Mining hierarchical relations in building management variables

ARTICLE · OCTOBER 2015

DOI: 10.1016/j.pmcj.2015.10.009

Luis Ignacio Lopera

Universität Passau

Mining hierarchical relations in building management variables

Luis I. Lopera Gonzalez, Oliver Amft

ACTLab, University of Passau, Germany
{*luis.loperagonzalez, oliver.amft*}@*uni-passau.de*

Abstract

We present a framework to relate variables as they occur in a modern building management system (BMS) that processes data from building-installed sensor and actuators. Our group mining framework extracts a unified event time series as changes in building management variables, derives prepositional variable association rules, and extracts hierarchical variable groups from the derived rules. Variable changes typically occur by either occupants interacting with the building or as response to outdoor environmental changes. As a user enters a building, a sequence of sensor activations will create a specific temporal event pattern that is mapped into a variable hierarchy by our framework. Similarly, as outdoor lighting changes, a variable hierarchy appears that relates variables to the change. To extract variable groups, we introduce a novel hierarchical transitive clustering (HTC) algorithm that constructs a rooted variable tree and then clusters the tree to represent variable group relations. HTC is parameter-free and works unsupervised. We evaluated the group mining framework in living-lab data recorded in different office environments during 14 months. As typical for BMS operation, variables in our dataset represent measurements and control states of building-installed devices and processed context information. HTC showed a correctness of over 0.91 and an average variable coverage of 75%, this improving variable coverage by 40 p.p. compared to previous work. We successfully detected alternative hierarchies and show how variables relate across office rooms.

Keywords: Activity, hierarchical activity recognition, activity discovery, prepositional rule clustering, unsupervised hierarchical clustering, building automation.

DOI: 10.1016/j.pmcj.2015.10.009

1. Introduction

Effective operation of modern buildings relies on interconnected ubiquitous devices. Examples of ubiquitous devices installed in many large buildings include movement detectors and light sensors, besides actuators, e.g., ceiling lights and air ventilation. The devices are represented as measurement and control variables in a building management system (BMS). The variables are then combined in rules or more advanced interpretation algorithms to infer building context states. For example, Milenkovic et al.[1] showed how several office user activities could be recognised from sensors already available in many buildings. Detected user activities are again variables. A BMS thus links measurements from sensors and operations of actuators. For example, occupancy measurements in a meeting room, indoor temperature, and outdoor weather conditions are used to control optimal operation of a heating, ventilation, and air-conditioning (HVAC) system in this meeting room.

The motivation of building owners and operators to further increase measurement and automation functions are manifold, including saving energy [2], and improve occupant comfort under energy efficiency constraints [3, 4]. Buildings, such as office towers may utilise about ten measurement and control variables per desk, resulting in 100 variables for a 10-occupant open office space. In a typical office building with ten floors and 500 desks, as many as 50.000 variables can be expected, not considering corridors, elevators, meeting rooms, etc. Priyadarshini et al. [5] reported an application example of wireless sensor nodes, where ubiquitous integration and rapid increase in devices and variables is well illustrated. For scalability, BMS development is thus migrating to a service-oriented architecture, as illustrated by Degeler et al.[6]. Nevertheless, a great deal of manual labour is required during commissioning to correctly relate sensors and actuators, derive context variables, and maintain consistency during a building's lifetime.

Whether newly build or refurbished, commissioning of a BMS is today performed by expert technicians that work together with device and system suppliers to correctly interlink variables. During maintenance, further effort is spent to correct variable naming errors and updating configurations upon device exchanges, failures, or upgrades. It is technically conceivable that any modern building-installed device would automatically identify itself, revealing its location and interrelation of variables with other devices. However, several constraints prohibit self-configuring features from being implemented in individual building-installed devices, including device cost, size, power consumption, and robustness for continuous operation over many years. Due to the cost of large BMS installations, sensors and actuators need to be affordable. For example, EnOcean (www.enocean.com) develops wireless self-powered sensors and actuators, including motion detectors and wall switches, which are even used in new building constructions to minimise physical network infrastructure. Therefore, methods that could mine variable relations from a central BMS based on the variables' data streams are sought. In earlier work, we showed that mining relations among BMS variables and grouping variables has many applications, including detecting missing or broken devices, detecting and correcting variable naming errors, and deriving new variables such as people count in office spaces [7].

We present a novel approach to derive groups and group hierarchy from variable association rules that are mined in event time series of variable states. Our hypothesis is that variables that relate to the same physical space in a building will provide events with temporal relation. Variables that have a causal relation, i.e., a variable changes in response to a change in another variable, should have a temporal dependency in states that could be extracted from their event time series. For example, occupants walking through a building space towards a desk will create changes in presence and activity variables and potentially in light actuator states, first for the building space, i.e. room, and then at the desk. For our approach, the variables could have different origin, including sensor measurements, actuator states, and derived states. Our approach considers state changes across all variable as one unified, temporal ordered event stream and derives variable grouping without supervision. As rules are mined, temporal relation of events provides the basis for variable groupings. Subsequently, variable hierarchy is determined to describe dependencies.

In particular, the paper provides the following contributions:

1. We present a group mining framework to extract variable association rules from the unified variable event stream and then extract variable groups from the rules. A novel parameter-free hierarchical tree clustering (HTC) algorithm is introduced to group variables into hierarchical structures. HTC works in two steps, where initially a rooted variable tree is constructed based on the extracted variable association rules and subsequently the tree is clustered and trimmed to represent variable group relations.
2. We evaluate our mining framework using actual BMS data of several months and derive variable grouping performance. The grouping performance is compared to expert generated groups and to groups derived from BMS configurations. The evaluation confirmed that HTC provides relevant groupings that are robust even if only subsets of the BMS data was considered as

source.

In our previous work, we devised a weighted transitive clustering (WTC) algorithm to discover variable grouping [7]. However, the groups derived by WTC provided no hierarchy information. As a result, when an environmental variable is associated with more than one room group, the linked room groups are merged into one. Additionally, to distinguish different hierarchy levels, e.g. room and desks, WTC needs to be run with different parameter settings. In contrast, the HTC algorithm presented in this work produces hierarchical groups that describe the relationship between rooms and desks. The hierarchical groups can also describe interconnections between groups created by environmental variables. In this work, we demonstrate that HTC can outperform the WTC algorithm. HTC finds groups that associate a majority of variables and identifies alternative relations between variables. Moreover, due to the hierarchical representation, variable relations could be inspected graphically.

