Final Project

Autonomous Software Agents - UniTn 2021/2022

Andrea Cristiano

229370

# Introduction

Rapid development of information technology has introduced a new generation of the software paradigm, named Agent Oriented Programming. In the following sections, a system model for managing tasks in smart homes using multi-agent solutions is proposed. The proposed solution organizes work and distributes tasks to individual family members as well as to additional subjects called agents. All of them will interact with each other and with the devices the house is equipped with. The document describes the application context and each relevant subject, human, software and hardware. The proposal is described with regard to the possibility of implementing a smart home project.

Intelligent agents are more and more present in our daily lives. They acquire data from the environment by using built-in sensors, such as cameras, microphones, and motion and temperature sensors. Information is processed to extract data that are then shared with the other devices on the network. Smart homes are an example of that. The agents are able to monitor the entire home in the absence of residents, analyzing and/or modifying existing conditions, like temperature and energy consumption. They make use of the house’s devices in order to increase the comfort of the resident’s life. In addition, in case of failure, such information can be shared with the tenants, with the producer of the system and with many other subjects not usually in close contact with the house (police, firefighter, etc.).

The document is divided into several sections to increase the readability. In [S](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6427340/#sec1-sensors-19-01249)ection 2, a description of the house is provided on different levels, with the addition of the house’s blueprint.  
[S](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6427340/#sec1-sensors-19-01249)ection 3 offers an overview on the devices with which the house is equipped. Section 4 introduces some metrics used to evaluate the impact of the agents on the house. Finally, Section 5 introduces the main subjects that interact in the context of the house: the agents and the residents.

The implemented simulation relies on knowledge-based agent framework. The adopted multi-agent-system includes:

* A house agent, which acquires all the possible information about the environment through the sensors or devices, but also through interactions with the tenants and the other agents.
* Two vacuum cleaner agents, one for each house level, which interact with the environment through the assigned vacuum cleaner device and the house agent.

The agent-based architecture also allows device agents to issue requests to the home system in order to supports automatic adaptation to changing conditions. Because of the general knowledge owned by the house agent, this last is also framed as mediator agent. It provides the surrounding conditions for agents to exist and that mediates both the interaction among agents and the access to resources. But not only that, it can expose one or more services, or can interact with devices, agents, and users in order to reach the goals.

On the contrary, the other agents are usually specialized in achieving one single goal or a set of goals closely related to each other, or even a set of goals belonging to the same application field.

Agents are thought to be embedded within devices that present some advanced capability (i.e., home appliances with some computational capabilities and memory availability).  
When necessary, they can encapsulate device status and properties that are used to cooperate with other agents, for example by making requests toward them. Such requests are generated after a sensor data gathering phase or when the internal status changes.

Ideally, the number of connected resources and agents may vary unpredictably, for example by connecting or disconnecting devices, or even if new users come into the house.  
That doesn’t need to redefine the communication and negotiation framework.

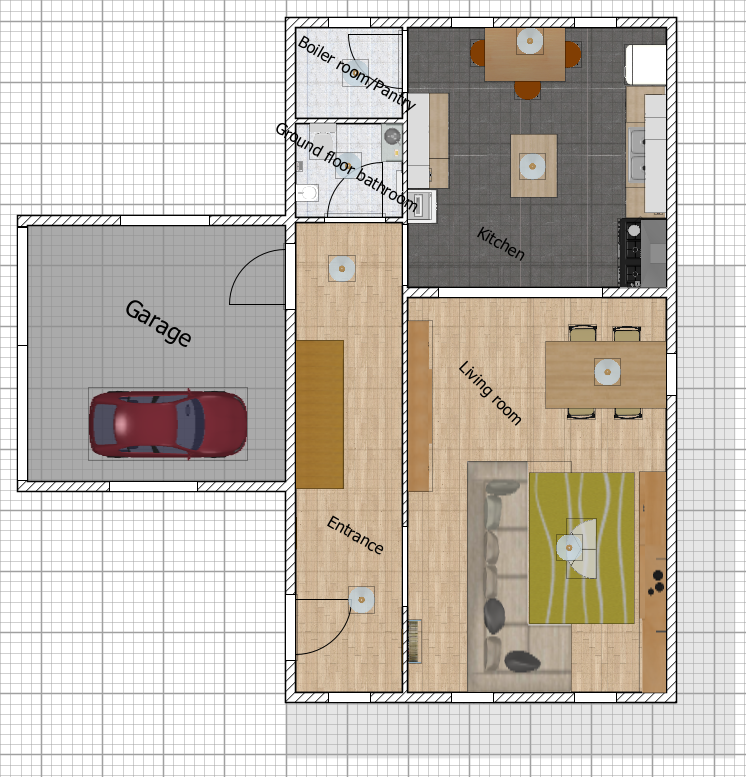
# House description and blueprint

The house is on two floors, furnished with a minimal but essential custom-made furniture.  
On the ground floor, there is the entrance. From here, it is possible to reach almost all the rooms located on the ground floor and, through the staircase, the rooms on the first floor.

