Modeling and Verification of Building Automation Control Systems

Alessio Bollea, Francesco Grella

February 2019

1 Introduction

The goal of this project was to design and validate a model of an intelligent Room Management System for climate regulation of a room with dimensions resembling those of a generic university classroom.
The work has been divided in two different parts:

- The first one consisted of the implementation, by using the Ptolemy II modeling software, of a realistic physical model of the environment and the effects of people inside the room on three fundamental parameters for climatic regulation: Temperature, CO2 Concentration and Humidity. A discrete control based on Finite State Machines has been developed for the three parameters.

- The second part, strictly based on the first one, was voted to build an intelligible learning model from simulation data and user preferences with respect to the conditions of the environment.

2 Ptolemy Modeling of the HVAC System

During the modeling phase several details have been taken into account from the physical point of view, in order to provide realistic data for simulation of the control process.

2.1 The User Model

The User Model allows to simulate the behaviour of people entering and leaving the room. At every time instant a random integer value is generated between -1 and 1; this simulates a single person that can enter or leave the room. This is done for every simulation tick to stress the system, but the model can be easily

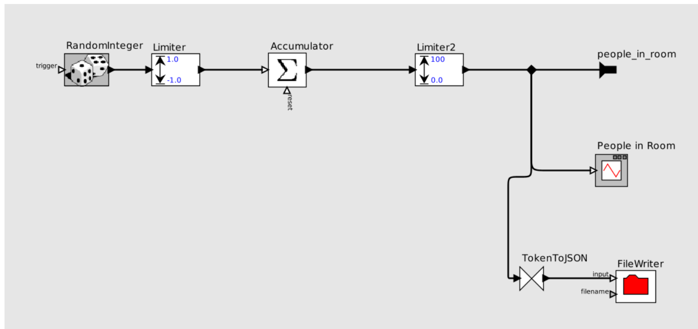

Figure 1: The user model

modified to represent a more realistic situation, where people stay for a longer time.
The generated value is added through the *Accumulator* to the previous number of people in the room to update it.

2.2 The Room Model

The room, which represents the physical environment in which the system is employed, is a standard-shaped room with a *600* m^3 volume: this notion, jointly with the pressure of the air fixed at *101325 Pa*, is crucial for the model because the following characteristic quantities are computed keeping it into account:

- The number of *Gas Moles* contained in the room:

$$n = \frac{PV}{RT_0}$$

  where P represents the Pressure, V the volume of the room, R is the *Perfect Gas Constant* fixed at 8.31 and T_0 is the initial temperature of the environment, which is assumed for hot seasons at 294.5 K.

- The *Mass of Dry Air (M)*, necessary for computations in the humidity model:

$$M = V \cdot \rho = V \cdot \frac{P}{R_{specific} T_0}$$

  where ρ is the dry air density and $R_{specific}$ is the specific gas constant for dry air

The room is implemented as an actor which contains three different sub-actors, one for each variable taken into account for the simulation (temperature, humidity and CO_2 concentration).

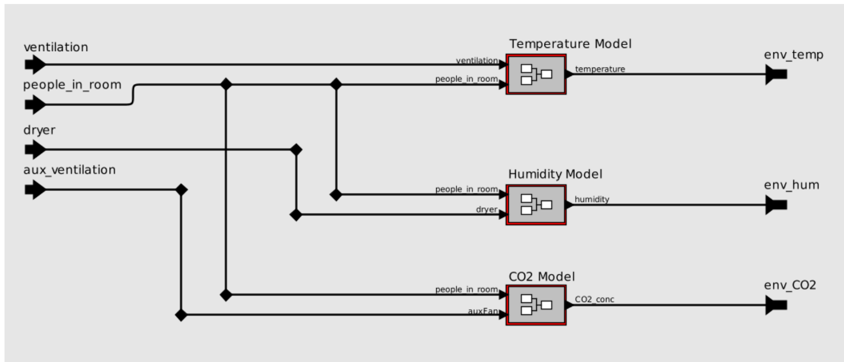

Figure 2: The room model

2.3 The Temperature Model

2.3.1 Temperature Equation

The temperature is the most important variable taken into account, because it's the parameter people are more sensible to. Every person in the room produce a certain quantity of heat that increases the temperature of the environment; this is known as **human heat gain**. Let's call λ the heat gain per person (Watt). The internal energy of a thermodynamic system, considering air as a bi-atomic molecule, can be computed as

$$U = \frac{5}{2} nRT \tag{1}$$

To better understand the temperature behaviour, a state space model is needed. By computing the derivative with respect to time on both members and isolating the temperature, it becomes:

$$\frac{dT}{dt} = \frac{2}{5nR} \frac{dU}{dt} \tag{2}$$

So the temperature depends on the power provided to the system. In the model, this power is considered as the sum of two different sources:

- The human heat gain, defined as the number of people in the room multiplied by the heat gain per person ($Pw^{\text{body}} = N\lambda$)
- The heat provided by the HVAC system (Pw^{HVAC})

By substituting these definitions in equation 2, we get:

$$\frac{dT}{dt} = \frac{2}{5} \frac{N\lambda}{nR} + \frac{2}{5} \frac{Pw^{\text{HVAC}}}{nR} \tag{3}$$

This equation would be accurate if n was constant; this is an assumption we can't make, so we should substitute its definition based on the perfect gas law in both the terms in the second member. But to avoid having a too complex equation with a non-linearity in the input, we decided to do it only in the first term of the sum and get:

$$\frac{dT}{dt} = \frac{2}{5}\frac{NT\lambda}{PV} + \frac{2}{5}\frac{Pw^{\text{HVAC}}}{nR} \tag{4}$$

Considering N and T as state variables for the system, we decided to linearize this equation, by computing the derivative of the second member over the state variables and substituting the equilibrium points, to make the implementation easier. The resulting equation, substituting the value of $\lambda = 63W$, is:

$$\frac{dT}{dt} = \frac{25.2}{63nR}N + \frac{25.2}{PV}T + \frac{0.4}{nR}Pw^{\text{HVAC}} \tag{5}$$

At the end, we computed the discrete-time version of this equation. Assuming that the input is constant in every instant between two consecutive time periods, we can substitute the derivative with its definition and get

$$T(k+1) = \frac{25.2 \cdot \Delta t}{63nR}N(k) + (1 + \frac{25.2 \cdot \Delta t}{PV})T(k) + \frac{0.4 \cdot \Delta t}{nR}Pw^{\text{HVAC}}(k) \tag{6}$$

which is the equation used in the model, where Δt is the sampling time.

2.3.2 Temperature Model implementation in Ptolemy

The temperature model is simply implemented as a sequence of mathematical actors, such as *Expression*, *MultiplyDivide* and *Gain*, to recreate equation 6 in a graphical way.

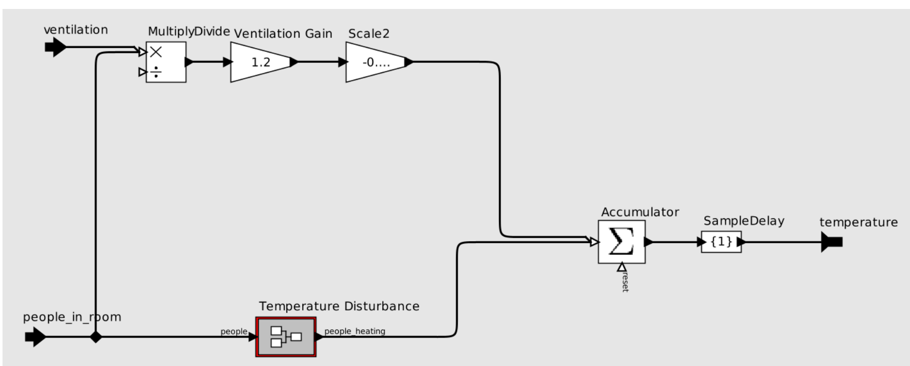

Figure 3: The temperature model

The *Accumulator* allows to store the temperature value every instant and update it, adding the input to it. It's a way to implement state-space models, where you want to represent the evolution of a variable.
The *Temperature Disturbance* block has only an expression to compute the heat produced by people in the room.

