Multi-Agent Architecture for Control of Heating and Cooling in a Residential Space

Domen Zupančič[1], Mitja Luštrek[2] and Matjaž Gams[2]

[1]*Robotina d.o.o., OIC-Hrpelje 38, 6240 Kozina*
[2]*Department of Intelligent Systems, Jožef Stefan Institute, Ljubljana, Slovenia*
Email: domen.zupancic@ijs.si

Energy demand in a smart grid is directly related to energy consumption, as defined by user needs and comfort experience. This article presents a multi-agent architecture for smart control of space heating and cooling processes, in an attempt to enable flexible ways of monitoring and adjusting energy supply and demand. In this proposed system, control agents are implemented in order to perform temperature set-point delegation for heating and cooling systems in a building, offering a means to observe and learn from both the environment and the occupant. Operation of the proposed algorithms is compared to traditional algorithms utilised for room heating, using a simulated model of a residential building and real data about user behaviour. The results show (i) the performance of machine learning for the occupancy forecasting problem and for the problem of calculating the time to heat or cool a room; and (ii) the performance of the control algorithms, with respect to energy consumption and occupant comfort. The proposed control agents make it possible to significantly improve an occupant comfort with a relatively small increase in energy consumption, compared to simple control strategies that always maintain predefined temperatures. The findings enable the smart grid to anticipate the energy needs of the building.

Keywords: multi-agent architecture, smart building, heating, user behaviour modelling, simulation, energy consumption, comfort experience

Received ; revised

1. INTRODUCTION

Heating is the largest consumer of energy in buildings. In 2010, 41 percent of energy consumed by buildings was spent on heating (27 percent for households and 14 percent for the tertiary sector). In buildings, more than 65 percent of this energy was used for space heating, 12 percent for domestic hot water (DHW), and 15 percent for electric appliances [1, 2].

The energy consumption of a household is closely related to occupant comfort. Therefore, the efficacy of a control system should be evaluated in terms of both energy consumption and occupant comfort. Control and management of building automation systems (BAS) is a complex problem, because it requires mathematical and physical background of building operations as well as knowledge about occupant behaviour and interaction with the building. Occupant comfort is evaluated only during their occupancy, making it possible to significantly reduce energy costs and energy consumption for HVAC or lighting during periods when the building or rooms are unoccupied. Therefore, a systematic approach to efficiently control the energy consumers in buildings should consider when to give more weight in the evaluation to occupant comfort, and when give more weight to energy and energy cost savings. Further, accurate forecasting of near-future building occupancy and a thermal dynamic building model together form the basis for an effective optimization of energy costs and occupant comfort. If a building that employs such a prediction method is connected to a smart grid, these methods can also enable the grid to anticipate the power demand and energy consumption of the building.

The control architectures for BAS, building management systems (BMS) and building energy management systems (BEMS) are used to manage the operation of each entity in a system and provide feedback about system states and analysis of operation. Control architectures are either centralised, decentralised, or distributed. Moroşan et al. [3] compared the computational efficiency of these architectures and concluded that the distributed architecture is less computationally demanding than the centralised architecture, while achieving the same effect.

This article presents the multi-agent control architecture for smart buildings, taking into account knowledge about human behaviour when scheduling the heating and cooling processes. The research is based on our previous work implementing the multi-agent architecture for a DHW system, which incorporated knowledge about user behaviour into water heater operations in order to reduce energy consumption and energy costs, while increasing comfort experience [4]. Under this system, energy consumption should be low compared to consumption in the "always on" strategy, and comfort should increase compared to the "on-presence on" strategy. In addition, the multi-agent architecture is considered the most appropriate for grids, because agents and grids are highly dynamic in nature. The related work is presented in Section 2. Section 3 presents the proposed multi-agent architecture and the operation of the proposed types of agents. The machine learning methods that identify the heating dynamics and forecast occupant behaviour are presented in Section 4. Section 5 presents the algorithms used to control the operation of an individual process. Experimental set-up is presented in Section 6, followed by results and discussions in Section 7. Section 8 concludes the paper and offers an overview of possible directions for further investigation.

2. RELATED WORK

There are several available approaches to reducing the energy consumption required for heating and cooling. One important approach in modern building construction is related to low-energy buildings that require minimal energy for space heating and cooling. In both new and existing buildings, methods exist for encouraging occupants to reduce heating or cooling energy consumption by decreasing set-point temperatures for heating, or increasing set-point temperatures for cooling, using a visual representation of the energy consumed [5, 6, 7, 8]. Methods also exist to encourage occupants or to apply management policy to control systems, drawing on self-reported data about occupant interactions with environment. Cho et al. proposed an appliance aware activity recognition mechanism for home, which notify users what unused appliance is or turn them off automatically [9]. Zhun et al. analysed the influence of occupant behaviour on building energy consumption and developed a technique for identifying energy-saving potential by changing user behaviour to reduce energy consumption; this is the opposite of the approach described in this paper, in which the building adapts to user behaviour [10].

Several recent studies - including the one described in this paper - have included data about occupants in the control schema for the thermostat set-point delegation. Mozer et al. implemented a neural network algorithm to predict occupancy, and expressed the comfort as a misery unit in dollars [11]. Scott et al. implemented an occupancy prediction algorithm, in which the occupancy was predicted for a 15-minute interval. The improvement of the new algorithm was compared with the predefined schedule, regarding energy consumption and comfort experience for individual rooms in an apartment [12]. Lu et al. [13] used occupancy sensors to obtain data regarding the behaviours patterns of the occupant, and also evaluated energy consumption and comfort experience where the comfort experience was evaluated as a miss time in minutes - that is, the total time during which the indoor temperature had not reached the set-point temperature within the tolerance of 1°C. Bapat et al. [14] used data about user occupancy to optimise the operation scheduling of home appliances, such as dishwashers and washing machines, in order to minimise the energy costs. Krumm et al. [15] used GPS data to predict human occupancy, but did not evaluate the effect of the algorithm results with any method for evaluating energy or comfort. Bayir et al. [16] developed a web-based service which, based on cell registration data, cell location changing patterns and training data, predicts location of the cell phone owner. Vrečko et al. [17] implemented the model predictive approach for calculating temperature rise time by heating, which is similar to the approach applied in the present study. In their approach, the data about occupancy is known in advance. These studies show that the energy consumption for heating, ventilation, and air-conditioning (HVAC) systems, DHW systems, and electric appliances can be significantly reduced by including data about occupant interaction with the environment into the control system architecture, such that the inhabitants' comfort experience improves or stays the same.

The occupant's comfort experience in an environment is an expression of how the occupant feels in the environment. The full comfort experience is composed of several aspects: visual comfort, acoustic comfort, indoor air quality, and thermal comfort [18]. In the present paper, the term *comfort* refers to thermal comfort. International definitions of thermal comfort are described in standards-defined by the International Organization for Standardization (ISO), the European Committee for Standardization (CEN), and the American Society of Heating, Refrigerating, and Air Conditioning Engineers (ASHRAE)-such as ISO EN 7730, CR 1752, or ASHRAE 55. The environmental parameters that constitute the thermal comfort are temperature (for example, outdoor air, indoor air, radiant), humidity, air velocity, and personal parameters (for example, clothing level, activity level) [19]. It is often assumed that environmental parameters are easier to obtain than parameters, that express human activities and behaviours. However, Lustrek et al. [20] and Kozina et al. [21], for example, presented a system that, based on recognised activities of a person using a smart phone, can estimate the energy expenditure expressed in Metabolic Equivalent of Task (MET) that can be transformed into activity level. However, the goal of our work was not to determine the appropriate values of environment set-point parameters, such as temperature, but rather to achieve a given set-point value at the appropriate time. For the evaluation of thermal comfort, we used the penalty approach, similar to that described in [3], where the differences between desired and actual temperatures were summarised over the simulation period, but only for the times at which the building was occupied.

For the study and analysis of system long-term behaviour, which requires between one month and one year of runtime data, the real-time experiment with real occupants in the real environment is desirable. For practical reasons, research scientists in the fields of BAS and BEMS sciences commonly use simulation techniques to evaluate the performance of control algorithms using a thermal dynamic model of a building - for example, [11, 22]. System simulation offers the following advantages in development compared to real tests: It is faster; it allows comparison between several control algorithms with the same weather situation and occupant behaviour; and the price of experiment is incomparably lower, which means that failed experiments have less negative impact on research budgets.

It is known from the control of dynamics systems theory that HVAC and DHW processes are (i) slow and (ii) time-delayed, which means that the system response is not achievable in the moment immediately after following actuation. Furthermore, these real systems are also non-linear, which affects the control system with additional uncertainties [23]. Therefore, a Model Predictive Control (MPC) approach is commonly used to predict the operation of the dynamic process. Our approach implements the approximation of rise time as a black box model, as defined in [24], where the black box model is a regression model used to predict the indoor temperature rise/fall time according to indoor temperature, outdoor temperature and difference between desired (set-point) temperature and indoor temperature.

