Future Technologies Conference (FTC) 2017
29-30 November 2017 | Vancouver, Canada

Knowledge Mining Architecture for Localized Optimization of Smart Heating Systems

Bolatzhan Kumalakov
Department of Computer Science, School of Science and Technology, Nazarbayev University
Astana, Republic of Kazakhstan
School of Engineering Management
Almaty Management University
Almaty, Republic of Kazakhstan
b.kumalakov@gmail.com

Lyazzat Ashikbayeva
Nazarbayev University Library and IT Services
Nazarbayev University
Astana,
Republic of Kazakhstan
lyazzat.ashikbayeva@nu.edu.kz

***Abstract*—Improvement in resource consumption is among the many important targets that smart heating systems are aimed at achieving. Such a system automatically manipulates a household's physical artefacts (such as radiators, heating boiler, etc.) and changes their operational regimes to achieve this goal. This system can formally be represented by the parameter optimization problem. Although substantial research in this area has already been conducted, there is room for improvement on a collective scale. We adapted the context-aware parameter optimization architecture for geographically distributed machines to integrate multiple-peer knowledge into local optimization. This approach is a novel because it redefines knowledge mining and interpretation functionality, and it employs clustering and machine learning algorithms. Current paper is an attempt to explore sensitivity of heating system's local optimization to the mined knowledge, as it indicates if the method is applicable at all. A computational experiment confirms such sensitivity and provides basis for the future research.**

Keywords—Distributed knowledge management; context recognition; smart heating

I. Introduction

A smart heating system (SHS) is an automated infrastructure that automatically tunes a household's heating infrastructure to utilize resources efficiently. It consists of three major sub-systems: user interface, sensor network and heating flow manipulator. The user interface provides household members with a way to interact with the SHS controller, view current data, set desired room temperature and specify a time at which the temperature should be achieved. The sensor network monitors the heating system state, including water temperature, room temperature, outside temperature, etc. All these data are then used by the controller to manage heating flow manipulators and achieve the desired goal.

Recently, energy conservation has been an active research domain (see [1]-[4]). While much has been done, we believe there is room for improvement on a collective scale. That is, if multiple households make a coordinated effort, small improvements in the efficiency of each household could result in substantial impact on the community level. When performing its job, an SHS monitors a set of parameters, such as flow restrictions within individual radiators (typically 1 to 5 in standard household radiators), heating time, etc., and finding an optimal set of parameters on a household level is not a trivial task. Furthermore, coordinating collective parameter optimization is an even more challenging problem. Different households may have different piping, room designs and isolation materials, which results in the need to optimize divergent parameters. The goal of this study is to extract and mine "parameter-performance" data from multiple SHSs, extract knowledge from these data, and apply the knowledge in individual households without interrupting continuing operations.

The idea for our solution is inspired by [5]-[6], where the authors presented the central, context-aware parameter optimization architecture for geographically distributed machines. We build on this by partially shifting the information mining functionality from the server to the SHS. We also employ the OPTICS algorithm [13] to derive context knowledge and the SARSA algorithm [14] to mine information from raw statistical data. Experimental results suggest that the resulting architectural solution has the potential to solve the SHS problem, but its applicability should be verified through future research.

The rest of the paper is organized as follows. The problem definition is presented in Section II. Section III discusses the research methodology and formulates the question. Section IV presents a proposed solution design, while Section V concentrates on the experimental design and results. Finally, Section VI concludes the paper and provides suggestions for future research.

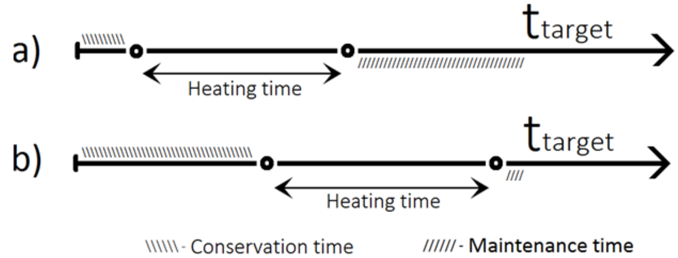

Fig. 1. The desired SHS behaviour on an individual household level, where a) is the first run and b) is the optimized behaviour. The goal is to minimize the time spent maintaining higher temperature and maximize conservation time to decrease fuel consumption.