2. Related work

In wireless sensors networks, device localisation is an active field of research. One common approach to localisation is device-based, i.e. using sensor node communication capabilities to estimate a node's relative location with respect to its neighbours [8, 9]. Aspens et al. [10] provide a detailed theory for automatically constructing graphs that represent the sensor location in a bi-dimensional or three-dimensional space. They considered the distance between neighbouring nodes known and assume fixed positions of beacon nodes. In practice, the location of beacons is manually configured in order to map a node's location to a physical location. Furthermore, beacons become critical path points in the system, as broken beacons may prevent correct node localisation.

Patwari et al. [11] illustrate how to obtain the distance between neighbouring nodes using time of arrival and received signal strength. The localisation techniques used in wireless sensor networks require that a node spends some energy in the task of localisation. In a highly ubiquitous and distributed environment, sensor nodes will have very restricted energy budget and on-sensor localisation might not be feasible. In addition, alternative communication media and protocols may require separate device-based localisation solutions. Since buildings are used for decades, different communication media may get installed as technology evolves. In this regard, modern BMSs are already design to be extensible and handle different protocols or media [6]. Instead of device-based localisation, it is therefore interesting to utilise the data available in a BMS.

While modelling energy consumption patterns of a building, Moreno et al. [12] investigated which building variables had the largest impact on individual energy-consuming appliances, e.g. HVAC. They proposed a clustering approach to infer a group of variables related to the energy measuring device in each appliance. To implement the clustering, snapshots of the building variables' values were used to compute features. The features would cluster variables for each appliance. In the work of Moreno et al., energy measuring devices represent the clusters to which other variables get assigned. In contrast, our variable grouping does not require a particular energy measuring device or appliance to be present.

Rule mining approaches are able to extract association of data from multiple modalities [13, 14]. For example, Koperski et al. [15] showed how useful rules can be extracted by mining a geographical database, e.g., house prices rise for houses closer to the beach. Furthermore, Ma et al. [16] showed that their rule mining approach was able to correctly handle many different datasets. Yin et al. [17] illustrated the difficulty of mining relevant rules, and the importance of balancing the support of the rules in order to get meaningful results. In our framework [7], we used the temporal rule association method proposed by Guillame-Bert et al. [18]. The framework receives a labelled stream of events and extracts rules of the form $A \Rightarrow B$, where B happens within an observation window after A. Temporal association rules can represent the behaviour of real systems that have delays between the time a sensor changes value and the time an actuator responds to that change. In addition to sensor actuator relationships, we consider relations created by a user moving from one context state to another, e.g., presence at the desk to computer work. The delayed actuator reaction and the time of a context transition are not possible to detect in a snapshot of the data. Thus temporal rule mining methods are better suited to extract relations in a building environment than variable clustering using data snapshots.

After obtaining variable relations, groups among all variables are extracted. When using WTC as group extraction method, we observed that it was not possible to differentiate structures within variable groups [7]. WTC uses thresholds to control grouping. We observed that the thresholds also controlled the hierarchical level of groups, e.g., room level or desk level. Grouping is thus challenging in building structures that share common elements from external environmental variables, e.g., sun light, or from joint building variables, e.g., presence in hallways. Consequently, WTC required a parametric search over the thresholds to extract each grouping hierarchy level, i.e., distinguish rooms or desks in a room.

In this work, we replace WTC with a novel group extraction algorithm that yields hierarchical group structures. These structures describe individual variable relations, thus solving the grouping in shared building structures. Unlike WTC that relies on thresholds, our new approach is non-parametric. We evaluate the resulting groups to compare with WTC, and discover additional variable relations that might be of interest.

3. Group Mining Framework

For our mining approach, we assume that sensors and actuators that are located, or related to the same physical space will produce temporally related events. Relations will appear in particular then, when variables have a causal relation. The event relations among variables may be caused by the user interacting with the space, but could as well be coming from weather changes and other sources of dynamics. For example, changes in outdoor lighting conditions may cause a BMS to respond by changing blind or lighting states. A change in outdoor light should be observable in most outdoor light sensors. Therefore, all outdoor light sensors will present a high temporal correlation of their change events. Moreover, the events of associated actuators will show correlations to the outdoor light events.[1]

Our group mining framework comprises three steps: (1) event extraction, (2) rule extraction, and (3) group extraction. In this work, we focus on the group extraction to obtain hierarchical variable groups that could be used in building management. Figure 1 illustrates the group mining framework. In this section, we describe the processing steps of our framework.

3.1. Event extraction

Event extraction is used to convert the data time series of building management variables into a unified time series of events. Events are extracted by detecting positive value changes in the variable data time series. Then, all events are labelled with the variable name and inserted into a single, temporal ordered event time series. The actual data values or other information about the detected changes are not used in the subsequent processing. Table 1 shows an example sequence of the resulting unified event time series.

Every variable v_i was processed for value changes at every sample time t using the condition $\frac{\Delta v_i}{\Delta t} > \Theta_i$. The threshold filter $\Theta_i > 0$ was used to filter out variable-specific noise.

[1] There may be different relations observable among the variables relating to individual building facades, depending on the building orientation.

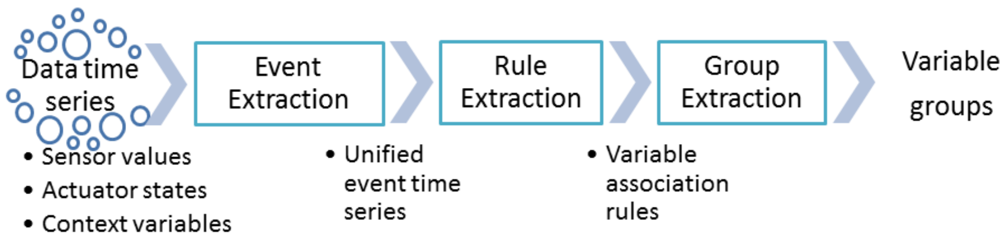

Figure 1: Group mining framework to derive variable group relations from the data time series of building management variables. The groups are derived by mining a unified time series of events for variable association rules. In this work, we focus on extracting variable groups using a novel hierarchical tree clustering (HTC) algorithm.

