FINAL PROJECT

Autonomous Software Agents 2021/2022

Giovanni Ambrosi

Introduction

The main target of this project is to handle in the best efficient way a house equipped with a panel system. Panels are installed on the roof and absorb solar energy generating electricity. This electric current is used for satisfying the house's consumption but can also be stocked in a battery pack installed inside the house. Based on the energy's consumption a control unit (briefly an agent) has three ways to use the panel's electricity:

1. Use for house demand (i.e. charging car, turning lights on);
2. Stock in rechargeable batteries if the house demand is totally covered;
3. Sell it to public authority;

Eventually the agent can buy (small) quantities of current from the public line if batteries are empty and panels disconnected.

The house is obviously composed of other devices controlled by some agents that are somewhat hierarchical structured. At the top of the pyramid, there is the house agent, which controls panels, and batteries through sensors. It decides how to split the use of energy between the devices and has perhaps the main control of the house.

The other agents belong to a lower level. For example, if the electric car has low charge the house agent will use more energy to allow its battery to reach a minimum charge level as fast as it can (the way of how to charge the battery is managed by its own agent). Otherwise, if the car's battery is enough charged (a threshold is chosen a priori) the house agent uses less quantity of electricity.

House description and blueprint

The house has two floors. At the ground level, we have the garage, the laundry room, the kitchen, one of the leaving rooms and the bathroom, while on the upper floor there are the bedrooms, the second living room and a 6th room where the batteries are installed (called simply battery room).

The house presents a main entry on the north side constituted of an electronic door with ID scanner. Once entered we have the leaving room provided with a TV screen, a sofa and a table used for every type of activity.

There are a second and a third accesses consisting of French doors in the kitchen and in the living room at the ground floor that guarantee a faster way to go to the garage, avoiding residents to pass through the main doorway each time.

The garage has an electric car parked inside connected to its charge column. It is divided from the rest of the house and is located on the south side.

The laundry room has a washing machine and other minor devices.

In the leaving room, the stairs lead up to the upper floor and have the bathroom next to them.

As said at the beginning, the roof is equipped with solar panels connected to an inverter which allows the system to feed the house and keep the count of kWs used, produced and sold.

Rooms

1. Kitchen

The kitchen is provided with a main light and it is the most used room of the house, used botfor eating and sometimes for working

As mentioned before the kitchen has an independent door to get into the garden but also a door accessible from the living room.

2. Bathroom

The bathroom is at the ground floor, next to the stairs and is provided with autonomous light for illumination.

3. Laundry Room

As mentioned in the description laundry room has only small devices. In the scenario it will be considered only for movements of the residents and the consumption coming from the devices of this room is averaged 0.5 kWs plus an additional kW.

4. Garage

The garage has the electric car and the charge column where the car can obviously charge.

It is used for parking and recharging only and eventually for other activities which are not relevant to the project.

