# Energy Efficiency Optimization Through Occupancy Detection and User Preferences

Vitor Oliveira Mansur
vitor.mansur@tecnico.ulisboa.pt

Instituto Superior Técnico

## ABSTRACT

Building Automation Systems aims to provide energy control and comfort management by controlling the HVAC systems. However, these systems use pre-set configurations which usually do not correspond to occupants preferences. Moreover, there is no way for occupants to specify their preferences to HVAC system. Indeed, existing systems take into account the number of occupants and the energy consumption, disregarding individual occupant preferences.

A system that can track the occupants preferences, and manage automatically the ventilation and heating levels accordingly to their preferences, should prove to be an innovation in the management of HVAC systems, allowing it to pool its resources to saving energy while maintaining user comfort levels.

This article studies the problem of automatically finding the appropriate set-point of HVAC system taking into account user preferences. A prototype solution implementation is described and validated The results were evaluated by simulation using occupant's votes. Therefore, it is expected to improve the occupant's experience and reduce the energy consumption by applying this concept into real environments. Our findings indicate that one of the algorithms is able to successfully maintain the appropriate comfort levels while also reducing energy consumption by comparing with a standard scenario.

## Keywords

Energy Savings, Users Preferences, Occupancy Detection, HVAC System, Thermal Comfort.

## 1. INTRODUCTION

Heating, Ventilation and Air Conditioning (HVAC) systems account for up to 60% of the total energy consumption of commercial buildings [14]. A Building Automation System (BAS) is used to save energy while maintaining user comfort [12]. Since occupants cannot control the HVAC thermal settings, they may experience discomfort while the system is probably cooling or heating above what is required and wasting energy. Presumably if occupants were given the opportunity to control the system, situations of excessive service delivery, that spend more energy, would not be so frequent.

A reasonable climate control solution employing intelligent systems is potentially able to minimize the energy consumption. This technique can be used to promote energy savings while maintaining the occupants comfort. However, it is important to note that the occupants opinion regarding comfort or discomfort, is typically not considered when controlling traditionally systems.

Studies show that the use of occupancy detection techniques with light and systems is beneficial to energy optimization, reaching savings values from 10% up to 50% [2, 7, 10]. However occupancy detection does not always have an acceptable accuracy rate. A promising approach is to use RFID systems to detect occupants with reliable accuracy, and retrieve occupants information such as their comfort preferences. This can lead to a higher occupancy detection accuracy and energy management control.

Energy management is concerned with finding sources of waste and act in order to improve the energy savings — a major problem on commercial buildings. Energy typically targets the system in order to reduce energy waste. However if the occupant is not in the loop, these improvements are under-optimal. Maximizing occupant comfort is a difficult task because preferences an be contradictory. Most existing systems only take into account a fixed mean temperature aiming at a comfortable ambient to the majority of occupants.

Even if the possibility to change system settings is given to occupants, probably it still would not be possible to reach a consensual setpoint value that would make every single occupant feel comfortable. In a system in which all occupants could give their opinion about the climate comfort, it may be possible to compute a value of these opinions that minimizes the overall discomfort and configure the system for it.

### 1.1 Motivation and objectives

This work develops a system which enables occupants of a given room to give their opinion on the configuration of the system. It calculates a setpoint temperature which best fits their suggestions and find the most appropriate actuation to minimize energy consumption while, at the same time maximize comfort.

The proposed system incorporates dynamic user feedback control loops in order to optimize the energy/comfort tradeoff. Furthermore, it provides a cooperative mechanism for users to agree on levels of comfort that decrease energy consumption. This system could expanded into a testing platform for validating different scenarios and algorithms related to the comfort and energy use.

## 1.2 Problem Statement

Energy Management Systems are mainly concerned on the energy consumption. Since occupant comfort can be contradictory, these systems typically have trouble on defining a optimal occupant comfort rate while minimising the energy consumption. There are system which aims into solving this problem, however, there is still a lack of solutions based on occupant preferences. To date, there is no solution platform **Maximizatizes Comfort** by collecting information on the occupant's preferences and give a setpoint which will control the temperature based on their feedback. **Encourage Occupants to save energy** by allowing them to see information on the other occupant's votes and actual inside and outside temperature of the room. **Store occupant's preferences** in order to make decisions in the future, it is first necessary to have information about the occupant's behaviour. **Make automatic Decisions** by using data on occupant's preferences which will enable an automatic configuration based on the data retrieved.

# 2. SOLUTION PROPOSAL

The proposed solution consists on the creation of a system which loads user preferences ubiquitously, and automatically sets the best comfort settings while trying to coerce them into saving energy, by providing users with feedback. This system will also give occupants the possibility to contribute with their input for changing the settings at any time by means of a computer or smartphone.

## 2.1 Overview

Since the users should be able to access the system in any device, and at any time, it was necessary to devise a solution whose architecture was independent from the programming language, which should also be available whenever the user wants. In order to make it available for users, it is appropriate to use a web server which accepts requests from users and is available on the network, centring all the relevant and complex processes into it. It was necessary to develop a simple system which could detect the occupants, and read the student's card. An solution using Arduino and a Near Field Communication (NFC) module could retrieve the users informations, read the student's card, communicate on the network with a web server and could also be utile into other situations.

## 2.2 Backend

An backend application which can abstract the access to the database, could possibilitate the development of a SOA [5] where any desired client should be able to request information on the database. Clients are able to make these requests using HTTP[1].

Several advantages could be attained by using this application, among them: The creation of an abstraction layers which simplifies the users interactions with the Database (DB), the client application is independent on the backbackend, this allows for a client application to be developed in other programming language. Only clients applications can start interactions by making requests, the application will be simpler since it only has to provide answers to requests.

In order to make a user-friendly Service Oriented Architecture (SOA) where an abstraction of the system's layers would be created, the user interface is developed using the Model View Controller (MVC) software architectural pattern, where the application is divided into three interconnected main parts, as the name suggests: Models, Views and Controllers. The Models are the representation of all the data on the system, i.e. the DB, the Views are those information displayed for the users, such as the layout, Hypertext Transfer Protocol (HTTP) and JavaScript code, and the Controllers are used to manipulate the Models [1].