Table 1: Example of unified event time series derived from building management variables by the event extraction step of our group mining framework.

Event timestamp	Event label
1353935850.335012	powerconsumption11
1353935868.344362	powerconsumption2
1353935898.467623	presence3
1353935900.453503	luxlevel3
1353935934.376489	powerconsumption1
1353935936.701680	powerconsumption4
1353935946.382288	powerconsumption3

3.2. Rule extraction

Rule extraction converts the unified event time series into a set of prepositional rules with a confidence assigned to each extracted rule. Rules were derived as variable associations based on event occurrences over an observation window. The resulting variable association rules are denoted as logic relations of the form $A \Rightarrow B$. When A is observed, B is expected to occur before a time laps τ. In addition, we obtain a rule confidence and denote the support of a rule. Here we consider rule confidence as follows: Given all events of label A, how many times $A \Rightarrow B$ occurred within τ. The concept of support can be defined as: Given all events with label B, how many B events can be produced by the rule $A \Rightarrow B$ within τ. Table 2 shows an example of the variable association rules.

To implement the rule extraction, we used the TITArl algorithm by Guillem-Bert et al. [18]. TITArl discovers rules in the event sequence by tracking the

Table 2: Example of variable association rules with confidence and support derived from a unified event time series by the rule extraction step of our group mining framework.

Rule	Confidence	Support
$blindsangle3 \Rightarrow dimmer2$	0.05	0.05
$dimmer1 \Rightarrow blindsheight2$	0.05	0.06
$blindsangle2 \Rightarrow blindsheight2$	0.05	0.11
$blindsangle1 \Rightarrow dimmer2$	0.05	0.25
$luxlevel1 \Rightarrow luxlevelout2$	0.05	0.10
$deskwork9 \Rightarrow presence3$	0.05	0.06
$door1 \Rightarrow statuslights2$	1.00	1.00
$powerconsumption1 \Rightarrow statuscomputer1$	1.00	1.00
$statusbeamer1 \Rightarrow presentation1$	1.00	1.00

temporal distribution of candidate event occurrences and their correlation over time. TITArl provides a validity period, indicating the rule lifetime across an event sequence. TITArl parameters were set identical as in our previous work [7]: $inputEvent$ predicate `".*"`, $outputEvent$ predicate `".*"`, time window future $15s$, time window past $15s$, minimum rule confidence 0.05, minimum support 0.05, minimum number of use 2. All other settings were left at their default configuration. We set confidence, support, and number of uses parameters to obtain a sensitive rule mining behaviour.

3.3. Group extraction

Group extraction intends to convert the varying variable association rules into variable groups according to their relationship. In this work, we consider that organising the variables in a hierarchical representation according to their relationship benefits variable grouping as the groups retain information about the variables' causal relation. We propose here the HTC algorithm that is detailed in the following section. The WTC algorithm, used previously, considered all elements in a rule as belonging to one group [7]. WTC also considered that a newly extracted rule would be part of an existing group, if the new rule shared elements with the existing group. WTC used a threshold on a new rule's confidence to decide whether to include the new elements from the rule in the existing group.

4. Hierarchical tree clustering (HTC)

To derive hierarchical variable groups our HTC algorithm operates in two steps. In the first step, a rooted tree is constructed from the variable association rules. In the second step the tree is clustered into hierarchical groups.

4.1. Constructing a rooted tree

In the first stage, we construct a rooted tree $G(N, \xi)$, where N is the set of nodes consisting of the variables in the rules plus a node *root* from where all branches grow. Set ξ is the set of edges constructed from pairing the premise of the rule to the conclusion. Every edge ξ is assigned a weight that is equal to the value of the rule confidence c. A *main branch* is defined as the sub-tree structure in which the top node is a child of the *root* node. Figure 3 (b) depicts the main branch for a four-people office room. The resulting tree is also called a rooted directional graph and is similar to an arborescense as defined by Mesbahi et al. [19], where there should exist no simple cycles, every node should have only one path leading to it, and all paths direct outward from the *root* node. For our tree construction, one exception to the definition was needed: Leaf nodes are allowed to have alternative paths. For the building variable tree, there may be shared elements at the leafs of the hierarchical structure. For example, two independent users presence variables can be related to one common ceiling light.

Subsequently, the tree is rearranged in two forms: (1) Nodes that are children of *root* and several other parents are removed from the *root* node. (2) Nodes that have multiple parents and are not a leaf node are promoted to be the parent of their parents. Promoting nodes enforces the arborescense. The rearrangement process is repeated until every node has a single parent, with the exception for leaf nodes. Figure 3 (a) shows a tree constructed from the BMS configurations, Figure 3 (b) illustrates the result of applying node promotion.

In building management data, node promotion has logical implications. Variables that are used to compute an additional variable are often promoted. For example, presence in the room is computed from the presence at the desks. Therefore, promoting room presence over the presence at the desks creates a correct logical hierarchy. In the rule mining step, rules will reflect the order of computation. Thus, node promotion is needed to reflect the logical hierarchy.

After the tree rearrangement is concluded, trimming is done by removing unwanted edges. We iterate over the main branches and do a one-to-rest comparison. We defined a normalised branch length $\overline{L}_B$, as shown in Equation 1, where $\overline{L}_p$ is the normalised length of path p, e_p is number of edges in the path p, c_i is the edge weight in path p, and β is the set of paths of the main branch B that have common nodes with other main branches. All paths in β are trimmed by enforcing Equation 2. Finally, if the trimming process yields loose branches, these loose branches become main branches. Orphaned nodes are discarded. Figure 3 (b) illustrates the trimming process by removing the connection between the door sensor, dimmers, and outdoor light sensor.

$$\overline{L}_p = \frac{\sum_{i=1}^{e_p}(1-c_i)}{e_p} \qquad (1)$$
$$\overline{L}_B = \max\left(\overline{L}_p \middle| \forall p \notin \beta\right)$$

$$\overline{L}_p \leq \overline{L}_B | \forall p \in \beta \qquad (2)$$

4.2. Tree clustering into hierarchical groups

The second and final step of the HTC algorithm is to convert the main branches into hierarchical groups. A hierarchical group consists of variables arranged in a head set, an elements set, and an array of children. The children are groups with the same properties. We define a *main group* as the hierarchical group derived from a main branch. In addition,we considered *group members* as set of variables contained in head and elements sets of a group, as well as the variables of all child groups.

