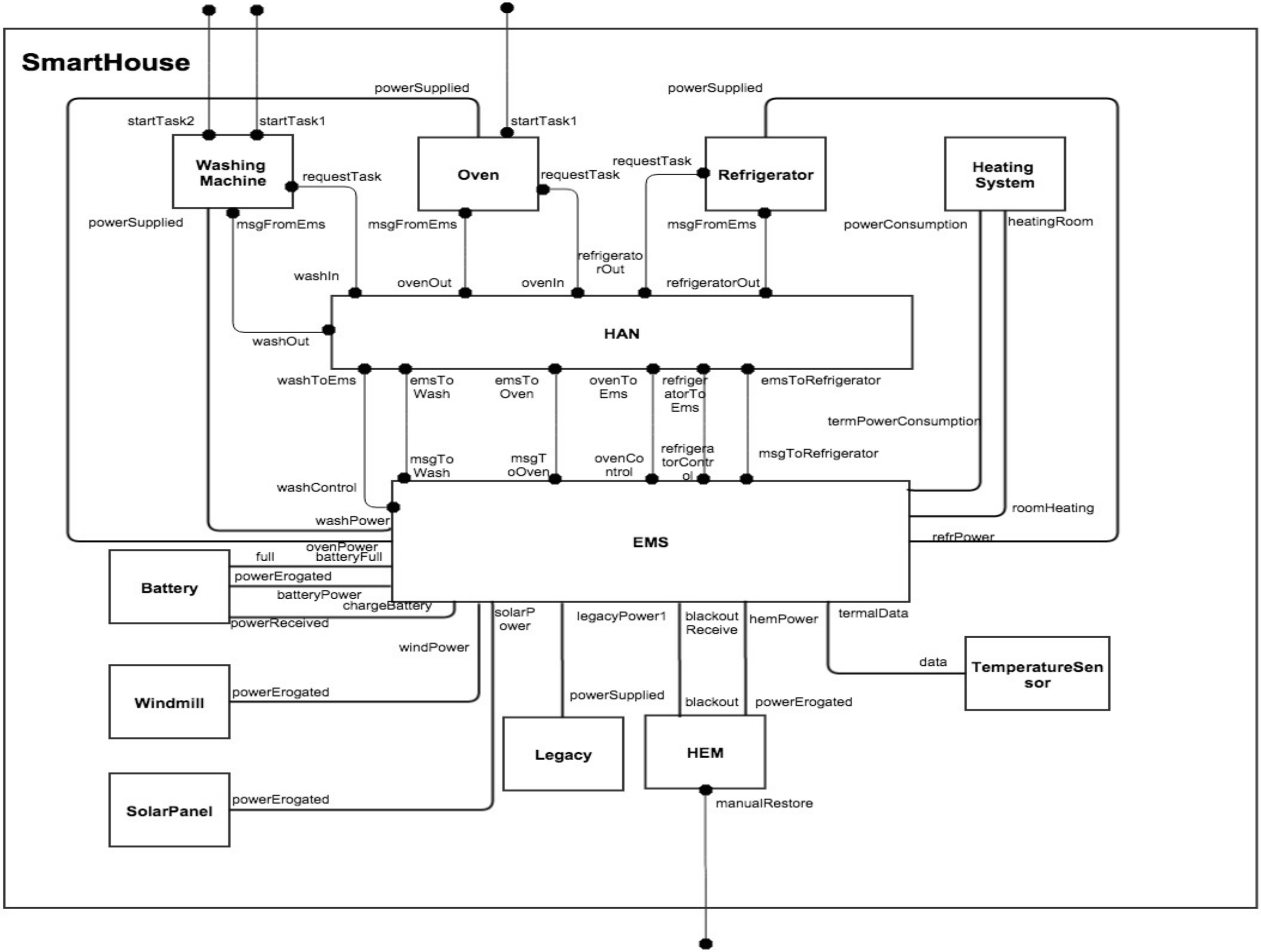
TRIO PROJECT: Smart House

Draft Solution

Filippo Pellolio
Andrea Rottigni

Class Diagram


SmartHouse
powerSupplied
powerSupplied
startTask2
startTask1
startTask1
Washing Machine
Oven
Refrigerator
Heating System
requestTask
requestTask
requestTask
powerSupplied
msgFromEms
msgFromEms
msgFromEms
powerConsumption
heatingRoom
refrigeratorOut
washIn
ovenOut
ovenIn
refrigeratorOut
HAN
washOut
washToEms
emsToWash
emsToOven
ovenToEms
refrigeratorToEms
emsToRefrigerator
termPowerConsumption
msgToWash
msgToOven
ovenControl
refrigeratorControl
msgToRefrigerator
washControl
washPower
roomHeating
ovenPower
refrPower
EMS
full
batteryFull
Battery
powerErogated
batteryPower
chargeBattery
powerReceived
solarPower
legacyPower1
blackoutReceive
hemPower
termalData
windPower
data
TemperatureSensor
Windmill
powerErogated
powerSupplied
blackout
powerErogated
Legacy
HEM
SolarPanel
powerErogated
manualRestore

Model Description

Device Inheritance

- The classes Refrigerator, Oven and Washing Machine inherit from the class device, in order to not replicate the code, that is basically the same.

Smart Devices

- They receive events from an human interface that we didn’t model.
- The execute task
- They communicate with the EMS through the HAN

Task

- A device can perform various tasks.
- We have decided to model a small number of tasks, but that can cover all the possible cases:
  - Must task
  - May task that can be shifted
  - May task that can be balanced
  - May task that can be shed

State Functions Meaning

- For the time being we haven’t specified in all the details all the states. For example, the state powerSupplied(p) can assume only one value of p at any instant.
- We decided to do that to not over specified our model.
- Of course, when we will translate our model for ZOT we have to take care of these “details”

Sensor

- We modeled only the temperature sensor because we can't think of useful functionalities that other kind of sensor can have in our system.
- We assumed that the temperature sensor is wired connected to the ems, and that there is a continuous exchange of data between them.

Heating System

- The heating system uses electrical power to warm ad cool the environment
- It doesn't use the HAN to communicate with the ems.
- The power given to the heating system can be balanced.
- It starts warming or cooling a room as soon as the temperature is below a certain threshold

Why only one legacy device?

- We modeled only one legacy device because basically there nothing they can do.
- For this reason model more than one legacy device is basically useless

Batteries and power suppliers

- For the time being we have assumed that the batteries can be loaded with any amount of power, but we can easily changed this by inserting some thresholds.
- Power suppliers are very simple, they produce energy in an unpredictable way.
- The energy produced is sent to the EMS, and store in the batteries if it is more than necessary, and the batteries are not full.

Blackout

- In case of overload, it is always possible to disconnect devices in order to avoid a blackout event, only if they are executing a may task.
- After a blackout the task is terminated to guarantee a liveness property. which prescribes that a device requiring power supply will eventually terminates its task correctly.

Liveness Property

- A device requiring power supply will eventually terminates its task correctly.
- If a task allows load shedding, may happen that it doesn’t complete its execution.

HAN

- We have modeled a very simple Home Area Network that simply introduces some delays.
- We don’t handle packets losses.

Where to go now?

- Step 1: refine some axioms
- Step 2: complete the documentation of the classes
- Step 3: translate our model to verify it with Zot