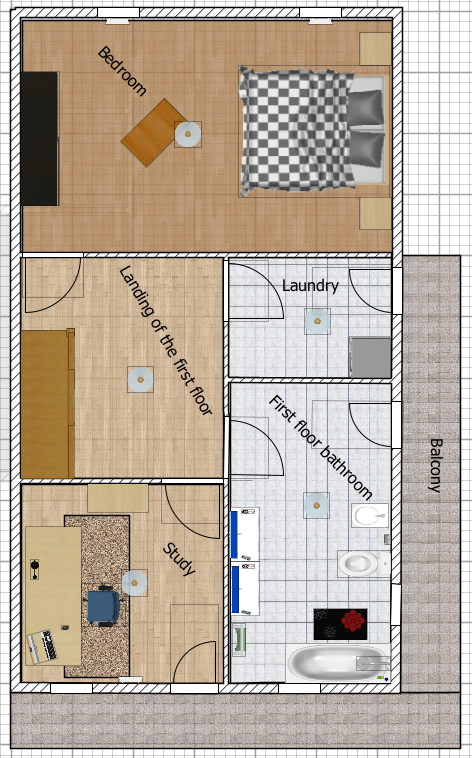
The rooms located on the ground floor are the kitchen, the living room and the ground floor bathroom. The living room and the kitchen are connecting rooms, but if there are guests, they can be made independent. All these rooms are reachable from the entrance. In addition, the garage has an independent access to the entrance. From the kitchen, it is possible to reach the boiler room where the geo-thermal plant is installed. This last room is also used as panty (note that this room can be reached only through the kitchen).

As mentioned above, the first floor can be reached only by the staircase. On the first floor there are the study, the laundry, the bedroom and the first-floor bathroom. All these rooms can be reached from the landing of the first floor. In addition, a big balcony connects the laundry to the study, from which it can be accessed. Moreover, a glass door connects the bathroom with the balcony

### Ground Floor



### First Floor



# Limitations

Due to the complexity of the real world, the implemented simulation accepts some limitations that help to simplify the description of the world.

The light sensors are simulated by defining specific hours of the day in which the house agent has to automatically turn on or off the lights inside the house. In general, two time slots have been detected: daily, from 07.00 to 21.30, and nightly from 21.30 and 07.00. During the nightly time slot, there could be situations in which the lights have to be turned off, for example when the tenants are sleeping. In order to do that, the house agent has been provided with a specific belief that let him know if the tenants are awake or not.

In relation with what has been said above, another limitation regards the behavior of the tenants during the sleep time: no activities or actions are performed from the tenants in that specific period of time, they just sleep until the next alarm sound.

The opening and closing sensors for the roll-up shutters have a simulated behavior that is similar to the light sensors. The opening and closing actions are managed by the time, instead of the presence of the sun or some other factor. The roll-up shutters are opened in the morning, when the alarm clock starts ringing, and are closed when the sleep time comes.

The variation of the temperature inside the house can be due to a very large variety of factors (i.e. the season of the year, the presence of open windows). Because of that, during the simulation, the variation of the temperature is randomly computed, between a range of 18 and 25 degrees.  
Then the desired temperature has been set to 22 degrees with a tolerance of ±2 degrees.

Something similar to the temperature variation has been done for the cleanliness of the rooms, which can be influenced by several different factors as well. Due to that, daily some rooms are marked randomly as dirty and clean-up activities are scheduled.

Regarding the other implemented devices, the motion sensors don’t present a real “physic” implementation, instead they are abstract elements of the house world. That choice is due to the way in which the tenants’ movements are simulated.

Another limitation is related to the actions performed by the tenants throughout the day. The actions are usually represented by the presence of one or more tenants into a specific room of the house. For example, there isn’t a real activity called lunch or dinner, but instead, if a tenant is in the kitchen between 13.00 and 14.00, or 20.00 and 22.00, we suppose that he/she is having lunch or dinner.

# Rooms

## Rooms’ configuration

The whole house is heated and cooled by the geo-thermal plant. Every room has underfloor heating, with an exception for the garage, where radiators are used. In general, the temperature is set between 19°C and 21°C during winter and between 21°C and 23°C during summer. However, each room has its own independent thermostat to control the temperature and different rooms can have different temperatures at the same time. For example, in the winter, the garage has a temperature set between 16°C and 18°C, because the tenants do not spend much time in there. While, in the same period of the year, the bathrooms temperature is set between 20°C and 22°C when the tenants wake up.

All the rooms except for the bathrooms are equipped with at least one camera. For obvious reasons, the bathrooms are only equipped with motion sensors. Some cameras are then installed outside, along the house’s perimeter, plus in front of the front door and the garage entrance. Those cameras are used to identify people and objects, their presence and their position.

All the rooms are equipped with ambient mics, which allow to collect sound information.  
The front door is instead equipped with a directional microphone. The outside of the house as well is equipped with the ambient mics, but because of the environmental sound, they are usually unused and turned off.

Every floor is equipped with a tablet to control the house’s functionalities. The tables are usually located at the entrance of the ground floor and in the landing of the first floor.  
Of course, the tables can be carried around the house. Every room has at least one wireless charging point for those devices that are equipped with it.

In addition, every room is equipped with a charging point for the vacuum cleaner, which is not forced to return to its main station if it has to be recharged.

The rest of the section will provide more details about each room of the house.

### Entrance

The entrance represents the first room in which people come across when they come from the outside of the house or from the garage. Through this room, it is possible to reach the first floor and all the rooms on the ground floor, except for the boiler room/pantry.

As said before, the tablet of the ground floor can be found here. The room is very simple, and it’s furnished with the essentials: coat-rack and typical furniture.