To convert the tree into groups, we analysed the main branches. Each node at the top of a main branch becomes the group head. The following nodes that have only one child also become part of the head set. The node elements are the leaf nodes which directly descend from the last head node. Any node derived from the last head node which contains more than one child, will be in the head set for the subgroup. The process is repeated recursively until all nodes are covered.

Groups can be of five different types: basic, alternative, shared, equivalent and overlapping. An *alternative group* comes from a branch that has all its paths ending in the leafs of another branch. When all paths of a branch end in different branches' leafs, then the resulting type is *shared group*. An *equivalent group* comes from branch that shares all leafs with another branch. An *overlapping group* derive from a branch that has some paths ending in their own leafs, and some paths that end in some other branches' leafs. Finally the *basic group* is a group that has no overlaps, but can have some leafs shared, or leafs appearing as part of an alternative group. Figure 2 illustrates all group types. Stand-alone trees, e.g. Figure 3 (b), are also considered basic groups in our work.

Figure 3 (c) shows an example of a main group corresponding to a room, cells corresponding to actual office desks in the room, and subcells corresponding to desk areas in front and at the sides of a computer screen. The hierarchical groups produced by HTC resemble the variable locality.

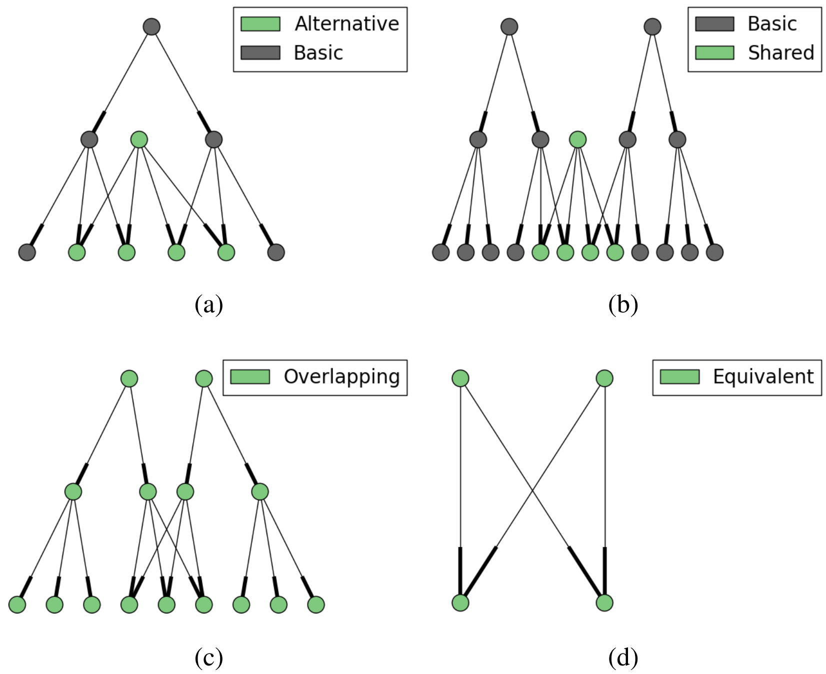

Figure 2: Illustrations of branches that generate different group types in our HTC algorithm. Please refer to Section 4.2 for more details.

From the rules that produce the tree in Figure 3 (a), a group is extracted that represents a room with cells and subcells subgroups. In contrast, when the same rules were used with WTC, the result is single flat group, e.g., Room 1, without information on cell or subcell association.

5. Evaluation methodology

For the evaluation of the HTC algorithm, we used the same set of rules mined for our previous analysis of the WTC algorithm [7]. However, we created a new ground truth and composed a complete hierarchical description of the living-lab data to validate the framework. Additionally to the new ground truth, we used configuration information from the BMS to validate the grouping stage of the framework. Finally, we took external environmental variables into consideration as they represent parallel hierarchies to the room organisation created from user-driven events.

