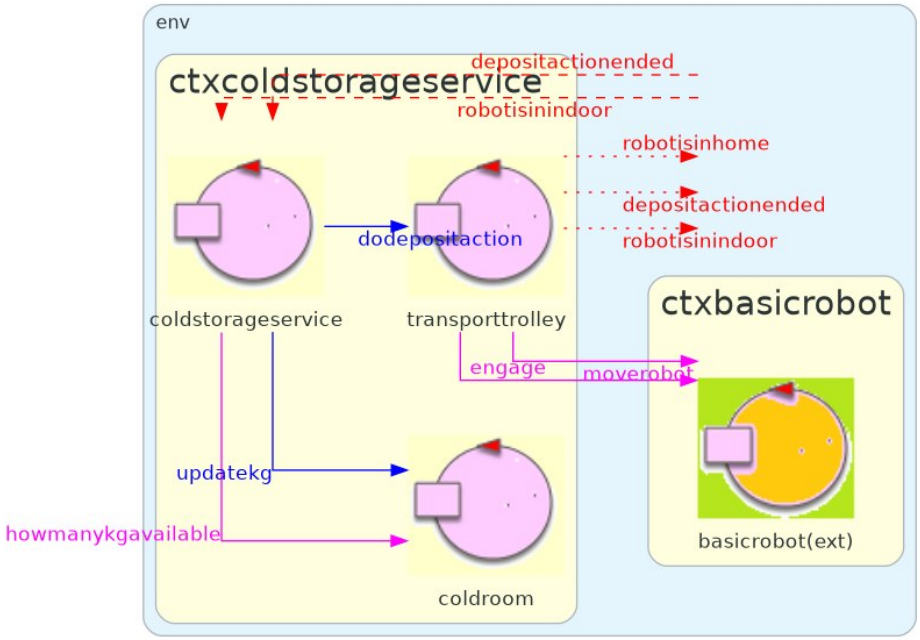
ColdStorageSprint3V0

Introduzione

Goal conseguiti nello sprint1

- prototipazione del core business ColdStorageService + TransportTrolley

architettura logica sprint1



coldstorageserviceArch

Goal dello sprint3

- introduzione del alarm requirement nel prototipo sviluppato allo sprint1

Requirements

The system includes a Sonar and a Led connected to a RaspberryPi.
The Sonar is used as an ‘alarm device’: when it measures a distance less than a prefixed value **DLIMT**, the transport trolley must be stopped; it will be resumed when Sonar detects again a distance higher than **DLIMT**.

The Led is used as a ***warning devices***, according to the following scheme:

> - the Led is **off** when the transport trolley is at HOME
> - the Led **blinks** while the transport trolley is moving
> - the Led is **on** when transport trolley is stopped.