## 2.3 Database

In order to store, manipulate and use information about the users, a system which could allow for this processes was needed. A DB System is an appropriate solution which will store and manage the data as intended by the solution main system.

The Data Base server is located in the same Local Area Network (LAN) as the Web Server and counts with the PostgreSQL DB System,[2] this system was chosen because the Middleware which controls the System is also using the PostgreSQL. Since the main system's database is in the same Data Base Server of the middleware's, it was decided then to use the PostgreSQL in order to avoid possible incompatibilities with the DB language. This results on the homogeneity of the Data Bases avoiding the installation of new databases in the same server, which may cause conflicts.

## 2.4 RFID System

It was intended to use a occupancy detection system which has a low cost, which has a user-friendly interface and it would also be interesting if the system could also read the student card. The RFID system is able to fulfil these problems and has many other advantages, among them: There is no need to users to buy tags, the student card will do. The student card does not need a battery, so it don't have any restriction to use it. There is no need to train the occupants to use the system, since the system is very user friendly. It can detect the user with a easy way which the students wouldn't bother to use. The RFID could be used on other situations, e.g.: replace the presence list.

Among many other systems, the Arduino could use a NFC and a Ethernet module to read the students card and communicate with the Web Application. The advantage to use the arduino is that it is possible to develop a lot of different systems using other shields[3]. This system could be improved in the future by using other types of sensors.

Since the room where it is intended to install the occupancy detection system has two doors, there were used two Arduinos. Each Arduino system has three shields: the first one is a ethernet shield, responsible to connect the Arduino into the LAN through a wired router. The second shield is the NFC antenna responsible to read information of the

[1] http://www.w3.org/Protocols/

[2] http://www.postgresql.org/docs/9.3/interactive/protocol.html

[3] http://www.arduino.cc/

RFID tags and the student's card, and the last shield is an LCD screen which is used to give a feedback for users.

Since there are two Arduino systems, there is the need for another system to aggregate the information of these systems and send it to the Web Application. To solve this problem, a Raspberry Pi, which is also connected into the same LAN, was used.

The Arduinos and the Raspberry has the role to transfer the tag information to the Web Server and give the server's response feedback to user on the Liquid Crystal Display (LCD) screen. In order to have their tags read, the users need to put their student card near the Arduino, since the read range of this system is only 6 centimetres.

# 3. DEVELOPMENT

Since it is not necessary a very powerful machine to run the application, to run the Web Server and the DB system it is necessary a server computer which is able to run the Java Virtual Machine and The PostgreSQL.

The Arduino programming language is an implementation of Wiring, which is very similar to the C programming language. The NFC module sends the information through the transmission jumper, the Arduino then resends it to the Ethernet shield, which encapsulates on a UDP packet and send to the Raspberry Pi. The UDP method is used since the LAN which connects all nodes, its a cabbed network, and the probability of loosing a packet due to collisions is very low.

## 3.1 Development Environment

The Grails[4] framework is used over the Groovy language which is dynamically compiled to JVM bytecode, and interoperates with other Java code and libraries [4]. It uses the Spring MVC under de hood [3], making it a faster and easy development of web application over the Java Language. The Groovy language also interprets the Pure Java Language.

## 3.2 DataBase

As mentioned on section 2.3, the project will collect large amount of data that will need to be retrieved to the learning algorithm in the future, this way the Data Base system is located on the same server as the Web Application, this improves the performance speed on retrieving and storing data. It stores information about: Users, Roles, HVAC variables, User's Inputs and the RFID tags.

## 3.3 Web Services

In order to develop a system which could integrate with other systems in such way that they should be independent from the programmed language the system needed to have a Web Service which as mentioned on section 2.2, will only take requests as a service.

The Web Service is invisible to users, its main function is to provide intercommunication among all applications and users such as the RFID system, HVAC services and all other third party software. It takes responsibility of Views of the Web Server. It is responsible for all the interactions among the systems which are also part of the solution.

The functionalities provided, can be accessed by either using a web browser, accessing common HTML pages, that can be accessed through any device with a installed web browser, or through other applications, or smartphones apps.**The Views** are responsible for showing the state of the system. Such as the ambient variables, number of people which voted and give means to user's to vote. The views are the main interaction that the users have with the system. **The Interactions:** The Web Service takes place between all the communication among the external system's and the Core Layer of the Web Server. All the data flow between the layers, first needs to be processed by the Web Service, and only after, the data is submitted to the core layer. The Web Service retrieves the relevant information that comes from these requests, and send the information to the Application Core, which will process the information and return the answer to the Web Service.

## 3.4 Application Core

The Application Core acts as the brain of the system. It is responsible for computing all the complex algorithms including the simulation and the learning algorithm. Some of the main functions are: **User vote and feedback:** Accept user's votes and send them the ambient variables. **Occupancy detection:** The task of knowing and deciding if a user is on the room or not, based on the information that is received from the RFID System. **Set-point calculation:** Collecting the occupant's votes, validate them and calculate a new set-point which minimize discomfort. **Learning algorithm:** Learn information about occupant's behaviour and predict a best decision to make. **Interact with HVAC System:** Send information about the new set-point and receive information about the ambient variables.

## 3.5 Algorithms

The solution contains functions and complex algorithms that need to be discussed. This section will enter in detail all the important algorithms that were implemented on the development of a solution. All the algorithms are processed by the Application Core.

### *3.5.1 Basic Services*

This algorithm can be divided in two steps, the first one is the process of providing to users the ambient variables, and collecting occupant's vote. The second step can be defined on the process that occurs on the Application Core which is the submission of the vote and the management of the queue.

On the first step, the system stands by waiting for a user to login, when it receives a user request, it starts by loading ambient variables from the DB system and returning this information. At this point, users are allowed to vote if they judge necessary. The vote will be sent to the Application Core which will deal with the process of submitting the vote. Figure 1 depicts this process.