II. Problem Definition

A. Desired System Behaviour

Residence areas in post-soviet countries are typically equipped with a local heating network. It consists of an autonomous station that heats circulating water and pipelines that run through the buildings in the network. In such a system, determining the monetary advantage of optimized power consumption within individual premises is not straightforward. Thus, motivating landlords to actively manage heat consumption is a challenging task, and even if some are convinced, it is not enough. Although rationalized heat consumption of an individual household surely adds value, it is the coordinated collective effort that maximizes it.

The solution to this problem is constant adoption of best heat conservation strategies across an entire heating network, such that slight improvements at the level of the individual household result in a considerable summative effect. The problem is that residents' commitments and lack of time or interest can jeopardize successful execution of such an activity. To overcome this obstacle, one might suggest building an analytical system that constantly evaluates consumption characteristics across houses and performs automated analysis to support local heating parameter optimization at every house.

Fig. 1 illustrates desired system behaviour on the level of a single SHS. The SHS here starts by heating the space in an inefficient way, as shown in Fig. 1(a). However, it learns how to meet user specifications (i.e., heat the room to the target temperature by the specified time) and maximize energy conservation time by utilizing knowledge, as shown in Fig. 1(b). Note that this example is illustrative rather than descriptive and does not provide any insight into the implementation details. Instead, it presents an end-user perspective on the design goals of the system.

B. Knowledge Management Requirements

In [7], Vayrynen et al. advocate data-driven performance analysis of industrial machines and experimentally justify the benefit of knowledge utilization in parameter tuning. Unfortunately, such knowledge is not universal and should be interpreted with the context of the specific application in mind.

Kannisto et al. attempt to solve this problem by designing a centralized, context-aware local optimization architecture [5], which collects feedback on operational parameters from a fleet of wood cutting machinery, calculates optimal values and distributes them across the entire fleet. Then, it is up to the machine to adjust the calculated values to its environment. Given a high number of physical machines, this process constantly forces parameters to the point of optimality, even if contexts evolve over time.

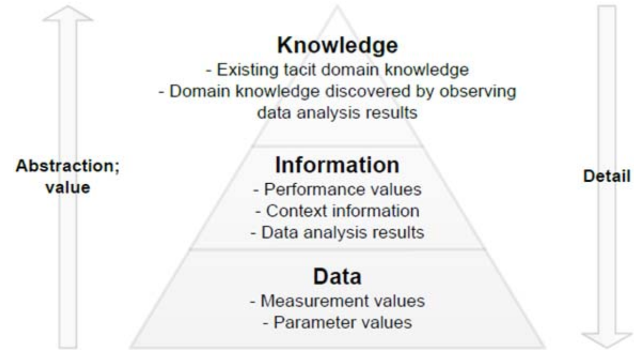

Fig. 2. Abstraction levels, which lay down the conceptual background for the knowledge mining approach, of the proposed SHS model. At the level of the individual SHS, we suggest retrieving all Information but only some Knowledge to mitigate context complexity.

In our case, unlike in the current literature, the heating system context consists of both fixed factors (radiator type, serviced household area, etc.) and changing factors (additional sources of heat, temperature outside, etc.). Hence, there is an additional complexity of not being able to compute a guaranteed optimal solution even for households with identical fixed factors. Thus, in addition to adopting the core idea, we must also determine how the system should mitigate additional complexity when mining knowledge.

First, let us refer to the "knowledge pyramid" [8], illustrated in Fig. 2. In Kannisto et al., physical machinery obtains parameters and generates feedback (corresponding to the data layer and part of the information layer in the pyramid), which is then analytically processed at the central server to derive knowledge (information and knowledge levels).

One intuitive solution to the mitigation problem is to let an individual machine seek its own sub-optimal parameter and let it choose between it and global optimum on a case by case basis. For instance, let us assume that the user wants a room warmed up by +20 degrees Celsius within some specified time, and the optimum (global) radiator temperature for that is +50 degrees. The individual SHS applies this value and achieves the result within target time +10 minutes. Thus, next time, the SHS uses +55 degrees, meets the targeted time and reports feedback to the server. The system might not necessarily update the global optimum due to statistical insignificance of the reported difference. However, the SHS does not override its local sub-optimum on the next run unless the new global optimum deviates significantly. Fig. 3 presents a schematic view of this concept.

