente del transport trolley come le seguenti informazioni riguardo al transport trolley in un dato momento:

- posizione nella service area;
- stato di movimento (**stopped**,**moving**).

Ticket

Essendo che vi è presente il TICKETTIME, associamo un **timestamp di emissione** ad ogni ticket generato, allo scopo di poter calcolare quanto tempo è passato dall'emissione al momento dell'inserimento del ticket stesso.

Sonar e led

Per l'utilizzo dei dati del sonar e il controllo del led verrà usato software fornito dal committente unito a software già disponibile alla nostra software house per abilitare la comunicazione con il transport trolley.

Verbs analysis

La seguente tabella mette in luce le **azioni** che vengono compiute all'interno della logica applicativa, evidenziando qual è l'entità che le compie e l'oggetto che fa parte dell'azione stessa.

Soggetto	Termine	Oggetto	Descrizione
Camion refrigerato	**Sends a request**	Peso FW	Il (guidatore del) camion invia una richiesta di deposito di **FW** kg di cibo tramite la ServiceAccessGUI.
	Drives to the INDOOR	-	Il (guidatore del) camion, in caso di richiesta accettata, va fino alla porta di INDOOR, prima della scadenza del ticket (TICKETTIME).
	Enters the ticket number	Ticket number	Il (guidatore del) camion, inserisce il numero del ticket tramite la ServiceAccessGUI.
	Waits	Messaggio charge taken	Il (guidatore del) camion attende fino a che il messaggio Charge taken non appare sulla ServiceAccessGUI.
	Leaves the INDOOR	-	Il (guidatore del) camion, in caso di richiesta di deposito rifiutata, o dopo aver atteso il messaggio, lascia la INDOOR.
ColdStorageService	**Accepts a ticket**	Ticket	Il servizio accetta il ticket che il (guidatore del) camion ha inserito.
	Avvisa il transport trolley	Peso FW	Il servizio comunica al transport trolley che c'è del cibo da depositare.
	Sends message *Charge taken*	Charge taken	Il ColdStorageService comunica alla ServiceAccessGUI che il transport trolley ha raccolto il carico.
Transport trolley	**Accepts a ticket**	Ticket	Il transport trolley accetta un nuovo ticket.
	Reaches the INDOOR	-	Il transport trolley raggiunge la INDOOR.
	Picks up the food	Peso FW	Il transport trolley prende il cibo da depositare dalla INDOOR.
	Goes to the ColdRoom	-	Il transport trolley si muove fino alla ColdRoom.
	Stores the food	Peso FW	Il transport trolley deposita il cibo nella ColdRoom, il completamento di questa

			azione determina la **fine dell'azione di deposito**
	Return to HOME	-	Il transport trolley ritorna alla HOME dopo aver depositato il cibo.
	Sends message *Charge taken*	Charge taken	Il transport trolley comunica al ColdStorageService di aver raccolto il carico.

Software fornito dal cliente

Il cliente fornisce il software `basicrobot` utile a controllare il DDR robot, oltre a software per gestire l'accensione e spegnimento di un led e la raccolta dei dati da un sonar.

Modello logico

In questa fase verranno presi in analisi i macro-componenti del sistema e le interazioni che avvengono fra essi, con il fine di delineare l'architettura logica del sistema.