Intelligent control of systems in smart buildings requires the collection and processing of large sets of heterogeneous data about sensor states, actuator actions, and occupant actions. The demand for HVAC and DHW systems varies between residential buildings and commercial or office buildings. Energy savings are easier to achieve in office buildings than in residential buildings due to higher predictability, and the violation of the thermal comfort experience is easier to determine. The proposed multi-agent architecture enables the simulation and prediction of occupant behaviour (for example, occupancy prediction) and of system behaviour (for example, time to achieve the desired room temperature). The obtained knowledge is used to create adaptive set-point schedules for heating in residential buildings. Further, the intelligent autonomous operation of individual agents and their ability to communicate with other agents enables automatic creation, optimization, and reconfiguration of the control schema for an individual device in a building, as well as for the whole building.

Control and regulation of individual subsystems, such as a single heating device, represent a matured research area with a wide range of scientific literature on control system design, and on several methods for implementing a particular controller for a specific individual subsystem. These methods often propose centralised approaches, resulting in complex and expensive computation methods for controller set-points delegation, especially for multiple-input/multiple-output (MIMO) systems. Operations such as algorithm implementation, complex behaviour model creation to be used by control, system organisation, user interfaces, data fusion, and other computationally expensive operations can be performed on spatially distributed computational entities, resulting in faster and more fault-tolerant operation. In a review of advanced methods for management of energy and comfort in the building environment, Dounis et al. [25] stated that the future control systems for building management will be intelligent and have a human-centric approach. Advanced control systems are composed of (i) a low-level feedback control part of each building's zone; and (ii) a high-level supervision and planning part, where the set-point schedules are generated and passed to the low-level part.

3. THE PROPOSED MACS ARCHITECTURE

The control architecture is implemented as a multi-agent control system (MACS), with several types of agents: sensor agents, control agents, routing agents, machine learning agents, and housekeeper agents. Layers in building automation and management are shown in Figure 1. The environment layer is located at the bottom and is composed of a building, representing the consumer in a smart grid; occupants, which interact with the building; and weather, which has an influence on the building dynamics. The physical layer is located above the environmental layer and is composed of physical entities, such as sensors and actuators, which are used to perceive the environmental states and to affect some of these states. The automation layer includes the physical controllers linking the physical layer and management layer. Controllers are used for closed-loop control, according to the algorithms, which are created and adapted by control agents in the management layer. When a simulation is used instead of a real environment, the environmental, physical, and automation layers are integrated in a simulation environment. The management layer is the top layer and represents a multi-agent system, composed of several types of agents. Each agent type must carry out appropriate tasks, and agent operations are specified as agent behaviours. During the runtime, each agent waits for the message and responds to it, meaning that the agent operation is data-driven or event-driven. Messages received by an agent can include information or a command. Routing agents are used to transform environmental states into a cloud of agent states. Routing agents are also denoted as interfaces between simulation environment and a control system. Sensor agents are used to obtain the sensor values denoted as simple states from physical sensors through routing agents and inferred values from various simple sensor states, denoted as complex states as a result of data fusion process, performed among several simple sensor states. Control agents represent virtual control entities that schedule actuator operation for controlled subsystems, such as room temperature. Control output is an input to the regulation feedback loop and delegates set-point values. Machine learning (ML) agents are used to create ML models when requested, using data prepared by sensor agents. Control agents use the ML models for control operations. A housekeeper agent, which is the top-level agent in a smart building, monitors and manages subordinate agents. The operation of each agent is explained in the following sub-sections.

3.1. Housekeeper agents

Housekeeper agents represent the highest instance in a building automation system. The housekeeper agent, which is used as an interface for the administration of the controlled system, is able to serve the information about the states of the environment and to manage the operation of individual control systems in the environment. The housekeeper agent also interacts with smart grid suppliers, such as a power plant or photovoltaic power plant, in order to provide data about current and future power demand. In this paper, the smart grid suppliers are denoted as a single city coordinator (CC) agent in the rest of the paper. We denote the i^{th} environment as H_i from a set of I environments $H_i \in \{H_1, H_2, ..., H_I\}$. For example, in the experimental set-up instantiation described in Section 6, H_1 is home and H_2 is workplace. Each environment has a corresponding housekeeper agent.

Definition 1. Housekeeper agent H_i is a tuple $\langle \mathcal{S}_i, \mathcal{C}_i, \mathcal{R}_i, CC \rangle$ where

- $\mathcal{S}_i$ is a set of all sensor agents in environment i
- $\mathcal{C}_i$ is a set of controller agents in environment i
- $\mathcal{R}_i$ is a set of routing agents in environment i (there is one routing agent only if an environment behaviour is simulated)
- CC is a city coordinator agent, which coordinates the negotiation for energy price between energy suppliers and consumers

3.2. Sensor agents

Knowledge about the operation of sensors and sensor systems is crucial for integration of sensing entities into a control system. Sensor agents represent the physical entity - that is, the sensor in an environment. The sensor state is the state of the environment variable (temperature, occupancy, power, etc.), which sensor agents can perceive and serve. There are roughly two types of sensor states: simple sensor states and complex sensor states. Simple sensor states are states that can be measured using standardised sensors and sensor systems with detailed information about sensing accuracy, drift, unit of sensed output, conversion factors, sampling frequency, and quantization for the digitalization process and other relevant information. Complex sensor states are inferred from simple sensor states, using expert knowledge for the sensing domain and machine learning methods used for estimation of sensor states, and consequently using additional processing and memory capabilities. The process by which information is extracted from several data sources is referred to as data fusion or multisensor data fusion [26]. We denote the complex sensor state as a sink of fusion process on simple sensor states. In addition to simple sensor states, the complex sensor states have meta-data about the accuracy of sensing algorithms and a set of sensor agents, which provide simple states available for data fusion. Sensor agents perform several tasks. First, they always maintain the most recent value of the sensor state. Second, they keep the history log of sensor states and generate data for machine learning. Finally, sensor agents are able to receive and process messages from other agents wishing to subscribe or stop receiving information about sensor states from routing agents, which provide the environmental data from physical sensors and can request that the sensor agent stop working.

Each sensor in an environment i is represented as a S_{ij}, where the first index i represents i^{th} environment and the index j represents the unique sensor number in that environment. Thus, an environment i has a set of J sensor agents $S_{ij} \in \{S_{i1}, S_{i2}, ..., S_{iJ}\}$.

Definition 2. Sensor agent S_{ij} is a tuple $\langle s, sm, \mathcal{C}, R \rangle$ where

- s is a state variable
- sm is meta-data, parsed from configuration file of that sensor agent
- $\mathcal{C}$ is a set of control agents, sensor agents and/or other agents, engaged with an agent S_{ij} for state variable value delivery, where a set $\mathcal{C}$ can include agents from any environment i
- R is a routing agent from set $\mathcal{R}_i$, which delivers the state variable from automation layer