Data in this case include both current sensor readings and local sub-optimal values. Aggregated context values represent knowledge from the cloud analyser and are used by the local context engine to determine which set of values to use and to provide input for utilized operational values.

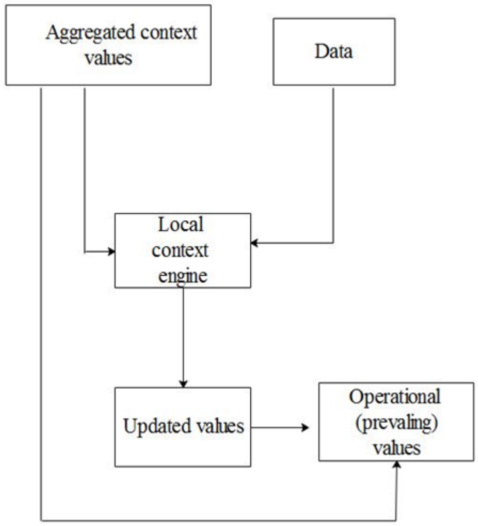

Fig. 3. Illustration of how context is utilized in generation of parameter values. Updated values on the diagram represent the idea that the SHS can pick values to learn that differ from both current sub- and global optima. This corresponds to changing the radiator temperature from +50 to +55 in the example.

The general idea here is that every SHS sticks to its own experience (best practice) unless there is a high potential benefit to changing, as promised by aggregated knowledge. The reason for this approach is that global optima do not necessarily better suit the unique context of an individual SHS but instead carry the risk of misbalancing the carefully tuned system. Fig. 4 illustrates the flow of information and resulting knowledge for this approach.

III. Research Context

A. Research Methodology

To carry out this project we employ the *constructive design science research methodology*, as described in [16]. The core idea is to break necessary work down into iterations. Each iteration takes an input hypothesis and tests it by implementing and evaluating a software artefact. The hypothesis is being proven or disproven results in novel knowledge, which is then used to formulate a new hypothesis, triggering the next iteration. This process continues until the overall research objective is met.

B. Research Question

The current paper presents experimental results of the iterative process that investigates the following hypothesis: “Globally optimized SHS context parameters have significant impact on local optimization but do not overrule local optima if the potential benefit is lower than the locally known result.”

This is of key importance because the validity of the entire proposed architecture depends on how sensitive parameter optimization is to the change in knowledge. Hence, at this stage we must answer the following research questions:

1) How does the system learn, represent and store knowledge?

2) How does an individual SHS interpret and utilize the knowledge?

IV. Solution Design

A. Context Recognition and Representation

The notion of context (also called environment) has been extensively studied in the setting of smart home systems [9]-[12]. It has been modelled using a range of methods from formal to heuristic. In our case, building a formal model would require knowledge of each household’s physical environment or at least its significant objects and their properties (such as materials used, area, etc.). Unfortunately, this is infeasible because major factors differ from house to house and estimating their influence would not be reasonable or straightforward.

Instead, we approach the problem by treating the context as a set of optimized parameters with relevant metadata. To determine these values, we consider system inputs and their resulting outputs. Under this definition, the main function of context is to serve as the aggregated experience of multiple lower level optimizations. The best way to conceptualize this is to imagine several people performing a repetitive task. Each person learns how to perform it best based on their experience and skills. However, if you compare their efficiency, you would likely discover that some people perform faster because they came up with a less costly method than the others. Then, sharing better practices with others brings the average execution time down for the entire population. Equation (1) presents a formal definition of the context.

$$\mathit{Context} = \{t_{out}, t_f, T_\mu, \Psi_\mu\} \tag{1}$$

Here, t_{out} is the outside temperature, t_f is the flow temperature, $T_\mu = \{\tau_0 .. \tau_\mu\}$, $\mu = 0...N$ is the set of heating speeds under flow restrictions μ, and $\Psi_\mu = \{\psi_0 .. \psi_\mu\}$, $\psi = 0...N$ is the set of fuel consumption values under flow restrictions μ. Each τ_μ is stored as a tuple:

$$\tau_\mu = (\mu, \varphi_\mu) \tag{2}$$

Where, μ is the restriction value, and φ_μ is the heating speed under the following restriction:

$$\varphi_\mu = \Delta t^\mu / \mathrm{time} \tag{3}$$

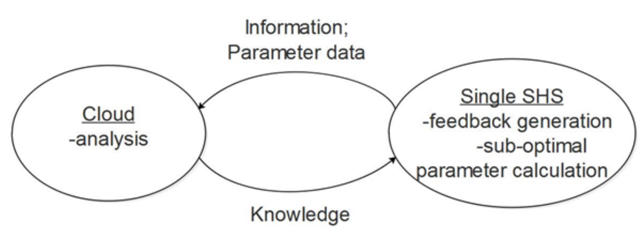

Fig. 4. Unlike in the literature, each single SHS also computes sub-optimal parameters and can choose which version to use. Parameter data here, if used, is feedback on global, optimized parameters. Information is a set of sub-optimal parameter values.

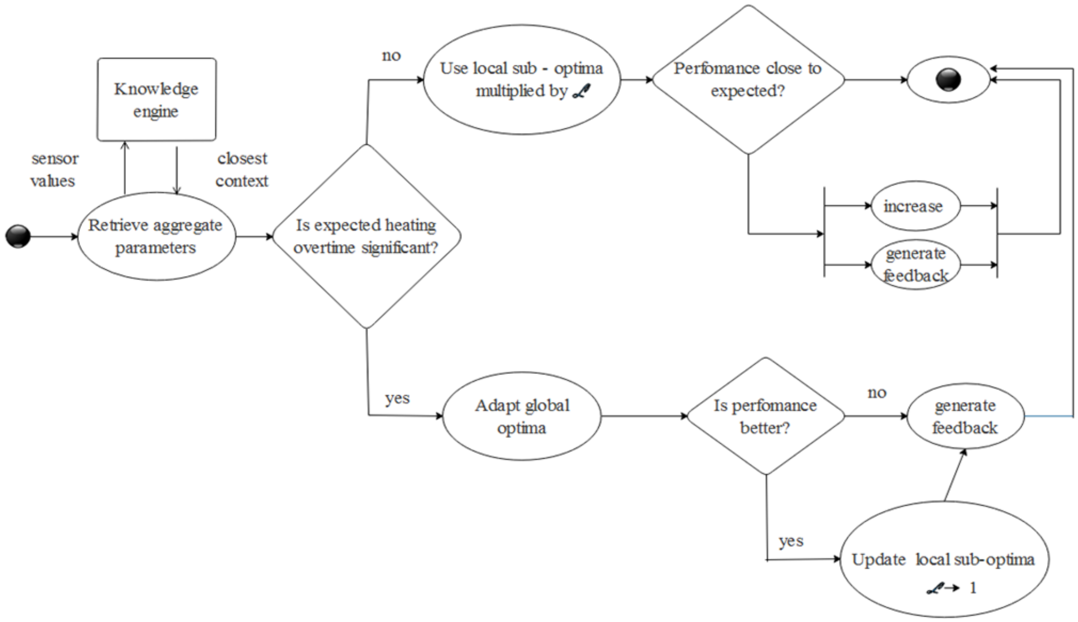

Fig. 5. Parameter feedback generation flow within a single SHS.

Here, Δt^{μ} denotes the temperature change over the considered time span. Each ψ_μ is also a tuple, which stores μ and the associated fuel consumption c_μ:

$$\psi_\mu = (\mu, c_\mu)$$

Context recognition occurs in two steps. First, when the system is initiated for the first time, a cloud based analytics tool receives credentials and {tout, tf, Tμ, Ψμ} parameters for each SHS. Then, the system runs an OPTICS algorithm [13] to identify clusters and their corresponding cluster centres. In this way, approximation values are derived based on heat and fuel consumption characteristics rather than the construction specifics of each individual household.

Second, a context recognition tool accepts SHS feedback to global parameters, and if more than 15% of cluster objects are flagged as different, it re-computes them using new sub-optimal values that come from each SHS.

B. Feedback Generation and SHS Learning

The feedback generation flow is illustrated in Fig. 5. First, the machine retrieves the context from the cloud knowledge engine. To do so, it reports current environmental characteristics, classifies the context according to known clusters, and returns values for the closest context.

