chronicles_sprint0_v3

IL TEAM:

Diego Bruno	Marco Crisafulli	Sebastiano Giannitti

GITHUB DEL TEAM

Introduzione allo Sprint0 e obiettivi attesi

Lo scopo di questo sprint, e di questo documento, è formalizzare i singoli termini del testo fornito da company e anche quello di fornire una prima visione di insieme del sistema da realizzare.
Come goal di questo sprint ci poniamo:

- Fornire una panoramica dei componenti già forniti dal committente e dei componenti da sviluppare.
- Produrre un piano di test preliminare da proporre e validare insieme al committente
- Chiarire eventuali dubbi relativi ai requisiti

Requisiti:

I requisiti del committente, che da questo momento per semplicità chiameremo “company”, sono disponibili al LINK.

Analisi dei requisiti

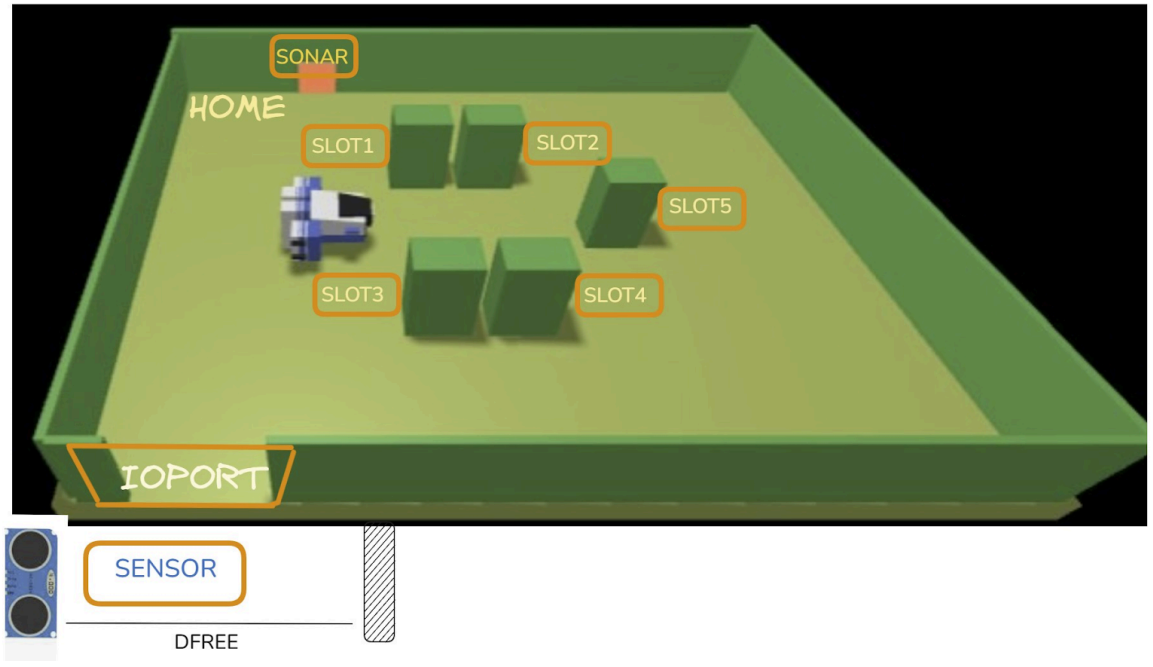
Con la finalità di analizzare i requisiti, partiamo dalla specifica fornita da company ed evidenziamo i componenti di interesse.
Specifica: *“A Maritime Cargo shipping company (from now on, simply company) intends to automate the operations of load of freight in the ship’s cargo hold (or simply hold). To this end, the company plans to employ a Differential Drive Robot (from now, called cargorobot) for the loading of goods (named products) in the ship’s hold.*

The products to be loaded must be placed in a container of predefined dimensions and registered, by specifying its weight, within a database, by using a proper service (productservice). After the registration, the productservice returns a unique product identifier as a natural number PID, PID>0.

The hold is a rectangular, flat area with an Input/Output port (IOPort). The area provides 4 slots for the product containers.”

