Anticipatory System for T–H–C Dynamics in Room With Real and Virtual Sensors

Martin Frešer
Department of intelligent systems
Jožef Stefan Institute
1000 Ljubljana, Slovenia
martin.freser@ijs.si

Božidara Cvetković
Department of intelligent systems
Jožef Stefan Institute
1000 Ljubljana, Slovenia
boza.cvetkovic@ijs.si

Anton Gradišek
Department of intelligent systems
Jožef Stefan Institute
1000 Ljubljana, Slovenia
anton.gradisek@ijs.si

Mitja Luštrek
Department of intelligent systems
Jožef Stefan Institute
1000 Ljubljana, Slovenia
mitja.lustrek@ijs.si

Ubicomp/ISWC'16 Adjunct , September 12-16, 2016, Heidelberg, Germany
ACM 978-1-4503-4462-3/16/09.
http://dx.doi.org/10.1145/2968219.2968442

Abstract
With modern technology and advanced models, it is possible to rather accurately anticipate the changes of weather parameters, such as temperature or precipitation, for a couple of days in advance. On the other hand, predicting dynamics of internal parameters, such as office spaces, can be tricky, as there are many variables that influence them, and we do not have information on their status. Being able to predict how parameters are changing would allow us to recommend appropriate actions to improve their work/living environment. In this paper, we present an anticipatory system, that is built from virtual sensors, which estimates number of occupants in the room and detects the state of the windows. Together with data from real sensors ontology, as coded knowledge, outputs reasonable actions that may improve internal environment. We built models, which anticipate dynamics of internal parameters temperature (T), humidity (H) and CO_2 concentration (C), regarding all combinations of actions.

Author Keywords
Smart office; Air quality; Monitoring; Ontology Recommendation

ACM Classification Keywords
H.2.8 [Database Management]: Database Applications

Introduction

The term Sick Building Syndrome (SBS) was coined by the World Health Organization in 1986 for buildings that cause the occupants various health and comfort problems, and is most commonly linked with air quality. The increasing availability of sensors to measure building parameters, as well as building automation to control them, makes the monitoring and control of air quality a highly relevant subject of investigation. It has been showed, that bad environmental conditions in office reduces work productivity resulting in around 20 billion USD loss in the USA [7]. Some of them are hard to subjectively detect – such as air pollutants, which decreases productivity by 6–9 % [20]. Another example would be inappropriate humidity, which may result in 5 % productivity loss [17] and it is also hard to detect. Somewhat easier to detect and also to correct is inappropriate temperature, but it also has negative effects on work productivity, 10 % [17]. New buildings typically already employ some kind of system that monitors and changes the environmental parameters automatically, but most of older buildings do not. Installing such systems would require a substantial investment and work, so it is reasonable to expect that many places will not undertake such renovations in near future. With this motivation, we build an intelligent system that is constantly improving environmental parameters in workplace and only uses a few additional plug-in components. For such intelligent system we need good anticipatory system, that is able to anticipate dynamics of environmental parameters regarding the actions occupants will take.

In this paper we present an anticipatory system, where we implement i.) Virtual sensors, which estimate how many occupants are in the room and detect the state of the windows. ii.) Coded knowledge in ontology, which outputs actions that are possible to take, to improve quality of environment and iii.) anticipatory models, which uses data from real and virtual sensors and anticipate the dynamics of internal parameters temperature (T), humidity (H) and CO_2 concentration (C) according to different set of actions, that ontology recommends. The next section of the paper presents related work, which is followed by a section describing our system, a section with experimental evaluation, and finally the conclusion.