The SHS computes expected heating time and cost using updated knowledge and then decides whether to use its local sub-optimal parameters or adjust to context parameters. Here, α is the learning coefficient, which determines willingness of the system to experiment with parameters. When α→0, the system is reluctant to learn, but as α→1, it becomes increasingly willing to experiment. All feedback, generated as shown in the diagram, is sent to the cloud to adjust the existing context class structure as described above. The idea here is that experimenting with local sub-optima will eventually cause feedback disruption that is large enough to identify new classes of contexts, and thus, the system will evolve. This phenomenon forces the analyser to reconsider and even abandon context classes if enough of the SHSs report them to be inefficient. In this way, the overall system tends to forget about the existence of classes which do not add value to the system.

To learn local sub-optima, however, SHS employs a modification of the SARSA algorithm [14] as follows:

```
Initialize Q(s, τ_μ);
for each episode do
        Read s;
        If connection then
                retrieve context;
                compute expected benefit;
                if benefit << last benefit then
                        adapt context;
                else
                        use local optima;
                end
        else
                use local optima;
        end
Choose a using s and Q(s, τ_μ);
while s is not terminated do

        if performance << expected then
        Q_t+1(s,τ_μ) <- Q_t(s,τ_μ) + α_high(r_t+1 + γQ_t(s_t+1,
                         τ_μ t+1) - Q_t(s,τ_μ))e_t(s,τ_μ);
        else
        Q_t+1(s,τ_μ) <- Q_t(s,τ_μ) + α_low(r_t+1 + γQ_t(s_t+1,
                         τ_μ t+1) - Q_t(s,τ_μ))e_t(s,τ_μ);
        end
        τ_μ <- τ_μ';
        end
    end
```

Where, s denotes the environment state (sensor data), and $Q(s, \tau_\mu)$ is a Q-values table, which stores mappings between known environment states and τ_μ. *a* denotes an action (SHS manipulator regimes, which implement τ_μ), α is the learning coefficient, and γ is the discount factor; *r* denotes a reward, and the weighted sum of error gradients is $e(s, \tau_\mu)$.

V. Experiment Design and Results

A. Infrastructure and Settings

To test the designed system, the following infrastructure was set up:

1) The sensor network collects the room temperature and the temperature of water flowing through the heating radiators separately for every room. Sensor devices are connected to the Arduino Nano microcontroller, which constitutes a sensor node and transmits data over Bluetooth connection to the Raspberry Pi 2 data node. The sensor network was implemented using architectural and technological solutions from [15].

2) Every radiator is equipped with an automatic valve that can reduce or restore water flow on command from Raspberry Pi 2 via Bluetooth connection. Control of these valves realizes the automatic manipulation system.

The SARSA algorithm is implemented using the Python 3 programming platform, run under the Raspberry Pi comparable Linux distribution. The cloud analyser software is implemented using Java and run on a PC connected to LAN with a data transfer speed of 10 Gb/sec.

It is important to note limitations of the experiment. First, software implementation is basic; a more sophisticated implementation is a subject for future work. At this stage, most controllers use “naive” logic and basic sketches (scripts), most of which were borrowed from multiple coding forums and open resources. The same applies to the analyser implementation. Second, data is collected from only two sensor networks, which may not cause enough context class divisions to make certain conclusions about overall system behaviour.

Nonetheless, at this stage, we are seeking to investigate if the system exhibits expected behaviour and whether the work being done maintains an appropriate perspective or not.

B. Results and Discussion

First, we test the prototype by taking readings from both sensor networks as is; then, we introduce a random multiplier, which takes values between 0.1 and 1 but only takes effect after initial the $Q(s, \tau_\mu)$ is computed, to test how sensitive the system is to context changes. Table 1 presents the statistics acquired over 97 days.

From the presented data, one might conclude that the system is sensitive to context changes, as it requires a many more optimization runs if cluster members generate a significant difference. For instance, when both sensor networks showed close results, collectively, they required 37 iterations to keep sub-optima values close to each other. Of these 37, 23 constituted initial learning, that is, cases with high α when both networks independently try new combinations and combine learned values into the knowledge base. The remaining 14 are corrective iterations that occur when the outside temperature changes, and only 6 of these used context as input due to expected additional benefit. In this case, when we used a random multiplier and re-ran the algorithm (after the system learned), the system required almost three times as many corrective iterations to adjust values and used context data 10 times more often.