There are two ways to submitting a vote. The first one by accessing the application directly through a web browser. The user can browse to the vote section and place a vote. The second way is by applications. Since the Web Server also receive connections from other applications and smartphones apps, it is possible to send information relative to the Web Server sending a JSON object through HTTP request. The Web Server does the same process after receiving either of these requests.

After the user vote, the Core starts by verifying if the user

[4]http://grails.org/

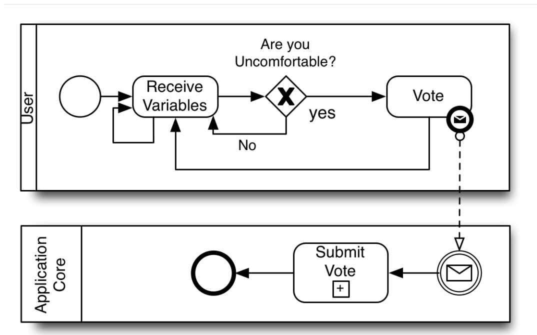


**Figure 1:** Users Interactions with Main Application. The users receive the variables and are able to vote. The Application Core receives the vote, compute it and finishes the process

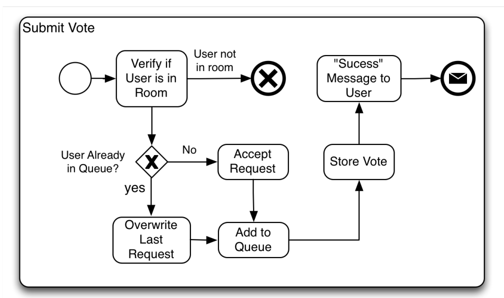


**Figure 2:** Process of storing vote. The system first verifies if the user is on the room, and then verifies if he is already on the queue. Store the vote and respond to the user.

is in the room, if it is not present, the process ends with an error message. Next it verifies if the user has already voted, if yes, it automatically overwrites the last request, adds it to the queue and stores it on the DB with all the relevant information of the status of the room. This process is depicted on figure 2.

The ambient variables which are on the DB are inserted by a routine process which sends the variables to the Application Core and then stores the received information. The set-point calculation algorithm is described on section 3.5.2.

### *3.5.2 Set-point Calculation*

The set-point calculation algorithm is responsible for gathering the occupants votes and calculate a set-point which tries to minimize the probability of votes occurrence, thus maximizing overall comfort on the room.

On a specific interval, the system verifies the queue, if it is not empty, it calculates the new set-point, cleans the queued requests and send the new set-point to the HVAC system. After this process, returns to the state where it waits again for recalculating the new set-point. Figure 3 illustrates the basic process of the requests.

This algorithm uses a equation to validate the proof of concept. If the system calculates the set-point based always on the majority of votes, it is natural that eventually it will reach a medium value where the majority of occupant's will feel comfortable.

It is assumed that a vote is a proof that a person is un-

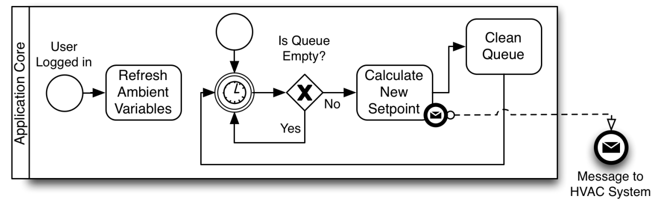


**Figure 3:** Main functions of the Application Core. Refresh ambient variables, calculates the setpoint periodically and cleans queue.

satisfied with the current temperature and wants to change it, then he can vote to increase or decrease the temperature. The vote calculation starts on the second part of the set-point calculation algorithm. Starts by collecting all the queued votes and calculating the number of people in the room. Then it verifies the percentage of votes relatively to the number of people in the room, if more than 10% of the present users has voted, it means that less than 90% of occupants are unsatisfied, then the system starts by calculating the percentages of votes. The difference between the votes that want to increase and decrease the temperature should be greater than 10%, if its not, then it only cleans the queue and do not change the set-point. The calculation decides if the set-point should increase or decrease the temperature considering the majority of the votes changing it from 1°C.

### *3.5.3 RFID System*

To detect the user presence, the data base has an information field which indicates if the user is or not present. When the RFID system detects a tag, the information reaches the Web Server adding a flag on the information field which means that the user is present.

The RFID System interacts with the Web Service as soon as a tag is read. Figure 4 depicts the flow of interactions that are triggered when a tag is read. The RFID System sends the information to the Web Service which will verify if the tag exists. If it does, the system will set the user present or not present depending on the current status of the tag owner.

The algorithm goal is to control the flow of people in the room, thus recording the historical behaviour of occupants and making possible for other algorithms that needs the number of people in the room as a variable to be calculated. The main algorithm consists in three stages, which are: the first one is to detect the RFID tag (or student card), the second one is to route this information through a Raspberry Pi and add a simple security layer to the information. Finally the last stage consists in the Web Server receive the information, adds a flag on the user database indicating that he is present, save the date time and hour that the user entered the room and return the response to the Arduino's LCD.

The Arduino has a Ethernet Module which communicates with the Raspberry Pi on the LAN, whose is also connected with the Web Server. The algorithm receives the 25 hexadecimal codes which each tag contains, since the Arduino has a very limited amount of RAM memory, the parsing and security functions are made by the Raspberry Pi on the second part of the Arduino. The hexadecimal codes are sent to the Raspberry Pi through a UDP channel then the Arduino waits for an answer, marking an end to the first step.