The illumination can come either from the window located on the wall near the front door, from the living room or from the lights located on the ceiling. The room is long enough to be required to install two lights.

The entrance is usually a crossing point, it is one of the most used room as a function of its own location.

### Kitchen

The kitchen is located on the ground floor and from there it is possible to reach the living room, the boiler room/pantry and the entrance. The kitchen has a wall in common with the bathroom, but it is necessary to go through the entrance in order to reach the bathroom from the kitchen, and vice versa.

The kitchen is equipped with typical instruments and devices. Near the small electrical appliances (toaster, blender, etc.), it is possible to find large household appliances: dishwasher, fridge, induction hob, oven and microwave and food processor. While the small electrical appliances can be plugged and unplugged easily, the large household appliances are always plugged, assuming that everting works as it should.

It is interesting to highlight that together with the typical kitchen’s furniture, the room is equipped with a kitchen island, located at the center of the room, and with a table on the other side of the room.

In the kitchen there are two windows that illuminate the room by natural light during the day. Some other light comes from the living room thanks to the communication between the two rooms. There is then a main light located in the middle of the room, over the kitchen island, plus additional separated lights, one over the table, and others over the kitchen surfaces and cooktop.

This is one of the most used room. Usually the tenants consume here their main daily meals.

### Boiler room and Pantry

This room is located on the ground floor, and it can be reached only from the kitchen. Here there is the geo-thermal plant and some shelves, since the room is also used as pantry. In this room it is possible to find the main station of the vacuum cleaner that is responsible for the cleaning of the entire ground floor.

Illumination is provided by one window that allows the sunrays to penetrate during the day, and one main light located in the middle of the room.

This is one of the lesser used rooms. Usually, residents go there if they have to take or place objects, if they have to take care of the vacuum cleaner, or in case of maintenance of the geo-thermal plant.

### Living room

The living room is located on the ground floor and from there it is possible to reach the kitchen and the entrance. In order to reach the bathroom, it is necessary to go through the entrance, and vice versa.

The room can be subdivided in two macro areas. The first, the bigger one, is the one in which the sofa and most of the furniture can be found. Among them, there is also the TV stand and a little bookcase behind the sofa. The other area is closer to the kitchen and it is occupied by a table.

In the living room there are three windows that illuminate the room by natural light during the day. Some light also comes from the kitchen and the entrance, thanks to the communication between the rooms. There is then a light located over the table, and another one over the sofa. Another light is located behind the sofa near the door between the entrance and the living room.

Usually, the residents spend a lot of time in the living room over the weekend. This room in equipped with a smart TV and a bed-sofa, and sometimes they watch television and then spend the night on the bed-sofa. Because of the bed-sofa, if the tenants have guests, the living room becomes the guest room.

### Ground floor bathroom

The ground floor bathroom is a very simple room, provided with the typical bathroom’s supplies.  
In this bathroom there is also the shower. Although in the room there is the underfloor heating, the ground floor bathroom is also provided with a towel warmer.

The illumination is provided by one window and a main light located in the middle of the room. Over the washbasin there is a mirror which has its own light.

In case of guests, the ground floor bathroom becomes the guests’ bathroom and the residence will use only the one on the first floor.

### Garage

The garage is located on the ground floor and it has an independent access to the entrance of the house.

The garage is large enough for two cars, but the residents possess just one car. This last therefore occupies one park area of the garage, while the other is used as a warehouse, with shelves and a worktable with various tools. In between the two, there is a charging station for electric cars. The same charging station can be used to recharge the electric bicycle.

Illumination is provided by two windows, one for each side, that allow the sunrays to penetrate during the day, and two main lights, located in the middle of each park area.

During the winter this is one of the lesser used rooms, while during the summer the tenants use it form many different activities.

### Landing of the first floor

The landing of the first floor allows the access to all the rooms situated on the first floor. It is connected to the entrance by a staircase.

As said before, usually the tablet of the first floor can be found here. The room is very simple and it is furnished with the essentials: coat-rack and typical furniture.

The illumination can come either from the windows located in the wall near the staircase and from the light located on the ceiling.

Like the entrance, the landing is usually a crossing point; it is one of the most used room as a function of its own location.

### Study

This room is located on the first floor, and it is reachable by the landing of the first floor.

Over the desk placed in the room it is possible to find a laptop with an external monitor and a desk light. On the wall in front of the desk, hangs a very large monitor that is connected with the main system of the house, with which it is possible to interact. The same monitor can also be connected with other devices and used for other reasons. The room is then completed by the typical office’s furniture.

The illumination can come either from two windows located on the wall near the desk, but also from the light located in the middle of the room, as well as from the desk light.

One of the residents is a software developer and he works three days per week from home.  
Because of that, the room is mainly used by him, although it can represent a useful workspace for all the residents.

By this room, through the glass door, it is possible to reach the balcony.

### First floor bathroom

The first-floor bathroom is bigger than the ground floor bathroom. It is provided by the typical bathroom’s supplies, but this time there are two washbasins, one for each residents, and it has a whirlpool bath. As for the ground floor bathroom, in this room there is the underfloor heating but also two towel warmers.

Two windows and one glass door provide the illumination. One main light is located in the middle of the room. Each washbasin has its own mirror and its own light. Moreover, there is another light on the ceiling over the whirlpool bath. The whirlpool bath also has a colorful light that can illuminate the water.

The first-floor bathroom is a frequently used room; only the tenants and no one else usually use it.