TABLE I. Experiment Results

Test item	**Sensor readings**	
	Real	***With multiplier***
Number of times context was used to compute optimal solution	29	68
Number of iterations required to learn local optima at the start	23	23
Number of iterations spent to adjust values	37	94

While the current result is promising, it should be reinforced with experiments on wider fleets of SHS. Moreover, when properly advanced, the current simplistic implementation of the tested prototype might add additional corrections to the influence of the context on local performance. Nonetheless, if the properties discovered here hold after accounting for these corrections, the derived architecture promises to deliver features that are targeted by current research.

VI. Conclusion

In this article, we propose an enhanced distributed context-aware parameter optimization architecture to first mine and then integrate knowledge into local parameter optimization for an SHS.

Practical novelty here comes from a redefinition of knowledge mining and interpretation functionality, which in turn redefines the corresponding architecture level design and algorithmic implementation. The core algorithms used are conventional OPTICS and modified SARSA.

The computational experiment suggests a high sensitivity of local parameter optimization to the change in mined knowledge, but this may be arguable due to the simplicity of the experiment. Thus, comprehensive conclusions would demand data from experiments that incorporate more complex measurements and parameters, such as fuel consumption.

As a result, future project efforts will be dedicated to enhancing the architecture, advancing its prototype implementation, and obtaining performance data that realistically reflect its operational characteristics in real settings.

References

[1] Y. Krozer, “Innovative offices for smarter cities, including energy use and energy-related carbon dioxide emissions”, Energy, Sustainability and Society, vol. 7 (1), 2017.

[2] A. Bhati, M. Hansen, C.M. Chan, “Energy conservation through smart homes in a smart city: A lesson for Singapore households”, Energy Policy, vol. 104, pp. 230-239, 2017.

[3] M.A. Fotouhi-Ghazvini, J. Soares, O. Abrishambaf, R. Castro, Z. Vale, “Demand response implementation in smart households”, Energy and Buildings, vol. 143, pp. 129-148, 2017.

[4] M.F. Ibrahim, M. Mohamed, B.H. Far, “Measuring the effectiveness of zonal heating control for energy saving”, IEEE International Conference on Systems, Man, and Cybernetics, pp. 132-136, 2016.

[5] P. Kannisto, D. Hästbacka, S. Kuikka, “System Architecture for Mastering Machine Parameter Optimisation”, Computers in Industry, vol. 85, pp. 39-47, 2017.

[6] P. Kannisto, D. Hästbacka, “Enabling centralised management of local sensor data refinement in machine fleets”, In Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, v.3, pp. 21-30, 2016.

[7] T.V. Vayrynen, S. Peltokangas, E. Anttila, M. Vilkko, “Data-driven approach for analysis of performance indices in mobile work machines”, In: DATA ANALYTICS 2015, The Fourth International Conference on Data Analytics, pp. 81-86, 2015.

[8] J. Rowley, “The wisdom hierarchy: representations of the DIKW hierarchy”, Journal of Information Science, vol. 33 (2), pp. 163-180, 2007.

[9] A.R. Pratama, G.D. Putra, “An Infrastructure-less Occupant Context-Recognition in Energy Efficient Building”, 6th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 2014.

[10] S. Franco, V.R. Mandla, K. Ram Mohan Rao, “Urbanization, energy consumption and emissions in the Indian context A review”, Renewable and Sustainable Energy Reviews, vol. 71, pp. 898-907, 2017.

[11] M. Khan, B.N. Silva, C. Jung, K. Han, “A context-aware smart home control system based on ZigBee sensor network”, KSII Transactions on Internet and Information Systems, vol. 11 (2), pp. 1057-1069, 2017.

[12] G.M. Toschi, L.B. Campos, C.E. Cugnasca, “Home automation networks: A survey”, Computer Standards and Interfaces, vol. 50, pp. 42-54, 2017.

[13] R.J. Campello, D. Moulavi, A. Zimek, J. Sander, “A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies”, Data Mining and Knowledge Discovery, vol. 27 (3), pp. 344-371, 2013.

[14] Y. Wang, T. Li, C. Lin, “Backward Q-learning: The combination of Sarsa algorithm and Q-learning”, Eng. Appl. of AI, vol. 26, pp. 2184-2193, 2013.

[15] C. Bell, “Beginning Sensor Networks with Arduino and Raspberry Pi”, Apress, p. 372, 2013.

[16] K. Piirainen and R. Gonzalez, “Seeking Constructive Synergy – Design Science and the Constructive Research Approach”, Lecture Notes in Coomputer Science, vol. 7939, pp. 59-72, 2013.