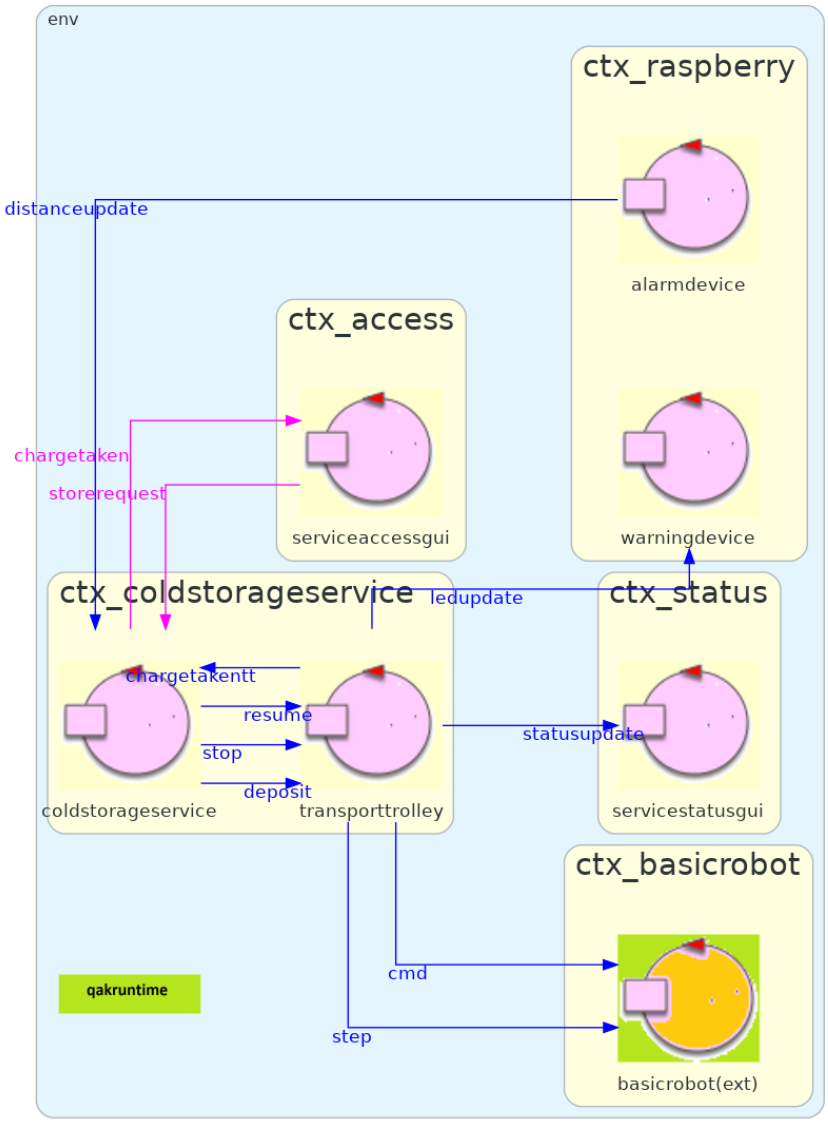
Abbiamo deciso di utilizzare il modello ad attori per questo progetto in quanto rispecchia i requisiti analizzati dal documento fornito dal cliente: il sistema è costituito da componenti **eterogenei** che interagiscono tra di loro su **base distribuita**; inoltre abbiamo a disposizione librerie e un linguaggio di metamodellazione che supporteranno lo sviluppo orientato agli attori.

Il linguaggio in questione è QAK, un linguaggio di modellazione eseguibile che permette di creare prototipi e sistemi funzionanti in breve tempo (rispetto a linguaggi tradizionali) semplificando gli aspetti più complessi della gestione delle interazioni tramite una struttura basata su attori che operano in contesti.
Oltre alla velocità di sviluppo possibile con QAK, la scelta deriva dal fatto che il cliente ha fornito software per pilotare un DDR robot (o robot virtuali) compatibile con QAK.