2.4 The Humidity Model

2.4.1 Humidity Equation

Humidity is another important variable to take into account.
Two different values of humidity can really change the way people perceive temperature, moreover, high humidity values can cause the making of mold, which can cause respiratory diseases.
For the model we started considering the **humidity ratio** x, defined as the ratio between actual mass of water vapor present in air and the mass of the dry air. Every person generates vapor; the quantity of vapor generated depends on some factors, but the most important is the physical activity performed by the person. The **vapor generation ratio** per person ξ on average, for a person at rest or performing office job or studying, is $1.94 \cdot 10^{-5} kg/s$.
Ignoring possible disturbances, water vapor mass remains constant in a room, unless vapor is generated by people or subtracted due to the activation of the HVAC system; so, the derivative of water vapor mass is equal to the sum between vapor generated by people (which is ξ multiplied by the number of people N) and the vapor mass added or subtracted by the HVAC system (u^{HVAC}):

$$\frac{dm}{dt} = \xi \cdot N + u^{\text{HVAC}} \tag{7}$$

From this equation we can get the humidity ratio equation, simply dividing both members by the total mass of dry air (M), that can be computed using the perfect gas law:

$$\frac{dx}{dt} = \frac{\xi \cdot N}{M} + \frac{u^{\text{HVAC}}}{M} \tag{8}$$

Then, we can find the discrete-time version of this equation, in the same way as the temperature equation (equation 6):

$$x(k+1) = x(k) + \frac{\xi \cdot \Delta t}{M} N(k) + \frac{\Delta t}{M} u^{\text{HVAC}}(k) \tag{9}$$

In the model we also compute the **relative humidity**, which is the humidity ratio x over the maximum saturation humidity ratio (the maximum possible amount of water vapor in the air at the current temperature). We need relative humidity because it's the value that must be controlled.

2.4.2 Humidity Model implementation in Ptolemy

The humidity model is implemented in the same way as the temperature model, where there's an *Accumulator* to store the humidity ratio value at every instant and update it adding its input.

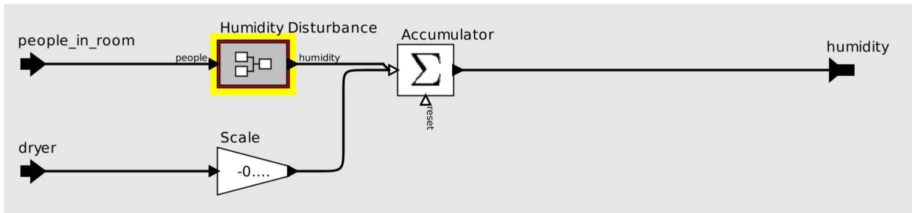

Figure 4: The humidity model

2.5 The CO_2 Model

2.5.1 CO_2 Equation

CO_2 concentration in the air is the most difficult variable to model, because a lot of different physical factors must be kept into account. For this reason, in our model we decided to use the equation found in the article "Dynamic of Changes in Carbon Dioxide Concentration in Bedrooms" by Piotr Batog and Marek Badura [1]. The equation is:

$$c(t) = \frac{\lambda V_{\mathrm{m}}}{QM}(1 - exp(-\frac{Qt}{V})) + c_0 \tag{10}$$

where:
$c(t)$: average CO_2 concentration at time t
c_0: initial CO_2 concentration
V: volume of the room, m^3 V_{m}: air molar volume, $2.24 \cdot 10^{-2} m^3/mol$
M: CO_2 molar mass, $4.4 \cdot 10^{-2} kg/mol$
Q: ventilation rate (our input variable), m^3/s
λ: CO_2 production rate, kg/s
The CO_2 production rate, in our model, depends on the number of people: it's the number of people at time t ($N(t)$) multiplied by the CO_2 production rate of a single person ($\gamma = 10^{-5} kg/s$). So equation 10 becomes:

$$c(t) = \frac{\gamma V_{\mathrm{m}} N(t)}{QM}(1 - exp(-\frac{Qt}{V})) + c_0 \tag{11}$$

This equation has three limitations:

- Q can't be 0, since it's at the denominator. This is a limitation, but in some norms, such as ASHRAE norm (American Society of Heating, Refrigerating and Air-Conditioning Engineers, which releases standards for heating, refrigeration and air conditioning), a minimum amount of inlet air flux is mandatory.
- Q should be constant, otherwise c can have strange behaviours. In our case Q is not constant but it's a step function (because the HVAC system can be enabled or disabled), so the output presents discontinuities.
- The equation is not in state-space form.

