

Agent-based Architecture for Demand Side Management using Real-Time Resources' Priorities and a Deterministic Optimization Algorithm

Luis Gomes ¹, João Spínola ¹, Zita Vale ² and Juan M. Corchado ³

¹ GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development
Polytechnic of Porto (IPP),
Porto, Portugal
lufog@isep.ipp.pt, jafps@isep.ipp.pt

² Polytechnic of Porto (IPP),
Porto, Portugal
zav@isep.ipp.pt

³ BISITE research Group, University of Salamanca, Salamanca 37007, Spain
Air Institute, IoT Digital Innovation Hub, Salamanca 37188, Spain
Department of Electronics, Information and Communication, Osaka Institute of Technology, Osaka 535-8585, Japan
corchado@usal.es

Abstract — Microgrids and smart grids are largely accepted concepts in power energy systems. They bring an innovative and distributed view to the old centralized system. This demands a more active participation from end-consumers, that can be achieved by using demand response and demand side management. In this paper it is proposed a solution for demand side management involving and agent-based architectures that was deployed in a small office. The deployment integrated an algorithm for generation and consumption balance with real-time contextual resources' priorities. The deployment's overall results, from a winter and a summer day, are presented in this paper.

Index Terms — demand side management, load optimization, load priority, real-time optimization

I. INTRODUCTION

The recent introduction of new energy generation technologies, mostly consisting of renewables, has triggered significant developments in future sustainable solutions for power energy systems [1]. Moreover, these developments gave rise to two important concepts which became the most promising solutions, namely, microgrids and smart grids. Microgrids are small energy infrastructures that comprise loads and distributed energy resources, they maintain a balance between these two by using renewable energy, such as wind turbines and photovoltaic panels [2]. The definition of smart grids concept is more extensive, since it includes microgrids, i.e. the concept of smart grids describes the evolution of power energy systems from their traditional form, centralized activity and top-down communication flow, to a decentralized operation which is achieved through complete communication between entities and the use of monitored and controllable infrastructure that allows real-time analysis and intervention [3].

Power energy systems are not only being revolutionized at the infrastructural level, where new low-voltage and distributed

generation resources are introduced and more efficient consumption loads are used, but also at the organizational level, changing the interaction and participation of end-consumers in the electrical grid management [4]. Consumers are being encouraged to actively participate in the smart grid and in the microgrid. One of the keys to the new paradigm is Demand Side Management (DSM) where consumers can change their consumption profiles [5]. Additionally, Demand Response (DR) programs have been gaining popularity due to its many applications; it corresponds to changes in the consumer's load consumption in response to price-signals or monetary incentives, sent by operators of demand response promoters/organizers.

In this context, energy management systems appear at several implementation levels, from operators to small consumers, and focus on two major applications: distributed or on-site generation [6] and demand response [4]. These systems are platforms that optimize the consumer's energy use according to their preferences, the monitored data, and the available generation, promoting efficient use of energy.

This paper proposes three contributions to DSM systems, by providing a complete solution that can be deployed and used in buildings. The three major contributions are: a real-time contextual resources' priority mechanism, an algorithm for generation and consumption balance, and an agent-based software architecture for DSM.

To enable DSM, the paper will first detail and propose how priorities can be given to individual electrical resources in real-time. The priorities will merge contextual sensor data and users' preferences. The proposed balancing algorithm will then use the priorities to optimize the energy consumption to be at the same level as the energy generation. The balance algorithm is integrated in an agent-based software that was deployed in a small office. The agent-based architecture enables not only

DSM but also the ability to integrate the controlled building in a community, such as microgrids and/or smart grids. This paper presents the real deployment of its contributions in an office.

This paper is divided into seven main sections. Following this first introductory section (Section I), Section II proposes equations for real-time contextual resources' priorities using users' preferences. Section III presents the proposed algorithm for generation and consumption balance. The proposed agent-based architecture for DSM is presented in Section IV. The deployment in a small office is detailed in Section V while the results are presented in Section VI. Section VII presents the main conclusions drawn from the results of this work.