By this room, through the glass door, it is possible to reach the balcony.

### Laundry

This room is located on the first floor, and it can be reached by the landing of the first floor.

Here there is the typical furniture of a laundry, some tools (iron, laundry basket, clothes peg, etc.) and devices. The main devices are the washing machine and the tumble dryer. Since they are large household appliances, they are always plugged. In this room, it is possible to find the main station of the vacuum cleaner that is responsible for the cleaning of the entire first floor.

Illumination is provided by one glass door that allows the sunrays to penetrate during the day, and one main light located in the middle of the room.

This is one of the lesser used rooms. Usually, residents go there if they have to take care of the status of the vacuum cleaner or to clean their clothes.

By this room, through the glass door, it is possible to reach the balcony.

### Bedroom

The bedroom is located on the first floor and it is reachable by the landing of the first floor.

With the living room and the kitchen this is one of the largest rooms of the house. The room is equipped with the typical bedroom’s furniture (wardrobe, bed, bedside tables, etc.). At the center of the room there is a little table and in front of the door, on the other side of the room, there is a small bedroom desk. On the right side of the wardrobe, in the corner of the room, is located a bedroom mirror.

The illumination is provided by two windows, located perpendicular to the bed. There is then a main illumination in the middle of the room. In addition, each bedside table has its own bedside lamp. Also the wardrobe has a light, but it is inside and in order to use that is it necessary to open at least one closet door.

Usually, this room is used by the tenants only, to sleep and to change their clothes.

# Devices

This section presents the devices available in the house. The following devices are intended as smart devices instead of agents because they are mainly performers and not decision-makers. Most of them collect information and share them with the agents present in the house or the resident (e.g. Sensor). Sometimes they can also perform some kind of computation on the data before sharing them.  
Other devices are instead used by the agents or by the residents to perform activities in order to reach some goals.

Most of the devices can be put into operation in a traditional way (for example a light can be turned on/off by using the switch on the wall), or by the house agent (for example by asking the agent to turn on/off the light or by using the tablet’s application). When you want to interact with the devices through an agent, it is mandatory that the device is plugged and connected with the agent. This last aspect is common to all the following devices unless otherwise stated.

## Light

Lights are the most widespread devices around the house. They make use of the Led technology, that allows to have great performances with low costs. Lights provide illuminations to the rooms when necessary, for example at night, which we consider always from 21.30 to 7.00 in the morning. Each light consumes in average 5 W of electricity when turned on.

|  |
| --- |
| **Status** |
| **isLightOn()** – returns true if the light is turned on, false otherwise |

|  |
| --- |
| **Actions** |
| **switchLightOn()** – turns light on  **switchLightOff()** – turns light on | |

|  |
| --- |
| **Preconditions** |
| **switchLightOn()** – light have to be turned off  **switchLightOff()** – light have to be turned on |

## Solar panel

Solar panels produce in average 2 kW per day. Each panel produces in average 200 W from 7.00 to 19.00, when it is sunny. In all the other cases, they don’t produce anything.

Solar panel is a passive device; it is not possible to make any kind of action on it.

|  |
| --- |
| **Status** |
| **getProducedWatt()** – returns how many watts have been produced |

|  |
| --- |
| **Preconditions** |
| There must be light |

## Vacuum cleaner

Vacuum cleaner consumes on average 0.85 kWh and it works on average 1.30 hours every day.  
House agent identifies the dirty rooms and saves the information inside the device, then it informs the vacuum cleaner agent that there are some dirty rooms. In this context the device is used as the one who executes the actions in the real world, but also as a tool used by the house agent to communicate with the vacuum cleaner agent.

|  |
| --- |
| **Status** |
| **getLocation()** – returns to the room in which the device is located  **updateGoal()** – update the list of the dirty rooms and the final location in which the device has to be led  **resetGoal()** – cancel the goal  **hasGoal()** – %  **getGoal()** – returns the goal  **hasGoal()** – returns if a goal has been defined  **isCleaning()** – returns if the device is cleaning |

|  |
| --- |
| **Actions** |
| **startClean()** – the device starts to clean  **stopClean()** – the device stops cleaning |

|  |
| --- |
| **Preconditions** |
| **startClean()** – the device should not be cleaning  **stopClean()** – the device should be cleaning |

## Roll-up shutter

The roll-up shutter consumes 290 W if completely opened when it is completely closed and vice versa.

|  |
| --- |
| **Status** |
| **isOpen()** – returns true if the roll-up shutter is opened, false otherwise |

|  |
| --- |
| **Actions** |
| **open()** – open completely the roll-up shutter  **close()** – close completely, if it isn’t already completely closed, regardless the actual openness level |

|  |
| --- |
| **Preconditions** |
| **open()** – roll-up have to be closed, there have to be nightly time slot  **close()** – roll-up shutter have to be opened |

## Thermostat

Usually, temperature is completely managed by the house agent. In large, if there aren’t any anomalies, the residents don’t interact with the device.

|  |
| --- |
| **Status** |
| **getTemperature()** – returns the current temperature of the room  **isCCSystemOn()** – returns true if the climate control system is turned on, false otherwise  **getMinTemperature()** – returns the minimum acceptable temperature  **getMaxTemperature()** – returns the maximum acceptable temperature |

|  |
| --- |
| **Actions** |
| **setTemperature()** – set the desired temperature for the room  **switchCCSystemOn()** – turns on the climate control system  **switchCCSystemOff()** – turns off the climate control system  **updateTemperature()** – simulate the variation of temperature over time |

|  |
| --- |
| **Preconditions** |
| **switchCCSystemOn()** – the climate control system should be turned off  **switchCCSystemOff()** - the climate control system should be turned on |

## Sensor

Sensor is a passive device; neither it is possible to make any kinds of actions on it, nor there are special precondition for its operation.