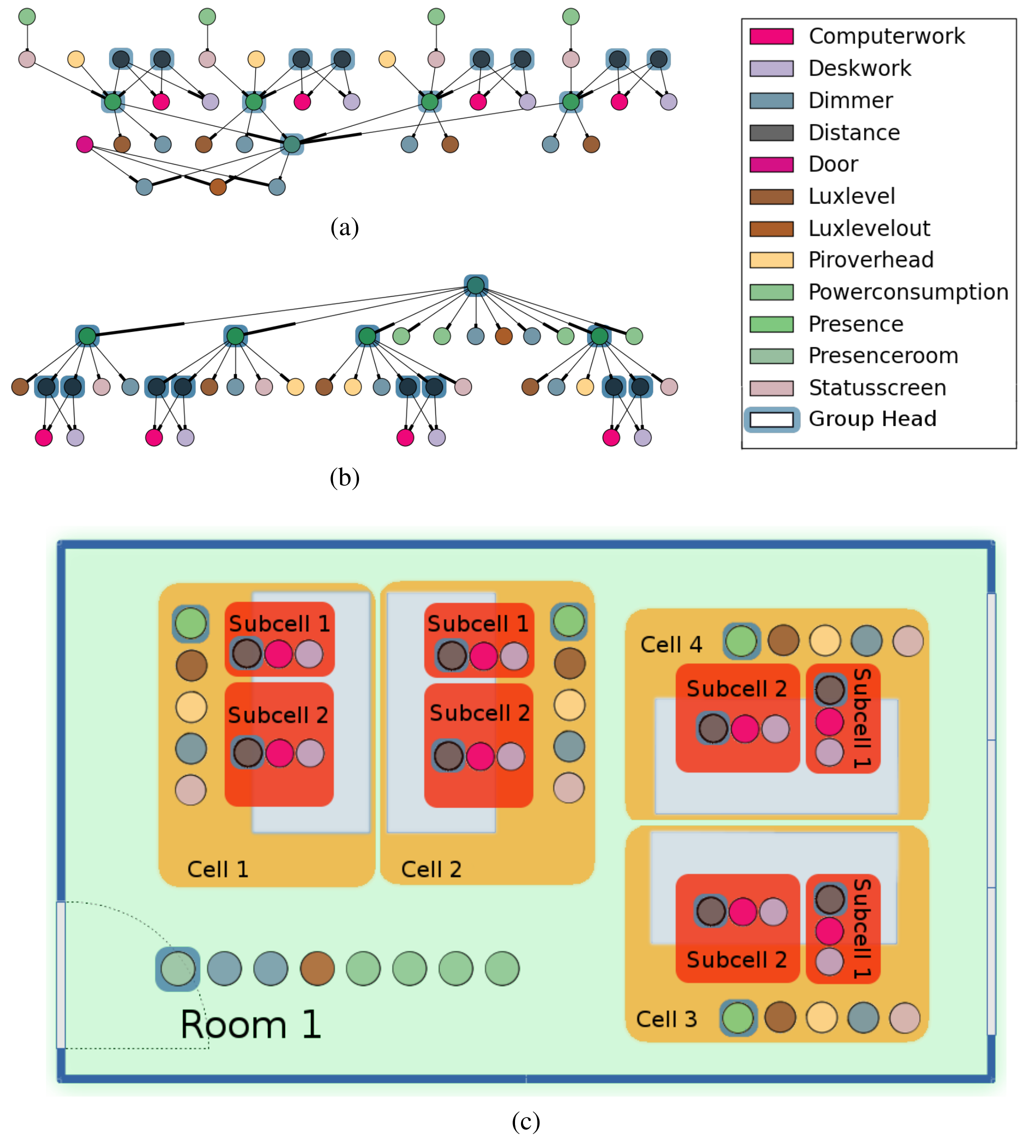

Figure 3: Illustration of the HTC algorithm steps. (a) Rooted directional graph constructed from system configuration rules of a four-people office room. (b) Processing result after arranging and trimming the tree. (c) Respective conversion of the tree into hierarchical variable groups. The groups are depicted in correspondence to the physical architecture of the room.

5.1. Evaluation Dataset

The GreenerBuilding (GB) living-lab dataset was recorded at the TU Eindhoven campus during 14 months. The living-lab consisted of 3 rooms; R1: a 4 people office room (14 months), R2: a ∼20 people meeting room (13 months), and R3: a 12 people open office space (3 months). In total 277 variables were recorded. The dataset contains sensors (outdoor light, room CO_2, humidity, temperature, etc.), actuators, and contextual variables. Context variables included room presence, desk presence, computer work, presentation, brain storming, etc. During the 14 months of recording, sensors were added, broken ones replaced, rules were updated, and configurations fine-tuned. We consider that the modifications are part of a real building life-cycle, and thus presents a rich set of test cases to evaluate our approach. The dataset is described in detailed in [7].

5.2. Ground truth and BMS configurations

We derived ground truth groups (GTG) manually considering user-driven and environmental dynamics to create a rooted tree. The tree was subsequently grouped into hierarchical groups. After grouping, GTG contained ten main groups in total. Two of the main groups represented R1 and R2 respectively. R3 had no variable that joined the available cells, i.e., no door or presence in the room variables. Thus, the other eight main groups represented cells from R3.

In addition, we created BMS configuration groups (BCG) by automatically parsing the BMS configurations and extracting rooted tree. Similarly to GTG, we grouped the tree into hierarchical groups using the group creation step of HTC. The configurations were selected from the final month of recording, where all system variables were present. The resulting groups were then manually verified to ensure correctness. The BCG represents a baseline for comparison to our HTC algorithm where all the edges' weights were set to one.

5.3. Evaluation metrics

Metrics of graph similarity could be applied to compare the ground truth with the mined groups [20]. However, graph similarity requires strict agreement in variable associations. The ground truth used in our evaluation contained all the variables ever recorded by the system, whereas individual periods would contain a subset of the variables only. In the BMS application domain, different objectives should be assessed: It is desirable that all available variables are used, and that each variable is associated correctly with its corresponding variables. Thus, relative coverage and correctness were chosen as metrics to evaluate performance.

Relative coverage measures how many variables are associated from those variables available in the test period. By measuring relative coverage, we can keep track of the effectiveness of the algorithm. Accurate grouping of a fraction of variables only could hinder applications of the algorithm.

Correctness assesses the quality of the groups rather than the similarity to the GTG. To establish the group quality, we define room level correctness and variable neighbour correctness. The room to which a variable belongs to is encoded in the variable's name, e.g., *dimmer1_R1*. The naming convention facilitates checking correctness at the room level. We defined three approaches to determine the label of a group: majority voting (MV), head majority voting (HMV), and sensor class (SC).

Correctness MV computes the label of the group by parsing the name of all variables in the group, including subgroups. The room label with the most occurrences becomes the group label. *Correctness HMV* only applies majority voting to the head of the main group. Correctness HMV is thus a stricter comparison than correctness MV and emphasises performance of the hierarchical approach. *Correctness SC* separates sensors in two classes: outdoor, and the rest, we want to measure how well is the grouping when we consider that external environmental sensors might link variables from different rooms. Therefore, correctness SC applies correctness MV for the groups that do not contain external environmental sensors, and otherwise considers groups to be correct. In each case, the respective correctness score is computed by the ratio between the number of variables that have the chosen group label and the total amount of variables in the group.