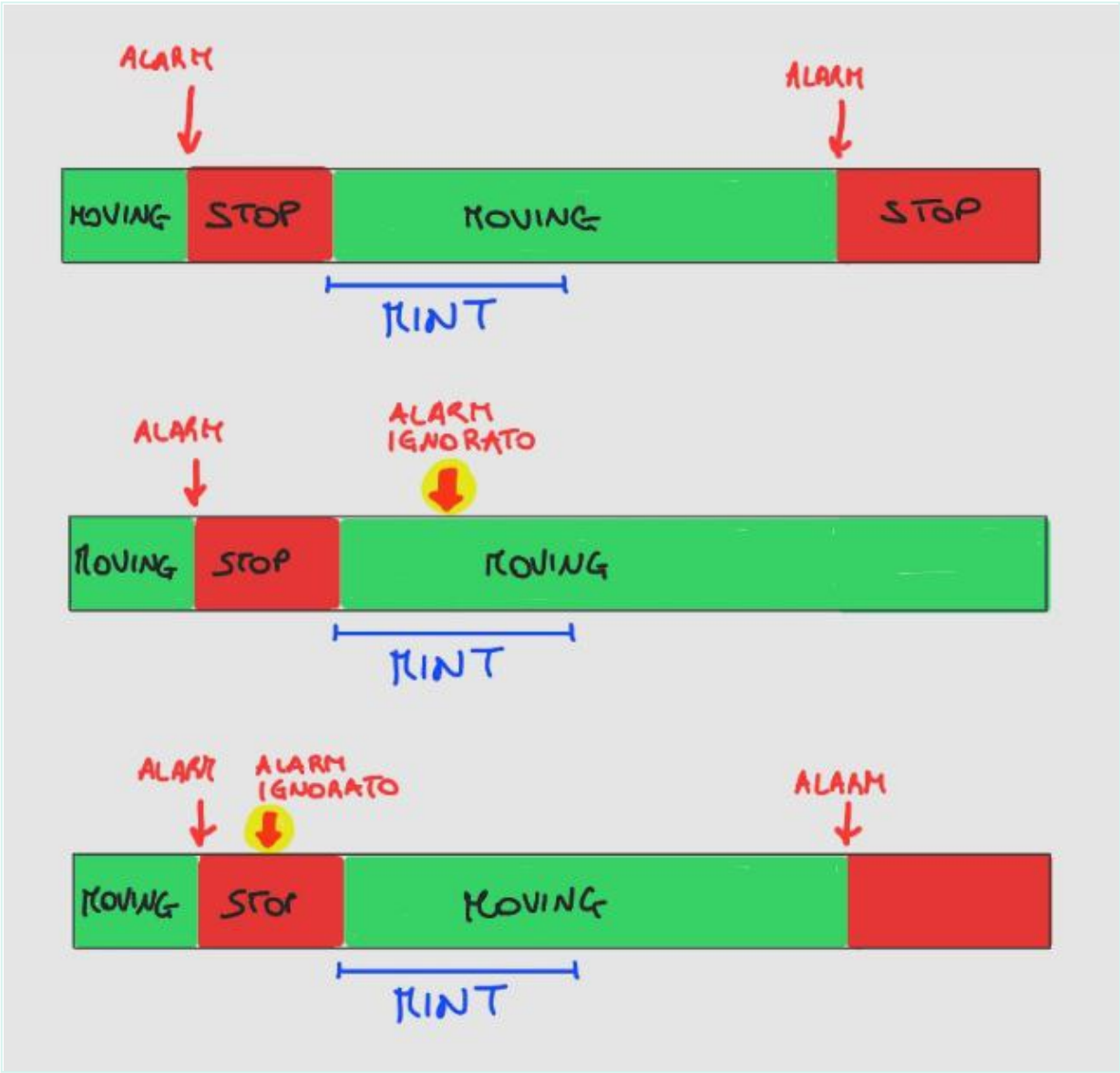
SERVICE USER STORY
5) While the transport trolley is moving, the Alarm requirements should be satisfied. However, the transport trolley should not be stopped if some prefixed amount of time (**MINT** msecs) is not passed from the previous stop.

Requirements analysis

- Per Sonar si intende il dispositivo a ultrasuoni HCSR04 e viene formalizzato a livello logico con l'entità Sonar.
- Per Led si intende un dispositivo che emette energia sotto forma di luce e viene formalizzato a livello logico dall'entità Led.

Dopo opportuni colloqui con il committente, possiamo affermare che :

- il Sonar e il Led sono su un raspberry che non dipende dalla service area.
- Il calcolo del delay di MINT parte da quando parte la endalarm dello stop precedente



Problem Analysis

Misurazione continua Sonar

Il dispositivo fisico Sonar HCSR04 potrebbe emettere dati spuri.

Alarm Basic Robot

Per fermare il robot si fa affidamento al comando "alarm", già formalizzato dai messaggi presenti in questa documentazione: BasicRobot23.html

L'evento endalarm

Quando il Transport Trolley riceve un evento di alarm, il Basic Robot poi gli invia **moverobotfailed** come risposta finale dell'ultimo comando. Dunque si deve tener traccia dell'ultima mossa finale del

Transport Trolley, dovendo così estendere il Transport Trolley sviluppato allo sprint1.

Architettura logica dopo analisi del problema

Progettazione

Misurazioni corrette Sonar

Si predispongono due CodedQActor usabili per costruire una pipe che ha sonar come sorgente-dati e che provvede a eliminare dati spuri (dataCleaner) e a generare (distancefilter) eventi significativi per il livello applicativo.