II. REAL-TIME CONTEXTUAL RESOURCES' PRIORITIES

The users' comfort is a variable that must be taken into account in resources optimization algorithms and is usually modulated as a weight representing the resource's priority. This can be seen in works such as [7], [8] and [9]. In [7] a real-time optimization is proposed using the forecasted renewable sources availability and market prices. In [8] an optimization to maximize users' comfort is proposed. A home energy optimization to minimize users' discomfort and energy costs while complying with residential electrical resources constraints is proposed in [9]. Because resources' priority varies in time and context, this paper proposes real-time contextual priorities for light resources and heating and cooling resources. The proposed real-time contextual priorities consider sensor data as well as user preferences.

The user preferences are values set by the building's administrative user. These can assume multiple unit measures, such as lux and degrees Celsius. The combination of multiple users' preferences must be done by the building's administrator. The combination of multiple users' preferences is not performed by the proposed system, because it lacks the ability to identify which users are located in a given zone. The anonymity of users prevents their identification and consequently the combination of multiple users' preferences.

A. Light Priority

Office buildings are mainly occupied during the day. However, lighting can have a big impact on the overall consumption. According to the National Optical Astronomy Observatory of United States of America, the illumination needed to perform office and computer work is 500 lux [10]. This value almost demands the use of artificial light in the majority of buildings.

This paper proposes equation (1) to calculate the priority of each light considering its context and users' preferences where:

- *occupancy* – is a binary number (1 or 0) indicating if there are users in the light actuation zone (i.e. zone where the light has impact in the visibility);
- *curLumen* – real-time zone's natural light (i.e. lux);
- *userPrefLux* – user preference for the light zone (i.e. lux);
- *elasticity* – percentage representing the elasticity of the user regarding his/her preference (i.e. how much the light can differ from the user's preference without having a negative impact);

- x – combination between real-time context values and user's preferences, calculated in equation (2);
- *smooth* – represents the opening of the formula curve, smoothing the variation between high and low priorities.

$$f(x) = \begin{cases} 1, & x < 0 \\ \text{occupancy} \times \frac{2}{e^{x/\text{smooth} \times (1-\text{elasticity})} + 1}, & x \geq 0 \end{cases} \quad (1)$$

$$x = \text{curLumen} - (\text{userPrefLux} - \text{userPrefLux} \times \text{elasticity}) \quad (2)$$

Figure 1 shows $f(x)$ of equation (1) for a user's preference of 500 lux and an *elasticity* between 0% and 10%. When elasticity is 0% (blue line), the light priority has maximum value (one) if the zone's lux is below user's lux preference (i.e. 500 lux). If a user has elasticity above 0%, it means that the user preference is not entirely mandatory and can be exceeded in extreme cases. The yellow area of Figure 1 represents a 10% elasticity.

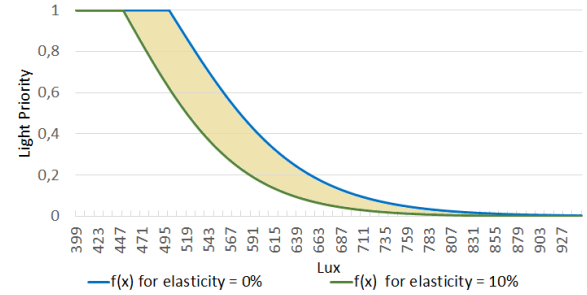


Figure 1. Light priority for *smooth* = 70

B. Heater and Cooler Priority

Other commonly used resource in buildings is the heating, ventilation and air conditioning (HVAC) resources. Their consumption can have a big impact in the building's overall consumption. Therefore, works such as [11], [12] and [13] propose and simulate HVAC optimization algorithms for home and offices. In [11] the users' comfort and HVAC efficiency are taken into account to minimize the HVAC consumption during demand response events by shifting resources consumption. A dynamic demand response controller is proposed in [12] for real-time pricing using user's threshold prices. In [13] an HVAC optimization algorithm is proposed taking into account technological, hygienic, acoustic, energetic, material, architectural, constructional, safety and reliability limitations.

The ability to control HVAC systems enables the use of such systems for demand response participation and real-time prices events, as seen in [11] and in [12]. Therefore, the application of real-time contextual priorities to HVAC resources benefits optimization and decision-making algorithms.

