

Waste Incinerator Service

Sprint info

Sprint name	Sprint 1
Previous sprint	Sprint 0
Next sprint	
QAK model	sprint1.qak
Developed by	Alessio Benenati Giulia Fattori

Sprint Starting Condition and Goals

In the previous sprint we focused on requirements analysis and we produced a simple base architecture of what could be inferred by the assignment text. in this sprint we will focus on the relationship between WIS and OpRobot, our goals are

- finding the best way to divide the buisness logic between the OpRobot and the WIS actor
- consequently choosing the right model (**Actor** or **POJO**) for the OpRobot
- producing a simple prototype of the system reproducing the functioning of this two entities

Problem Analysis

Questa sezione ha come scopo definire la relazione che sussiste tra WIS e OpRobot, in particolare come suddividere la business logic tra questi due componenti e di conseguenza decidere come classificare OpRobot (POJO o Actor). Al termine di questo Sprint sara' disponibile un primo prototipo del sistema funzionante.

WIS and OpRobot

Based on the requirements, we understand that the code for controlling the basic robot is provided by the client. Specifically, the basic robot is modeled as a service that can be interacted with through messages such as `moverobot` :
`moverobot (TARGETX, TARGETY)` after engagement.

The BasicRobot autonomously handles movement requests to specific coordinates using an internal representation of the ServiceArea as a set of cells with a size of DR. Therefore, it is the responsibility of the WIS to store the coordinates of each point of interest in order to control the BasicRobot.

It remains to be determined whether it is truly necessary to model the OpRobot, or if its responsibilities can be incorporated into the WIS. Introducing an additional actor, OpRobot, which controls the basic robot and is managed by the WIS, would not significantly improve the isolation of responsibilities. On the contrary, it could create confusion regarding the division of business logic among the various actors.

For this reason, at this stage, it is more practical to keep the business logic concentrated solely within the WIS, without modeling the OpRobot. However, this decision may be subject to change in the future if the analysis reveals new responsibilities for the WIS, making it necessary to separate tasks (Single Responsibility Principle).

System Observability

da requisiti il WIS necessita di essere notificato in caso di determinati cambiamenti di stato del sistema (ad esempio l' `endOfBurining` dell'Incinerator). Questo sottointende la necessità di modellare una qualche forma di comunicazione tra i vari componenti del sistema (Scale, MonitoringDevice ed Incinerator) ed il WIS stesso. una prima soluzione potrebbe essere quella di prevedere un serie di richieste di stato effettuate dal WIS ai singoli componenti, tuttavia questo approccio si rivela inefficace in quanto prevede che il WIS 'sprechi' molto tempo di calcolo nell'interrogare i vari componenti. risulta invece meno dispendioso adottare il pattern observer facendo sì che siano i componenti stessi ad avvisare il WIS in caso di cambiamenti di stato. Tuttavia il compito del WIS consiste principalmente nel pilotare il BasicRobot, pertanto non è interessato a tutte le modifiche di stato del sistema, gli basta sapere che le condizioni per dare inizio ad un ciclo di movimento dell'OpRobot siano verificate per questo motivo si consiglia l'introduzione di un nuovo attore `WIS_Observer` che si occupi di osservare il sistema per intero e di notificare il WIS in merito alla validità tali condizioni

Future Sprints