Componenti di interesse

productservice	Componente per la registrazione dei prodotti. Non può essere rappresentato da classi o oggetti. Abstraction gap da colmare.
cargorobot	Entità dell'applicazione che si avvale del software basicrobot fornito dal committente.
products	Prodotti da caricare nella stiva, ogni prodotto ha un peso. Entità, oggetto.
hold	Area rettangolare, con una IOPort per il prelievo del prodotto e 4 slot di carico per ospitare i prodotti. Rappresentabile come una classe con le sue proprietà.



"In the picture above:

- *Gli **slots** depict the hold storage areas, when they are occupied by product containers*
- *The **slots5** area is permanently occupied, while the other slots are initially empty*
- *The **sensor** put in front of the IOPort is a sonar used to detect the presence of a product container, when it measures a distance **D**, such that **D < DFREE/2**, during a reasonable time (e.g. **3** secs)."*

slots	slot per il carico merci (1-5) identificati da un sistema di coordinate. Classe con proprietà di posizione e stato binario.
sensor	sensore che rileva la presenza di prodotti quando la distanza D del prodotto dal sensore è < DFREE/2 con DFREE valore numerico. Abstraction gap da colmare.

"The company asks us to build a software systems (named cargoservice) that:

1. *is able to receive the **request to load** on the cargo a product container already registered in the productservice.*
   The request is rejected when:
   - *the product-weight is evaluated too high, since the ship can carry a maximum load of **MaxLoad>0 kg**.*
   - *the hold is already full, i.e. the **4 slots** are already occupied.*

cargoservice	Riceve la request to load di un prodotto contenuto nel productservice e tale richiesta viene rifiutata o se il prodotto ha un peso troppo alto o se la stiva è già piena. Abstraction gap.

2. *If the request is accepted, the cargoservice associates a slot to the product **PID** and returns the name of the reserved slot. Afterwards, it waits that the product container is delivered to the ioport. In the meantime, other requests are not elaborated.*

cargoservice	Associa uno slot al PID del prodotto e fornisce il nome dello slot riservato

3. *is able to detect (by means of the sonar **sensor**) the presence of the product container at the ioport*

Sonar sensor	Sensore Sonar reale che rileva la presenza di prodotti (products) presso la ioport. Locato su un nodo esterno in grado di montare un sensore.
ioport	I prodotti (products) saranno presenti presso la ioport. Classe con proprietà di posizione e stato.

4. *is able to ensure that the product container is placed by the cargorobot within its reserved slot. At the end of the work:*
    - *the cargorobot should returns to its HOME location.*
    - *the cargoservice can process another load-request*

cargoservice	Si assicura che il prodotto sia posizionato dal cargorobot nello slot associato al PID rispettivo. Successivamente può processare altre richieste. Il cargorobot è autonomo? Conosce la posizione degli slot? Requisito non esaustivo.
cargorobot	Dopo aver posizionato il prodotto torna alla posizione HOME. Come torna in HOME? Necessaria analisi del problema.
HOME	Posizione base del cargorobot, identificata dal sistema di coordinate

5. *is able to show the current state of the hold, by means of a dynamically updated **web-gui**.*

web-gui	*pagina web che mostra lo stato corrente della hold, aggiornata dinamicamente. Come recupera le informazioni sullo stato della hold? Come vengono mostrate? Analisi del problema.*
hold	*la stiva (hold) avrà uno stato degli slot, modellabile come classe.*

6. *interrupts any activity and turns on a led if the sonar sensor measures a distance **D > DFREE** for at least **3** secs (perhaps a sonar failure). The service continues its activities as soon as the sonar measures a distance **D <= DFREE**."*

sonar sensor	Interrompi ogni attività del cargoservice e accendi un led se D > DFREE per almeno 3 secondi, eventuale sonar failure. Abstraction gap
cargoservice	il servizio riprende quando il sonar D <= DFREE. Abstraction gap

Abstraction Gap

La nostra software house dispone di un linguaggio di modellazione chiamato QAK. Il qak ci fornisce gli strumenti necessari a colmare l'abstraction gap.
Introduciamo l'attore qak:
Un attore qak è un componente attivo che:

- nasce, vive e muore in un *contesto* che può essere comune a (molti) altri attori;
- ha un **nome univoco** nell'ambito di tutto il sistema;
- è logicamente attivo, cioè dotato di flusso di controllo autonomo;
- è capace di inviare messaggi ad un altro attore, di cui conosce il **nome**, incluso sè stesso;
- è capace di eseguire elaborazioni autonome e/o elaborazioni di messaggi;
- è dotato di una sua coda locale (**msgQueue**) in cui sono depositati i messaggi a lui inviati

Procediamo a riformulare l'analisi dei requisiti.