5. Battery room

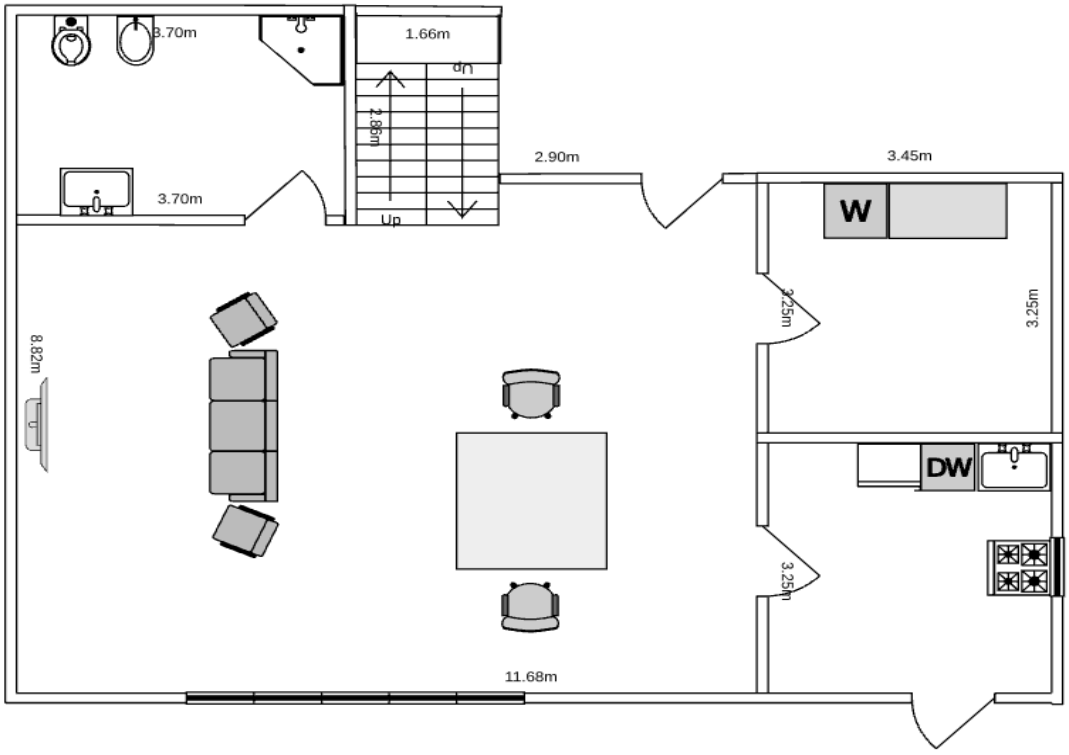
This is the core of the house, right to the stairs once get to the upper floor. Batteries are fixed on the wall.

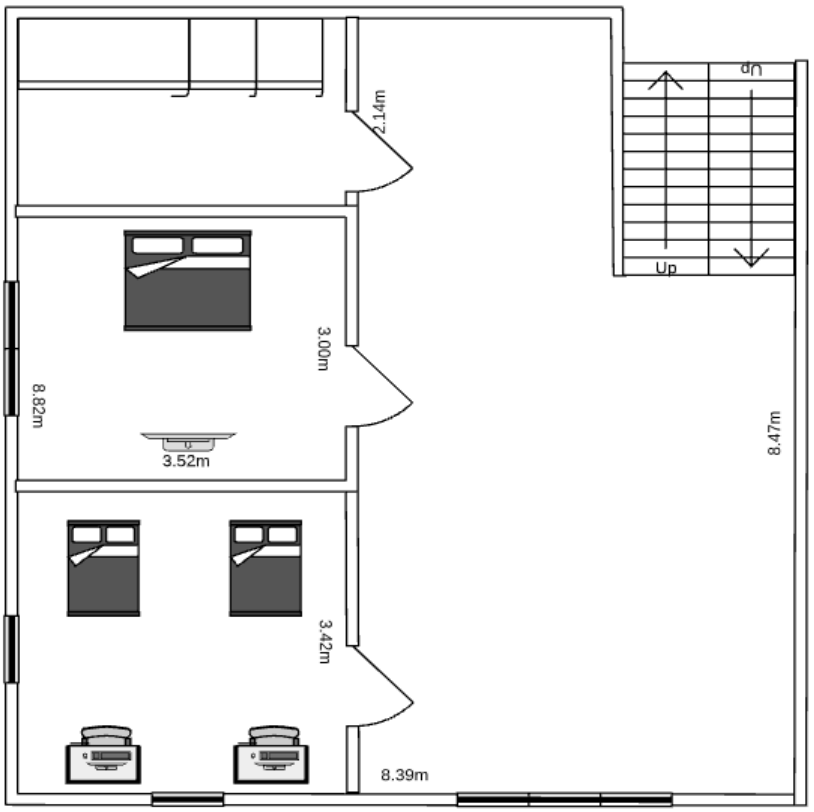
As mentioned in the general description batteries have the function of feeding the house during the night, when solar panels don’t work, allowing to spare money.

The access into this room is intended only for cleaning it up with a vacuum cleaner robot and for solving technical issues of the batteries.