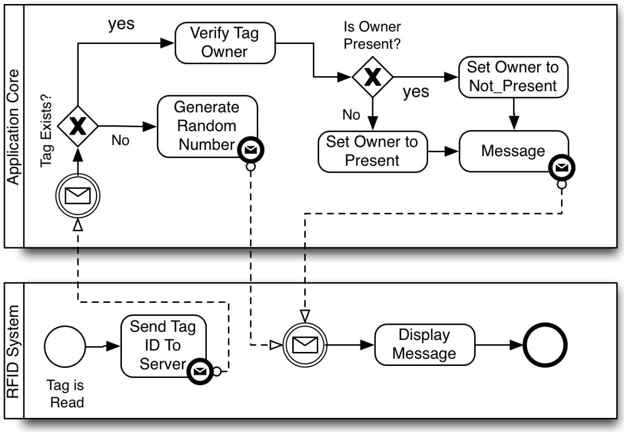


**Figure 4:** RFID System's interaction with the Main Application. It starts by sending the read tag ID to the Core, which verifies if the owner exists and defines it as present or abscent.

The second step starts with the Raspberry receiving the datagram and verifying its size, if the size is less than 25 it discards it, because it means that the datagram came with an error, or the message sent by the Arduino may not be of a tag reading. The parsing phase eliminates the first 17 hex codes and creates a string object containing the 8 important hex codes whose is encapsulated on a JSON object and sent to the Web Server as a HTTP request. Finally the last step of the process is to verify the tag received, compare the string with the one contained on the database.

### *3.5.4 Learning Algorithm*

The idea to make the users configure the HVAC System with votes, can be improved in order to it to automatically configure itself. First it is needed to the system to gather all the votes, identify patterns into it and learn them to take decisions in the future.

A common learning algorithm is able to identify patterns among thousands of input data. In this particular case, the system won't have this amount of data to identify patterns. And the data analysed will probably have three different types of input data: the ones who want to raise the temperature, the ones who doesn't want to change and the ones which wants to lower the temperature. After this analysis, a learning agorithm which could separate these data into clusters accordingly to their similarity is the K-means algorithm [8]. Using this algorithm makes possible the auto configuration of the room's set-point without needing the occupants to vote.

The learning algorithm has as a goal to configure the set-point accordingly to a prediction based on the actual status of the room. The Web Server manages to do that by loading the information about the room and comparing it to a set of previously calculated scenarios. Then it has a decision algorithm that defines which set-point is the most indicated.

This algorithm is intended to provide more comfort to users, since they are detected as inside the room, it should be possible to automatically calculate the most appropriate set-point which will minimize the occupants discomfort. This is possible through a unsupervised learning algorithm which uses the clustering technique.

The k-means algorithm is a method of cluster analysis which divides into $k$ clusters a set of $n$ observations, each cluster $S$ are associated with a centroid $C$. Each observation should belong to the cluster with the nearest mean. Considering a set of data points $S = \{X\}$, letting $d(X, Y)$ be the distance between two vectors $X$ and $Y$.

The k-means algorithm can be divided in two main phases, the first once consists in mapping the clusters and the second one is the computing of the new centroid. First, the algorithm starts by mapping the clusters by choosing $k$ random values and define them as the new centroids, then it iterates and calculate a new centroid [13].

## 3.6 Simulations

For a better understanding of the system's functionality and a pre-validation of the project's objectives, a low cost solution which imitates the user's and temperature behaviour can give preliminary results to ensure that the system works as intended and verify if the results are similar to the ones expected. Among various simulations system, the chosen one is the Energy Plus simulation software. Energy Plus is a simulation software designed to engineers and architect to size HVAC equipments in order to optimize energy performance [9]. This software can be powered with the Ptolemy II framework [11] , it is a open-source framework which allows the use of Java developed actor to interact with the Energy Plus simulation as it goes. Actors are able to change variables on the running simulation, such as number of people in room, set-point and turn on or off the HVAC System. With this powerful framework it is possible to communicate with the Web Server on the simulation run. This section will explain each step of the simulation process among with the results.

### *3.6.1 Overview*

The project's goal is to define a set-point which minimize the occupants discomfort and validate the possibility to save energy using this method. The simulated environment is set within a room of 144 square meters with four windows and one door. The timezone and weather used is on the Lisbon city in Portugal. The month chosen was January, in the winter.

The simulations takes on account that occupants arrive at 8:00h and leave for lunch at 12:00h returning at 13:00h. Then they leave at 18:00h. Each occupant has a unique comfort interval and can be absent. These equations used will be explained furthermore in this section.

The Energy Plus takes the same role as the HVAC System, and the process uses the same principle, first it starts by simulating outdoor and indoor temperatures and adding it to the DB notifying the Application Core that is has been updated. The Application Core will verify the indoor temperature and simulate the discomfort among occupants to retrieve voting results and storing it. The next step is to calculate the new set-point based on the votes, the result will be sent back to the Energy Plus software. At this point, the system will simulate the temperature according to the new set-point received from the Application Core. The overall procedure is depicted on Figure 5.

### *3.6.2 Role of the Application Core*

It was necessary the development of a interface which

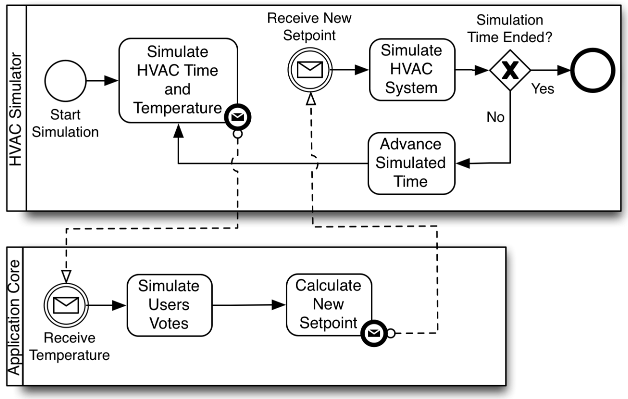


**Figure 5:** Overview of Simulation Process. The simulation framework starts by simulating temperature, sends the data to the Core which will simulate votes and calculate a new setpoint, responding the new value to the HVAC simulation again. This process happens until the simulation time is over.

would make possible the communication of the Application Core and the simulation software. While the Application Core takes on the simulation of users and vote, the other softwares are used only to provide simulated information about the HVAC system and room temperature.