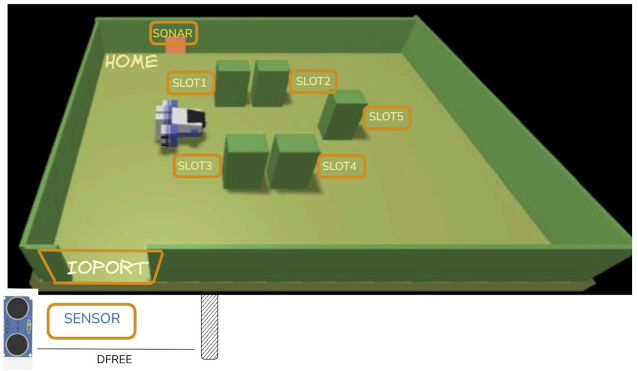
Tabella riassuntiva

Componente	ID requisito	Requisito (sintesi)	Dettagli operativi
productservice	PRD-1	Registra prodotti e restituisce **PID > 0**	Attore
products	P-1	Ogni prodotto ha un **peso**	Classe (PID, nome, peso)
hold	H-1	Area rettangolare con **IOPort** e **4 slot**	Classe o attore. Necessaria analisi del problema.
hold / slots	H-2	Uno **slot è permanentemente occupato**, gli altri inizialmente vuoti	Classe o attore.

hold / slots	H-3	Slot identificati da **coordinate** / nome univoco	*Guardare sezione HOLD*
hold / IOPort	H-4	Punto di arrivo del container prima del posizionamento	*Guardare sezione HOLD*
sensor (sonar)	SNS-1	Rileva presenza container: **D < DFREE/2 per ~3s ⇒ presente**	Attore.
sensor (sonar)	SNS-2	**Guasto**: **D > DFREE per ≥3s ⇒ failure**	
LED	LED-1	Indicatore di failure sonar	**ON** se SNS-2; **OFF** al ripristino (D ≤ DFREE). Azionato da `cargoservice`.
cargorobot	ROB-1	Posiziona il container nello **slot assegnato**	Attore.
cargorobot	ROB-2	Ritorna alla posizione **HOME** a fine lavoro	
cargoservice	CRS-1	Riceve richiesta di carico per **PID** registrato	Attore
cargoservice	CRS-2	**Rifiuta** la richiesta se **peso > MaxLoad**	
cargoservice	CRS-3	**Rifiuta** la richiesta se **stiva piena** (4 slot occupati)	
cargoservice	CRS-4	Se accettata la richiesta, **associa** uno **slot** al PID e **ritorna il numero dello slot**	
cargoservice	CRS-5	**Attende** il container all'IOPort; **nel frattempo altre richieste non sono elaborate**	
cargoservice	CRS-6	**Assicura** il posizionamento del container nello **slot**	

		riservato tramite `cargorobot`	
cargoservice	CRS-7	**Interrompe** attività e accende **LED** se SNS-2	
web-GUI	GUI-1	Mostra **dinamicamente** lo stato corrente della **hold**	Pagina web. Necessaria analisi del problema.

Analisi della hold



r	1	1	1	1	1	1
1	1	X	X	1	1	1
1	1	1	1	X	1	1
1	1	X	X	1	1	1
1	1	1	1	1	1	1
X	X	X	X	X	X	X

La hold è modellata dal basicrobot attraverso una matrice 7x6.
La dimensione di ogni cella corrisponde alle dimensioni del robot, quindi il robot occuperà in ogni momento un posto sulla mappa.
Gli '1' corrispondono a superficie calpestabile dal robot, le X corrispondono ad ostacoli o muri.

HOME	1	1	1	1	1	1
1	SLOT1	X	X	SLOT2	1	1
1	1	1	1	X	SLOT5	1
1	SLOT3	X	X	SLOT4	1	1
IOPORT	1	1	1	1	1	1
X	X	X	X	X	X	X

Ai fini dei requisiti abbiamo ritenuto opportuno indicare sulla mappa delle celle in particolare:

- HOME (0, 0)
- SLOT1 (1, 1)
- SLOT2 (1, 4)
- SLOT3 (3, 1)
- SLOT4 (3, 4)
- SLOT5 (2, 5)
- IOPORT (4, 0)

Ognuna di queste celle è identificata tramite coordinate (riga, colonna).