Related work

The monitoring and control of indoor environment parameters is a popular topic in smart–building research. Its main focus is the trade–off between the occupants' comfort in terms of environment quality, and energy consumption [16], [18]. Our goal is instead to satisfy regulatory requirements (which presumably ensure a high comfort and effectiveness of workers), while we do not focus on energy consumption. To achieve this goal, we have to tackle two problems: (i) recommending appropriate actions to improve the environment parameters, and (ii) estimating the parameters required for this that cannot be sensed directly with the commercial weather station at our disposal. Some of the related work on the monitoring and control of building parameters focuses on single parameters. An example is the detection of air pollutants in the indoor environment [12], which can be extended with suggestions on actions that will improve the environment [5]. Multiple parameters can be tackled by predictive control techniques, which first estimate the values of the studied parameters in the future [15], and then suggest the most efficient actions to improve them. This technique reports accurate results in terms of energy efficiency, but less so in terms of comfort. Its main weakness is that to set up the equations for the predictive models, one must have detailed information about the building characteristics (e.g., room sizes and building materials). Our system can be used in any indoor environment with no information

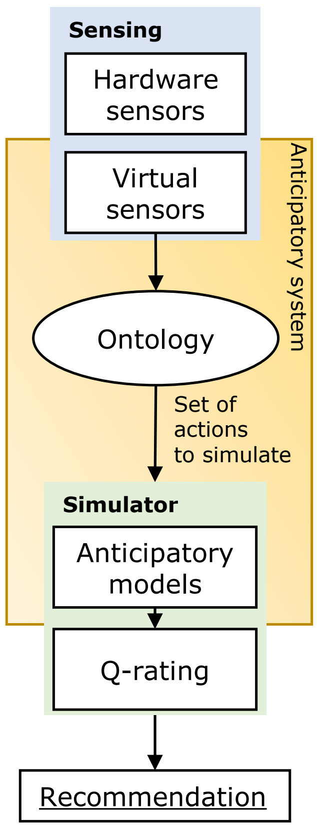

Figure 1: System architecture

about its characteristics beyond the list of devices that can be used for control (e.g., windows, heater and humidifier). Such information can easily be provided by any user. Important parameters that cannot be sensed directly with the weather station should be estimated. These are the number of occupants and state of the windows. A correlation between the T–H–C values and the occupancy state or window state has been reported [3][10], so it can be modelled. For example, Han et al. [10] report 0.96 root mean squared error (RMSE) for estimating the number of occupants solely from CO_2 concentration using Hidden Markov Models, while others used additional sensors but report a similar error [4]. We did not come across any relevant methods for window state detection. Our system estimates both parameters from the T–H–C values; the error of the occupancy estimation is comparable or lower than in the referenced work.

System description

The intelligent system to improve T–H–C has three major components – a sensing component, an ontology, and a simulator, as shown in Figure 1. The sensing component is composed of hardware (real) sensors and virtual sensors. The hardware sensors measure and return raw parameter values, while virtual sensors use machine learning on the raw parameter values to estimate parameter values or device states that cannot be sensed directly. The outputs of the sensing component are fed into the ontology. The ontology infers which actions can improve the state, based on the current state and present devices. The list of actions is fed into the simulator. The simulator is composed of anticipatory models and the quality rating module (Q–rating) which anticipates the T–H–C values for all combination of actions and evaluate the resulting T–H–C states (values range between -1 and 1). The action resulting in the best state is finally recommended by the system. In this work we present components of anticipatory system: Ontology, virtual sensors and anticipation of the T–H–C parameters. We also carried out an experiment of the whole recommender system and evaluated it over the course of 3 time periods in winter time. The details about experiment and the Q–rating will be published in future.

Sensing

There is a growing market of commercial devices [19], [6], [1] with integrated environmental sensors, which can monitor environment quality. We used the commercial weather station NetAtmo^{TM} [19] composed of an indoor and outdoor module. The indoor module measures the indoor temperature, humidity, CO_2 concentration and noise, while the outdoor module measures the outdoor temperature, humidity, and air pressure. The measurements occur every five minutes, so we define five minutes as one time step. We also implemented two virtual sensors: the occupancy estimator, which estimates the number of occupants in the room, and the window state detector, which detects whether the window is closed or open. These serve in place of physical sensors, which would increase the cost of the system and be more difficult to install than the weather station, thus being inconsistent with our design philosophy. The occupancy estimator works in two steps. First, a classification model classifies whether the room is currently occupied or not. If it is, a regression model estimates the number of occupants. The models were trained on the real numbers of occupants with machine learning. While the occupancy of the room influences CO_2 concentration in a pretty straightforward manner (more occupants $\rightarrow$ higher rate of CO_2 increase), the dynamics of the T–H–C parameters in relation to window actions are more complex; in part because they are affected not only by windows but also by doors, whose opening may cause a draft. However, we are only interested in windows being opened for at least a moderate amount of time to cause a relevant effect. In order to detect such