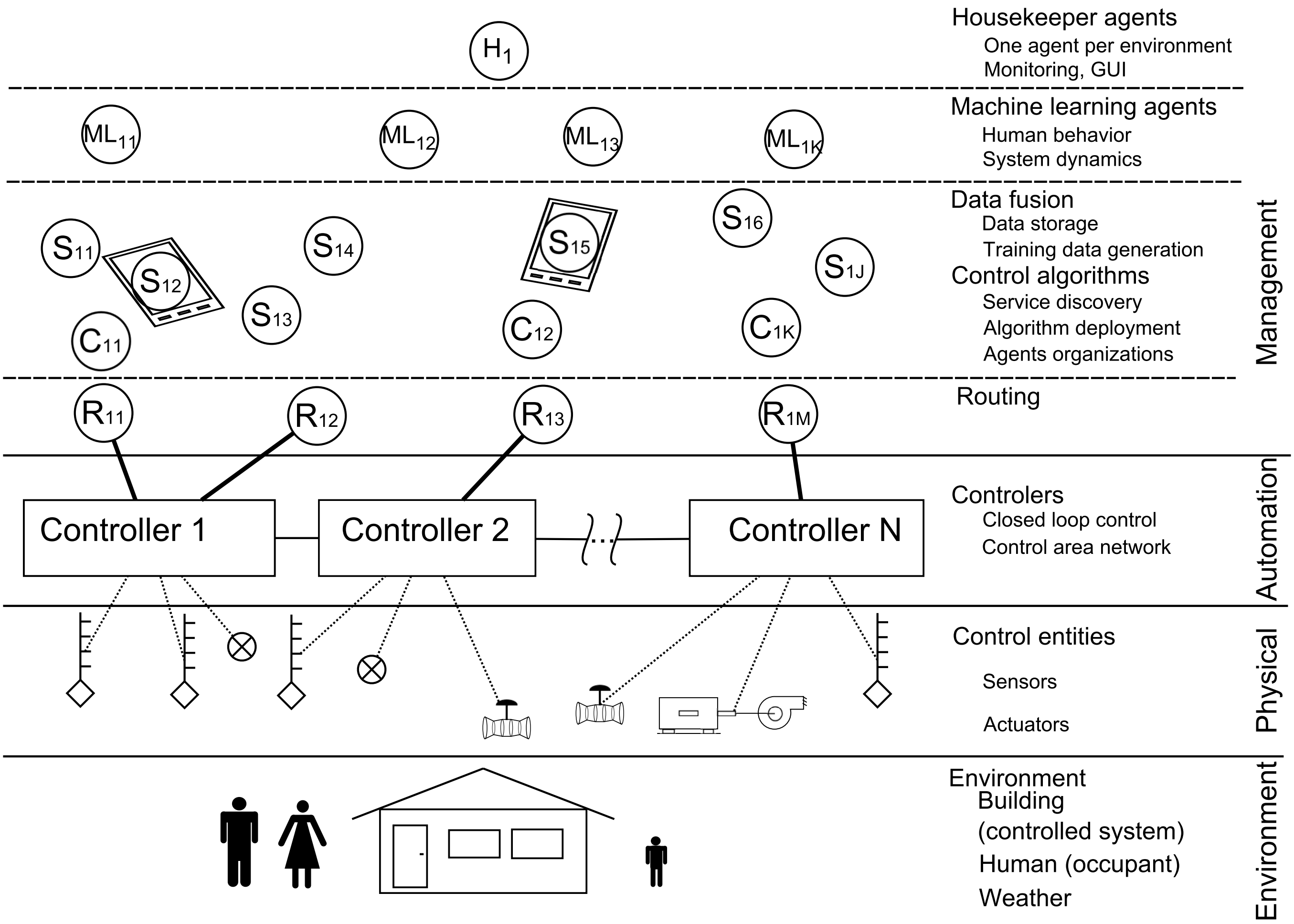


FIGURE 1. Layers in building automation and management

3.3. Control agents

Control agents represent software entities (threads), where the algorithms for defining set-point values are implemented as control behaviours. Control agents have a predefined set of control behaviours. There are simple behaviours, such as On and Off behaviours (where the set-point temperature is always set to a predefined high and low temperature respectively); Schedule behaviours (acts according to schedule instructions); Sense behaviours (acts according to sensed occupancy state, so that high set point temperature is applied if the building is occupied, low temperature is applied when the building is not occupied, and sleep temperature is applied when the occupant is sleeping); and complex behaviours, which exploit machine learning models to define the set-point values according to MPC method, such as control based on occupancy prediction in buildings and control based on identified behavioural patterns of heating dynamics of a building. Control behaviour can be a combination of predefined set of behaviours, such as Schedule behaviour in combination with MPC for predicting the time needed to heat a room.

The notation for control agents is similar to the notation of sensor agents; each control agent in an environment i is denoted as C_{ik} from a set of K control agents $C_{ik} \in \{C_{i1}, C_{i2}, ..., C_{iK}\}$.

When a control agent implements control behaviour, it first subscribes to the appropriate sensor agents for information delivery about the desired sensor states. If a control algorithm needs data about the outdoor temperature, indoor temperature, and occupancy, it engages with the appropriate sensor agents for information delivery. The control agent creates a control organization, which is composed of itself and appropriate sensor agents. If a control agent performs MPC, it will periodically activate a ML agent that, based on historic data, creates a ML model. Until the ML agent returns the new model, the old ML model is used. The control agent will use Sense behaviour if no model is available yet. The control agent passes the set-point values to the appropriate physical control entities in an automation layer through the routing agent, and saves set-point value in a control agent history file.

Definition 3. Control agent C_{ik} is a tuple $\langle c, cm, R, B \rangle$ where

- c is a set-point variable
- cm is meta-data, parsed from configuration file of that control agent
- R is a routing agent, which delivers the set-point value to the automation layer
- B is a tuple, which defines the type of control behaviour and sensor agents, which are needed for operation of that behaviour, defined according to Definition 4

Definition 4. Behaviour B is a tuple $\langle b, \mathcal{S}, ML\rangle$ where

- b is a control behaviour from a set of control behaviours, explained in Section 5
- $\mathcal{S}$ is a set of sensor agents from any environment i, which are needed for state variable values delivery for control behaviour implementation
- ML is a machine learning agent, used for machine learning model creation

3.4. ML agents

Machine learning agents are instantiated by control agents. When a control agent implements a model-predictive control, it employs a ML agent to perform machine learning and create a classification or regression model. There are two types of learning in the proposed architecture: (i) learning human behaviour and (ii) learning system dynamics (that is, identification). Data used for learning human behaviour is generated by sensor agents for occupancy at home or at work. Data used for learning system dynamics is created by control agents (such as temperature set-point values) and appropriate sensor agents (for example, temperatures achieved by changing set-point values) and are finally merged into a training dataset by the ML agent.

Each ML agent in environment i is denoted as ML_{il} from a set of L ML agents $ML_{il} \in \{ML_{i1}, ML_{i2}, ..., ML_{iL}\}$.

Definition 5. ML agent ML_{il} is a tuple $\langle C, \mathcal{S}, ml, tp\rangle$ where

- C is a control agent, which activated the ML agent
- $\mathcal{S}$ is a set of sensor agents, which store relevant data for the machine learning procedure
- ml is a machine learning algorithm, used for ML model creation
- tp is a machine learning model template, consisting of the list of attributes and a class value

3.5. Routing agents

Routing agents represent an interface between the multi-agent control architecture and the simulation environment or the real environment. Routing agents accept values that are obtained from a physical environment through the physical and automation layer and distribute these values to the appropriate sensor agents. On the other side, routing agents pass control parameters, variables, and schedules - computed by control agents - to the automation layer, where the new values are used as set-point values to perform feedback control in an automation layer.

Routing agents are used both for simulation purposes and for real operation purposes. In simulation, the routing agent creates and manages mapping from the simulation environment to the control system and vice versa. In real operation, the routing agent represents an interface between the controller and the control system. Each routing agent is denoted as R_{im} from a set of M Routing agents $R_{im} \in \{R_{i1}, R_{i2}, ..., R_{iM}\}$.

Definition 6. Routing agent R_{im} is a tuple $\langle \mathcal{C}, \mathcal{S}, cfg\rangle$ where

- $\mathcal{C}$ is a set of control agents, which compute set-point values
- $\mathcal{S}$ is a set of sensor agents, which provide state variable values
- cfg is a socket configuration of either the simulation environment or the controller in a physical environment

4. MACHINE LEARNING

Machine learning is an important procedure in MPC, since the model created as a result of machine learning algorithms can be used to predict the operation of a modelled process. In the present paper, the machine learning procedure was used to create a human behaviour model for predicting the occupancy of a building and creating a regression model to predict the time required to heat the room from the current temperature to the temperature denoted as the comfort temperature. All the machine learning procedures were implemented using the Weka data mining software [27].

4.1. User behaviour modelling for occupancy prediction

User behaviour modelling was performed using machine learning on real data about occupancy at work and at home over longer periods of time. The data about occupancy was transformed into a time series, with a frequency of one minute, where the occupancy state could have two values: 0, signifying absence; and 1, signifying presence. Based on that data, with a corresponding date-time value, the instance for machine learning $occInst$ can be created using the following list of attributes. There are two types of attribute values used: (i) unsigned integer values; and (ii) nominal values, which are defined in parentheses.

- Minute in a day (unsig. integer)
- Day in a week (1,2,...7)
- Weekend day (0,1): 1 for Saturday and Sunday, 0 otherwise

- Month in a year (1,2,...,12)
- a.m./p.m. (0,1): 1 represents time before midday, 0 represents time after midday
- Number of days before Saturday (0,1,2,3,4,5)
- The elapsed time in minutes since the occupant left home (unsig. integer)
- Number of occurrences in a day that the occupant left home (unsig. integer)
- Occupancy at home exactly one week earlier (0,1)
- A daily sum of minutes that the person was at home (unsig. integer)
- The elapsed time in minutes since the occupant left work environment (unsig. integer)
- Occupancy at work exactly one week earlier (0,1)
- A daily sum of minutes that the person was at work (unsig. integer)
- Occupancy at work (0,1): 1 represents presence and 0 represents absence at work
- Occupancy at home(0,1): class value to be predicted, 1 represents presence and 0 represents absence at home.

For user behaviour modelling, the following three agents are needed: Sensor agent S_{16} representing occupancy at home; sensor agent S_{21} representing occupancy at work; and a ML agent ML_{13}. Note that the agent identifiers used in that paragraph are taken from experimental set-up instantiation, described in Section 6. S_{16} and S_{21} are logging occupancy values. When the ML_{13} agent receives a request to create an occupancy model, it reads log files of sensors S_{16} and S_{21} and merges them into a training dataset, using the attributes listed above. After the training dataset is created, the ML_{13} agent performs machine learning using one of the following algorithms: C4.5 decision trees, RandomForest, or K nearest neighbour algorithm with $K = 5$ nearest neighbours. Algorithms C4.5 and kNN are implemented in two variants: (i) C4.5 and kNN, representing the classification for nominal class value as a result, which can be either 0 or 1; and (ii) the C4.5_dist and kNN_dist, representing the distribution for class value 1, which is a continuous value between 0 and 1. Other parameters remain at default values as defined in Weka. When the classification model is created, it can be used by control agents C_{11} and C_{12} in learning behaviour.

The occupancy prediction model is created upon request by the control agent, using past occupancy instances *occInst*. This study involved performing an incremental evaluation of the occupancy prediction model using the following approach: at the end of each day, the classification model was created based on historical data from previous days, and tested on the next day. After that next day, the new training instances were added to the set of previous training instances, and so on. From those testing instances, we removed the instances representing the moment at which the person was present at home for more than one hour, and marked these as non-interesting instances for testing. This data was removed because, if the occupancy happened, and the prediction model did not forecast it early enough, the control agent would perform the reactive component of learning behaviour, which operates as a Sense behaviour. In sum, near-future human occupancy forecasting in a building is interesting only when the building is not yet occupied. Occupancy prediction model was evaluated for algorithms C4.5, kNN, and RandomForest, where the output of prediction model is a nominal class value 0 or 1.