The room analysis is insufficient to evaluate the information gained from the hierarchical grouping. Therefore, we consider correctness from a variable's neighbourhood point of view too. Equation 3 defines the neighbourhood of x with respect to a set of groups G, where $gm(g)$ are the group members of g.

$$\mathrm{Ne}(x, G) = \{\mathrm{gm}(g) \forall g \in G | x \in g.head \vee g.elements\} \tag{3}$$

Correctness at the group level (G) computes for a given variable x, the ratio of the number of elements in the intersection between $\mathrm{Ne}(x, G_d)$ and $\mathrm{Ne}(x, GTG)$ over number of elements in $\mathrm{Ne}(x, G_d)$, where $Ne(x, G_d)$ is the neighbourhood set of x with respect to the groups extracted from the data , and $Ne(x, GTG)$ is the neighbourhood set of x with respect to the GTG. Equation 4 shows how the total score of $G_d$is computed.

$$C_G = \mathrm{mean}_{\forall g \in G_d} \left(\mathrm{mean}_{\forall x \in g} \left(\frac{|Ne(x, G_d) \cap Ne(x, GTG)|}{|Ne(x, G_d)|} \right) \right) \tag{4}$$

Moreover, we want to measure the effect of allowing the external environmental sensors to mix different variables together. We therefore measure correctness at the group level with sensor class selection (GSC). Similarly to correctness SC, correctness GSC considers a subgroup being correct, which contains at least one external environmental sensor and otherwise applies correctness G. In either case, to compute the total score of a group, the score for all variables is averaged. Finally, to compute the score of a grouping result the score of all groups are averaged.

Applications in the BMS domain depend greatly on the correctness of the groups [7]. For example, in autonomous configuration applications mistakes in the group extraction will create user discomfort, e.g. when light sensors and switches may be linked to wrong overhead lights. Similarly, in error detection application erroneous variable grouping would require additional effort from technicians in reviewing the application's error reports.

5.4. Group quality analysis

In addition to correctness and relative coverage, we measured group quality by counting how many groups are created according to the group types defined in Section 3, e.g., basic, alternative, etc.

Groups containing external environmental sensors may interact with other groups. By using a naming convention that differentiates groups with and without external environmental sensors, we measured their occurrence as a ratio. Any group containing external environmental sensors were named *Out_variable-type_id*, e.g., *Out_light_001*, while other groups were named *Space_label_id*, e.g, *Space_R1_0001*.

5.5. Comparative analysis

We compared the performance of HTC against the WTC method and therefore used the same rules as mined in our previous work [7]. Rules are derived in periods of one month. We evaluated the performance on individual months and compared results against GTG and BCG. Against GTG, an absolute benchmark on the grouping capabilities is obtained. BCG serves as a baseline of a basic functional hierarchical grouping.

Subsequently, we evaluated performance at the end of the recordings by accumulating period rules. The rules of periods were added sequentially to the processing.

6. Results

BCG achieved a relative coverage of only 33% and produced 15 additional groups compared to the three main groups in GTG. While correctness at room level was perfect for BCG, correctness G was 0.96.

The average correctness results at the room level were: HMV = 0.91 ± 0.07, MV = 0.93 ± 0.06, and SC = 0.96 ± 0.04. Figure 4 shows the correctness results for each evaluation period. An almost identical correctness HMV and MV suggest that errors are related to the group members or their children, rather than to the head set. In other words, the head set is of the same label as the majority of the group members. The evaluation of the correctness grouping from a variable perspective yielded: G = 0.66 ± 0.11 and GSC = 0.72 ± 0.1. HTC achieved an average relative coverage of $75\% \pm 6$ p.p.. The correctness SC metric shows that the system can correctly separate user generated hierarchies from external environmental variable hierarchies. Figure 4 shows all four metrics for the individual periods.

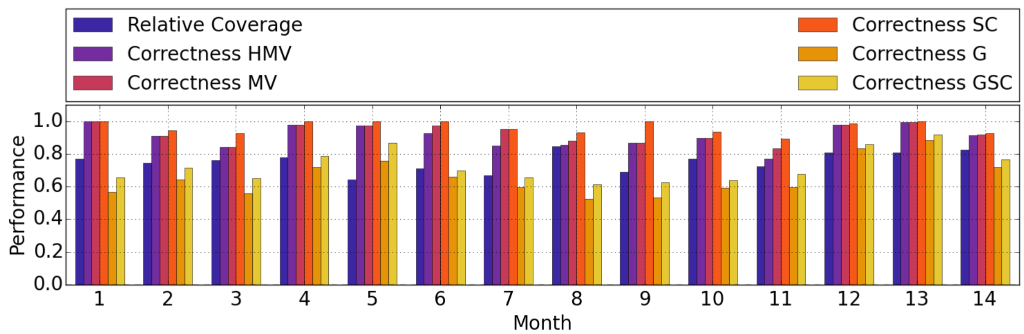

Figure 4: Performance results for relative coverage and correctness derived on individual monthly periods of the dataset. The correctness metrics are described in Sec. 5.3

The per-period group quality analysis is summarised in Table 3, where the distribution of group names with respect to group types is shown. There were overlaps between *Out light* and *Space* groups. Thus, outdoor light sensors were associated with variables from different rooms, which is consistent with previous observations [7]. The average number of basic groups is similar to the number of groups obtained with GTG. Equivalent and alternative groups may occur as a result of incomplete information from the rule extraction stage.

Figure 5 illustrates the cumulative system performance, i.e. when data from all preceding monthly periods was accumulated. Due to the installation changes (device additions, replacements) during the living-lab recordings, system performance varies. At the end of the last period, i.e. month 14, relative coverage was 96%, and correctness metrics were: HT=0.95, MV=0.95, SC=0.97, G=0.78, and GSC=0.81.

Finally, we analysed group quality in the cumulative setting. A total of 46

	Basic	Alternative	Equivalent	Overlapping
Out CO_2	0.07	–	–	–
Out humidity	0.07	–	–	–
Out light	1.57	–	–	0.80
Out temperature	0.07	–	–	–
Space	11.93	1.40	2.33	3.00

Table 3: Per-period group quality analysis showing average number of group name per period assigned to each group type. As expected, *Out light* groups overlap with *Space* groups. The average number of basic groups is similar to the number of groups obtained by the GTG.