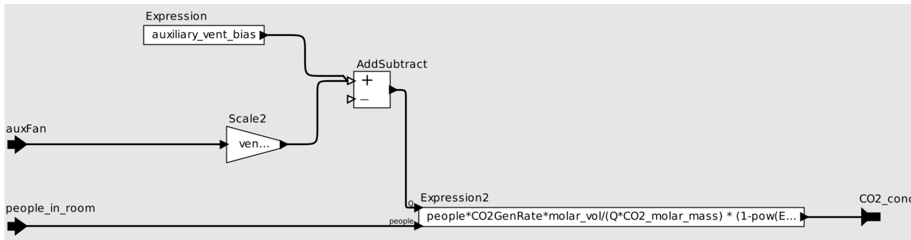

Figure 5: The CO_2 model

2.5.2 CO_2 Model implementation in Ptolemy

Since equation 11 is not in state-space form, the implementation here is slightly different, in fact we only need an *Expression* to compute the CO_2 level for every simulation step.
As said before the ventilation ratio Q (the control variable) can't be 0, so we suppose to have always a minimum ventilation ratio, even if the control is disabled. This ventilation ratio is increased to the maximum value when the control is enabled.
This mechanism is implemented by multiplying the control variable value (which can only have value 0 or 1) by the maximum ventilation ratio and then adding the minimum ventilation ratio

2.6 The Room Management System

The **Room Management System** is the control system for the three variables.
It's implemented as an actor that takes as input the three quantities and generates three different outputs; every output is a digital signal for enabling or disabling the control of a particular variable. In fact, we suppose that there are three different devices, where every device works independently from the others. These devices are an air conditioning system for temperature, a dryer for humidity and a fan for CO_2.
Internally, it's composed by three independent extended finite state machines; every state machine is in charge of generating the control signal for a single variable of the room, so there is a machine that controls the air conditioning system, one that controls the dryer and one that controls the CO_2 fan.
All the state machines are implemented in a very similar way: they start in a state where the output is 0; if the variable exceeds the desired value plus an hysteresis the machine goes in a state where the output is 1. Here, if the variable goes under the desired value minus an hysteresis, the machine goes in the previous state where the output is 0. The only difference is that the temperature controller has this mechanism implemented twice, one for cold seasons (heating) and one for hot seasons (cooling); the two parts are connected by two immediate

transitions to an initial state where a variable called *isWinter* is checked. One of the two immediate transitions is taken basing on the value of that variable.