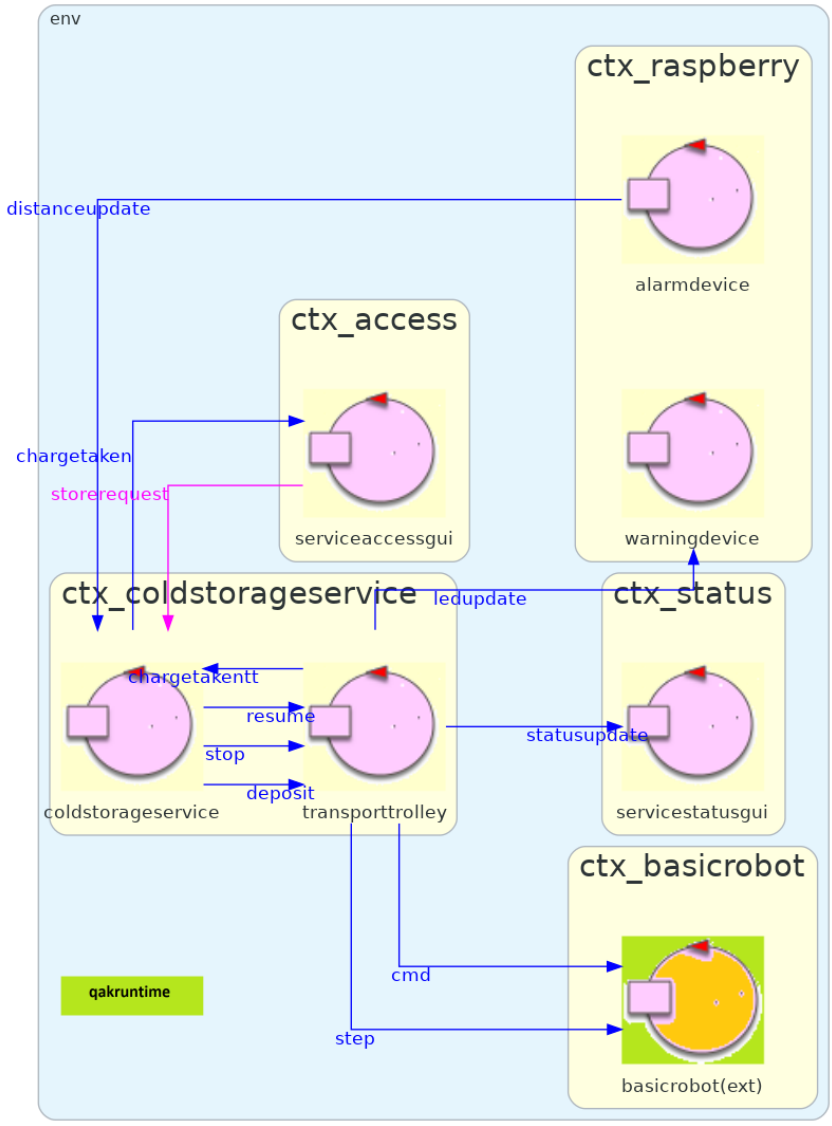
Charge taken

Nell'analisi dei requisiti abbiamo identificato una **discrepanza** riguardo al messaggio **charge taken**: non è chiaro se il messaggio charge taken debba avere semantica richiesta-risposta o semantica dispatch (non attende risposta).
Il committente, in un secondo colloquio, non ha dato nessuna preferenza riguardo alla tipologia di messaggio.

Abbiamo quindi deciso di realizzare due versioni del **modello dei requisiti**, rimandando ad una futura discussione con il committente (mostrandogli i due schemi possibili) la scelta di una determinata semantica per i messaggi **charge taken**.



chargetaken - Request-Reply



chargetaken - Dispatch

Problem analysis

Dalla fase di analisi dei requisiti sono emerse diverse problematiche da affrontare, ma sono stati anche individuati con chiarezza dei componenti che andranno realizzati.
Di seguito, è riportata la lista dei macro-componenti individuati, le cui descrizioni sono già state definite in fase di analisi dei requisiti:

- ColdStorageService
- TransportTrolley
- ServiceAccessGUI
- ServiceStatusGUI
- RaspberryPi
  - Warning device (led)
  - Alarm device (sonar)

Suddivisione in Sprint

Abbiamo pensato di utilizzare la tecnica **divide et impera**, separando quindi l'analisi dell'applicazione in parti più piccole ed analizzandole una ad una.
Per fare ciò si è dovuto prima identificare quale fosse il **core business** dell'applicazione, basandosi sull'analisi dei requisiti appena redatta. Abbiamo scelto come tale l'interazione tra *ColdStorageService* e *TransportTrolley*, in quanto da essa si realizza il cuore della logica applicativa richiesta dal committente. Il focus sul *TransportTrolley* implica anche la presa in considerazione del concetto di *basicrobot*, già definito in analisi dei requisiti e fornito dal committente.