**User comfort range:** Each time the system restarts, it counts with a tool which will create up to 40 unique users. Each user have a comfort interval which means that is the best comfort value that this user will have. Every user will eventually vote to reach the values in their comfort range.

The comfort range of each unique user is defined based on the equality from Humphreys and Nicol [6] defined on equation 1 where $T_c$ is the comfort temperature and $T_{out}$ is defined by the average outdoor temperature from the previous 30 days, in that case 13°C.

$$T_c = 24.2 + 0.43(T_{out} - 22) \times e^{(\frac{T_{out}-22}{24\sqrt{2}})^2} \quad (1)$$

In order to make a unique comfort range to each occupant, the result value is added to the equation 2 where $r_i$ is the result from the equation 1 and $r_f$ the lower value of the comfort range, where it subtracts 2°C and adds a random number which goes through 0 and 4. The comfort range should be from 1°C, then is just add 1 to $r_f$ in order to retrieve the upper limit of the comfort range.

$$r_f = (r_i - 2) + rand(0..4) \quad (2)$$

**Day simulation:** In order to simulate days, the application provides a simple mechanism which introduce a date on the end of each day simulation. It will not only update the days, but some other important functions as removing all the occupants, and calculating a probability of the users to be present on the next day. Using this system, it is possible to simulate a entire month.

**User presence simulation:** To make the simulations with a reasonable resemblance to the reality, the number of occupants in a room are are also simulated. Each time that the system resets the days, users are inserted in the room, but each user has a probability of not being in the room. This calculation tries to simulate people which by any reason does not attend to classes.

**User probability of vote:** To simulate a user voting, each time that the indoor temperature goes different from the user's comfort interval, it votes. But a user would not vote every time it feels uncomfortable, it should have a probability of voting. This probability is calculated using the poisson discrete probability distribution with mean 4 times a day.

**New Set-point Calculation:** There is no need to simulate the set-point, since the algorithm which will calculate the set-point is exactly the same as the one that will be used to calculate the true set-point on a production stage. The main difference is that this function will return the set-point to the Ptolemy II framework instead of returning it to the framework which controls the HVAC system.

### 3.6.3 Energy Plus

The Energy Plus software is the base system which simulates a room, and the temperature inside and outside the room. It takes on account the temperature that the human bodies irradiates inside the room, the sun's temperature, and a 12000W heater.

The Ptolemy II framework uses a simplified interface to work with the Energy Plus simulations, but it has the power to add complexity to the simulation, as to put and remove people from the room or turning the heating system on and off while the simulation is running.

The Ptolemy II framework is used to interact with the simulation as it runs by means of programming. The Ptolemy II framework, calculates all the temperature and HVAC related data. While the Application Core will only simulate users entering the room, voting and calculate the best setpoint while trying to minimize the occupants discomfort.

The framework uses Actors to automate and change variables whilst the Energy Plus simulation is running. The base simulation uses two actors, the first one to turn on and off the heating system, and the second one to add people to the room. The simulation starts at 0:00h, and starts the heating system at 7:00h, people are set to arrive at 8:00h. Lunch time is set between 12:00h and 13:00h. The people leave the room at 18:00h and the heating system goes off at the same time.

The Application Core is responsible to create users, simulate votes and calculate an optimal set-point, the Ptolemy II framework will work as a interface to connect both softwares. It starts by calculating a base temperature for a common winter day on Lisbon, then calculates the inside temperature and sends it to the Application Core, which will receive the internal temperature as an input parameter.

The Application Core will simulate the people feeling uncomfortable and voting as described in section 3.6.2, then it will calculate the new set-point and send the response to Ptolemy II, which will then calculate again the temperature of the room based on the new set-point. This cycle repeats itself on a time period of 10 minutes, and it will end once the simulated time reaches the 0:00h again.

### 3.6.4 Integration with Energy Plus

The Application Core and the Ptolemy II framework are both developed on a Java environment and there is no tool to assemble them. All the Ptolemy's actors are developed in Java language. Then an intercommunication interface should be made in order to make the integration of both systems.

The first step was to develop an Actor compatible with the Ptolemy II framework, its documentation and website have almost no information. A reverse engineering was made in order to develop an Actor which will receive the room's temperature as a input and return the new set-point as an output.

The first step is to define the input and output ports, and their function. The first input port is able to receive the simulated indoor temperature. The Actor then opens a socket connection to communicate with the Application Core, this communications calls the simulation algorithm which will receive as a parameter the temperature and waits for the response.

At this moment, the users are already created and they receive the temperature as a input and each user has a probability of voting as was explained on section 3.6.2. After collecting all the votes the system calculates the new set-point and returns it to the Ptolemy II Actor which at this moment is still waiting for a response in order to close the socket connection. The return message is the newest calculated set-point which will be the output of the Actor and input for the next iteration of the simulation which will be 10 minutes later..

#### *3.6.5 The Role of the Learning Algorithm*

The Learning Algorithm is needed on a post phase of the simulations, withdrawing the results of the simulations, and it defines the next decision of the new set-point. This step is important to validate if the k-means algorithm proposed on section 3.5.4 is a valid solution on the learning and decision making phase of the solution.

The algorithm retrieves the information on each vote made and save on the DB during the submission of a user request. The information retrieved by the k-means algorithm is calculated and returns different centroids. Each centroid is related to a system decision. When it activates, it calculates once per day the new centroids updating all the new requests and labelling each one of them.

Once the system has the clusters calculated, it takes the information on current variables such as the indoor temperature, hour of day, number of occupants and calculates the cluster that fits better these variables. Then verifies which is the decision associated with the cluster and then decides it.

At the beginning of each day, the algorithm calculates again the new centroids based on all the previous votes and the now new ones made on the previous day, this methods makes possible of using the algorithm to calculate the set-point and at the same time using data to train it to achieve better results.

The k-means decision algorithm is adapted on the Ptolemy II Actor, which will take total control over the set-point calculation. The users are still able to vote, but at this point their vote will not interfere with the set-point calculation, it is only for the purpose of training the Learning Algorithm.