4.2. Identification of room heating and cooling dynamics

Room heating and cooling dynamics were also identified using machine learning on historical data trends during simulation runtime. Each instance for machine learning consists of the following numeric attributes, given as rational numbers:

- Outdoor temperature (numeric) in $°C$
- Indoor temperature (numeric) in $°C$
- Set-point temperature (numeric) in $°C$
- Rise time coefficient (numeric) in $minutes/°C$, which represents class value.

Rise time coefficient (K_{RT}) is a linearised factor, because we assume that the indoor temperature rises (falls) in a linear manner when heating/cooling. K_{RT} is calculated using:

$$K_{RT} = (T_{sHi} - T_{in})/t_{RT} \tag{1}$$

where T_{sHi} is the desired comfort temperature, T_{in} is the indoor temperature and t_{RT} is the time needed to heat or cool the room from T_{in} to $T_{in} = T_{sHi}$. Figure 2 shows the process of attribute extraction to create an instance from historical data trends of sensor states and states of set-point values. Here, the instance *rtInst* is created using the itemised attributes above.

For rise time regression modelling, the following three agents are needed: Sensor agent S_{11}, representing the indoor temperature; sensor agent S_{12}, representing the outdoor temperature; and control agent C_{11}, representing the set-point temperature value. The process of creating the regression model is similar to that in the previous subsection, except that the ML agents ML_{11} or ML_{12} are employed for creating the regression model for rise time (used by heating) and fall time (used by cooling), respectively. Fall time differs from rise time only by detecting the negative set-point temperature change instead of positive set-point temperature change, as depicted in Figure 2.

Regression models are built during runtime after each change of set-point temperature, when the new instance *rtInst* was created. From the simulation run, we obtained M true rise time t_{RTT} values and the

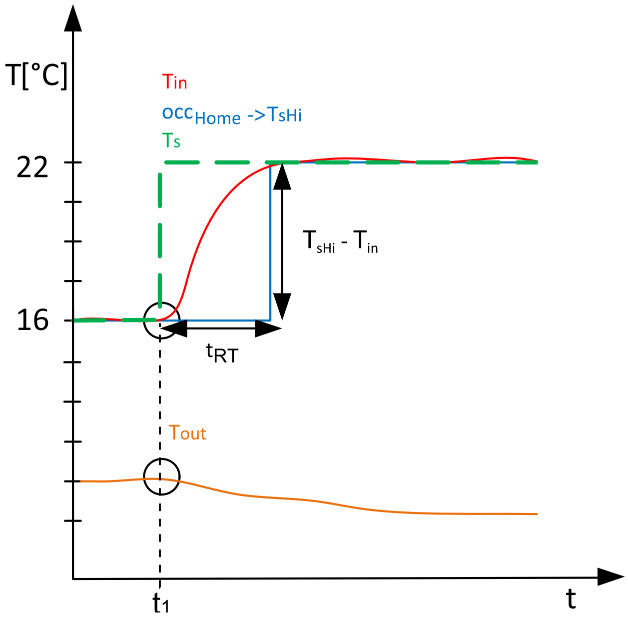


FIGURE 2. Attribute extraction for training data for rise time regression model creation

corresponding M predicted rise time t_{RTP} values. We then computed the Mean Absolute Percentage Error (MAPE), defined as:

$$MAPE = \frac{100\%}{M} \sum_{i=1}^{M} \left| \frac{t_{RTT_i} - t_{RTP_i}}{t_{RTT_i}} \right| \quad (2)$$

Results of rise/fall time regression models are shown in Section 7.1.

5. CONTROL BEHAVIOUR

Control algorithms are implemented as control behaviours of control agents, which affect the appropriate control entity as defined in Definition 3 and 4. Control agents read temperature set-point parameters, which are T_{sHi}, T_{sLo} and T_{sSleep}. T_{sHi} represents the desired indoor temperature at which the occupant feels comfortable at home. T_{sLo} value represents the temperature that should be used when the occupant is not present at home. T_{sLo} value is used to reduce energy consumption for heating or cooling. When the outdoor temperature is low and heating is necessary, T_{sHi} value is higher than T_{sLo}. When the outdoor temperature is high and cooling is necessary, T_{sHi} value is lower than T_{sLo}. T_{sSleep} value represents the temperature at which the occupant feels comfortable when sleeping.

We have defined five control behaviours for control agents: On, Off, Schedule, Sense and Learning behaviour (explained in subsections 5.1-5.4). During the initialisation phase of a control behaviour, an agent searches for the appropriate sensor agents, that are needed for implementation of depicted behaviour. The following subsections describe the operation of control behaviours.

5.1. On and Off behaviour

On and Off behaviours are the most trivial control behaviours, since the set-point temperatures are both static (see Table 1). On behaviour is the most wasteful but the most comfortable. In contrast, Off behaviour is the most economical but the least comfortable. The two behaviours are used to obtain extreme points in energy consumption and in comfort experience. There is only one set-point value in each of the two behaviours: T_{sHi} for On behaviour and T_{sLo} for Off behaviour.

5.2. Schedule behaviour

Schedule behaviour represents the most widely used control type in building automation systems. Schedule behaviour works according to a predefined schedule for each day in a week, where the predefined set-point value is applied for each minute in a day. Control operation according to schedule is often used in conventional control systems.

Schedule is defined as Sch_{ik}, which represents the schedule used by control agent C_{ik} for each minute in a week. The weekly schedule is composed of daily schedules, by which each day in a week can have a unique schedule. Schedule behaviour was not used for comparisons in this paper, although it is implemented and prepared for usage within the proposed control architecture.

5.3. Sense behaviour

Sense behaviour is an example of reactive behaviour. Sense behaviour uses rules equipped with sensor states to change set-point values over time. One example of Sense behaviour by heating is a scenario in which the heater ensures a high set-point temperature when the home is occupied and the occupant is awake, a sleep set-point temperature when the home is occupied and the occupant is sleeping, and a low set-point temperature when the home is not occupied.

Sense behaviour initialisation includes sensor agents, which provide sensor states during runtime, and rules that define the impact of sensor states on the set-point calculation.

5.4. Learning behaviour

Learning behaviour represents a control algorithm that adapts during its runtime according to historical observations of human and system behaviour. That behaviour is composed of anticipative and reactive components, as proposed by Abras et al. [28]. However, the anticipative component in the present study used machine learning methods to extract knowledge about human behaviour. Another anticipative component uses machine learning methods to perform system dynamics identification for MPC. The anticipative component is used to forecast occupant behaviour and the dynamics of heating or cooling processes in a building. The reactive component of a behaviour is similar to either Sense or Schedule behaviour, which - in the case of violation of some constraints - applies one of the behaviours explained in the previous two subsections.

Algorithm 1 presents an implementation of the behaviour for a room heating example, based on the following environmental state information: outdoor temperature (T_{out}), indoor temperature (T_{in}), occupancy at home (occ_{home}), and occupancy at work (occ_{work}). These values are obtained for each simulation time-step from appropriate sensor agents. Here the occ_{home} and occ_{work} variables represent home and work occupancy state for one person. Algorithm outputs the room set-point temperature value T_s, which is then passed to the controller in the automation level to achieve the desired indoor temperature.

Algorithm 1 Control agent implementing Learning control behaviour

```
occ_Home ←
occ_Work ←
T_in ←
T_out ←
if (occ_Home == present) then
    T_s ← T_sHi
else if (occ_Home == sleeping) then
    T_s ← T_sSleep
else
    Create rtInst                              ▷ see Section 4.2
    K_RT ← classify(rtInst, RtClassifier)
    t_RT ← (T_sHi − T_in)/K_RT                 ▷ see eq:1
    Create occInst                             ▷ see Section 4.1
    occAtRt ← classify(occInst, OccClassifier)
    T_s ← T_sLo + occAtRt ∗ (T_sHi − T_sLo)
end if
```

Figure 3 shows an example in which the T_s was scaled according to the distribution for future home occupancy, using the Isotonic Regression model to predict rise time and kNN_dist classifier for occupancy prediction. It is clear that the occupancy was predicted approximately one hour earlier than it really happened. In such examples, the system consumes more energy than the Sense behaviour would. On the other hand, the comfort experience is better, because the occupant enters home with already comfortable temperature, which is not a case with Sense behaviour. The corresponding lower figure presents the power rates of the chiller P_c, P_h and P_{sh}.

5.5. Evaluation of control behaviour

To evaluate comfort, we defined a discomfort index IC, which computes the comfort experience for the simulation run as follows:

$$IC = \sum_{t=0}^{n} sign(T_{sHi} - T_{in}(t)) * |T_{sHi} - T_{in}(t)| * occ_{Home}(t)) \quad (3)$$

where n is the total number of time steps in the simulation (number of minutes); T_{sHi} is high set-point temperature, which represents comfort temperature; and $T_{in}(t)$ is the indoor temperature in time step t. By heating, function $sign(x)$ returns 1 for $x > 0$ and 0 for $x \leq 0$. By cooling, function $sign(x)$ returns 0 for $x \geq 0$ and 1 for $x < 0$. Function $sign(x)$ is used to exclude the time steps, where the indoor temperature is at least as high as the set-point temperature used for heating, and at least as low as the set-point temperature used for cooling. The discomfort index represents the sum of per-minute differences of the actual temperature from the desired temperature - meaning the smaller the index, the greater the comfort.