For a real-time operation, this paper proposes the application of equation (3) to calculate the priority of each HVAC considering the context and users' preferences, where:

- *userPreferenceTemp* – user's preference expressed in degree Celsius;
- *curTemp* – zone's temperature;

- *elasticity* – percentage representing the elasticity of the user regarding his/her preference;
- *x* – combination between real-time context values and user's preference, equation (4);
- *smooth* – represents the formula curve opening, smoothing the change between high and low priorities.

Equation (3) application changes according to the type of resource used. If a heater is used, equation (3) only applies for $x < 0$, if a cooler is used, equation (3) can only be used for $x > 0$. For situations where equation (3) cannot be used, $f(x)$ will have the value 0; for example, when a heater is used and $x > 0$ the heater's priority will be 0.

$$f(x) = 1 - \frac{4e^{\frac{x \times \text{smooth} \times (1 - \text{elasticity})}{2}}}{\left(e^{\frac{x \times \text{smooth} \times (1 - \text{elasticity})}{2}} + 1\right)^2} \quad (3)$$

$$x = \text{currentTemp} - \text{userPreferenceTemp} \quad (4)$$

Figure 2 shows $f(x)$ of equation (3) using a user temperature preference of 24 °C. The yellow area represents a 10% elasticity. Figure 2 represents an air-conditioner that can heat and cool the room/zone. Otherwise, the chart should not be symmetric. The blue line represents que normal HVAC priority, while the green line represents the minimum possible priority using a 10% elasticity.

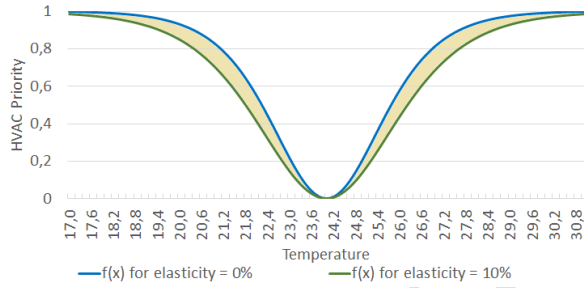


Figure 2. HVAC priority for *smooth* = 2

III. PROPOSED ALGORITHM FOR GENERATION AND CONSUMPTION BALANCE

The proposed algorithm for generation and consumption balance is an evolution of the algorithm present in [14] and [15]. The new proposed algorithm is able to run in real-time in a single-board computer, while the previous version was built to reduce energy consumption according to a target and was not prepared to balance generation and consumption in real-time.

The algorithm takes into consideration the software and hardware limitations of a single-board computer and the limitation of the GNU Linear Programming Kit (GLPK). GLPK, used as solver, allows the solving of large-scale linear programming and mixed integer programming problems. Java Optimization Modeler (JOM) is also used to provide a layer between Java language and GLPK, simplifying the solver usage and configuration.

The proposed approach considers two types of resources: discrete and variable. As to the mathematical formulation used for optimization, it considers the available on-site generation (P^{Gen}), the current consumption (P^{Cons}) and two demand

response programs, load reduction (P_i^{var}) and load curtailment (P_i^{dis}). The first is applied to dimmable controlled loads (i.e. variable), while the latter acts on loads that can only be turned on/off (i.e. discrete). Equation (5) shows the objective function used for balancing the consumption and generation and considering the minimization of impact on consumer priorities. In equation (5), n_{var} represents the number of loads capable of load reduction and n_{dis} represents the number of loads capable of load curtailment. User priorities are represented by W_i^{var} and W_i^{dis} . The w is a weight that prioritizes the balance between generation and consumption.

$$\begin{aligned} \text{minimize} = & \sum_{i=1}^{n_{var}} |P_i^{var}| \times W_i^{var} + \sum_{i=1}^{n_{dis}} |P_i^{dis}| \times W_i^{dis} + \\ & w \times \left[P^{Gen} - P^{Cons} + \sum_{i=1}^{n_{var}} P_i^{var} + \sum_{i=1}^{n_{dis}} P_i^{dis} \right] \end{aligned} \quad (5)$$