## 4. EVALUATION AND METHODOLOGY

In order to evaluate the proposed solution, a simulation of an application to a real life scenario was made. The real life scenario consisted on a room where occupants are used the proposed solution and were able to cast a vote with the intent of setting the temperature which best fitted their comfort. The system would calculate the new setpoint and adjust the HVAC system accordingly. The data collected in order to evaluate the system corresponds to the overall daily comfort value, the daily mean energy consumption in KWh and the mean daily setpoint.

Three variations of the main scenario were used. In the first scenario the temperature within the room did not vary throughout the day, independently on the occupant's votes. This scenario was used as baseline of comparison to the reminder scenarios. The other two scenarios were: A scenario where the system adjusts the room temperature based on the occupants votes and a scenario where the room temperature is adjusted based on the Learning Algorithm.

### 4.1 Baseline

As mentioned, in this scenario the room temperature does not vary throughout the day. In order to calculate the mean comfort temperature equation 1 was used. Where $T_{out}$ was set at 13°C and as a result $T_c$ was nearly 21°C. Simulations were then made using fixed setpoints on 20°C, 21°C, 22°C, 23°C and 24°C. In figure 6 the blue bars represent the mean energy waste, going on a total mean of 78,33KWh with a confidence interval of maximum 105,55 KWh and minimum of 52,78 KWh. The red line represents the mean daily comfort rate with total medium of 73,40%.

Analysing the results displayed in figure 6-(a), it is possible to see that there is a high standard deviation (17,16KWh on energy spendings, and 3,31% on the mean comfort rate, see table 1). This is due to the fact that the occupants comfort values decreased on simulations where the setpoint was fixed on 23°C and 24°C. In order to try to attain results where the comfort is maxed, other simulations were made using as a fixed setpoint at 20°C, 21°C and 22°C, where the temperature will be near the mean comfort of the occupants. Figure 6-(b) depicts the results of the second baseline simulations, where blue bars represent the electric consumption and the red line represents the comfort rate. Table 1 provide comparison of the results of both simulations. The results of the second simulation are sightly better, where the mean energy consumption is 67,89KWh with a 95 comfort interval from 61,54KWh to 74,25KWh compared with 67,69KWh and 88,96KWh on the first simulations. The standard deviation of 9,73KWh and 17,16KWh on the fist one.

### 4.2 Simulation Tests

With the informations gathered in section 4.1 it was possible to simulate the occupants votes, and evaluate the results in order to verify if they would behave as expected. As mentioned, this simulation would simulate the adjustment of temperature based on the occupants votes.

Figure 7-(a)represents the results on the simulations and table 2 displays the same data, so as to facilitate a direct comparison with the data displayed in table 1. On the simulations the setpoint was always around 21°C as expected, and the comfort value has a 73,6%, which is almost 2% less than the second simulation made on section 4.1. This margin can be disregarded for being too low.

### 4.3 Learning Algorithm

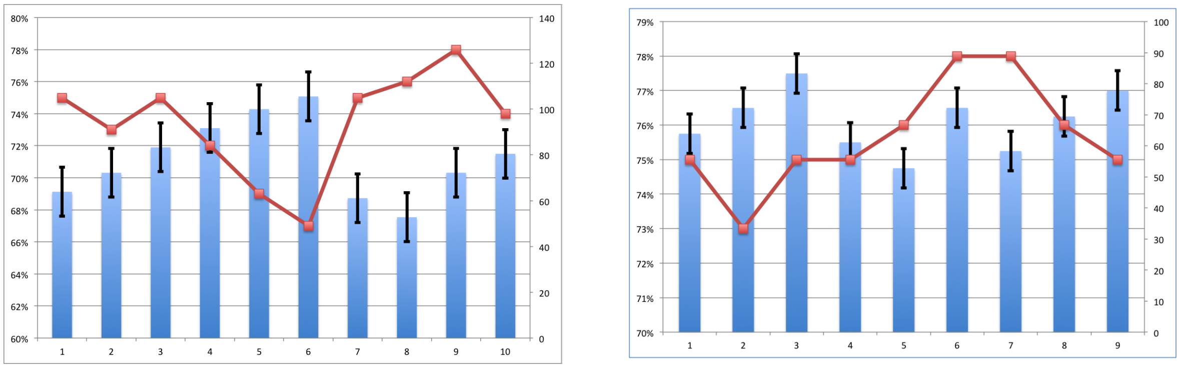


**Figure 6:** Baseline simulations results. Where the red line shows the percentage of the computed comfort, and the blue bars shows the the energy consumption in KWh of each simulation. And on the left there is the results of the first baseline simulations (a) and on the right the results on the second one (b).

| | First Simulation | | Second Simulation | |
|---|---|---|---|---|
| | Energy Use (KWh) | Comfort % | Energy Use (KWh) | Comfort % |
| Mean | 78,33 | 73,40 | 67,89 | 75,67 |
| Std Deviation | 17,16 | 3,31 | 9,37 | 1,58 |
| Superior CV | 88,96 | 75,45 | 74,25 | 76,70 |
| Inferior CV | 67,69 | 71,35 | 61,54 | 64,96 |

**Table 1:** Comparison of the results of the two baseline simulations

The Learning Algorithm was tested in three different scenarios. The first one was intended to test the k-means algorithm using 5 days simulations to train it and 9 centroids ($k = 9$). The second one used 10 days to train it and 9 centroids($k = 9$). The last one also used 10 days to train it, but with only 3 centroids ($k = 3$).

The results on the first two scenarios are presented in table 3. By analyzing the table it is possible to see that, the results sightly better than the baseline, but after analysing the standard deviation and confidence interval, it is notable that the achieved results are still heterogenous in both tests. The mean energy waste has achieved a mean of 54,99KWh and 52,49KWh. This was due to the fact that the k-means decision algorithm was lowering the temperature, thus less energy was being used to warm the room, this explains the low comfort rate in both simulations.