L'evento endalarm

Il Transport Trolley deve gestire i casi di fallimento dovuti alla risposta **moverobotfailed** data dal basic robot quando viene emesso un evento **alarm**.
Quando entra nello stato di robotfailed, bisogna aspettare

1. un comando/evento di fine dell'alarm (endalarm). In questo caso, sarà il Sonar a dover emettere l'evento endalarm. Una volta ricevuto l'endalarm, il Transport Trolley deve ripristinare lo stato dell'ultimo comando al basicrobot. Per fare ciò il Tranport Trolley deve tenere traccia dello stato dell'ultimo comando inviato al basic robot.
2. una risposta di fine della mossa precedente (moverobotdone). In questo caso, Transport Trolley è già fermo, dunque l'evento di alarm viene ignorato si attende il comando successivo.

```
State robotmovefailed{
println("$name | robot failed to move $lastmove") color red
[# endInstant= System.currentTimeMillis();
deltatime= (endInstant - startInstant);
#]
println("$name | end-start= $deltatime >/< MINT=$MINT") color red
}
Goto ignorealarm if [# deltatime <= MINT#] else alarmconsidered


State ignorealarm{
println("$name | alarm ignored --> resume to $lastmove") color red
}Goto moverobottoindoor if [# lastmove == "moverobottoindoor" #] else option2
State option2{} Goto moverobottohome if [# lastmove == "moverobottohome" #] else option3
State option3{} Goto moverobottocoldroom


State alarmconsidered{
println("$name | alarm considered!!! --> wait for moverobotdone or endalarm") color red
emit robotisstopped: robotisstopped //serve per led
} Transition t0 whenReply moverobotdone -> endalarmstate
whenEvent endalarm -> endalarmstate
```

```
State endalarmstate{
println("$name | resume to $lastmove") color red
[# startInstant= System.currentTimeMillis(); #]
}Goto moverobottoindoor if [# lastmove == "moverobottoindoor" #] else option2
State option2{} Goto moverobottohome if [# lastmove == "moverobottohome" #] else option3
State option3{} Goto moverobottocoldroom
```

Il timer

Dai requisiti si deve predisporre un timer che parte dall'ultima endalarm effettuata. Per fare ciò si adopera la classe Java System con il metodo:

```
System.currentTimeMillis()
```