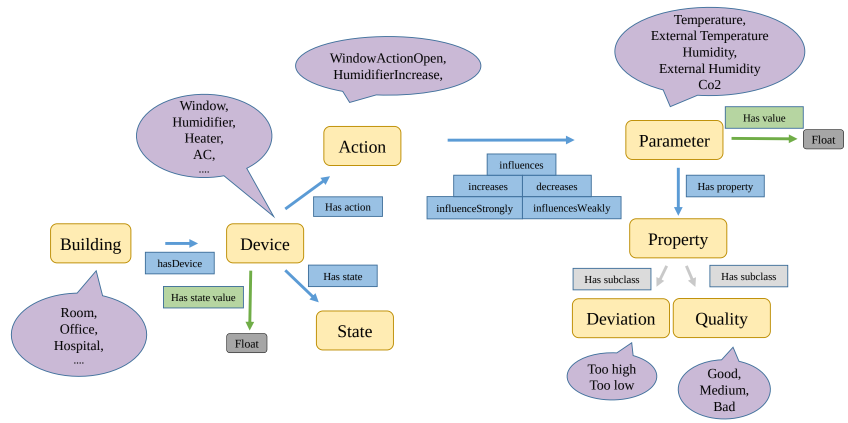

Figure 2: Ontology description, where classes are in yellow boxes, example of instances are in purple clouds and object properties (relations) are in blue boxes. Relation of subclasses are indicated in gray boxes, while data-relations are in green boxes.

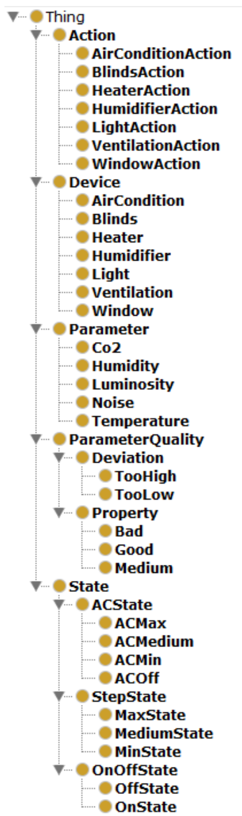

Figure 3: Exhaustive class hiearchy overview

window state and not the "false positives" originating from people entering or leaving the office, a window is considered open if it is detected as such in two consecutive time steps. The same approach is used to detect if the window is closed.

We used machine learning to model both virtual sensors. Since their outputs are based on the outputs of the hardware sensors, they also work with the time step of five minutes. For each time step, we take the hardware sensor values as well as features extracted from the historic data of the last 20 time steps, and feed them into a machine–learning algorithms. The classification models are trained with the Support Vector Machines algorithm and the regression model with the Support Vector Regression algorithm, both implemented in the Weka suite [9]. We have extracted relevant features, which are important for successfully trained models:

- Last measured values of indoor T–H–C
- Last measured values of outdoor T–H
- "First derivate" of each parameter, calculated over the last n time steps (n = 3, 5, 20) with the least square linear regression [8]. This feature intuitively gives average change of parameters in given time-line.
- "Second derivate" of each parameter, again calculated over the last n time steps (n = 3, 5, 20) with the least square linear regression [8], intuitively giving us the speed of the dynamics of the parameters
- The number of time steps since the last window action. This is important because parameter values change faster right after a window action was taken and then asymptotically approach a new equilibrium value.