The algorithm intends to minimize the gap between generation and consumption, when the available generation is lower than actual consumption. This ensures that even though it is not possible to supply all loads with on-site generation, the operational costs are only the necessary ones. In equation (6), the constraint related to the balancing is considered, by looking at the available generation and demand response programs. Variable w_p is a percentual weight given to the generated energy to avoid a match overfitting between consumption and generation. The photovoltaic generation has overtime fast fluctuations, if an overfitting match is performed, there is no guarantee that the consumption will be kept below generation.

$$w_p \times P^{Gen} \geq P^{Cons} - \sum_{i=1}^{n_{var}} P_i^{var} - \sum_{i=1}^{n_{dis}} P_i^{dis} \quad (6)$$

Equation (7) reflects the load curtailment program where loads participating in it are turned on/off. The modeling of this program considers a binary variable (X_i^{dis}) that the optimization uses to define if i load will be turned on or turn off. The $P_i^{dis_cons}$ can take two different values, if consumption is bigger or equal than generation, then $P_i^{dis_cons}$ is the real-time consumption of i load, otherwise $P_i^{dis_cons}$ is the possible consumption of the load represented with a negative number – in other words, if a load consumes 500 W, then $P_i^{dis_cons} = -500$.

$$P_i^{dis} = P_i^{dis_cons} \times X_i^{dis}, \forall i \in \{1, \dots, n_{dis}\} \quad (7)$$

Equations (8) and (9) are constraints implemented to ensure absolute values for the load reduction variable ($|P_i^{var}|$) and the load curtailment variable ($|P_i^{dis}|$). These constraints are needed because GLPK is used with JOM that do not provide a syntax for absolute values in the objective function.

$$\begin{aligned} P_i^{var} & \leq |P_i^{var}| \\ -P_i^{var} & \leq |P_i^{var}| \end{aligned} \quad (8)$$

$$P_i^{dis} \leq |P_i^{dis}| \quad (9)$$

$$-p_i^{dis} \leq |p_i^{dis}|$$

The proposed algorithm is light enough to run in a Raspberry Pi 3 Model B in less than two seconds. This was the main concern and objective within the development of this algorithm. The load priorities, represented by W_i^{var} and W_i^{var} , are achieved using Section II equations (1) and (3) for real-time contextual resources' priorities.

IV. PROPOSED AGENT-BASED ARCHITECTURE FOR DEMAND SIDE MANAGEMENT

The use of Multi-Agent Systems (MAS) in microgrids appears to be a good fit, where each microgrid's player is represented by an agent of the MAS. This paper will focus on the architecture of a building's agent considering previous works. The proposed agent-based architecture will be able to be integrated in a MAS; of a microgrid or smart grid. However, the focus of this paper is to analyze the proposed agent-based architecture running in standalone to manage the electrical resources of a building.

The proposed architecture, shown in Figure 3, is divided into six interconnected layers that enable the full energy management of a building. Depending on the agent goal and desired functionalities, some layers can be left outside the implementation scope. Figure 3 also presents examples of technologies that can be used to implement each layer. The next subsections will describe the layers from the bottom-up.

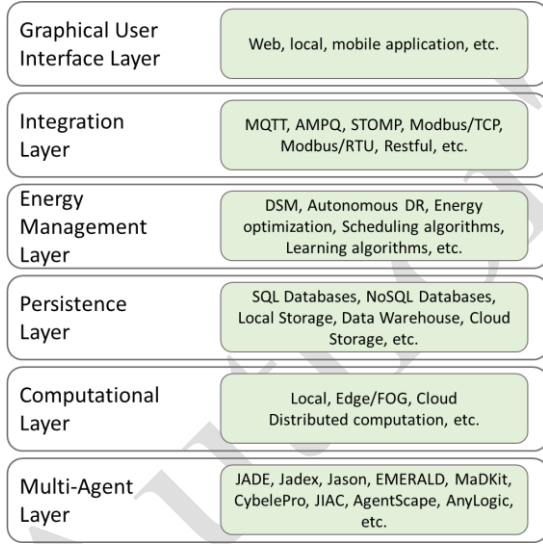


Figure 3. Agent Architecture Layers

A. Multi-Agent Layer

The Multi-Agent Layer is implemented using a framework, such as JADE or Jadex. Agent communications are implemented in this layer in order to allow it to communicate with other agents. The registrations and connectivity in MAS are also implemented in this layer.