# Metrics

## Cost of electricity

Buying electricity is more expensive during the day. For that reason, when possible the energy produced by the solar panel is used.

|  |  |  |
| --- | --- | --- |
| **Segment** | **Hours** | **Price** |
| F1 | 08.00 – 19.00 | 0.31612 €/kWh |
| F2 – F3 | 19.00 – 08.00 | 0.2519 €/kWh |

Produced electricity is sold at 0.15 €/Wh. The overproduced electricity is stored into batteries, when these last are full it is sold.

## Cleaning time

|  |  |
| --- | --- |
| **Room** | **Time (hours)** |
| Entrance | 0.25 |
| Kitchen | 0.45 |
| Boiler room/Panty | 0.15 |
| Living room | 0.35 |
| Ground floor bathroom | 0.20 |
| Garage | 0.25 |
| **Ground floor** | **2.45** |
| Landing of the first floor | 0.20 |
| Study | 0.25 |
| First floor bathroom | 0.30 |
| Laundry | 0.20 |
| Bedroom | 0.35 |
| **First floor** | **2.10** |

# People and agents

In this section intelligent and autonomous entities in the house are introduced.

## People

The house is inhabited by two residents: Adam and Ashley.

Adam is a software developer. In general, he spends three days per week working from home, while the other two he works in the office. He wakes up at 6.45 in the morning and starts to work at 7.30 am. Lunch break is from 01.00 to 01.30 pm and then he works until 4.30 in the afternoon. It may happen that he is away for two or three weeks because of the work. When he goes to the office, he uses the car, in all the other cases the car remains available to Ashley. He does sport three times per week: Monday, Wednesday and Thursday, from 7.00 pm to 8.30 pm.

Ashley is a doctor and she works is the city’s hospital. In general, she wakes up at 06.45 am and works from 07.45 am to 05.00 pm with one hour of lunch break. Sometimes she has a night shift, when it happens she spends that day resting and the same for the morning after. She does sport three times per week, but she frequents the gym and she decides week by week when going to train.

Usually, when it is possible, Adam and Ashley have a dinner together. They order pizza once a week, typically on Friday evening. On Saturday, they like to go around doing their hobbies, and the day finishes going out to dinner. They spend at least one weekend per month away from home, having a cultural trip or an excursion. On Sunday, they are used to spend the day with their families or their friends or simply by resting at home.

The main effort was primarily aimed to develop the agents and the actions that they can perform during the daily simulation. Because of that, the action performed by the people into the house is limited to their movement around the house itself.

## Agents

### House agent

The House Agent (HG) receives and computes all the information received by the different sensors present in the house (light sensors, motion sensors, cameras, thermostats, etc.). On the basis of the received data, it decides what action to take and communicates that decision to the selected performer device, or devices.

HG assists the residents in their daily life, by managing most of the processes that should be otherwise handled by them. The tenants can interact with HG simply by saying “Hey Sophie”, or “Hi Sophie”, (the name of HG) followed by the request, by using the tablets present into the house, by using their smart phone, in which is installed the same application that in installed into the tablets, or even by using the computer present into the study. Every activity is done trying to minimize consumption and cost.

Some examples of activities that can be handled by HG are:

* turning on/off the climate control system;
* opening/closing the roll-up shutter;
* redirecting the produced electricity to the battery or selling it;
* remembering something for the resident, for example if they have an appointment, or if the dishwasher has ended its work;
* taking notes for the resident, for example the shopping list;
* starting processes that involve other devices. Some processes are decided from the resident and HG is only an actuator, for example starting the washing machine;
* communicating with the other agents in the house;
* turning on/off the light.

The list could of course go on forever.

Here below a more specific description of some activities is stated.

HG regulates the house temperature based on the period of the year, the outside temperature and the habits of the resident. Although the residents can change the temperature in any moment, with the experience HG is able to satisfy the needs of the residents, which usually don’t intervene.

By recognizing the presence of the residents, HG turns on/off the light in the rooms following their movement. If one of them is between two rooms, only the nearest lights are switched on. If one resident wants to look inside a room without entering in it, he can just ask HG to switch on the light.

In addition, HG also works as anti-theft, by analyzing all the information received by the other devices. In particular, this activity is done day and night, both if the residents are at home or not. HG decides autonomously if it has to communicate the presence of possible thieves to the residents only or also to the police station.

Although there is only one HG, it is able to have different conversation in different rooms at the same time. For example, if Adam is working in the study an Ashley is in the living room, they have different conversation with HG in the same moment. It might seem that there are four different subjects, two for every conversation. On the contrary, if the residents are both in the kitchen and they want to talk with HG, this conversation involves three subjects only.