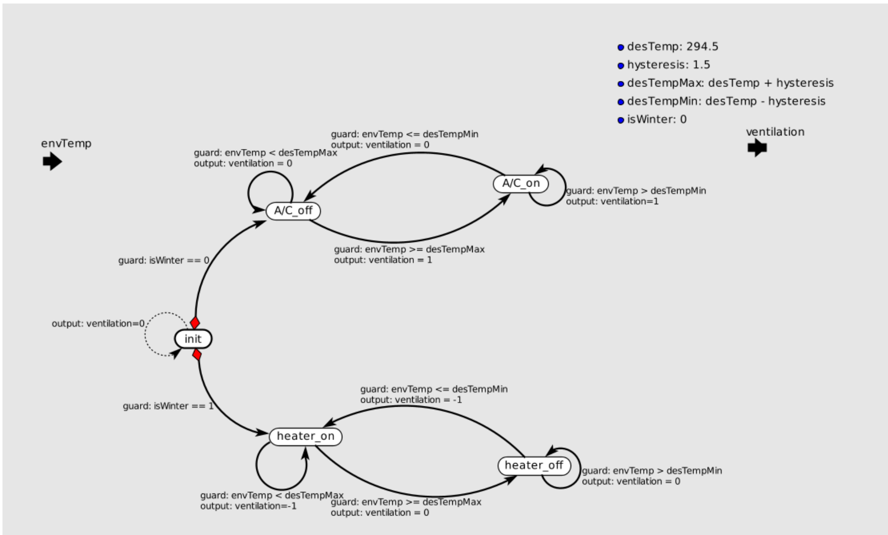

Figure 6: The temperature controller

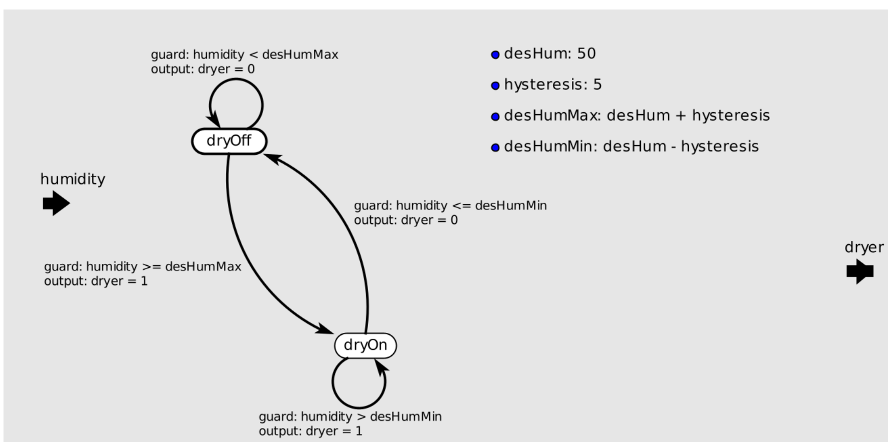

Figure 7: The humidity controller

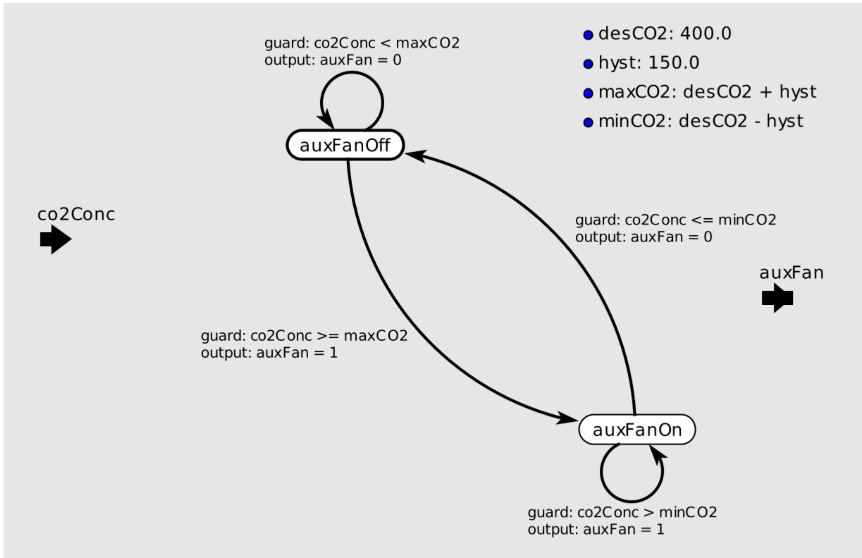

Figure 8: The CO_2 controller

2.7 Top-Level Model

All these parts are combined to form a closed loop architecture: the new values are computed in the *Environment Actor* (or *Room Actor*) and sent to the *Visualization* block, where values are converted in other units (kelvin to celsius for temperature, kg/kg to ppm for CO_2 and m^3/m^3 to % for humidity) and then visualized using *Sequence Plotters*. The same values are also brought to the *Room management System* to generate the control signals for the next simulation step.