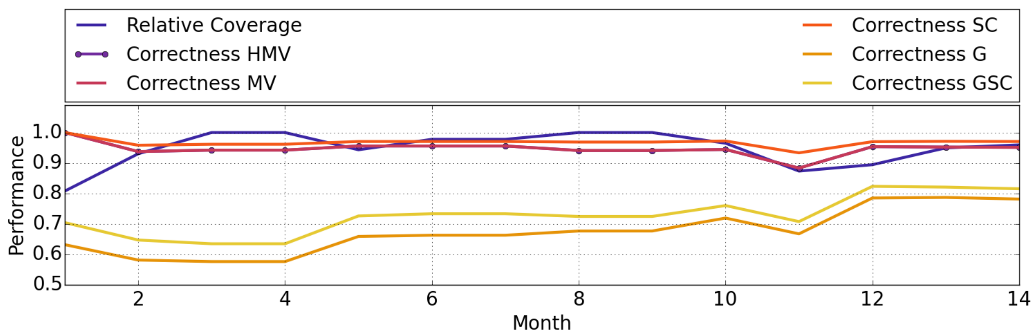

Figure 5: Performance results of the cumulative analysis for relative coverage and completeness. Performance variations could be explained by installation changes (device additions, replacements) during the living-lab recordings.

basic *Space* groups were obtained at the end of the last period. In conjunction with the high correctness scores, we interpret that the hierarchical groups contain correct variables, but that there was missing information in the mined rules to assemble the variables in the expected ten main groups.

WTC achieved an average relative coverage of $\sim 22\%$, whereas HTC reached 75% for the same data. For the cumulative analysis, WTC finished with a relative coverage of $\sim 70\%$, whereas HTC scored 96%. HTC was able to use more variables per grouping than WTC. At the room level, correctness metrics were 0.91 for HTC, and ~ 0.94 for WTC. However, correctness SC yielded an average of 0.96 for HTC, indicating that in the overall grouping of room level variables both algorithms have similar performance. HTC finished with a score of 0.95 in the cumulative analysis for correctness at the room level. In comparison, WTC reached only 0.75 for correctness, indicating that HTC was able to improve its performance as new information became available.

7. Discussion

The framework performance is closely related to the quality of mined rules and in turn, the stability of the monitored system is key to obtain good rules. Figure 5 shows a performance drop at month 11, which is due to adding office room R3 to the BMS. R3 inserted noisy sensor data, resulting in random actuation. The noisy sensors produced irrelevant variable changes and events, which subsequently yielded false associations and rules. In HTC, we did not apply filters on the incoming rules. The quality of group structures was therefore dependent on the quality of the extracted rules. Nevertheless, subsequent analysis months show that the performance recovered as variable errors declined. We used several variants of the performance metrics correctness and relative coverage to evaluate the framework and HTC algorithm. With these performance metrics we could ensure that the framework is suitable for different applications within a BMS.

In addition to the framework evaluation, we performed a group quality analysis, The group quality analysis provided insight into the internal structure of the resulting group sets. In the per-period analysis we found that the external environmental sensors related to air quality (humidity, CO_2, temperature) formed basic groups. The basic groups consisted of variables of the same type, e.g. all temperature sensors, or a combination of CO_2, temperature and humidity. The tendency of external environmental variables of producing basic groups could be explained with event delays. While user-driven changes in building context occur within a second, a change in air quality due to user activity could take several minutes. However, changes in external environmental variables are reflected without delay. For example, outside temperature events are almost simultaneous with indoor temperature events. Similarly, *Out light* basic groups contain light related sensors and actuators. The specialisation of the group is explained by a higher number of outdoor light events with respect to presence changes, i.e., during working hours only a handful of presence changes will occur, whereas the outdoor light could change many times during the same period.

Alternative and equivalent group types were not considered in the GTG, however, they are present in the BCG. BCG has only 33% relative coverage, thus, the resulting groups are incomplete. Similarly, 46 basic *Space* groups were found at the end of the cumulative period compared to ten in GTG and 25 in BCG. The results can be explained by incomplete information in the rules. Nevertheless, the relative coverage and correctness results indicate that correct variables are grouped. We can interpret the result as follows: Applications that depend on correct hierarchical information, i.e., people counting estimation, would obtain reduced performance due to the accumulating basic groups. However, applications that depend in the general association of the variables, i.e., cell variable finder, will function properly.

During the recording months, many device modifications were performed, e.g. adding new sensors and replacing broken devices. In addition, rules were updated, and configurations fine-tuned, resulting in changes to variable behaviour. The changes may have negatively affected the rule mining and group extraction. However, such modifications are part of the building life-cycle and thus correspond to an even longer evaluation in a real building. In the living-lab dataset, three office rooms with different properties were included: meeting room, four-person office, open space office. We consider these room types representative for office buildings in general.

A threshold filter was applied to the unified event time series with the goal to prevent continuous variables, e.g, desk power consumption, from triggering new events at every change in measured value. In this work, thresholds were set manually for selected variable types, e.g. for desk power consumption to distinguish computer on and off events, for light intensity sensor to detect on/off states in overhead lighting, etc. The threshold filter was not used for variables that may have several states, including external light intensity and temperature measurements. If the threshold filter would be omitted for any variable, additional events may be created that are uncorrelated with actual building state changes. As a result, the rule extraction may produce irrelevant rules. The tree trimming step, described in Section 4.1, was designed to eliminate connections created by any sources of noise and thus would eliminate parts of the tree that were created due to the noise events. A combination of threshold filter and tree trimming in our framework however still yielded trees that better related to actual context changes than the tree trimming step alone. Based on our results, we consider the thresholds robust enough to be applied for any variable of the same type. Thus, to implement our framework, filter thresholds could be specified in the BMS for each variable type and applied accordingly. Another option is to estimate filter thresholds from variable observations over a short operation time of the building, e.g. one week. As building states change will be reflected in the variables, robust estimation of boundaries between two variable states will become feasible.

In building environments, one of the strongest motivators to install or retrofit high sensor densities is energy saving due to more fine-grained automated control. An energy saving study conducted in our living-lab [21] showed that even the energy consumed by the sensors and actuators affects energy saving. Therefore, adding additional functionality to devices may not be feasible under energy saving restrictions. HTC deals with energy-constrained devices as the method does not impose additional functions on sensors and actuators.

Applications such as the variable failure management [7], benefit from the hierarchical groups extracted by HTC. In several cases, the extracted groups contained multiple outside light sensors that could be used interchangeably to control ceiling lights and blinds states if the original sensor fails. The group mining framework provides the basis to handle failure exchanges automatically.

In general, the mining framework could be extended to other processes, which can be described by a temporal sequence of events, and where the resulting associative temporal rules qualitatively describe a property of the process. The resulting hierarchical groups will indicate the intrinsic hierarchy of the variables. For example, if the framework is used in an automated production line, the resulting groups will indicate the association of sensors to each step of the production line.