Questo modello di HOLD sfruttato dal robot rappresenta una mappa dello spazio virtuale, ma gli slot e la IOPORT non sono note al robot se non come delle semplici coordinate che noi forniremo.
Nel modello proposto piu' avanti sfrutteremo un componente per tenere "traccia" dello stato degli slot.

CargoRobot

CargoRobot è l'attore che utilizza il software basicrobot24, fornito dal committente.
Questo componente è sviluppato in linguaggio QAK.

Il basicrobot è in grado di riconoscere dei messaggi ben formattati che esprimono una direzione e un numero di unità. Il robot si muoverà nella direzione specificata per il numero di unità indicato.

Queste mosse sono supportate anche da un algoritmo di pathfinding a* che permette di trovare la strada verso delle coordinate (x,y) come specificate nella sezione HOLD e muovervi il robot.
Il robot rimane allineato con la griglia della HOLD perchè al termine di ogni percorso torna in HOME e fa delle manovre per allinearsi nuovamente alla griglia.

Ci preoccuperemo perciò di sfruttare il modello di comunicazione request/reply incluso nel linguaggio qak e compatibile con il robot:

```
Request moverobot    :  moverobot(TARGETX, TARGETY)
Reply moverobotdone  :  moverobotok(ARG)                    for moverobot
Reply moverobotfailed:  moverobotfailed(PLANDONE, PLANTODO) for moverobot
```

ProductService

L'attore productService è fornito dal committente, sviluppato in linguaggio qak e si occupa di fornire il PID e peso dei prodotti registrati in database (anch'esso fornito dal committente).

```
Request createProduct : product(String)                    //String J
Reply   createdProduct: productid(ID) for createProduct   //String JSO

Request deleteProduct  : product( ID )
Reply   deletedProduct : product(String) for deleteProduct

Request getProduct : product( ID )
Reply   getProductAnswer: product( JSonString ) for getProduct

Request getAllProducts : dummy( ID )
Reply   getAllProductsAnswer: products(  String ) for getAllProducts
```

Apprendiamo inoltre il formato dei prodotti in database e come ci verranno restituiti:

```
//String JSON '{"productId":31,"name":"p31","weight":311}'
```

Sonar

Il sonar è un attore qak fornito dal committente e completo di alcune delle funzionalità richieste, andiamo a vedere come emette i dati:

```
Event sonardata      : distance(D)   //emitted  by sonardevice
```

Questi dati verranno emessi come degli eventi qak su un contesto locale al sonar e notiamo che questo software può sfruttare un broker mqtt (molto utile ai nostri fini) a livello di sistema (da documentazione qak):

```
mqttBroker "192.168.1.214" : 1883 eventTopic "logkb"
```

Chiaramente gli ip saranno oggetto di modifica.

Testing

Dall'analisi dei requisiti emerge un quadro parziale del nostro componente che abbiamo individuato come attore. Possiamo stilare un primo piano di test sulla base delle informazioni trattate.