To evaluate energy consumption, the power rate is integrated over time, according to the following equation:

$$E = \sum_{t=0}^{n} P(t) * \Delta t \quad (4)$$

where $P(t)$ is the power rate of heating and cooling appliances, and Δt is the duration of the simulation time step.

We used the Sense behaviour as a baseline for evaluating the Learning behaviour. To compare different Learning algorithms, we devise the index

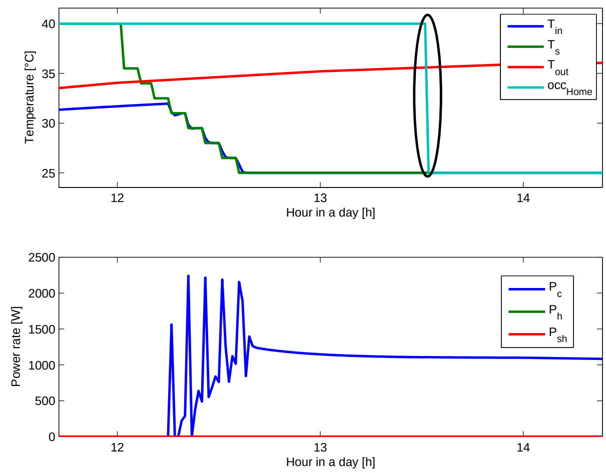


FIGURE 3. An example in which the T_s was scaled according to the probability for occupancy at t_{RT}. The figure above shows T_{in}, T_{out}, T_s and occ_{home}, where occ_{home} is transformed to set-point temperature units (25°C for presence, 40°C for absence) for better representation. The black ellipse highlights the moment at which the building becomes occupied. The lower part of the figure shows the corresponding power rates P_c, P_h and P_{sh} of the chiller, heater, and supplementing heater, respectively.

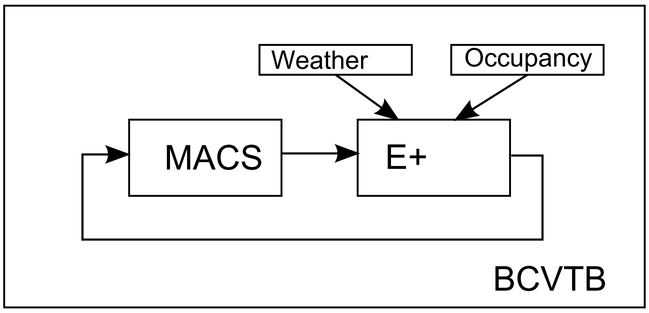


FIGURE 4. Coupling of MACS with the simulation model

$D_{Learning}$:

$$D_{Learning} = \frac{IC_{Sense} - IC_{Learning}}{E_{Learning} - E_{Sense}} \frac{E_{Sense}}{IC_{Sense}} \tag{5}$$

where IC and E are given according to 3 and 4, respectively. The $D_{Learning}$ index represents the ratio between the normalised decrease in discomfort (which should be as large as possible) and the normalised increase in energy consumption (which should be as small as possible). So that the greater the index, the better the algorithm implemented in Learning behaviour.

6. EXPERIMENTAL SET-UP

The experimental set-up is shown in Figure 4. Building Controls Virtual Test Bed (BCVTB) is a free co-simulation software that can be used for coupling of different simulation modules [29]. BCVTB was used to couple a simulation model (represented as E+ block) of a building with the Multi-Agent Control System (represented as MACS block). The weather data, data about occupancy at home and occupancy at work, and set-point values - computed by agents in MACS - are all inputs to the E+ model. The E+ model outputs environmental states, which are forwarded to MACS.

6.1. Simulation model of a building

Experimental set-up included the selection of the model of packaged terminal heat pump, which comes with the installation of the EnergyPlus [30] software. The building is represented as a single-floor, L-shaped structure with three heating zones. Only the west zone, representing a floor area of approximately 36m^2, was considered for indoor temperature control experiments. The cooling coil total cooling capacity was rated at 3700W and COP 3.00. The coil performance was defined using total cooling capacity function of temperature, total cooling capacity function of flow fraction, energy input ratio function of temperature, energy input ratio function of flow fraction and part load fraction correlation curve. The heating coil total heating capacity was rated at 3700W and COP at 2.75. The coil performance was defined using heating capacity function of temperature curve, heating capacity function of flow fraction curve, energy input ratio function of temperature curve, energy input ratio function of flow fraction curve and part load fraction correlation curve. The coefficients for both heating and cooling functions remain default as defined in obtained model. The power rate of each coil is dynamic and depends on the dry-bulb temperature, wet-bulb temperature and air flow rate. The location of the simulation was Ljubljana, Slovenia, so the corresponding weather file for that city was used for the experiment.

6.2. MACS

The control system was instantiated according to the proposed Multi-Agent Control Architecture implemented in Java Agent Development Environment (JADE) [31]. JADE is a software framework for developing agent applications for interoperable intelligent multi-agent systems. For experimental set-up, MACS was deployed using the following sets of agents:

- Housekeeper Agents
  $H_i \in \{H_1, H_2\}$.
  1. Building Home
  2. Building Work
- Control Agents
  $C_{1j} \in \{C_{11}, C_{12}\}$.
  1. Heater
  2. Chiller
- Sensor Agents
  $S_{1j} \in \{S_{11}, S_{12}, S_{13}, S_{14}, S_{15}, S_{16}\}$
  1. Indoor temperature
  2. Outdoor temperature
  3. Heater power rate
  4. Supplementary heater power rate
  5. Chiller power rate
  6. Occupancy in Building 1 (Representing occupancy sensor at home)
- Sensor Agent
  $S_{2j} \in \{S_{21}\}$
  1. Occupancy in Building 2 (Representing occupancy sensor at work)
- ML Agents
  $ML_{1l} \in \{ML_{11}, ML_{12}, ML_{13}\}$
  1. ML Heater for t_{RT} regression modelling
  2. ML Heater for t_{FT} regression modelling
  3. ML Occupancy Building 1
- Routing Agent
  $R_{1m} \in \{R_{11}\}$
  1. Routing BCVTB-MACS

C	T_{sHi} [°C]	T_{sLo} [°C]	T_{sSleep} [°C]
C_{11}	21.5	5	18.5
C_{12}	25.0	40	26

TABLE 1: Controller set-point parameters

Person	Period	Number of days
Person 1	22.07.13 - 25.11.13	126
Person 2	02.08.13 - 20.11.13	110
Person 3	21.07.13 - 30.08.13	40
Person 4	01.08.13 - 19.10.13	79
Person 5	01.09.13 - 31.10.13	60

TABLE 2: Human behaviour datasets used for evaluation

For the experiment, the set-point parameters for C_{11} and C_{12} are listed in Table 1. There are three set-point values: T_{sHi}, T_{sLo}, and T_{sSleep}. T_{sHi} and T_{sSleep} values are denoted as the indoor temperature values at which the occupant feels comfortable during presence and sleeping, respectively. Each control agent implemented either On, Off, Sense, or Learning control behaviour. When a control agent performed a Learning control behaviour, the agent created and launched the appropriate ML agent, which was used to create classification or regression models, as described in Section 4. Classification models for occupancy prediction were created with $C4.5_{dist}$, $C4.5$, $RandomForest$, kNN, and kNN_{dist} (where $k = 5$) algorithms. Regression models for rise time forecasting were created using Multilayer Perceptron, Gaussian Processes, and Isotonic Regression. Results of the models are presented in Section 7.

6.3. Human behaviour dataset

To obtain the human behaviour dataset, we chose five volunteers to record date-time stamp of the start and stop times of certain activities - sleeping, home occupancy, and work occupancy - over a period of at least one month. The data about activities at home was logged manually using a smart phone logger, and the data about occupancy at work was logged utilizing the RFID system installed at our research institute. According to Section 3.2, we consider the occupancy state to be a complex state, because the result of occupancy detection or activity recognition is not necessarily the result of a single simple sensor; rather, it requires additional processing of data from various sources. The scope of our work did not include processes of occupancy detection, person identification, estimation of the number of occupants in a building, or activity recognition to recognise sleeping activity. In the experiment, we assumed that only one person uses an apartment. The data was transformed into the time series, with one-minute time stamps. Table 2 presents the details of the dataset for each person.

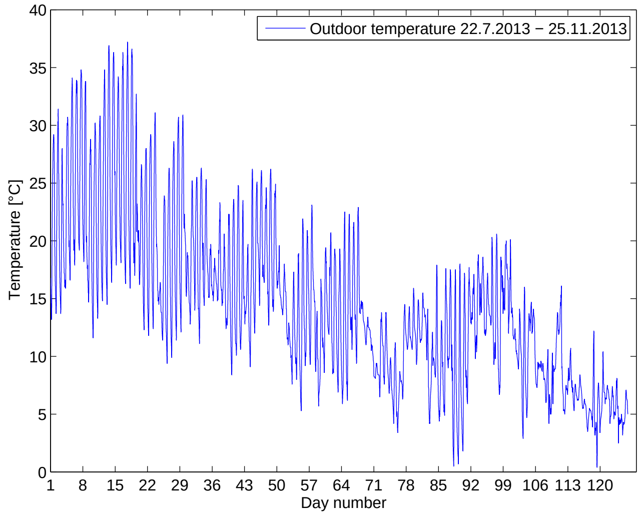


FIGURE 5. Outdoor temperature for weather station Ljubljana, Slovenia between 22 July 2013 and 25 September, 2013.