The motivation of this layer comes from previous works where energy management systems for buildings were integrated in MAS for microgrids and smart grids players modeling [16-19]. In [16] a MAS for microgrid is proposed for distributed consensual optimizations where communication delays are

considered. A distributed cyber physical microgrid model for real-time tasks is proposed in [17]. In [18] a MAS for islanded microgrids is simulated. A survey regarding the deployment of MAS in smart grids is presented in [19] where several agent layers are presented. This paper's agent-based architecture will be part of the Prosumer Agents layer identified in [19].

B. Computational Layer

The computational capability of the agent is usually taken for granted while using today's computers. However, to deploy agents in end-consumer's buildings, smaller and cheaper computational platforms should be taken into consideration, such as single-board computers that have smaller computational power. The Computational Layer determines the possibilities of the agent to process complex operations, for instance, forecasting algorithms.

The Computational Layer allows the usage of distributed computing, edge/FOG computing and cloud computing [20]. A combination of computing technologies is used to empower the agent processing speed and quality. The integration of real-time data and IoT devices, in the Integration Layer, demands significant computational power to process everything. By using external computational resources, more complex and efficient algorithms for energy management can be used. This layer will orchestrate local and remote data processing.

Security measures must be considered when dealing with external computation, this is a significant issue when dealing with end-consumer data that can jeopardize user security and privacy. The security of data passing from one side to the other must be addressed [21].

C. Persistence Layer

To build an efficient system for energy management and DSM, is important to store data. In energy management systems, is important the use of: forecasting algorithm as seen in [22], data analysis as seen in [23], and user profiles as seen in [24]. Therefore, the Persistence Layer is needed to feed algorithms with historical data.

The Persistence Layer is divided into two levels: local storage, and external storage. The local storage stores the data in the same hardware where the agent is running, for instance, in the same computer. The external storage uses external hardware to store the information of the agent, for instance, a server in the cloud. The external storage is used for older data – freeing the local storage memory – and/or to store data backups.

D. Energy Management Layer

Regarding building's energy management, the most important layer of the proposed architecture is the Energy Management Layer. This layer integrates algorithms for DSM [25], autonomous demand response [26], energy optimization [27], scheduling algorithms [28], learning algorithms [29] and all other methods that the developer wants to enable in the architecture. This layer will accommodate the business logic of each agent and is seen as the intelligent layer. For this paper, a DSM resource optimization (Section III) will be implemented in this layer.

E. Integration Layer

Internet of things (IoT) devices are almost anywhere, and they can be used for energy management systems [30-32]. In [30] is demonstrated a human-in-the-loop (HiTL) system that uses IoT devices to enable sensing, data transmission and signal receivers for resources control. In [31] an energy management system using IoT devices is proposed using REST communications and MQTT protocol. In [32] an energy management system with IoT devices is proposed and simulated. The proposed system of [32] uses OpenADR to bridge between grid- and facility-side. The integration of IoT devices in our proposed energy management system is done in the Integration Layer.

The Integration Layer is responsible to communicate and integrate IoT devices using multiple communication protocols. This layer will communicate to all the building's resources and is responsible to monitor and control all of them.

The success of the Integration Layer depends on two main aspects: the number of available and implemented communication protocols, and the number of resources/devices with monitoring and/or control capabilities on the building. The developer can take care of the first aspect, by developing the appropriate protocols. Regarding the second aspect, the end-consumer must provide enough controllable resources.

F. Graphical User Interface Layer

The Graphical User Interface Layer is responsible for users' interactions. This layer responds to the need of having a graphical interface for users, as stated and analyzed in [33] and [34]. This layer enables the users' real-time interaction with the agent-based architecture. Users are considered to be the persons that live/work/"are in" the building where the architecture is deployed. The interface complexity and compatibilities depend on the functionalities that each developer makes available.

The interface can be made locally, using a screen to show the interface of the agent, or externally, using an API consumed by an external device(s), such as a computer or a smartphone.