Since both results are equally heterogenous, it can be deduced that using more days of training is not enough to achieve more homogenous results. This problem could be due to the high number of clusters, since there are many clusters, the k-means may have trouble to identify clusters which are similar to each other.

One possible solution would be change the number of clusters. If the observations are too homogenous, more clusters should solve the problem. However, if the observations are too homogenous, probably using less clusters could divide them more precisely. Since 9 clusters is a relatively big number when compared to the total amount of data to analyse as input, it was decided to use less clusters, and the number was reduced to 3.

For the last scenario, the same data was used as input, the results are depicted on figure 7-(b) where it can be seen that the results are more homogeneous, however the comfort curve is still lower than the one displayed in figures 6 (a) and (b) from the baseline simulations. Through an analysis of table 4, the results prove to be near the expected but still needs some improvement. Comparing the data between tables 4 and 1, it is possible to determine that the mean energy use is slightly lower. This could mean that the system configured the HVAC System to be too cold, an assumption which can be confirmed by the lower value on the user comfort rate which is 70,78% against 75,67% on the fist baseline.

## 5. DISCUSSION

Considering that on a first moment, a baseline scenario was drawn in order to simulate an real life situation where the HVAC system is configured on a fixed temperature. The second baseline simulation is chosen to be the one to compare results, because this simulation was set a setpoint which was near the user mean comfort value.

On the first scenario, the most relevant retrieved data is related to the energy spent, which is notably 18,19KWh less than the baseline, accounting for an energy reduction of 26,79%. This results suggests that it is possible to save energy only by allowing the occupants to vote on the temperature which more suits their needs.

The k-means algorithm has been used on three different configurations, the first and second simulations was intended to prove that the learning days could change the accuracy of the decisions. Even failing to prove this concept, the third scenario prove that in this particular case the accuracy is defined by the configuration variables used on this system. In this case, by using less number of clusters, the accuracy was increased. This is due to the data collected from the votes, the data probably is very heterogeneous, having a big distinction among each cluster, in that case, only three clusters could define more precisely a decision.

The results on the the last scenario using the k-means algorithm suggest that it is possible to allow a room to configure the HVAC system based on a machine learning technique. Despite the comfort rate, ehich is approximately 5% lesser than the baseline, the energy consumption had a decrease of 5KWh on the mean consume. This suggests that

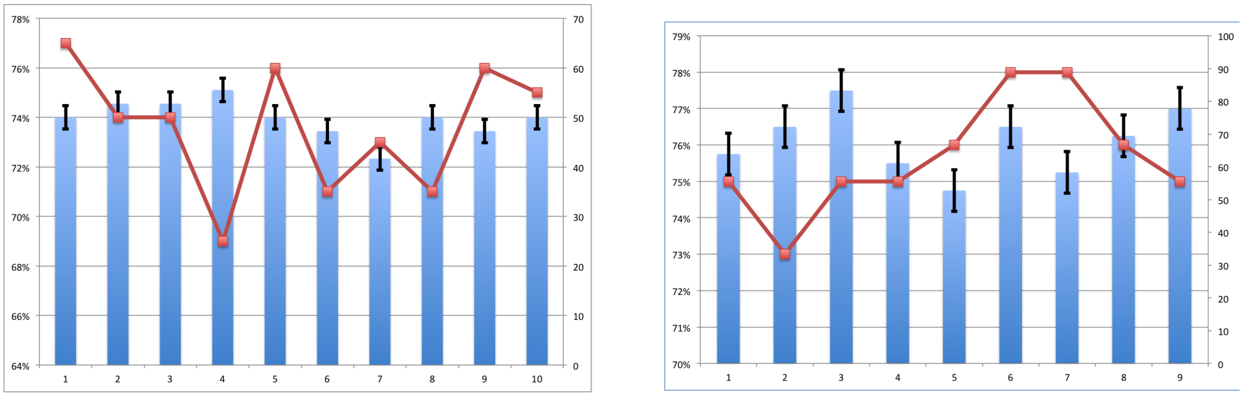


**Figure 7:** Simulation results using the occupant's votes to configure the setpoint on the left (a), and on the right (b) is the second k-means simulation results, using $k = 3$. Where the red line shows the percentage of the computed comfort, and the blue bars shows the the energy consumption in KWh of each simulation.

| | Energy Use (KWh) | Comfort % | Mean Setpoint |
|---|---|---|---|
| Mean | 49,72 | 73,6 | 20,997 |
| Std Deviation | 3,81 | 2,59 | 0,35 |
| Superior CV | 52,08 | 75,21 | 21,21 |
| Inferior CV | 47,36 | 71,99 | 20,78 |

**Table 2:** Results on the simulations using occupants votes.

there was a reduction on the energy consumption while abstaining from the occupant comfort rate.

# 6. CONCLUSIONS

This work presented several contributions, of which it can be highlighted the creation of a prototype application which interacts with simulation applications and could be used in order to test various types of setpoint calculation and learning algorithms. Our hypothesis was that developing a system which gives occupants the appropriate means to control the HVAC system, results in the minimization of the energy consumption while maximizing comfort.

## 6.1 Retrospective

The use of BAS systems are becoming everyday more common, since energy management is a major global concern, there is a raising interest in system which could promote energy savings.

Even with plenty of system which explores the same area of study, there still remains numerous questions yet to be answered. One of which is the possibility of allowing AmI Systems to control building's energy management system. This works proves to be possible to give the chance to occupants to give their opinion on a HVAC system and still provide reduction on the energy consumption. However, using a learning algorithm to control this category of system proved to be delicate, in which it is possible however, cautions and excessive testing needs to be taken, in order to start using in a real environment. This works also contains the development of a prototype system which makes possible the occupancy detection through a small electric boards which is able to read the university student card.

## 6.2 Achievements

This thesis contributed on the area of home and building automation systems with emphasis on energy management, with testing and validation based on simulations. It is addressed to the lack of system which could configure the HVAC system based on the preferences of the occupants. The main achievements of this work are:

**An Occupancy Detection system** based on the Arduino technology which is capable of reading the student card, thus could be used on many other practical applications.