Controllo del Led tramite Stato del Transport Trolley

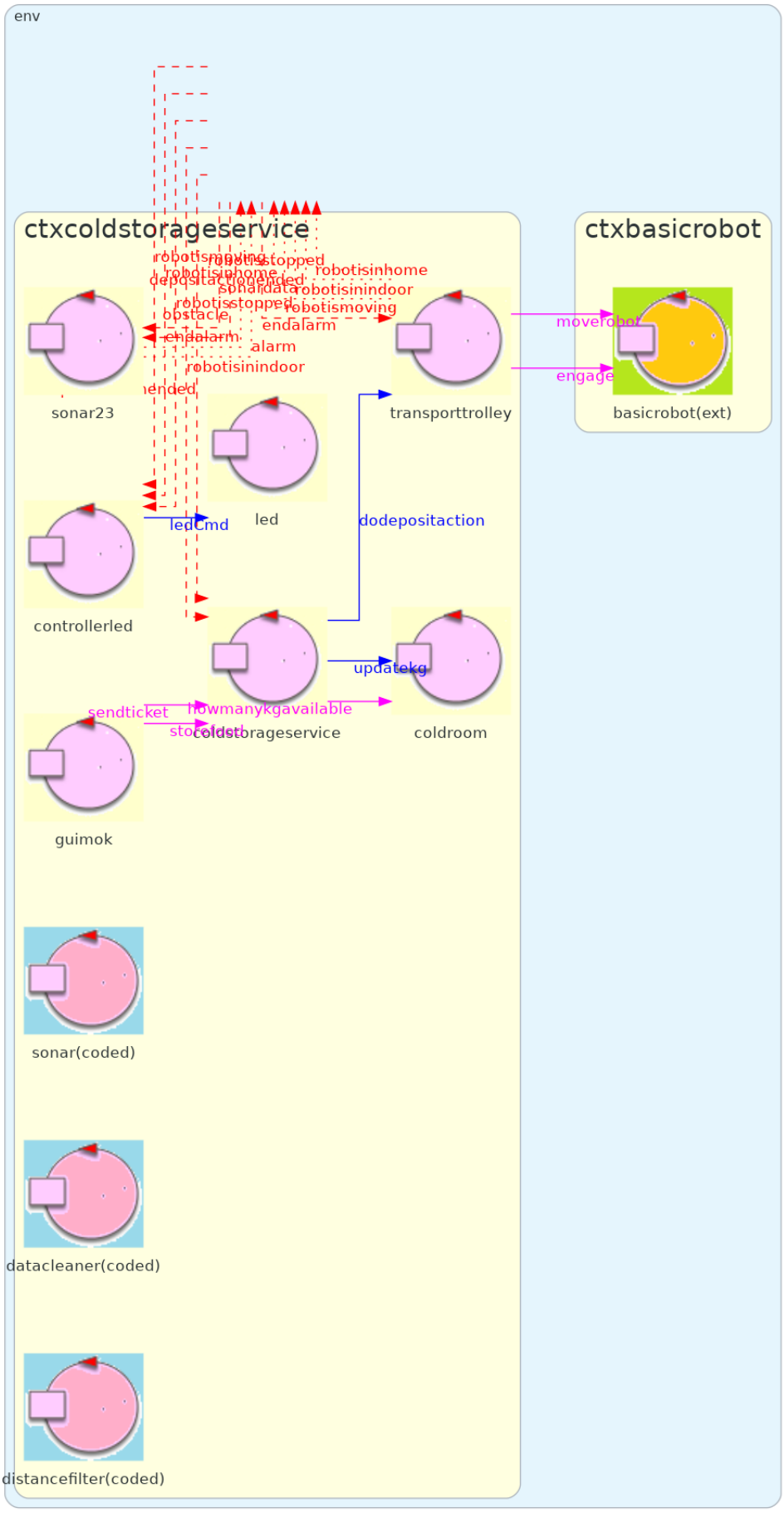
Dai requisiti il Led deve modificare il proprio stato a seconda dello stato del robot (se in movimento oppure se è in home oppure se è in stato di allarme). Questo è possibile se il Transport Trolley emette degli eventi sullo stato del robot ricevuti dal ControllerLed.

```
QActor controllerled context ctxcoldstorageservice{

State s0 initial{
println("${name} | START")
forward led -m ledCmd : ledCmd(off) //robot nasce in home
}

Transition t0 whenEvent robotismoving -> ledblink
                          whenEvent robotisinhome -> ledoff
                     whenEvent robotisstopped -> ledon
...
}
```

Il sistema è composto da:

- ***Sonar*** : effettua la misurazione e emette eventi. E' inserito in un contesto diverso perchè è specificato nei requisiti che è presente un Raspberry alla quale sarà collegato.
- ***Led***: viene inserito nello stesso contesto di Sonar
- ***ControllerLed***: entità che legge gli eventi del Transport Trolley per comandare il Led

Codice dell'architettura: https://github.com/RootLeo00/robot-coldstorage/tree/main/sprint3/src/coldstorageservice.qak


env
ctxcoldstorageservice
ctxbasicrobot
robotisstopped
robotisinhome
depositactionended
sonardata
robotisinindoor
obstacle
robotismoving
endalarm
alarm
moverobot
engage
sonar23
transporttrolley
basicrobot(ext)
led
ledCmd
dodepositaction
controllerled
updatekg
sendticket
howmanykgavailable
coldstorageservice
coldroom
guimok
sonar(coded)
datacleaner(coded)
distancefilter(coded)

coldstorageserviceArch

By Caterina Leonelli email: caterina.leonelli2@studio.unibo.it, GIT repo: https://github.com/RootLeo00/sw-eng.git

By Matteo Longhi email: matteo.longhi5@studio.unibo.it GIT repo: https://github.com/carnivuth/iss_2023_matteo_longhi.git