6.4. Weather data

Weather data was obtained from the Slovenian Environment Agency[3]. Figure 5 shows the outdoor temperature for Ljubljana, Slovenia for the period between 22 July, 2013 and 25 November, 2013.

7. RESULTS AND DISCUSSIONS

Our evaluation of the operation of the control system was based on the quality of rise/fall time regression models (using algorithms Gaussian Processes, Multilayered Perceptron and Isotonic Regression), occupancy prediction models (using algorithms $C4.5$, $RandomForest$, and kNN), and the evaluation of energy consumption with respect to an occupant's comfort experience (using algorithms On, Off, Sense, and all variants of the Learning algorithm). The regression model was always conducted within one second. The creation of the occupancy prediction models took approximately $0.1s$, $2.9s$, and $6.8s$ for kNN, $C4.5$, and $RandomForest$, respectively, to conduct 178560 training instances representing the training data for the past 124 days. For experiments, the control agent always waits until the ML agent create new models. For real-time implementation, the control agent uses an old model until the new one is created. The following subsections present the results obtained for each individual problem and a discussion about assumptions used and limitations encountered in our experiment.

7.1. Rise time forecasting

Regression models for rise time forecasting were evaluated using MAPE according to (2). The instances

[3]Slovenian Environment Agency, Web page: http://www.arso.gov.si/en/

	Rise time	Fall time
Gaussian Processes	36.0	24.7
Isotonic Regression	**29.3**	**19.7**
Multilayer Perceptron	37.0	24.5

TABLE 3: MAPE for each of the regression algorithms for rise time and fall time. Best values are bolded.

Person	No. arrivals	Early forecasts (%)		
		C4.5	RF	kNN
Person 1	126	**62 (49)**	59 (47)	59 (47)
Person 2	78	43 (55)	**44 (56)**	33 (42)
Person 3	33	**17 (52)**	15 (45)	14 (42)
Person 4	37	**15 (40)**	14 (38)	14 (38)
Person 5	45	**30 (67)**	26 (58)	24 (53)

TABLE 4: Occupancy prediction results. Values in bold indicate the best result for each person.

were obtained from the simulation of Sense control behaviour for Person 1. In 126 days, there were 64 instances for rise time and 166 instances for fall time. Table 3 presents the MAPE for each of the regression algorithms for rise time and fall time. According to these evaluations, we decided to choose the Isotonic Regression algorithm for the further simulations, because it has the lowest MAPE for both rise time and fall time prediction models - 29.3% and 19.7%, respectively.

7.2. Occupancy prediction

Occupancy prediction was evaluated before simulating the heating process. Results are shown in Figure 6. For that evaluation, we used only the $C4.5$, kNN, and $RandomForest$ classification algorithms. The interesting parts of each sub-figure are indicated by the green circles, which represent the transition from absent to present occupancy state. The best transitions are blue → light-blue, signifying that the classifier made an exact prediction and that there were no misclassified instances. Also of interest are the transitions from blue → red → light-blue, signifying the occupancy was predicted too early. In such cases, Learning behaviour turns on heating or cooling before the occupant's arrival, thereby improving the comfort experience but increasing energy consumption. $C4.5$ algorithm made the most early forecasts, as shown in Table 4. For four of the five people, the $C4.5$ algorithm predicted occupancy accurately or slightly early.

7.3. Energy and (dis)comfort evaluation

The simulation model of a building was used to simulate energy consumption and comfort. The data about occupancy was used for each person, and the following control behaviours were used to control the heating and cooling processes:

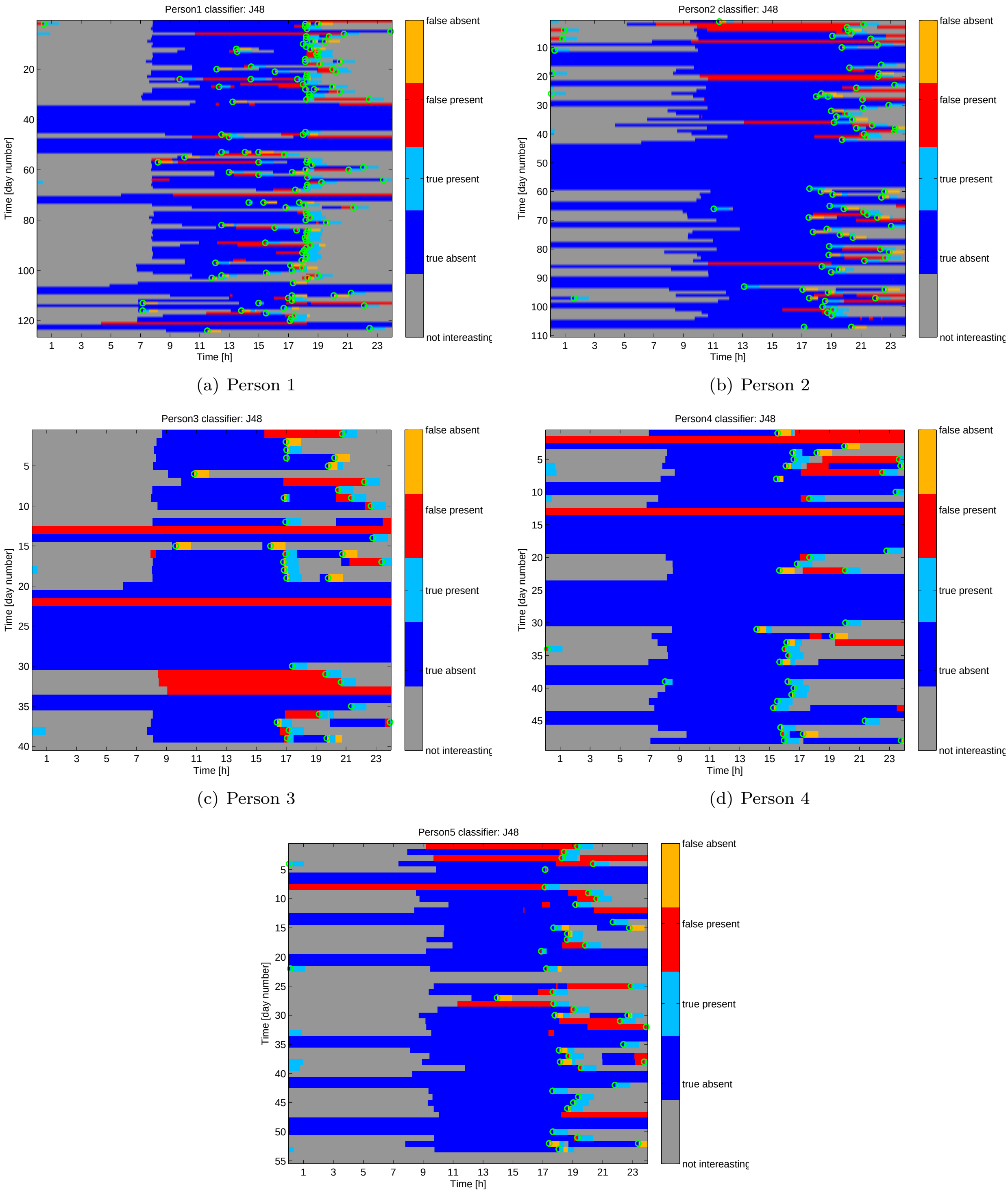
$D_{Learning}$	Classification			Distribution	
	$C4.5$	RF	kNN	$C4.5\_dist$	kNN_dist
Person 1	**372.42**	0.99	9.15	1.93	9.15
Person 2	**818.67**	0.68	0.66	3.78	6.01
Person 3	**136.78**	1.24	0.92	0.37	3.58
Person 4	**151.44**	0.80	0.68	2.20	1.75
Person 5	**357.97**	0.79	0.75	0.95	28.85

TABLE 5: D index representing comfort vs. energy improvement of different versions of Learning behaviours with respect to Sense behaviour. Values in bold indicates the best result for each person.

- On
- Off
- Learning with $C4.5$, $C4.5\_dist$, $RandomForest$, kNN, and kNN_dist where $k = 5$
- Sense

Results are shown in Figure 7. Each sub-figure presents the results for simulation control behaviours. As expected, On behaviour results in the highest energy consumption and the best comfort experience, while Off behaviour results in the lowest energy consumption and the worst comfort experience. The red line indicates the Pareto front, which links the non-dominated points on the chart. Non-dominated points are those that are not worse than any other point, in terms of both comfort experience and energy consumption. For Learning behaviour, the most promising results were obtained using the Learning_$C4.5$ classification algorithm for occupancy prediction, if the user prefers to spend less energy, or an algorithm like the kNN algorithm (both variants) or RandomForest algorithm, if the user prefers a higher comfort experience. Note that the Y axis, which represents the value of the D index, uses a logarithmic scale for better visualisation. In practical terms, Person 5 spent 2.43 GJ of energy for heating and cooling and achieved an ID of 577 using Sense behaviour. When using the Learning_kNN_dist algorithm, he spent 2.45 GJ (0.02 GJ more) of energy and achieved the ID of 440 (133 less). In other words, Person 5 spent 0.8 percent more energy and achieved 23.7 percent better comfort experience, which results in D_{kNN_dist}=28.85.