La tecnica appena descritta ci ha permesso di dividere i requisiti da analizzare secondo un ordine di priorità, concretizzato nella definizione di quattro *Sprint*, ognuno volto all'analisi di una diversa problematica:

1. **Sprint1**: ColdStorageService e TransportTrolley (analisi del core business)
2. **Sprint2**: ServiceAccessGUI
3. **Sprint3**: requisiti di allarme
4. **Sprint4**: ServiceStatusGUI

Questa suddivisione preliminare verrà confermata di volta in volta nei successivi Sprint, sulla base delle nuove problematiche che sorgeranno e dei relativi modi per affrontarle.

Valutazioni in termini di tempo

Il fatto di aver acquisito un'ampia visione di contesto sull'applicazione (derivante dall'analisi dei requisiti) e di aver suddiviso il lavoro in macro-parti (corrispondenti agli Sprint), ci ha permesso di definire con un buon margine di errore il costo in termini di tempo per la realizzazione dell'applicazione.
Il tempo che verrà impegato è poi strettamente dipendente da come verrà suddiviso il lavoro tra i membri del team di sviluppo (2 in questo caso), con possibilità di parallelismo nell'analisi di problematiche diverse.

Abbiamo scelto di affrontare lo *Sprint1* assieme, in modo tale da essere certi di analizzare quello che è il core business dell'applicazione nella maniera più precisa possibile (pagando però in termini di tempo).
Per i successivi Sprint, ovviamente al netto di eventuali nuove problematiche che sorgeranno, abbiamo invece deciso di dividerci l'analisi di *Sprint2* e *Sprint3*, che affrontano problematiche differenti. Per quanto riguarda lo *Sprint4*, pensiamo possa essere svolto il parallelo allo *Sprint2*, in quanto trattano problemi molto simili. Tuttavia, decisioni defintive verranno prese e/o confermate nei successivi *Sprint*.

In sintesi, per come abbiamo pensato la suddivisione del lavoro, questi sono i costi valutati in termini di tempo:

1. **Sprint1**: 2 persone - 20 ore
2. **Sprint2**: 1 persona - 11 ore
3. **Sprint3**: 1 persona - 16 ore
4. **Sprint4**: 1 persona - 5 ore

Impiego totale in termini di tempo: 42 ore in 2 persone.

Architettura logica

Sulla base dell'analisi dei requisiti non siamo ancora in grado di definire un'architettura logica completa del sistema, che verrà rimandata ai successi Sprint, a valle di un'analisi più approfondita dei componenti e delle interazione presenti fra essi.
Per ora, si utilizza come riferimento il **Modello dei requisiti** definito in fase di Analisi dei requisiti.

Test plans

Per quanto riguarda i piani di test per l'applicazione, verranno rimandati agli *Sprint* successivi, relativamente ai particolare problemi che verranno affrontati.

Introduciamo un discorso sul *Camion refrigerato* e sul suo ruolo all'interno dell'applicazione (particolarmente in fase di test).
Come già detto in fase di analisi dei requisiti, quest'entità rappresenta l'utente umano che interagisce con il sistema (in particolar modo con la *ServiceAccessGUI*). Nonostante anche questo aspetto verrà trattato negli *Sprint* successivi, consideriamo utile, ai fini di test, modellare il *Camion* come **entità mock** che simula l'interazione con il sistema. Successivamente, questa *mock* verrà scartata e sostituita da unità di test (ad esempio *JUnit*, *NUnit*,...) per il testing automatizzato delle varie funzioni della nostra applicazione.

GIT repo:
https://github.com/RiccardBarbieri/ColdStorageService