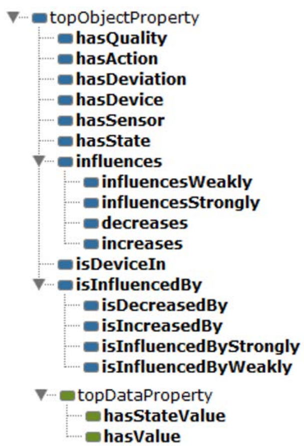

Figure 4: Exhaustive object (in blue) and data (in green) property hierarchy overview

Ontology
The motivation for representing the knowledge with an ontology is modularity. The simplicity of removing or adding new devices, actions, parameters as well as relations (e.g. if some unforseen relations are found experimentally afterwards) enables the system to be rapidly upgraded, often without any additional software development of the other components. Devices, actions, and domain expert knowledge used by our system are encoded in an ontology using the Web ontology language (OWL) [2] with the open-source software Protége [13]. The reasoning is done with the descriptive logic reasoner Pellet [20].

Figure 2 shows the structure of our ontology: a device has one state and one or multiple actions, which influence one or multiple parameters. The relation influence has two disjoint sub-relations: increases and decreases. Each parameter has a certain value, which is saved as float. It also has evaluation properties and can be evaluated in terms of quality (good, medium, bad) and deviation from good quality (too high or too low). The reasoner infers evaluation properties of parameters from their current values, and infers which actions can be taken to improve the quality based on whether the deviation of a parameter is too high/low and an action increases/decreases the parameter. While the influence of some actions is straight-forward (if we increase the heater, the temperature raises), the influence of window actions is complex and depends on external parameters. We handle such cases with SWRL rules [11] (e.g., if a window is closed and the external temperature is lower than internal, the action open the window will lower the internal temperature). After the reasoner infers all the relations, we use a SPARQL query [14] to search for the actions that change parameter quality from medium or bad to good. We present an exhaustive list of hiearchy in Figures 3 and 4, where we can see skeleton of the built ontology. The output of the ontology component is a list of actions that may improve the T–H–C quality. These actions are fed into the simulator.

Simulator
The simulator is composed of a anticipatory module, which anticipates the future changes of the T–H–C parameters, and the Q-rating module, which evaluates the quality of the environment according to the anticipation. The overall task of the simulator is to simulate the effect of all the actions suggested by the ontology on the T–H–C parameters, and to return the action that results in the highest Q-rating. We build models using 2 months of historic data. At anticipatory phase, we construct features using data for 20 time steps in history.

Anticipatory models - for individual parameters
The actions retrieved from the ontology can influence one or multiple monitored parameters. For example, turning up the humidifier influences only the humidity, whereas opening a window influences all the monitored parameters. To anticipate the values of the monitored parameters for each suggested action, we developed four machine-learning models for each of them: the temperature, CO_2 concentration, and relative humidity. Four models are needed to anticipate the values for 15, 20, 25 and 30 minutes in the future, so that the simulator can consider different durations of the recommended actions (e.g. whether it is better to open the window for 15 or 20 minutes). The anticipatory models for all three parameters are using the same features as the models for virtual sensors and additionally output of the virtual sensors. The historic data of all the parameters from the previous 20 time steps along with the extracted features are fed into a regression algorithm, which outputs the anticipated value.

Figure 5: Devices used in the experiment. Indoor NetAtmo™ module (top left), humidifier (top right), window state sensor (bottom left), and application interface for self–reporting the occupancy, labeling the state of the used devices and receiving the recommendations (bottom right).

We have evaluated multiple regression algorithms experimentally and selected the Support Vector Regression (SVR) for all the anticipatory models, as it produced the best results on the experimental data using 5-fold cross–validation. Using cross–validation, we also selected most appropriate parameters for SVR. For result presentation, we evaluate anticipatory models which anticipates for the 20 minutes in the future.

Experimental evaluation

Dataset

Three offices, A (43 m^2), B (27 m^2), and C (20 m^2) were equipped for data collection and evaluation. During the working hours (on work days between 9.00 and 17.00), the average number of occupants per office was: 2.6 ± 1.5 (max 9) in A, 2.0 ± 0.9 (max 7) in B and 1.6 ± 0.9 (max 7) in C. All offices were equipped with NetAtmo™ indoor and outdoor modules (top left in Figure 5), which measure several environmental parameters including T–H–C, a humidifier (top right in Figure 5), window and door sensors that detected the window state as opened or closed (bottom left in Figure 5), and with a smart–phone with an application for self-reporting the occupancy, labeling the state of the devices (e.g., humidifier is on or off) on bottom right corner of Figure 5.

