

WasteIncineratorService

Intro

WasteIncineratorService is the final project assigned by Prof. [anatali](#) for the major course of Software Engineering.

You can find the requirements of the application [here](#).

QAK

The majority of the project has been modeled using QAK (Quasi Actor Kotlin), a meta-model created at UNIBO. QAK has its own DSL developed using xText that compiles directly into Kotlin code.

QAK allowed us to design the application with a higher level of abstraction, introducing the following main concepts:

- Actor: active entity modelled as finite state machines capable of sending and receiving messages.
- Context: an environment that contains some actors and abilitates them to communicate with other actors both in the same or in another context
- Interactions: abstractions of the main communications strategies (requests, dispatches, events).

We chose to use QAK because it helps bridge the abstraction gap, allowing us to maintain a higher level of technology independence during the initial phases of development.




You can find a detailed description of QAK [here](#).

Development process

We adopted a Scrub inspired development process, where the main assignement was divided in a series of sub-problems each faced during in a Sprint.

At the end of each Sprint we produced an executable version of the system covering some of the requirements.

Sprints

Sprint Name	Description	QAK	UserDoc
WIS_Sprint0	requirements analysis	sprint0.qak	 sprint0.md  sprint0.pdf  sprint0.html

Usage

Credits