6. Bedroom(s)

Bedrooms have own lights. Based on the presence of people and time the agent will turn lights on/off (as for the other lights in the house). Bedrooms are used mainly for sleeping and eventually for studying (these activities are not distinguished in the final code)


3.70m
1.66m
Up
1.86m
2.90m
3.45m
W
3.70m
Up
3.25m
3.25m
8.82m
DW
3.25m
11.68m


2.14m
Up
3.00m
8.82m
3.52m
8.47m
3.42m
8.39m

Devices

Vacuum cleaner

The house is provided of two vacuum cleaner robots, one per level. They know the map of the entire house, both floor-0 and floor-1.

Status: status is generally either on or off.

Actions: the actions that they can perform are Clean and moveTo with their obvious meaning

Consume: not relevant

Prerequisites: -

Lights

Lights provide illumination to the rooms at every time of the day and are autonomous.

Status: Light status can be on or off.

Methods:

1. **switchOnLight**: turns light on and updates consume of the house (+1)
2. **switchOffLight**: turns light off and updates consume of the house (-1)

Consume: each light consumes 1 kW (unit measure is not real) when on, 0 otherwise

Electric car

The electric car has a battery of 54Kw, but its capacity is reported through a percentage parameter (10%,20%...) and has to be connected to the charge column every time it is parked in the garage. This is a sort of security system in order to know always the charge status.

Status:

- 0% < charge < 40% need to be recharged in DC_mode (Direct Current);
- 40% < charge < 100 % charge mode switches to AC_mode (Alternate Current) where the recharge process is slower in terms of time and consume;

Methods:

1. **switchOncharge**: switch the charge status to on
2. **switchOffcharge**: switch the charge status to off
3. **get_charge**: verifies the status of the charge and calls methods charge_AC_mode or charge_DC_mode
4. **decrease_charge**: decreases the charge of a battery in order to simulate its use during a day (i.e. there might be days where the car is used less and for that reason its battery decreases less)
5. **charge_DC_mode:** charges the car faster than AC_mode. When the charge reaches level 40% it calls method charge_AC_mode that keep charging the car
6. **charge_AC_mode:** invoked if the battery of the car is more of 40% or called by the method above. It keeps charging the car till 100% then calls switchOffCharge.

Consume: 5kW if DC_mode on, 3 kW otherwise.

Prerequisites: car connected to the charge point (automatically done). DC_mode and AC_mode are chosen based on its current capacity. DC_mode or AC_mode are automatically selected by the agent.

Solar panels

The entire system has a capacity of 10 kWh and has an average production of 4kWs from 6.00 am to 18.00 pm. After that the energy producted is set to 0. As described at the beginning the electricity is primarily used for feeding the house. Only if panels produce a surplus is stored in the batteries and eventually sold.

Status

- Capacity which indicates the current production of the panels

Methods:

1. **Production_during_time:** simulates the production of current basing on the clock. From 00.00 to 6.00 it is considered 0 (panels disconnected) then the average production is 4 till 18.00 and then set to 0 again
2. **Start_store_energy(diff):** starts storing the surplus of energy producted into the battery system