Improvements of the control agent's Learning behaviour with respect to Sense behaviour are shown in Table 5. If the D index is higher than 1, then the relative decrease of IC is higher than the relative increase of E. If the value of IC equals 1, the relative increase of E is the same as the relative decrease of IC. Values of D index lower than 1 show that much more energy is required for a relatively small improvement in comfort. We intend to achieve D values as high as possible. In our experiments, the Learning algorithm always provided greater comfort at the expense of an increase in energy consumption.



(a) Person 1

(b) Person 2

(c) Person 3

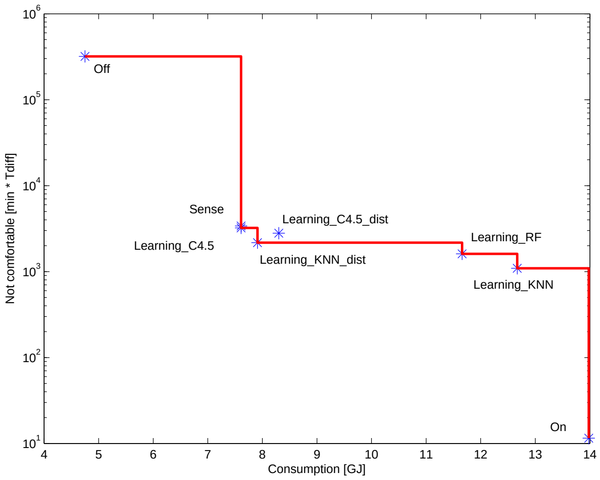
(d) Person 4

(e) Person 5

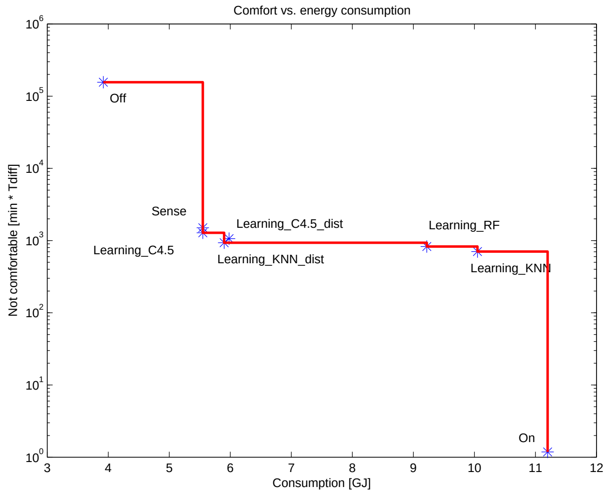
FIGURE 6. Occupancy prediction results for five people. The X axis shows the time of day with one-minute periods. The Y axis shows the day numbers from the start of simulation. The colours orange, red, light blue, dark blue, and grey represent the occupancy prediction results as false absent, false present, true present, true absent, and not interesting periods, respectively. Green circles represent the transition from the real absent occupancy state to the real present occupancy state.

7.4. Assumptions and limitations

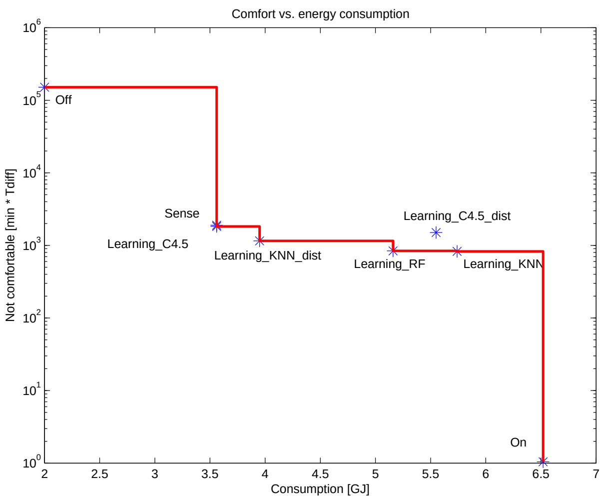
Activity recognition, occupancy detection, and person identification constitute a wide research area, in which the trade-offs between the unobtrusiveness, accuracy of results, complexity of installation and price of equipment are of central importance. In our experiment, we assume that there is one person included



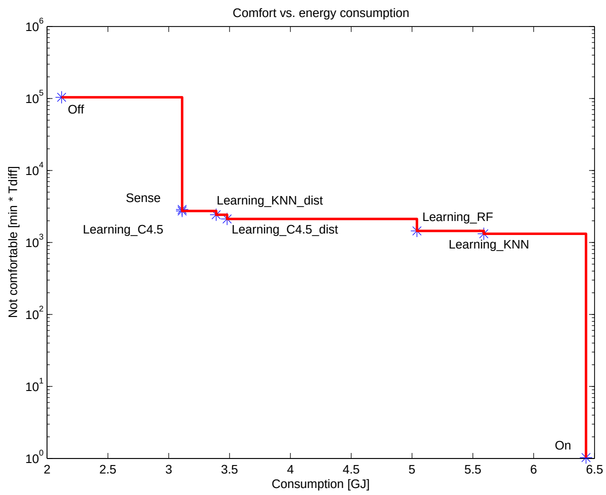
(a) Person 1



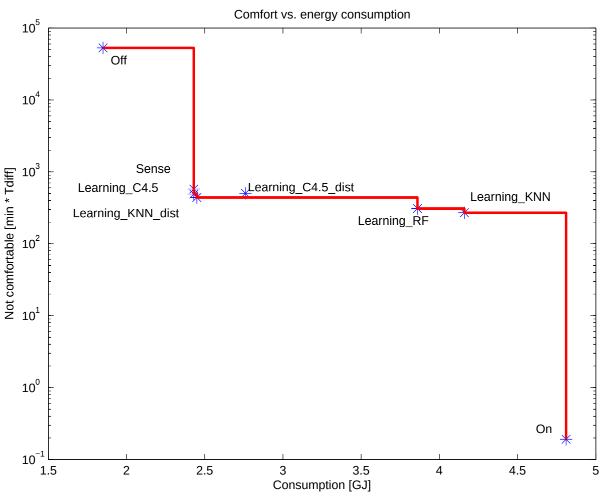
(b) Person 2



(c) Person 3



(d) Person 4



(e) Person 5

FIGURE 7. Pareto curve for five people, where the X axis presents electric energy consumption by heat pump for heating and cooling, for the whole simulation period in GJ. The Y axis presents the index *IC* for the whole simulation period. Each blue star on a figure presents an occupancy prediction algorithm. The red line presents the Pareto front, which connects non-dominated points on the graph.

as a source of human behaviour data for the control system. If we assume more people, the occupancy state is similar to the occupancy state of one person. In our experiment, one conditioned zone represents the whole apartment. A problem arises if we assume two people in the same conditioned zone, where one person is sleeping and another person is awake, watching TV. In such circumstances, the mean value of the T_{sHi} and the T_{sLo} could be used as a set-point value. The system assigned to recognise the activities of each person in an apartment should be able to distinguish between people and consider the appropriate set-point values given by the negotiation process, as Kwak et al. [5] did for commercial buildings.

The indoor set-point temperature parameters at which the occupant feels comfortable are fixed values, as shown in Table 1. The thermal comfort experience for one simulation is expressed as a sum of minutes, multiplied by the difference between the current temperature and the comfortable temperature for each time step, during which the building is occupied. Ciegler et al. [32] performed the regulation of PMV by adapting the indoor and radiant temperatures, but they also made certain assumptions about variables that are dynamic in reality, such as fixed values for relative humidity, or fixed schedules for the occupant's clothing insulation and activity level. To summarise, we assumed the T_{sHi} as a static value for the temperature at which one or many people in an apartment feel comfortable while awake and T_{sSleep} while sleeping.

Nguyen et al. [33] conducted a survey of systems that are used to perceive the activity of the user at home. Many approaches for detecting occupancy, identifying the occupant, or recognizing activities at home were already examined including a Radio Frequency Identification (RFID) system, video camera and/or microphone supervisory system, move detectors, door opening sensor, pressure sensors in floor, smart phone in relation to wireless home network, and other systems. Before the installation of such a system, it is worth considering the issues of privacy, the accuracy of such system, the price, the complexity of installation, reliability, and other aspects. Lu et al. [13] used a combination of passive infrared motion (PIR) sensors installed in rooms and magnetic reed switches on doors to detect occupancy and sleeping - this system is cheap, unobtrusive, and simple to install. The data about occupancy at work was obtained using the RFID system, installed at our institute. The data about home occupancy was obtained manually; the volunteer used a smart phone application logger to enter the data for the occupancy and sleeping events. Presumably, we could obtain similar results when using an installation, as did Lu et al. [13]. It is worth mentioning, that the set-point delegation highly depends on the data provided by sensor agents. If the states are wrongly perceived, the control operation will consequently result in either excessive energy consumption or comfort loss. The set-point values could be changed manually during operation, because the occupant has the highest priority in set-point delegation.