Specifically, in the implemented scenarios the agent takes care of the following activities:

* **Light**, by using the information received by the movement sensors and the light sensors, HG decides if it is necessary to turn on/off the lights around the house.
* **Alarm clock**, HG turns the alarm on every morning and sets the new alarm for the next morning following the habits of the tenants.
* **Temperature**, through the information shared by the thermostats HG decides when to turn on/off the climate control system.
* **Cleaning**, every time at 10.30, HG checks for the cleanliness status of every room of the house. If at least one dirty room is found, HG informs the vacuum cleaner responsible for the house level in which the dirty room is located.
* **Consumption**, at the end of the day HG computes the total electricity consumption in kWh for that day.

### Vacuum cleaner agent

The vacuum cleaner agent (VCG) interacts with the vacuum cleaner device, which is able to move around the house autonomously when a specific cleaning task is assigned. The beliefs received by VCG are filtered in order to remove all those information that are not relevant. Because of that, the procedure of cleaning the house starts with the house’s cleanliness check action, performed by HG. Only then, VCG receives the information that the house must be cleaned. As said before, VCG will then find the list of the dirty rooms stored into the vacuum cleaner device.

The weekly schedule could be different from the weekend schedule, that’s because of the habits of the tenants. For example, sometimes it happens that the residents spend the Saturday night sleeping on the bed-sofa. HG will inform VC about the presence of the tenant and VCG will then postpone the cleaning for that room.

VC reacts to the information received by HG but also to what it perceives. For example if it gets in touch with something that is unknown, it reports that to HG and asks him more details in order to understand what could be the best action to be taken.

Specifically, in the implemented scenarios the agent takes care of the following activity:

* **Cleaning**, after having received the information that the house has to be cleaned, VCG puts in place everything necessary to clean the dirty rooms.

As mentioned, the house is equipped with two VC, one for the ground floor and another for the first floor, the above description is in common to both the vacuum cleaner agents.

# Planning agent

The planning agent is an agent (or a part of that), which, given some information, formulates a plan to achieve the given goal. These information are mandatory for the formulation of the plan, and are usually grouped in two main files named domain file and problem file.

* **Problem** contains a restricted description of the world. Restricted because only the information relevant for the formulation of the plan is stored here.  
  Inside Problem three main sections can be found:
  + **objects**, here it is possible to find the list of the subjects involved in the planning process.
  + **init**, the status of the world is listed here. With status a broad spectrum of meanings is intended, like for example the individual characteristics of the objects, or the relation between some of them.
  + **goal**, the status of the world that is to be achieved is defined here.
* **Domain** contains all the elements that can be used to reach the goal.  
  Inside Domain two main sections can be found:
  + **predicates** represent properties of the object we are interested in. Predicates are used to check if those properties are true or false.
  + **actions** are those activities that can be used to change the status of the world.  
    Every action is composed of a list of parameters, necessary to perform the activity, a list of preconditions, that must be respected to put the activity in place, and finally a list of effects, that are the consequences of the application of the activity.

## Vacuum Cleaner planning agent

The vacuum cleaner planning agent (VCPG) aims to generate the domain and the problem files.  
In this specific situation, those are then sent to an online planner which, once found a plan, returns it to the VCPG. Once received the plan, VCPG puts it in place.

By following the structure above, it is possible to give a specific example of the domain and problem files generated by VCPG.

**References:**

* **planning** and **pddl** folders contain that part of the framework that is responsible for the definition of those activities that allow the planning
* **VacuumCleanerActions.js** and **HouseActions.js** are those files that contain all the element necessary to generate the **Domain.pddl** and **Problem.pddl** files
* **pddlData** is the folder in which **Domain.pddl** and **Problem.pddl** files are stored

### Problem

#### objects

The vacuum cleaner device has to clean the rooms of the house.  
In this section there is the list of those rooms.

#### init

In this section it is possible to find the types of objects involved and their relationship.

#### goal

Finally, this section reports the list of the goals.

### Domain

#### predicates

The properties that every parameter of the action must have.

#### action

The possible activities that can be performed. Only one action is reported below as example.

Implementation

In the previous part of this reports, the focus was mainly on the description of the idea behind the project and on the elements that compose this last. The given descriptions are in general low in details about how the thing have been developed and implemented, except for Planning agent section.

On the contrary, the intension of this part of the report is instead the one of try to go a little deeper into the description of the structure of the source code and the used strategies. Keeping in mind that this isn’t a punctual documentation of the implemented code, sometime it possible to find references to some pieces of the source code, or some of them can be directly reported into the report.

Finally for those part of the framework on which the project relay on, and that are not directly developed with the remained part of the code (see Framework section), please refer to the official documentation.

# Sensors and agent perception

As already said, the agents perceive the environment through information that can come for the sensors, from the other agents or from the users.

One of the simples sensor that is possible to find is the clock. Although could be weird to think about the clock like a sensor, it is the only tool, or at least the easiest one, that is able to keep track of the passage of the time. Since some activities depends on the time (i.e. the morning alarm clock) the house agent has the capability to observe the progress of the time. For the simulation scope, the time has been reduced to days, hours and minutes, and it advanced by 5 minutes every 50 milliseconds.

Near the alarm clock there are other actions that are related with the progress of the time: the lighting, the opening and closing of the roll-up shelter and the temperature, although in a different way as the others.

Instead of simulate the variation of the illumination, which should be observed by a specific sensor in the real world, into the simulation the house agent again observes the progress of the time. By relaying on a scheduling it decides when it is necessary to turn on/off the light or open/close the roll-up shelter.