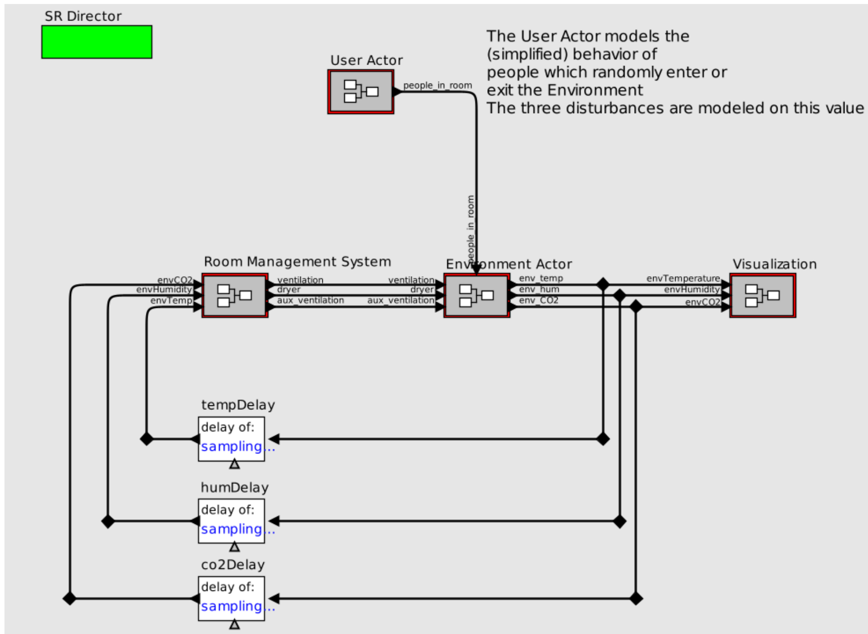

Figure 9: The top level of the model

3 Data Analysis

The verification part of the project has been centered on data analysis and identification based on artificial intelligence techniques, in particular by using *Decision Trees*.

The choice of using that particular learning model is due to the fact that is more intelligible than other well-known techniques like neural networks, and because through the *GraphViz* Python library the tree can be visualized graphically on image.

Two models have been built and trained using output data from the Ptolemy model, one for implementing a simulation and prediction of the *Preferences of the Occupants* and the other one intended for learning the *Logic of the Controller* and predict control actions.

Moreover, in both cases, an optimization has been performed during the decision tree building. In particular a 'best-tree selection' has been implemented, with a function that checks different depths and returns only the tree which has the least amount of wrong predictions.

3.1 User Preferences

The preferences which can be virtually expressed by the occupants of the room have been modeled with a random distribution with discrete values between -3 and 3 to take into account the fact that in presence of a certain temperature or humidity level the evaluation depends on the feelings of the person, so it could be different between individuals.

From the idea of merging the preferences in a single variable, to reduce the dimensionality, we have computed the *Humidex*, an index which merges humidity and temperature information for computing the preferences: given temperature $\boldsymbol{T}$ and relative humidity $\boldsymbol{H}$ the index is computed as follows.

$$humidex = T + \frac{5}{9}(6.112 \cdot 10^{\frac{7.5T}{237.7+T}} \cdot \frac{H}{100} - 10) \tag{12}$$

It must be said that the humidex is relevant only for preference values related to a neutral or hot perception (0,1,2,3), so if humidex value is over a certain threshold.
For low humidex values and so for preference values associated to perception of a colder temperature, the temperature value must be checked too.
So if the humidex value is below 30 the temperature is checked and negative preference values are given along the scale of 'cold discomfort', instead the index is checked to determine the level of discomfort for hot environments.

During this part of the project the CO_2 was neglected since we are considering a natural human perception, in fact it seems difficult that a human can distinctly perceive changes in all the variables, especially for CO2 which is perceived when it's very high and anyway it must be kept always under control in respect of the norms.
The dimensions of training and test set have been fixed to 70 and 30 percent of the overall dataset, and for training the model only two features have been used: Room Temperature and Humidity percentage.
The target variable is the Combined Preference, so the preference computed by keeping into account the Humidex in the way explained before; the results evaluated by performing the prediction on the test set have shown a satisfying classification of the user preferences.
An example of decision tree for preferences prediction can be seen in figure 10.