V. OFFICE DEPLOYMENT

For this case study, the proposed architecture of Section IV was implemented using open-source libraries, with licenses such as LGPL 2.1, Apache 2.0 and GPLv3. Figure 4 shows the main libraries and technologies used for each layer of the implemented system. It was used Java programming language; for developing; and a Raspberry Pi 3 Model B; for deployment. For this case study an SSH shell was used and no additional graphical user interface was developed. In the Integration Layer were implemented: Modbus/TCP and RESTful; able to control 100% of the controllable resources available. The proposed algorithm for generation and consumption balance; of Section III; was integrated in the Energy Management Layer. The optimization runs locally inside the Computational Layer (i.e. the Raspberry Pi 3 Model B processing unit).

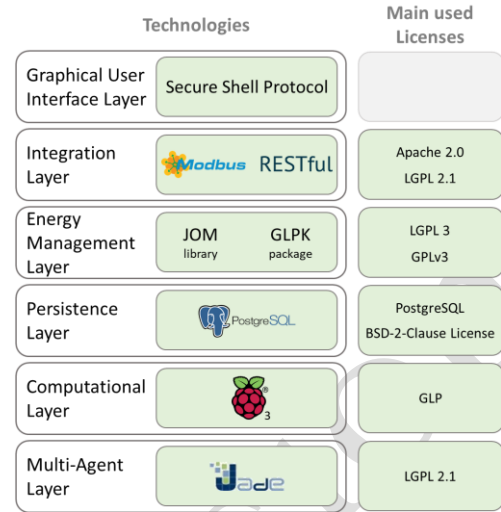


Figure 4. Case Study Implementation

For this case study was used an office with photovoltaic generation located in Portugal. The office has a total of 2.1 kW photovoltaic generation that can be used for self-consumption. For the purposes of this case study, the office was divided into 5 hierarchical zones; as seen in Figure 5. Table 1 lists the resources with individual monitoring in each zone as well as their control type and consumption/generation power. Besides these resources with individual monitoring, the office has a total of 270 W general uncontrollable consumption that corresponds to two desktop computers, four monitors and one laptop. This consumption is measured in real-time using one energy analyzer in the office's electrical board.

Table 1. Resources description

| Zone | Resource | Monitoring Technology | Control Type | Control Technology | Power (W) |
|------|--------------|-----------------------|--------------|--------------------|-----------|
| 1 | Photovoltaic | Energy Analyzer | none | none | 2,100 |
| 2 | Lamp 2.1 | Energy Analyzer | variable | DALI | 100 |
| | Lamp 2.2 | Energy Analyzer | variable | DALI | 100 |
| | Lamp 2.3 | Energy Analyzer | variable | DALI | 100 |
| 3 | Refrigerator | EnAPlug [35] | none* | EnAPlug | 120 |
| 4 | Lamp 2.4 | Energy Analyzer | variable | DALI | 100 |
| | Heater | TP-Link HS110 | discrete | TP-Link HS110 | 1,000 |
| 5 | Lamp 4.1 | Energy Analyzer | variable | DALI | 100 |
| | Lamp 4.2 | Energy Analyzer | variable | DALI | 100 |
| | Television | EDIMAX Smart Plug | variable | EDIMAX Smart Plug | 135 |

* has on/off control but is not used in this case study

Many of the resources in Table 1 use energy analyzers for consumption monitoring. These energy analyzers are connected to a main programmable logic controller (PLC) located in the office's electrical board. The communication between the PLC and the energy analyzers is done using Modbus/RTU protocol. Despite this high cost installation, the case study also uses smart plugs available on the market, such as TP-Link HS110 and EDIMAX Smart Plug. The Environment Awareness Smart Plug proposed in [35] is also used to combine energy metering with context data.

Figure 5 shows the representation of the office divided in 5 zones where there are 2 working desks – Zone 4 and 5 – and 2 common areas – Zone 2 and 3. Zone 1 represents the main/root zone with the generation resource. The zones are hierarchical where each one has a ‘parent’ zone and where zone 1 is considered the root zone – i.e. without parent.

The resources’ priorities; represented by W_i^{var} and W_i^{dis} in equation (5); are calculated in real-time using equations (1) and (3); depending on the resource type. Sensors were installed in the office in order to provide data to be used in equations (1) and (3). Table 2 presents the sensors installed. Sensors of zone 3 are the ones inside EnAPlug. The keyboard and mouse sensors used in zones 4 and 5 were developed in Java. This software uses a connection with an Arduino Nano; installed in the working desk; that has a luminosity sensor and a movement

sensor. The combination between keyboard, mouse and movement sensor identifies the use of the working place.