The idea is a little bit different for the temperature. Although the variation of this last is simulated every two simulation hours, the variation of the temperature itself is captured by the thermostats present in every room, and notified to the house agent. Notified the variation of the temperature the house agent compares the new temperature with the desired one, and decides if it is necessarily to turn on the climate control system.

Near the thermostat and the clock, it is possible to find another sensor, the movement sensor.  
In this case it doesn’t present a real class in the code that represent the sensor device, but instead when during the simulation the tenants are moved around the house, the movement itself is notified to the house agent.

The majority of the devices that are present into the house, but also the house and its parts themselves, have a specific class that define their structure. The main strategy to keep track of the described variation is done by letting the house agent to be notified when specific variables change their value.

In general the agents relies on element that are called beliefs. Those beliefs are the representation of specific status of the environment that are known by the agents, and that are stored in their memory. Beliefs are simple elements in the sense that they can be only true or false. Into the project, they are represented by strings, on which the agents can perform some actions:

* define them as true or false
* check if they are true or false

The majority of the action that can be performed by the agents themselves, or the devices or the tenants contribute to the creation/modification of the agents’ beliefs.

**References:**

For those actions that depends on the progress of the time, the observed variable is the “mm” which keep track of the progress of the minutes. The progress of the time is observed in HouseManager.js.

* House agent is defined in **HouseAgent.js**
* Clock is defined in **Clock.js**
* Thermostat is defined in **Thermostat.js**  
  The observed variable is “**temperature**”  
  The variation of the temperature is observed in **ThermostatManager.js**
* Light is defined in **Light.js**  
  The action that can be performed by the house agent are in **LightManager.js**
* Person is defined in **Person.js**  
  The observed variable is “**in\_room**”  
  With respect to the movement of the resident, the actions that change the beliefs of the house agent are in **MovementSensor.js**

# Agents acting in a shared environment

As described in the previous paragraph, the agents can observe specific variables of the devices or the representation of the users. Their beliefs can change as a consequence of actions performed by the agent itself but also by the other actors of the environment.

In order to better explain that, it is possible to look at the example of the lights.

By observing the progress of time, the house agent can identify the moment of the day in which it is necessary to turn the lights on into the house. As already said, this is an adopted solution that simplifies the scenario. For the simulation, two time slots have been identified: from 07.00 to 21.30, in which the lights are not necessary, and from 21.30 to 07.00, in which the lights are necessary. The observation of the progress of time helps the house agent to understand in which time slot it is.

Starting from 21.30, the house agent turns on the lights of the room where the tenants are and will then turn on and off the lights according to the movement of the tenants into the house. However, during the night time, if the tenants are sleeping, the house agent should not turn on the lights and it should instead take care of turning off all the lights in the house, in order to avoid consumes. In order to avoid turning on the light during the night if the resident are sleeping, the belief of the agent that identifies if the tenants are awake is set to false. By changing that belief the house agent turns off all the lights and will avoid to turn them on until its beliefs are updated with new knowledge.

When the next morning comes and the alarm clock rings, the update of the beliefs will contribute to let the house agent perform new actions, like opening the roll-up shelters or turning on the light. That will change the beliefs of the house agent and so on and so forth.

# Agents interaction and coordination

In the context of the simulated scenario, the interaction between the agents that is noteworthy, in the one that involves the house agent and the vacuum cleaner agents. This interaction take place with the aim of clean the house when it is necessary. The peculiarity of this interaction is that it is done by using the vacuum cleaner devices as an intermediary.

Keeping in mind that the ground floor vacuum cleaner device can work only on the ground floor, and vice versa for the first floor vacuum cleaner device, below it is referred as a generic vacuum cleaner agent and device.

At 10.30 in the morning it is simulated random dirtying of some rooms into the house, in both the house level. The house agent keep track of that, and create a list containing the names of the rooms that have to be cleaned. At the same time, the beliefs of the house agent change, removing the beliefs about the cleanliness of those rooms that are not more clean and adding the beliefs about the dirtiness of those rooms. The change of the house agent beliefs is notified to the vacuum cleaner device, which update its own beliefs.

Keep in mind that not all the changes in the house agent beliefs affect the change of the beliefs of the vacuum cleaner agent. All the notification about the variations in the house agent beliefs are filtered in order to identify those beliefs that are relevant for the vacuum cleaner.

Once the list of the dirty rooms has been draw up, it is stored into the vacuum cleaner device.  
A specific variable of the vacuum cleaner device is then set to true and the house agent has done its work. The above variable is observed by the vacuum cleaner agent, that when it is notified that the variable has changed its value, proceed to understand if it is necessary to clean the rooms or not.

Other details on what happens at this point have been already described into the Vacuum **cleaner agent** and **Planning agent** sections.

**References:**

For those actions that depends on the progress of the time, the observed variable is the “mm” which keep track of the progress of the minutes. The progress of the time is observed in HouseManager.js.

* House agent is defined in **HouseAgent.js**
* Clock is defined in **Clock.js**.
* The check of the cleanliness of the rooms in done in **CleanerManager.js**
* Vacuum cleaner agent is defined in **VacuumCleanerAgent.js**
* Vacuum cleaner device is defined in **VacuumCleaner.js**
* The observed variable is “**clean**”
* The variable is observed in **VacuumCleanerManager.js**
* Thevacuum cleaner beliefs filter is defined in **VacuumCleanerSensor.js**

# Scenarios

The implemented scenario described a standard week for a couple that lives in the smart house.