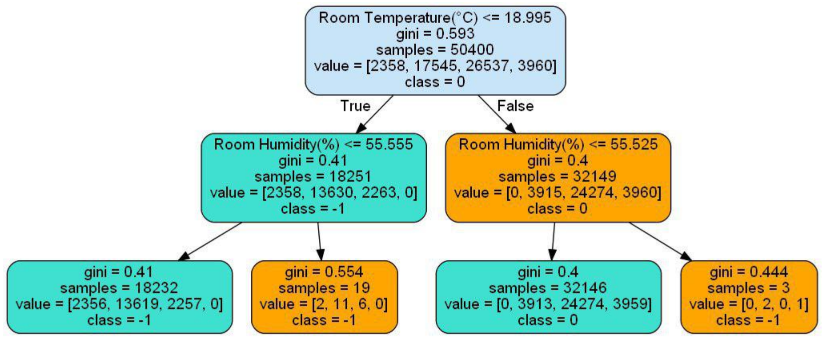

Figure 10: Decision tree for preferences prediction

3.2 Predictive Control

A similar approach has been used for predictive control, in particular there have been used the values of Temperature, Humidity and CO_2 levels as well as the control values obtained during a simulation of the model.
The control signals like said before represent 'On' or 'Off' behavior with '1' and '0' values, and the objective of this part of the project is to validate the model by having a Decision Tree which learns the logic of the three controllers implemented as finite state machines.
By using the same division between training and test set, this time there have been performed three different classifications for each one of the tree controllers, since in the model they are decoupled.
For each one of the tree variables three features have been used, the quantity of interest (Temperature, Humidity or CO_2), the amount of people in the room and the control value at the previous instant, while the target is identified in the actual control values.
From our tests we have found out that the prediction works quite well even without using previous inputs as feature, but when using them the amount of errors in predictions drops drastically: we think that this behavior is due to the fact that the control signal assumes the same value for a very large amount of samples.

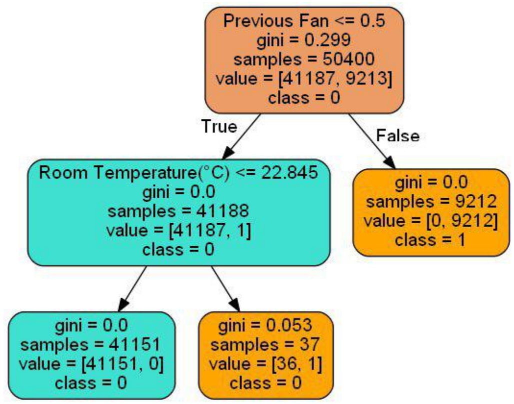

Figure 11: Decision tree for predictive control with previous inputs

4 Conclusions

To sum up whole the details described along the report, what we could get along the project is a working and sufficiently reliable model of an HVAC system, with a certain amount of flexibility through the parameters interface, which allows for instance to modify the volume of the room.
The simulated system should support the implementation of a proper Cyber-Physical System based on Internet of Things technologies: the choice of neglecting the reciprocal effects of the three controls on the other variables has been used for a simplification of the model itself, the idea of decoupled control behind it can suggest an implementation based on a main controller and distributed microcontrollers for local environment monitoring and control of the devices.
Moreover the idea of having the occupants expressing their preferences, thorugh a dedicated smartphone app, on the conditions of the environment will lead to a more 'vivid' classroom whose temperature, humidity and CO2 conditions will be adjusted in real time according with them.

The results of the simulations in Ptolemy and the outcomes of predictions by using the Decision Tree models are quite satisfying and for the actual state could be a basis for further improvements and implementations.

4.1 Future Work

Actually the model can properly be extended and improved in many ways, for instance the CO2 behavior has to be better refined to be better reliable, in particular the disturbance equation can be revised in order to obtain a smoother behavior.
Secondly the model has the possibility to behave even in cold seasons, due to the structure of the state machine in the temperature control; anyway the effects on

the model were beyond the scope of this project, since other factors like a 'cold' disturbance coming from the external environment and a steeper decreasing of the parameters when air conditioning is active have to be taken into account.

References

[1] Marek Badura Piotr Batog. "Dynamic of Changes in Carbon Dioxide Concentration in Bedrooms". In: *Procedia Engineering* 57 (2013), pp. 175–182. DOI: `https://doi.org/10.1016/j.proeng.2013.04.025`.

[2] Engineering ToolBox. *Human Heat Gain.* URL: `https://www.engineeringtoolbox.com/persons-heat-gain-d_242.html`. (accessed 02.02.2019).

[3] Engineering ToolBox. *Humidity Ratio of Air.* URL: `https://www.engineeringtoolbox.com/humidity-ratio-air-d_686.html`. (accessed 04.02.2019).