Experiment and results

The experiment was set to first evaluate and validate the developed machine-learning models (virtual sensors and anticipatory models). For the evaluation of the models we split the data of each office into the training set (70 %) and test set (30 %). The evaluation was done for each office. To validate the models, we used the leave-one-office-out approach. The classification models were evaluated and validated in terms of accuracy (ACC), and the regression models in terms of mean absolute error (MAE) and root mean squared error (RMSE). The results are presented in Table 1. We can see that we achieved a better validation RMSE score for estimating number of occupants (0.8) compared to related work (0.96). Intuitively, the MAE score of 0.46 means that we made less than half a person mistake in a room where people often enter and leave. The evaluation of the virtual sensors shows that the accuracy of window state detection is above 81 % (91 % in evaluation). Anticipatory models experiments show that the error slightly increases during the validation, however, the results are still comparable to the evaluation results. We can see that, in average, we make roughly 0.5 % MAE anticipating humidity in 20 minutes time. While this result may appear surprisingly accurate, we should stress that the changes of humidity only become significant over prolonged time intervals. Results for temperature and CO_2 are also encouraging, making half a degree and around 50 ppm error on average, which allows us to use this system to give quality recommendations to the users. In Figure 6 we show summary of results using presented anticipatory system on intelligent recommender system which gives best evaluated set of actions as recommendation to the user of the system. We evaluated system across 3 periods of time. We can see that we achieved better overall Q-rating (combined score from T–H–C parameters) in offices that were equipped with our system than in offices that were not. Further details of the analysis will be published in future, including also the data for all four seasons.

Conclusion

We developed an anticipatory system that contains two major components. First, virtual sensors take data from a commercial weather station (measuring internal and external temperature, internal and external humidity, and internal concentration of CO_2) and estimate the number of occupants in the room and detects state of the window

Models	ACC		MAE		RMSE	
	E	V	E	V	E	V
Window state [%]	91	81	✗		✗	
Number of occupants	✗		0.60	0.46	1.20	0.80
Ant. model (T [℃])	✗		0.41	0.46	0.54	0.56
Ant. model (H [%])	✗		0.58	0.33	0.9	0.47
Ant. model (C [ppm])	✗		54.7	44.7	104	60.7

Table 1: Evaluation (E) and validation (V) results for the developed models. Results for window state detection is reported in terms of accuracy (%). Results of estimation of number of occupants and anticipatory models are presented in terms of error (MAE, RMSE).

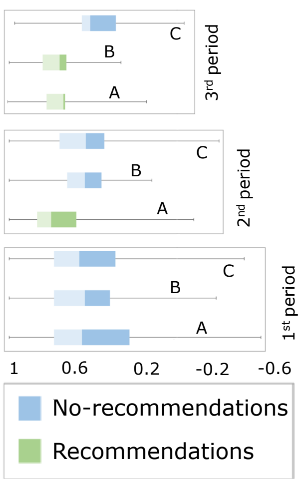

Figure 6: Overall Q–rating per office for all three periods. Blue boxes are offices without recommendations and green boxes are offices with recommendations.

(open/closed). Output of the first system together with hardware data is fed into ontology, representing the source of knowledge, which produces a meaningful set of actions to simulate. The second component is a simulator which simulates the values of the T-H-C parameters in near future, based on potential actions and data provided from virtual and hardware sensors. Machine–learning algorithms were used to construct models behind both virtual sensors and the simulator. Initial results based on testing the system on real data show that both components of implemented system are comparable or better than the related work. Using such implemented system, we are able to provide recommended actions that improve the quality of work/home environment. With accessibility of the hardware component we are able to bring "smartness" in "non-smart" households without an expensive intervention.

In future, we plan to detect states of additional devices (such as humidifier, air-condition and heater) through virtual sensors and to add additional environmental parameters into consideration (at least noise and light).