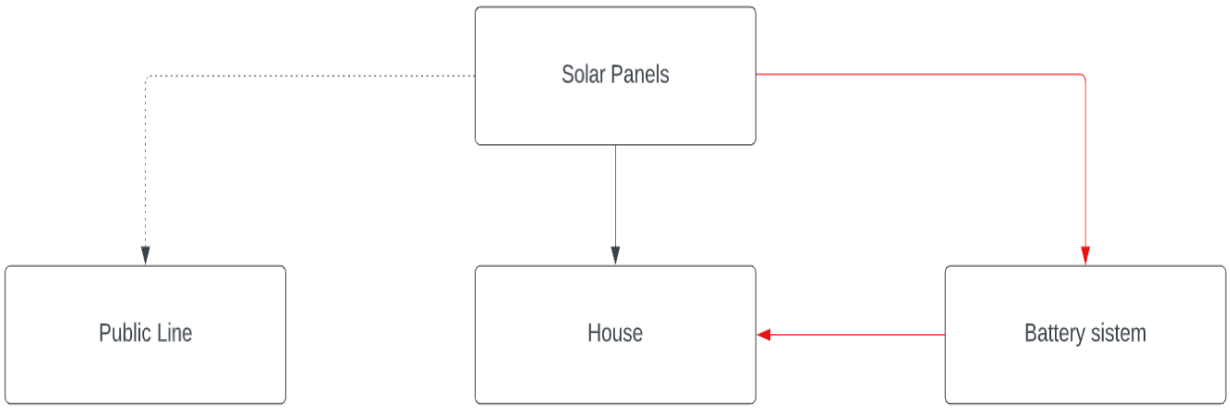
Battery system

When the night has come the solar panels stop working. The feeding of the house swaps to the batteries and the pack has a capacity of 5kW. When they are full the surplus of the panels is sold to the public authority.

The devices is controlled by the same agent of the panels and has a series of parameters that indicate some features as the current charge, current sold and bought.

Methods:

1. **Use_batteries**: decreases the capacity of the battery system basing on the parameter given as input
2. **Is_full():** returns true if the batteries are full (15 kWs)
3. **Is_empty():** returns false if the capacity is 0
4. **Start_store(diff):** method invoked by the solar panels, starts increasing capacity of the system of an amount depending on the parameter given as input



In the figure above there is a simple scheme of the devices. The arrow in black indicates the primary function while the arrows in red explains the alternative way to feed the house.

Metrics

Sell cost of electricity

Selling electricity is paid 0.20 €/kWh while buying it costs 0.10 €/kWh. The house is intended for selling electricity only or buy small quantities. At the end of the day a final report of current bought and sold is showed on the console.

People and agents

People

Residents in the house include two people, Stefano and Elisabetta. Stefano is out from 8.00 am to 19.00 pm and goes to work by car, it is perhaps important that at least 40% of the battery capacity is stored. Elisabetta works in a smart modality and has to take care of the house, so in her daily schedule movements in the house are considered.

The action they can perform is MoveTo that allows residents to move through the rooms. But every action or method in the daily schedule (called **brutal force**) is intended as “taken by a human action”. For example at 8.00 Stefano is going to work by car. He needs to switch off the charge so in the schedule it is performed the action switch_off_charge, method of car class, but it is formally performed by Stefano and not by the agent.

House agent

The house agent controls panels and battery system.

In particular, it has to choose when to stock the energy and when to sell it. The idea is that solar energy and battery resources are enough for the agent to not buy electricity (or buy small quantities) and so try to minimize the costs of the house. Its work is summarized in the four bullet points below:

1. The current is **primarily** used for feeding the house;
2. If the panels produce **more energy** than the demand it is stored in the batteries;
3. When the batteries are **full** (invoking the method is_full()) the eventual surplus is sold to the public authority, which obviously pay the owner for the service (0.20 €/kWh);
4. If batteries are empty (method is_empty()) then the agent buys electricity.

The four actions above are chosen by a sensor that looks at the production of the panels

The consume of the house is tracked by a sensor and is computed as the sum of the consumptions of lights and car plus an additional consume of 0.5 kWs and/or other random consumes in the daily schedule

Lights Agent

Lights agent helps people in the house to switch on and off lights automatically. The behavoiur of the agent is the following:

1. Switch on lights of a room if a person is in there
2. Switch off lights of a room if a person is no more in there

The consume of the lights is communicated by the agent to the house and the house agent can look at it to compute the total consumption of the electricity.

At the end of the day every light is in the ‘off’ status, otherwise residents would sleep during night with the light on.

Car agent

Depending on the status of the battery the agent has to choose the type of the charge. The car is automatically plugged in the charge column as soon it gets parked in the garage. The two types of modes are:

1. DC mode (fast mode) and consumes 5 kWs;
2. AC mode (normal mode) and consumes 3 kWs;
3. The overall consume is communicated to the house through a specific parameter.