For example, we envision that the mining framework could be applied in crowd sourcing applications. Participating users will no longer have to provide location using energy-expensive methods like GPS. Rather, the location could be estimated from the relation of low-power sensor measurements in smartphones.

8. Conclusions

The HTC algorithm successfully created hierarchical variable groups that resembled those of GTG and the BCG baseline. We showed how the grouping at low hierarchy levels, i.e. at cell and subcell levels, resembles corresponding groupings of GTG, which is a substantial improvement over the WTC algorithm. Moreover, HTC does not require tuning parameters, whereas WTC required a parametric search to find each hierarchy level. HTC extracts all hierarchy levels simultaneously from the variable association rules. When compared with WTC, HTC performed similarly or even better in all performance metrics used. The additional information stored within the hierarchical groups and the ability to describe variable interactions, led us to conclude that HTC is a better choice for the variable group extraction stage of the framework.

HTC yielded variable relations that were not considered in GTG or BCG baseline. These relations consisted mainly of external environmental sensors that created overlapping groups with room-related variables. These unexpected relationships can become useful for optimising system performance. For example, we found instances that associated external light sensors to user-operated windows. The association can be understood as an indirect measure of room temperature and user comfort. Consequently, the BMS could adjust the HVAC to compensate accordingly.

Acknowledgement

This work was kindly supported by the EU FP7 project GreenerBuildings, contract no. 258888.

References

[1] M. Milenkovic, O. Amft, Recognizing energy-related activities using sensors commonly installed in office buildings, in: SEIT 2013: Proceedings of the 3rd International Conference on Sustainable Energy Information Technology, Procedia Computer Science, Elsevier, Elsevier, 2013, p. 669–677. doi:10.1016/j.procs.2013.06.089.

[2] A. De Paola, M. Ortolani, G. Lo Re, G. Anastasi, S. K. Das, Intelligent management systems for energy efficiency in buildings: A survey, ACM Comput Surv 47 (1) (2014) 13:1–13:38. doi:10.1145/2611779.

[3] P. Jaramillo-Garcia, L. I. Lopera Gonzalez, O. Amft, Using implicit user feedback to balance energy consumption and user comfort of proximity-controlled computer screens, J Ambient Intell Humaniz Comput (2014) 1–15doi:10.1007/s12652-014-0222-2.

[4] J. Lu, D. Birru, K. Whitehouse, Using simple light sensors to achieve smart daylight harvesting, in: BuildSys 2010: Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building, BuildSys '10, ACM, New York, NY, USA, 2010, pp. 73–78. doi:10.1145/1878431.1878448.

[5] R. Priyadarshini, S. Reddy, R. Mehra, Design, development and deployment of multi-sensor node based wireless sensor network for campus monitoring, International Journal of Applied Engineering Research 10 (3) (2015) 5599–5615.

[6] V. Degeler, L. I. Lopera Gonzalez, M. Leva, P. Shrubsole, S. Bonomi, O. Amft, A. Lazovik, Service-oriented architecture for smart environments (short paper), in: Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference on, 2013, pp. 99–104. doi:10.1109/SOCA.2013.26.

[7] L. Lopera Gonzalez, O. Amft, Mining relations and physical grouping of building-embedded sensors and actuators, in: Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on, 2015. doi:10.1109/PERCOM.2015.7146503.

[8] A. Payal, C. Rai, B. Reddy, Analysis of radial basis function network for localization framework in wireless sensor networks, in: Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference -, 2014, pp. 432–435. doi:10.1109/CONFLUENCE.2014.6949349.

[9] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, N. Correal, Locating the nodes: cooperative localization in wireless sensor networks, Signal Processing Magazine, IEEE 22 (4) (2005) 54–69. doi:10.1109/MSP.2005.1458287.

[10] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. Anderson, P. N. Belhumeur, A theory of network localization, Mobile Computing, IEEE Transactions on 5 (12) (2006) 1663–1678.

[11] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, R. J. O'dea, Relative location estimation in wireless sensor networks, Signal Processing, IEEE Transactions on 51 (8) (2003) 2137–2148.

[12] M. V. Moreno, L. Dufour, A. F. Skarmeta, A. J. Jara, D. Genoud, B. Ladevie, J.-J. Bezian, Big data: the key to energy efficiency in smart buildings, Soft Computing (2015) 1–14.

[13] G. J. Wang, S. G. Jin, Application of association rule mining technology in collection and management of wireless sensor network node, in: Applied Mechanics and Materials, Vol. 685, Trans Tech Publ, 2014, pp. 575–578.

[14] N. Mohd Khairudin, A. Mustapha, M. H. Ahmad, Effect of temporal relationships in associative rule mining for web log data, The Scientific World Journal 2014.

[15] K. Koperski, J. Han, Discovery of spatial association rules in geographic information databases, in: Advances in spatial databases, Springer, 1995, pp. 47–66.

[16] B. L. W. H. Y. Ma, Integrating classification and association rule mining, in: Proceedings of the 4th International Conference on knowledge discovery and data mining(KDD-98), 1998.

[17] K.-C. Yin, Y.-L. Hsieh, D.-L. Yang, M.-C. Hung, Association rule mining considering local frequent patterns with temporal intervals, Appl. Math 8 (4) (2014) 1879–1890.

[18] M. Guillame-Bert, J. L. Crowley, Learning temporal association rules on symbolic time sequences, in: Proceedings of the 4th Asian Conference on Machine Learning, ACML 2012, Singapore, Singapore, November 4-6, 2012, 2012, pp. 159–174.

[19] M. Mesbahi, M. Egerstedt, Graph theoretic methods in multiagent networks, Princeton Series in Applied Mathematics, Princeton Univ. Press, 2010.

[20] M. Nikolić, Measuring similarity of graph nodes by neighbor matching, Intell. Data Anal. 16 (6) (2012) 865–878. doi:10.3233/IDA-2012-00556.

[21] L. I. Lopera Gonzalez, U. Großekathöfer, O. Amft, An intervention study on automated lighting control to save energy in open space offices, in: Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on, 2015, pp. 317–322. doi:10.1109/PERCOMW.2015.7134055.