Acknowledgements

This research was cofunded by the European AAL project Fit4work.

REFERENCES

1. Airboxlab. 2016. foobot. (2016). `http://foobot.io/`, accessible June 8th, 2016.
2. Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. 2004. *OWL Web Ontology Language Reference*. Technical Report. W3C, http://www.w3.org/TR/owl-ref/.
3. Adorkor Bruce-Konuah. 2014. *Occupant window opening behaviour: the relative importance of temperature and carbon dioxide in university office buildings*. Ph.D. Dissertation. University of Sheffield.
4. Luis M Candanedo and Véronique Feldheim. 2016. Accurate occupancy detection of an office room from light, temperature, humidity and CO 2 measurements using statistical learning models. *Energy and Buildings* 112 (2016), 28–39.
5. Xuxu Chen, Yu Zheng, Yubiao Chen, Qiwei Jin, Weiwei Sun, Eric Chang, and Wei-Ying Ma. 2014. Indoor air quality monitoring system for smart buildings. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 471–475.
6. CBSR d.o.o. 2016. CubeSensors. (2016). `https://cubesensors.com/`, accessible June 8th, 2016.
7. William J. Fisk, Doug Black, and Gregory Brunner. 2011. Benefits and costs of improved IEQ in U.S. offices. *Indoor Air* 5, 21 (2011), 357–367. DOI: `http://dx.doi.org/10.1111/j.1600-0668.2011.00719.x`

8. Francis Galton. 1886. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland* 15 (1886), 246–263.
9. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11, 1 (2009), 10–18.
10. Zhenyu Han, Robert X Gao, and Zhaoyan Fan. 2012. Occupancy and indoor environment quality sensing for smart buildings. In *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*. IEEE, 882–887.
11. Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, and others. 2004. SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission* 21 (2004), 79.
12. Yifei Jiang, Kun Li, Ricardo Piedrahita, Xiang Yun, Lei Tian, Omkar M Mansata, Qin Lv, Robert P Dick, Michael Hannigan, and Li Shang. 2013. User-Centric Indoor Air Quality Monitoring on Mobile Devices. *AI Magazine* 34, 2 (2013), 11.
13. Holger Knublauch, Ray W Fergerson, Natalya F Noy, and Mark A Musen. 2004. The Protégé OWL plugin: An open development environment for semantic web applications. In *The Semantic Web–ISWC 2004*. Springer, 229–243.
14. Ilianna Kollia, Birte Glimm, and Ian Horrocks. 2011. SPARQL query answering over OWL ontologies. In *The Semantic Web: Research and Applications*. Springer, 382–396.
15. D Kolokotsa, A Pouliezos, G Stavrakakis, and C Lazos. 2009. Predictive control techniques for energy and indoor environmental quality management in buildings. *Building and Environment* 44, 9 (2009), 1850–1863.
16. Prashant Kumar, Claudio Martani, Lidia Morawska, Leslie Norford, Ruchi Choudhary, Margaret Bell, and Matt Leach. 2016. Indoor air quality and energy management through real-time sensing in commercial buildings. *Energy and Buildings* 111 (2016), 145–153.
17. L. Lan, P. Wargocki, and Z. Lian. 2012. Optimal thermal environment improves performance of office work. *REHVA European HVAC Journal* (January 2012), 12–17. DOI:`http://dx.doi.org/10.1111/j.1600-0668.2004.00278.x`
18. Claudio Martani, David Lee, Prudence Robinson, Rex Britter, and Carlo Ratti. 2012. ENERNET: Studying the dynamic relationship between building occupancy and energy consumption. *Energy and Buildings* 47 (2012), 584–591.
19. Netatmo. 2016. Netatmo. (2016). `https://www.netatmo.com/`, accessible June 8th, 2016.
20. David D. P. Wyon. 2004. The effects of indoor air quality on performance and productivity. *Indoor Air* 7, 14 (2004), 92–101. DOI:`http://dx.doi.org/10.1111/j.1600-0668.2004.00278.x`