8. CONCLUSIONS AND FUTURE WORK

This paper presents the Multi-agent control architecture, composed of several types of agents used to manage processes in a smart building. Each agent type has its own duties and is related to other agents in an architecture. The proposed formalization of agent instances enables instantiation of a control system for various types of heating systems, thereby demonstrating the modularity of our system. We present the control algorithm for heating and cooling set-point delegation, which is implemented as a control behaviour of an agent. The proposed control algorithm uses machine learning methods to identify the dynamic processes in buildings and to learn human behaviour. The machine learning algorithms were instantiated for heating and cooling processes and used for MPC control to decrease energy consumption. Furthermore, the occupancy prediction of an occupant was performed to minimise energy consumption and to increase energy savings. Both types of learning were evaluated using five datasets, obtained from volunteers and through simulation of the heating and cooling process of a building, using a packaged terminal heat pump as the system for heating and cooling. In many cases, the proposed Learning behaviour of an agent decreased discomfort (improved comfort) significantly while slightly increasing the energy consumption with respect to Sense behaviour. The algorithm-provided prediction of the need for heating and cooling can also be passed on to the smart grid, allowing it to anticipate the energy needs of the building.

Areas of further investigation include the multiple-occupant problem, in which the set-point values differs for each occupant. Furthermore, the multiple-zone conditioning problem should also be explored, as it presents the opportunity to further reduce energy consumption by conditioning only occupied rooms. The addition of various models representing energy suppliers and energy storage systems in cooperation with other heating systems - such as air-to-water or water-to-water heat pump - is needed in order to enrich results obtained with the learning algorithm. We already develop control algorithms that take into account energy availability and price provided dynamically by the smart grid. Finally, real-time evaluation using a real building is already being planned where the impact of our control architecture will be evaluated using physical sensors and actuators.

ACKNOWLEDGEMENTS

Operation partly financed by the European Union, European Social Fund. Operation implemented in the framework of the Operational Programme for Human Resources Development for the Period 2007-2013, Priority axis 1: Promoting entrepreneurship and adaptability, Main type of activity 1.1.: Experts and researchers for competitive enterprises.

REFERENCES

[1] BPIE (2011) Europe's buildings under the microscope: Scenarios and strategies to 2050. Technical report. Buildings Performance Institute Europe, Brussels, Belgium.

[2] IEA (2010) Energy technology perspectives 2010: Scenarios and strategies to 2050. Technical report. International Energy Agency, Paris, France.

[3] Moroşan, P.-D., Bourdais, R., Dumur, D., and Buisson, J. (2010) Building temperature regulation using a distributed model predictive control. *Energy and Buildings*, **42**, 1445–1452.

[4] Zupančič, D., Luštrek, M., and Gams, M. (2013) A Network of Sensor and Actuator Agents for Building Automation Systems. In Bosse, T., Cook, D. J., Neerincx, M., and Sadri, F. (eds.), *Human Aspects in Ambient Intelligence*. Atlantis Press, Paris, France.

[5] Kwak, J.-y., Varakantham, P., Maheswaran, R., Tambe, M., Jazizadeh, F., Kavulya, G., Klein, L., Becerik-Gerber, B., Hayes, T., and Wood, W. (2012) Saves: A sustainable multiagent application to conserve building energy considering occupants. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, Valencia, Spain, 4-8 June, pp. 21–28. IFAAMAS, Richland, SC.

[6] Ouyang, J. and Hokao, K. (2009) Energy-saving potential by improving occupants behavior in urban residential sector in Hangzhou City, China. *Energy and Buildings*, **41**, 711–720.

[7] Holmes, T. G. (2007) Eco-visualization: Combining art and technology to reduce energy consumption. *Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition*, Washington, DC, USA, 13-15 June, pp. 153–162. ACM, New York, USA.

[8] Darby, S. (2006) The effectiveness of feedback on energy consumption: A review for DEFRA of the literature on metering, billing and direct displays. Technical report. Environmental Change Institute,University of Oxford, Oxford, UK.

[9] Cho, W.-T., Lai, Y.-X., Lai, C.-F., and Huang, Y.-M. (2013) Appliance-aware activity recognition mechanism for iot energy management system. *The Computer Journal*, **56**, 1020–1033.

[10] Yu, Z., Fung, B., Haghighat, F., Yoshino, H., and Morofsky, E. (2011) A systematic procedure to study the influence of occupant behavior on building energy consumption. *Energy and Buildings*, **43**, 1409–1417.

[11] Mozer, M., Vidmar, L., and Dodier, R. H. (1997) The neurothermostat: Predictive optimal control of residential heating systems. *Advances in Neural Information Processing Systems Volume 9*, Denver, Colorado, USA, 2-5 December, 1996, pp. 953–959. The MIT Press, Massachusetts, USA.

[12] Scott, J., Bernheim Brush, A., Krumm, J., Meyers, B., Hazas, M., Hodges, S., and Villar, N. (2011) Preheat: controlling home heating using occupancy prediction. *Proceedings of the 13th international conference on Ubiquitous computing*, Beijing, China, 17-21 September, pp. 281–290. ACM, New York, USA.

[13] Lu, J., Sookoor, T., Srinivasan, V., Gao, G., Holben, B., Stankovic, J., Field, E., and Whitehouse, K. (2010) The smart thermostat: Using occupancy sensors to save energy in homes. *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, Zürich, Switzerland, 2-5 November, pp. 211–224. ACM, New York, USA.

[14] Bapat, T., Sengupta, N., Ghai, S. K., Arya, V., Shrinivasan, Y. B., and Seetharam, D. (2011) User-sensitive scheduling of home appliances. *Proceedings of the 2nd ACM SIGCOMM workshop on Green networking*, Toronto, Canada, 15-19 August, pp. 43–48. ACM, New York, USA.

[15] Krumm, J. and Brush, A. (2011) Learning time-based presence probabilities. *Pervasive Computing, Lecture Notes in Computer Science*, San Francisco, USA, 12-15 June, pp. 79–96. Springer-Verlag, Berlin, Germany.

[16] Bayir, M. A., Demirbas, M., and Cosar, A. (2011) A web-based personalized mobility service for smartphone applications. *The Computer Journal*, **54**, 800–814.

[17] Vrečko, D., Vodopivec, N., and Strmčnik, S. (2009) An algorithm for calculating the optimal reference temperature in buildings. *Energy and Buildings*, **41**, 182–189.

[18] Frontczak, M. and Wargocki, P. (2011) Literature survey on how different factors influence human comfort in indoor environments. *Building and Environment*, **46**, 922–937.

[19] Olesen, B. W. (2004) International standards for the indoor environment. *Indoor Air*, **14**, 18–26.

[20] Luštrek, M., Cvetković, B., and Kozina, S. (2012) Energy expenditure estimation with wearable accelerometers. *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, Seoul, Korea, 20 - 23 May, pp. 5–8. IEEE.

[21] Kozina, S., Gjoreski, H., Gams, M., and Luštrek, M. (2013) Three-layer activity recognition combining domain knowledge and meta-classification. *Journal of Medical and Biological Engineering*, **33**, 406–414.

[22] Adolph, M., Kopmann, N., Lupulescu, B., and Mller, D. (2014) Adaptive control strategies for single room heating. *Energy and Buildings*, **68**, 771 – 778.

[23] Garcia, C. E., Prett, D. M., and Morari, M. (1989) Model predictive control: theory and practice–a survey. *Automatica*, **25**, 335–348.

[24] Juditsky, A., Hjalmarsson, H., Benveniste, A., Delyon, B., Ljung, L., Sjberg, J., and Zhang, Q. (1995) Nonlinear black-box models in system identification: Mathematical foundations. *Automatica*, **31**, 1725 – 1750.

[25] Dounis, A. I. and Caraiscos, C. (2009) Advanced control systems engineering for energy and comfort management in a building environment – a review. *Renewable and Sustainable Energy Reviews*, **13**, 1246–1261.

[26] Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. (2013) Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, **14**, 28–44.

[27] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009) The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, **11**, 10–18.

[28] Abras, S., Pesty, S., Ploix, S., and Jacomino, M. (2008) An anticipation mechanism for power management in a smart home using multi-agent systems. *Information and Communication Technologies: From Theory to Applications, 2008. 3rd International Conference on*, Damascus, Syria, 7-11 April, pp. 1–6. IEEE.

[29] W., M. (2011) Co-simulation of building energy and control systems with the building controls virtual test bed. *Journal of Building Performance Simulation*, **4**, 185–203.

[30] Crawley, D. B. e. a. (2001) Energyplus: creating a new-generation building energy simulation program. *Energy and Buildings*, **33**, 319–331.

[31] Bellifemine, F., Bergenti, F., Caire, G., and Poggi, A. (2005) JADE – a java agent development framework. In Bordini, R., Dastani, M., Dix, J., and Fallah Seghrouchni, A. (eds.), *Multi-Agent Programming*. Springer, New York, USA.

[32] Cigler, J., Prívara, S., Váňa, Z., Žáčeková, E., and Ferkl, L. (2012) Optimization of predicted mean vote index within model predictive control framework: Computationally tractable solution. *Energy and Buildings*, **52**, 39 – 49.

[33] Nguyen, T. A. and Aiello, M. (2013) Energy intelligent buildings based on user activity: A survey. *Energy and buildings*, **56**, 244–257.