For reasons of space, here below is shown a compact simulation of the scenario, representing only the first day of the week, Monday.

The day starts at midnight, but only at 6.30 in the morning the tenants wake up and the life in the house starts.

The simulation updates the temperature of the rooms every 3 simulation hours, and this activity is the only one that is perpetrated for the whole day, regardless all the other factors.

From when the residents wake up, the house agent follows they movement and takes care to turn on/off the lights. At 7.00 in the morning the lights are not more necessarily and from that moment the house agent avoids to turn on them.

From 7.00 to about 17.00 tenants spend their time working, with a break for the lunch around 13.00.

At 10.30 the house agent check the cleanliness of the rooms and share the results with the vacuum cleaner agents. Those last takes care to clean the identified rooms, if there are any.

After work the tenants gather in the living room and spend the rest of the day between that room and the kitchen. At 22.30 the resident prepare themselves for the night and at 23.30 they are going to sleep.

At 21.30 are necessary one again. The house agent turns on the lights for those rooms in which there are people inside.

At 23.55 the house agent computes the total energy consumption for the day.

## Running the scenarios - Logs

[Compact and/or extended version of logs from running the scenarios]

## Additional scenario

…

# Source code organization

Inside the **src** folder is organized all the source code that compose the project.  
In turn the **src** folder is then organized into the following sub folder.

* bdi
* pddl
* planning
* utils
* myWorld

## Framework

The code that compose the framework used to develop the project is contained into four of the previous five folders:

* bdi
* pddl
* planning
* utils

For the content of those, please refer to the official documentation.

## myWorld

The folder that gives the name to the title of this section contains all the code developed in order to realize the project as it is and bring to life the simulated scenario.

The content of this folder is the subdivided into four subfolder:

* Agents
* Classes
* Goals\_Intentions
* Utilities

and one file **Scenario.js**. This last contains part of the initialization of the agents as well as the scheduling of the activities performed by the resident during the week. The timer, is started and ended from here.

The rest of the section continues with a brief description of each file contained into the above subfolders. Brief because, as already said in the introduction, the scope is to present the structure of the project and not give a punctual documentation about the code.

## Agents

It contains two subfolders:

* House:
  + **HouseActions.js** – it contains the class that execute generic plan. Near that there are the definition of the action necessary to perform the cleaning plan.
  + **HouseAgent.js** – it contains the definition of the house agent with all the elements that it needs in order to reach the goal. The instantiation of the house agent is also in this file.
* VacuumCleaner, which contains
  + **VacuumCleaner Actions.js** – it contains the definition of the action that the vacuum cleaner has to perform in order to perform the cleaning plan.
  + **VacuumCleanerAgent.js** - it contains the definition of the vacuum cleaner agent with all the elements that it needs in order to reach the goal. The instantiation of the vacuum cleaner agents is also in this file.
  + **VacuumCleanerSensor.js** – the beliefs notified by the house agent are filtered in order to select those who are relevant for the vacuum cleaner agent. The sensor is contained in this file.

## Classes

It contains two subfolders:

* Devices:
  + **Device.js** – it contains the definition of a generic device, with some default characteristics. It is never used directly, but instead it is extended from the other classes contained in this folder.
  + **Light.js** – it contains the definition of the light device class.
  + **Thermostat.js** - it contains the definition of the thermostat device class.
  + **VacuumCleaner.js** – it contains the definition of the vacuum cleaner device class.
* house\_config – the project is initialized by using the json files containing all the information about the structure of the house, the presence of the device and the tenants into the house. The folder contains the following files:
  + **Devices.json**
  + **People.json**
  + **Rooms.json**
* **House.js** - it contains the definition of the house class. Here is where the json files are read and the house, the rooms, the devices and the people are created and initialized.  
  The instantiation of the house is also in this file.
* **Person.js** - it contains the definition of the person class.
* **Room.js** - it contains the definition of the room class. The devices are closely related with the rooms, every room has its own dictionary containing the list of the devices present in it.

## Goals\_intentions

* **CleanerManager.js** – it contains the intention and the goal to check the cleanliness of the house.
* **ConsumptionManager.js** - it contains the intention and the goal to compute the total energy consumption of the house and its devices.
* **HouseManager.js** - it contains the intention and the goal to execute specific action during the course of the day.
* **LightManager.js** - it contains the intentions and the goals of turn on/off the lights when specific situations occur.
* **MovementSensor.js** - it contains the intention and the goal to notify to the house agent, the movement of the resident into the house.
* **Retry.js** – if for some reason the cleaning goal cannot be achieved it is rescheduled for at most four times.
* **ThermostatManager.js** - it contains the intention and the goal to notify to the house agent, the variation of the temperature inside the rooms of the house.
* **VacuumCleanerManager.js** - it contains the intention and the goal to inform the vacuum cleaner agents that the house has to be clean.
* **WakeUpManager.js** - it contains the intention and the goal to turn on the alarm clock, set one for the next day and make other morning routines.

## Utilities

* **Calendar.js** - it contains the mapping from the number of the day of the week, or the number of the month of the year, to its own name.
* **GlobalUtilities.js** - it contains some static method, organized into a class, that can be useful all around the code.
* **Logger.js** – it contain a logger class that create a file log containing all the information showed in the terminal window, for every run scenario.

# GitHub repository link

<https://github.com/andy295/Autonomous_Software_Agent_Project>