# Waste Incinerator Service

## Requirements Analysis

### Structure

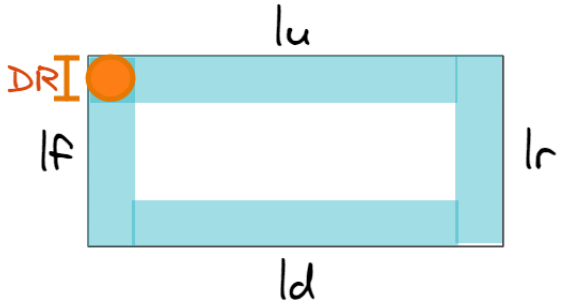
analyzing the natural language requirements text we found out the following entities that should be somehow modelled:

- ServiceArea
- WIS
- OpRobot
- DDRRobot
- Home
- Incinerator
  - BurnIn port
  - BurnOut port
- WasteIn
- WasteStorage
  - Scale
- RP
  - WRP
- AshOut
- AshStorage
- MonitoringDevice:
  - Sonar
  - Led

***Service Area Model***

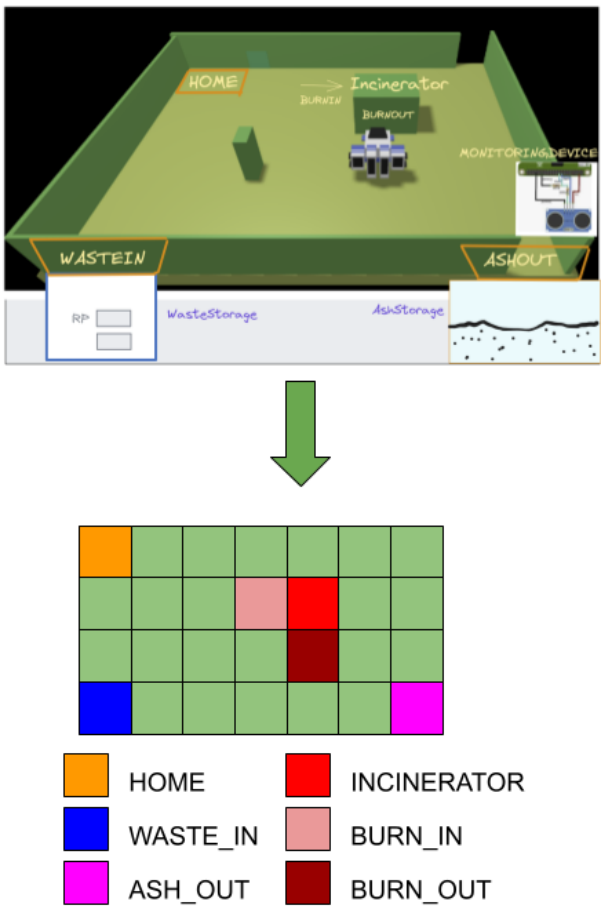
PROF

the **ServiceArea** is modelled as an Euclidean space delimited by its edges(similar to what has been done in the BoundaryWalk and RobotCleaner projects):




- the **perimeter edge** has length `lf+ld+lr+lu`
- being the ServiceArea rectangular we have `lf=lr && ld==lu`
- we define `DR=2R` being `R` the radius of the DDRRobot circumscribable circle

Given this model we have that **Home**, **BurnIn**, **BurnOut**,**WasteIn**, **AshOut** are all modelled as collections of cells in the serviceArea




***DDRRobot model***

The **OpRobot**, defined in the requirements as the robot controlled by the WIS, makes use of a DDRRobot (and its control software) given by the customer, we link the detailed definition of DDRRobot and its qak control software.

PROF

## Interaction

- activationCommand (WIS?) -> (Incinerator) // not evincible from requirements
- endOfBurning (Incinerator)->(OpRobot,WIS) // probably an event
- BurnInPortInfo (OpRobot? (fixed) )->(Incinerator) // emerges from requirements since otherwise the incinerator cant know when to start
- BurnOutPortInfo (Oprobot|WIS?)->(Incinerator) // emerges from requirements since the incinerator cannot burn if there are ahes
- scaleInfo (Scale)->(WIS) // not ncess. a message, could be a method to analyze during Problem
- sonarInfo (Sonar)->(WIS) // not necess. a message could be a method to analyze during Problem

## Behaviour

## Test

## Priority

- core buisness = Incinerator => probably i'll have to analyze the MonitoringDevice before the raspberry and the Scale

### SSGUI

- WasteStorage state
- AshStorage state
- Incinerator state
- OpRobot state

## Problem Analysis

### Entities Models

- ServiceArea -> other serviceAreas models
- WIS -> service (sends/recives messages)
- OpRobot -> service (given as service)
- DDRRobot -> service (sends/recives messages)
- Home -> colections of cells inside the serviceArea
- Incinerator -> actor (sends/receives messages) || pojo
    - BurnIn port
    - BurnOut port
- WasteIn -> collections of cells || coordinates
- WasteStorage -> context?
    - Scale -> actor (if sends/recives messages) || pojo (if Scaleinfo is retrieved using a method)
- RP -> pojo
    - WRP -> pojo attribute || config param of Scale actor, so that Scale sends the number of RPs, not the weights
- AshOut -> collections of cells || coordinates
- AshStorage -> context? (outside service area?)
- MonitoringDevice: -> actor? || context?
    - Sonar -> actor (sends/receives messages) || pojo
    - Led -> actor (receives messages) || pojo (is quite simple)