A sensor is thought to verify the charge of the battery and to inform the agent which will perform the best action invoking he right method.

The DC mode has a double speed with respect to the AC mode and it applied when the charge is less than 40%. The house agent helps the car agent to charge the car using primarly the current coming from panels, otherwise batteries have to charge the car (this is the case where house might buy electricity).

Planning agent

Planning agents are the two vacuum cleaners in the house, called VaccumCleanerAgent_0 and VaccumCleanerAgent_1. They are responsible for the respective floor indicated in the number at the end of their name (0 for ground floor and 1 for upper floor).

In the scenario the ground floor is the first to be cleaned up and the agent is activated at 8.00. Then once it has finished its work the other vacuum cleaner gets activate.

The two agents are thought as two robots, but the lightAgent (the one that manages the lights) is not able to detect the presence of them in a room and perhaps does not turn lights on. This is coherent with the target of the project which is spare energy as much as possible.

Implementation

Sensors and agent perception

Each agent perceives the environment through a sensor that updates its beliefs. The house is provided by the following sensors

Sensor	**Agent**
PersonSensor	LightAgent
LightSensor	LightAgent
CarSensor	CarAgent
VacCleanSensor	VacuumCleanerAgent(s)
VacPresSensor	VacuumCleanerAgent(s)
PanelSensor	HouseAgent

Sensor and implementation for planning agent

Vacuum Cleaner Agents perceive the environment as a set of rooms where they can move and can perceive the room in one of the two status, clean or dirty. The movements between rooms are allowed thanks to the doors connecting them and for that reason agents can move only between communicating rooms. The agents cannot go up and down through stairs and each floor of the house is provided of its own Vacuum Cleaner Agent.

Sensor and implementation for other agents

Light Agent perceives the presence of a person in a room through PersonPersSensor and turns on the light of that room and switches off the light of the room left by the person.

Example: Stefano goes from living room to kitchen, light agent turns on the light of the kitchen and switches off the light of the living room.

Furthermore, the agent can rely on the LightSensor which is used in cases lights are switched on/off by residents (brutal force) in order to updates its beliefs.

I have assumed that at the beginning of the day all lights are off. Once the residents start moving in rooms the lightAgent activates

House Agent controls solar panels and battery system with its sensor PanelSensor. It receives a signal from that about the production of the panels and decide if store the current produced in the batteries or sell it.

In general House Agent knows only if the panels are producing electricity or not and the status of the batteries.

Car agent uses the sensor CarSensor that informs it about the status of the car. Each time the car is in the garage it is plugged in the charge column (in the scenario this action is implemented in the daily schedule, brutal force) and agent observes the status of the battery. Once the battery is charged enough the agent just switches off the charge.

Its knowledge is perhaps limited to the garage and the car and has no need to know the house blueprint, which is fundamental for the vacuum cleaners

Agents acting in a shared environment

As said before **House Agent** governs the panels and batteries. At the beginning of the day panels are disconnected and the Agent uses the batteries to feed the house which has a minimum consume of 0.5 kWs. As the time proceeds the panels start producing and the Agent receives information about the consumption of the house, which can increase if a light turns on or the car gets in charge. The actions of this agent are summarized in the following:

00.00 <= time < 07.00

The sun does not have raised yet, House Agent use batteries through the method use_batteries() and the consumption of the house is 0.5 kWs. If batteries are empty a message is logged out on the console and the Agent starts buying current (prize 0.10 €/kWh).

07.00 <= time < 19.00

Panels produce 4 kWs per hour, the sensor of consumption of the house tracks each consume of the devices. If it is less than 4 kWs then the House Agent invokes the method start_store() meaning

that the energy not used id stored in the batteries. Once the battery system is full the current is sold with a prize of 0.20 €.

19.00 <= time < 00.00

Panels decrease their production till 0 kWs and the Agent uses the batteries again. At 23.30 the Agent computes a daily report of the current bought and sold.