RequestToLoad

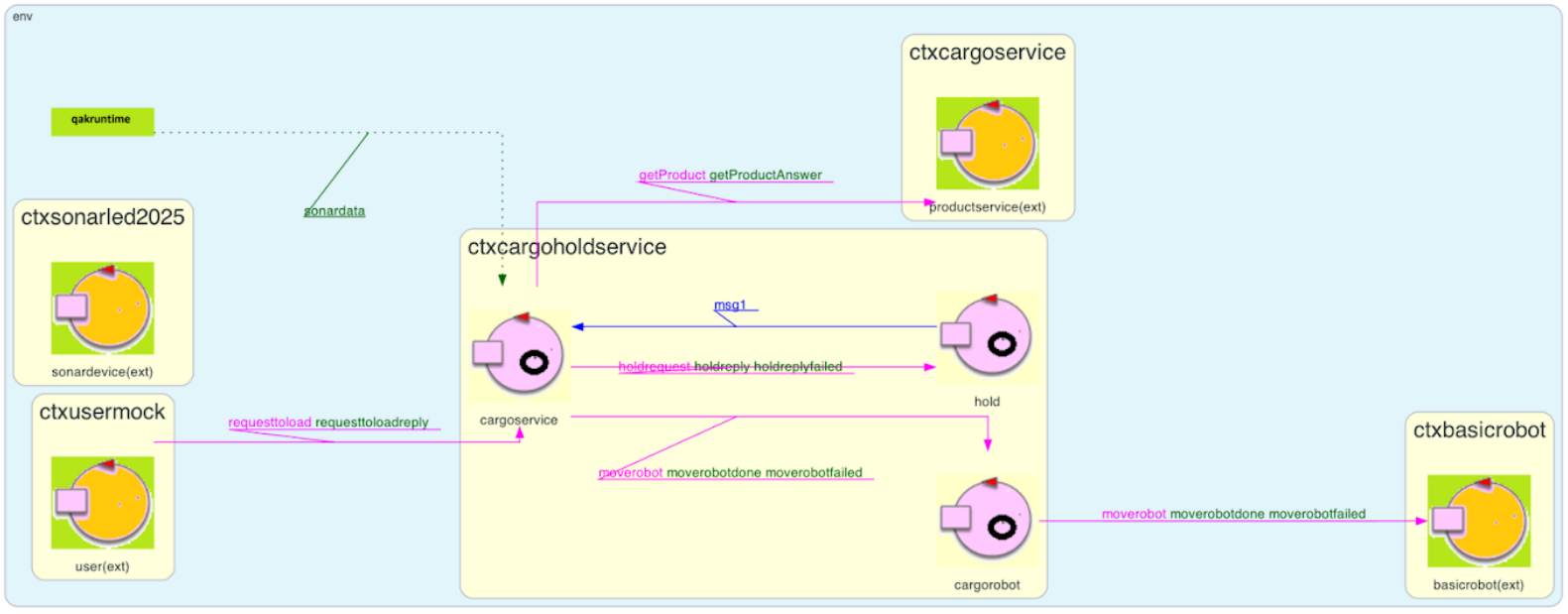
CRS-1 (PID registrato)	CRS-2 (Peso > MaxLoad)	CRS-3 (Slot occupati)	Esito
			Richiesta accettata
			Richiesta rifiutata
			Richiesta rifiutata
			Richiesta rifiutata

Proposta di modello

Date le osservazioni e le preliminari analisi svolte finora risulta possibile formulare un modello, o meglio, una prima iterazione di esso.
Notiamo a primo sguardo che l'attore cargoservice si assume molte responsabilità, come se fosse un macro attore e ciò non è una buona pratica.
Il linguaggio qak fornisce dei meccanismi di delega la cui utilità verrà valutata nei successivi sprint.
Per il resto osserviamo diverse richieste (compatibili con i componenti forniti dal committente) emesse verso contesti esterni e la presenza di un componente hold nello stesso contesto del cargoservice.
Contiamo perciò dei contesti esterni con dei componenti forniti dal committente:

- contesto ctxbasicrobot con cargorobot
- contesto ctxcargoservice con productservice
- contesto ctxsonarled2025 con sonardevice

Abbiamo poi inserito un contesto esterno ctxusermock per mimare un utente o un operatore che usa il sistema, non abbiamo inserito un contesto per la webgui in quanto potrebbe essere lo stesso contesto ctxusermock ad essere sfruttato come interfaccia utente / webgui compatibilmente con i requisiti.
Inoltre al fine di avere una webgui che rappresenti correttamente il sistema, il sistema deve funzionare correttamente, quindi ci occuperemo di sviluppare questo componente in seguito.
Il componente hold tiene traccia dello stato degli slot ed è pertinente nel contesto del nostro componente cargoservice che è oggetto di progetto.



cargoserviceArch

Piano di Lavoro

L’obiettivo è quello di dividere la progettazione in 3 sprint, con la finalità di ottenere 3 prodotti separati dove il successivo incapsula il precedente (immaginiamo il tutto come l’armatura di Ironman). la divisione sarà la seguente:

1. gestione della richiesta di carico, la registrazione dei prodotti e l'avviamento del robot (40h/uomo).
2. IOManager, in questa fase si svilupperà la parte di gestione del sonar, il rilevamento del product container nella porta di IO e l'interfacciamento hardware dei dispositivi fisici di sonar e led con il sistema.(30h/uomo)
3. Realizzazione della GUI (20h/uomo)