**The Development of an Platform** which could simulate users on a room, communicate with a HVAC simulation framework, and also can be upgraded to use machine learning algorithms to configure the HVAC system through simulations. And their results which suggests that it is possible to configure the HVAC system of a room by taking into account the occupants preferences, while consuming less energy.

**The Machine Learning technique** applied on the platform, whose results suggests that is also possible to configure an HVAC system by loading the occupants preferences and deciding which is the best setpoint that minimizes the energy consumption maximizing comfort.

## 6.3 Future Work

The implementation of this work involved several distinct technologies, making the main challenge for this project, to provide a simple and user-friendly interaction between them. However, it can be upgraded to be supported by new technologies. On the current implementation, the system uses a web server which is able respond inputs of all sorts of client applications; a relational DB model; an RFID system composed of Arduino and Raspberry Pi; interconnection with a simulation platform; and a learning algorithm capable of taking decisions.

**Web Server:** The Web Server in this moment, is interconnected with an simmulation application in which is the one responsible for providing all the presented results of this

| | $k = 9$ and Days=5 | | | $k = 9$ and Days=10 | | |
|---|---|---|---|---|---|---|
| | Energy Use (KWh) | Comfort % | Setpoint | Energy Use (KWh) | Comfort % | Setpoint |
| Mean | 54,99 | 67,26 | 20,11 | 52,49 | 66,44 | 20,24 |
| Std Deviation | 28,72 | 2,27 | 2,84 | 39,74 | 2,38 | 4,45 |
| Superior CV | 72,79 | 68,66 | 21,87 | 77,12 | 67,91 | 22,99 |
| Inferior CV | 37,19 | 65,85 | 18,35 | 27,85 | 64,96 | 17,48 |

**Table 3:** Results on the simulations using K-means algorithm, with 5 and 10 days of training.

| | $k = 3$, Days=10 | | |
|---|---|---|---|
| | Energy Use (KWh) | Comfort % | Setpoint |
| Mean | 62,22 | 70,78 | 21,67 |
| Std Deviation | 4,18 | 4,01 | 0,41 |
| Superior CV | 64,81 | 73,26 | 21,93 |
| Inferior CV | 59,63 | 68,29 | 64,96 |

**Table 4:** Results on the simulations using K-means algorithm, using $k = 3$.

work. It could be enhanced by adding a connection with a real HVAC controller system. In which would provide the use of a real world testbed.

**Architecture Expansion:** The actual architecture has been planned and developed for being used on an university. However, the presented architecture makes possible the expansion for other applications, such as meeting rooms, cinema and theatres, commercial centres among others.

**Client Applications:** The solution was developed by using technologies which will allow support by any type of client applications. However, besides the RFID and Simulation systems, in the actual moment there isn´t any client application. Mobile application for users could be developed in order to make this system more user-friendly and more acceptable for occupants.

**RFID System:** Taking in account that the RFID System has only been tested on labs, and it was still not been tested with real users. In the actual moment, to validate this system with real users, it needs to be installed and tested.

**Arduino and Raspberry Pi:** These system are responsible to managing the occupancy detection on the room. However, there are many other means of detecting occupants in a room. These systems could be improved by the adding of new sensors in which would validate different occupancy detection means.

**Simulations:** At this moment, the interconnection with the simulation framework is at a very initial state. The Energy Plus program is a very powerful tool to simulate various types of ambients, and this feature could be studied furthermore in order to improve the possible results on simulation that this application can provide.

**Learning Algorithm:** Besides it was possible to prove the viability of using an learning algorithm to control the room's setpoint, the tested learning algorithm had its results below expected. However, since there is of existence other learning algorithms, it would be interesting the results and comparison on various different learning algorithms and approaches using the simulation framework.

## 7. REFERENCES


[1] L. Avraham, Rayfield, and J. T. Web-Application Development Using the Model/View/Controller Design Pattern. *IEEE Enterprise Distributed Object Computing Conference*, pages 118–127, 2001.

[2] M. Boman and P. Davidsson. Energy saving and added customer value in intelligent buildings. *International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, 1:505 – 516, 1998.

[3] Grails. Grails Reference Documentation. `http://grails.org/doc/latest/guide/spring.html`. Accessed 04 May 2014.

[4] Grails. Grails website. `http://grails.org/`. Accessed 12 Apr 2014.

[5] T. O. Group. What is SOA? `http://www.opengroup.org/soa/source-book/soa/soa.htm#soa_definition`. Accessed 04 May 2014.

[6] M. Humphreys and J. Nicol. *Understanding the adaptive approach to thermal comforts*, volume 14 of *ASHRAE Technical Data Bulletin*. 1998.

[7] L. Klein, J.-y. Kwak, G. Kavulya, F. Jazizadeh, B. Becerik-Gerber, P. Varakantham, and M. Tambe. Coordinating occupant behavior for building energy and comfort management using multi-agent systems. *Automation in Construction*, 22:525–536, Mar. 2012.

[8] D. MacKay. Information theory, inference and learning algorithms. pages 316 – 322, 2003.

[9] U. D. of Energy. EnergyPlus Energy Simulation Software. `http://apps1.eere.energy.gov/buildings/energyplus`. Accessed 04 May 2014.

[10] K. Padmanabh and A. M. V. iSense: a wireless sensor network based conference room management system. *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 37–42, 2009.

[11] U. Regents. Ptolemy Project. `http://ptolemy.eecs.berkeley.edu/ptolemyII/`. Accessed 04 May 2014.

[12] S. Soucek and G. Zucker. Current developments and challenges in building automation. *e & i Elektrotechnik und Informationstechnik*, 129(4):278–285, aug 2012.

[13] Z. Sun and Y. Wun. Multispectral Image Compression Based on Fractal and K-Means Clustering. *2009 First International Conference on Information Science and Engineering*, (1):1341–1344, 2009.

[14] R. Yang and L. Wang. Development of multi-agent system for building energy and comfort management based on occupant behaviors. *Energy and Buildings*, 56:1 – 7, 2013.