The car Agent simply disconnects the car if it is enough charged using the method Switch_OFF_charge(). The idea is that Stefano, who is the main user of the car puts it in charge once he is back to home. The Agent is informed that the car is plugged in and verifies its charge deciding what type of modality to use (AC mode or DC mode explained in the device section).

The other agents have independent behaviour and do not have constraints on their actions.

Vacuum Cleaner Agent moves between rooms and clean them with clean action. The rooms are considered dirty every morning and the Vacuum Cleaner moves through the rooms of its floor cleaning them up.

In order to avoid that the two Cleaners overlap their actions the first is activated in the morning, while the second in the afternoon.

Light Agent turns on and off lights through its methods turn_on(), turn_off() and works in the both floors (no distinction between ground floor and upper floor as for the cleaning agents)

Agent interaction and coordination

As mentioned in the first section the agents are sort of hierarchy structured. House Agent decides how to use the energy of the panels. Furthermore, if the Car Agent has to charge the car the House Agent **chooses from what device take the energy needed**.

Light Agent has to turn on and off light depending on the presence or not of people in a room **but doesn't take into account the presence of the robots.**

Definitely the main interaction is between house and car agents.

Scenarios

The daily schedule starts with the residents moving in the house. The Light Agent detects their moves and starts switching on/off the lights.

hh 00.00 < hh < hh 06.00

Residents are sleeping and the rooms are considered dirty from the Vacuum Cleaner Agents.

hh 6.00

Stefano wakes up and plans to move from bedroom_1 to kitchen and bath_room. All this operations take 1.30 hours and then he goes to garage.

The rooms are putted in the 'dirty' status in order to ensure that the vacuum cleaners solve their task.

hh 6.30

Elisabetta wakes up and moves from bedroom_1 to kitchen going through living_room_1 and living_room_0

hh 8.00

The car is taken by Stefano and the charge is switched off, consume of the house decreases if the car was charging otherwise the consume remains constant. Once the car is disconnected the consume of the house depends on the lights and some additional consumes I set in the daily schedule.

Meanwhile the vacuum cleaner of the ground floor (VacuumCleaner_0) starts cleaning

hh 8.30

Elisabetta goes in the laundry_room, uses the washing machine (its consume is quantified in 1kW) and then spends her entire day walking in the house for different activities (not explicitly mentioned).

hh 10.00

The vacuumCleanerAgents starts cleaning rooms up. VaccumCleanerAgent_0 has to clean all the ground floor, VaccumCleanerAgent_1 the first floor. After they finish their planning all rooms are cleaned and the two robots turns off until the day after. The VaccumCleanerAgent_0 goes into the laundry_room once it turns off while the VaccumCleanerAgent_1 in the bedroom_2.

hh 10.00 < time < hh 19.30

In this time only movements of Elisabetta are considered. In the log file I provided the lights keep changing their status from on and off.

hh 19.30

Stefano comes back to home, the car is parked in the garage and the carAgent verifies its status. Depending on the charge the agent decides the best way to full the battery.

The VacuumCleaner_1 starts cleaning

hh 21.45

Residents are in the bedroom_1, all lights are turned off.

hh 22.45

Vacuum_Cleaner_1 finishes to clean rooms. It will be activate the day after, once the rooms will be dirty

hh 23.30

houseAgent prints its daily report about the electricity.

Additional: every hour the houseAgent shows the production of the panels and what type of action is performing (charging batteries or not). Moreover, it shows the consume of the house and the amount of electricity stored in the batteries (this can be observed in the log file).

At certain times I manually increased and decreased the consumption of the house in order to make the simulation more real and dynamic.

In the ImplementedScenario.log I reported only the main points of the day and I brefly comment them

The comments can be recognized because they are marked in this way

---

//COMMENT

In the planningScenario.log instead it is reported a simulation of all day till the tasks are completed. At the end of the file there is a briefly comment on that

Source code organization

In src/myworld all files are implemented

In ReportandLog log files and report are loaded

https://github.com/GiovanniAmbrosi/232252_Ambrosi_Giovanni_ASA2022/upload/main