Table 2. Sensors description

| Zone | Sensor | Units |
|------|--------------------------|-------|
| 3 | Refrigerator temperature | ° C |
| | Refrigerator humidity | % |
| | Temperature | ° C |
| 4 | Ambient light | Lux |
| | Movement | [0,1] |
| | Keyboard | [0,1] |
| | Computer mouse | [0,1] |
| 5 | Ambient light | Lux |
| | Movement | [0,1] |
| | Keyboard | [0,1] |
| | Computer mouse | [0,1] |

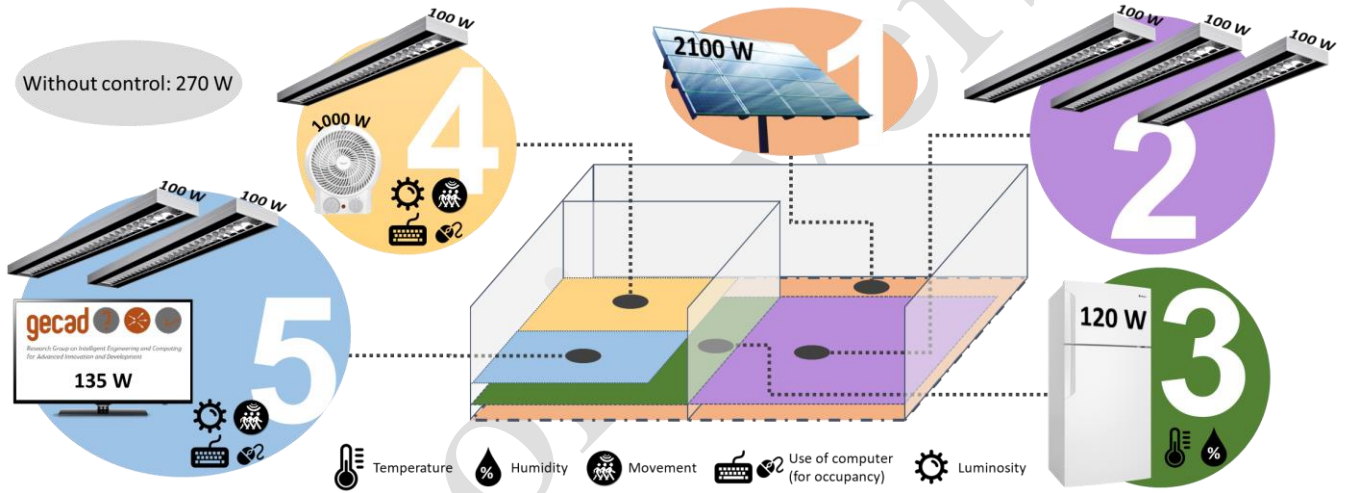


Figure 5. Office's zones and loads

The sensors are used as inputs for the real-time contextual resources’ priorities defined in Section II which in turn are used as inputs for the balancing algorithm defined in Section III. The sensor integration, resource integration and optimization are processed in a Raspberry Pi 3 Model B using the agent-based architecture proposed in Section IV.

The real-time contextual resources’ priorities use users’ preferences; that in this case study are divided by zones. The users’ preference for temperature is 24° C for Zone 4 – that is the only zone with a heater. The users’ preference for ambient light is 500 lux for Zones 4 and 5 and 280 lux for Zone 2. The elasticity values are set to 0, meaning that all the preferences should be strictly obeyed by the system. Regarding the smooth value, in this case study, a smooth value of 70 was used for light resources, and a smooth value of 4 was used for heating and cooling resources.

To provide a real deployment of the system, some rules had to be implemented. These rules aim the efficient deployment of the optimization algorithm considering real-time environments. Table 3 lists the restrictions and conditions implemented. In the

scope of this paper, restrictions are rules that must be true for the algorithm to be executed, where all of them must be true – if at least one is false, then the algorithm will not be executed. Conditions are, in the scope of this paper, rules that when true activate the algorithm, where at least one of them must be true – if at least one is true, then the algorithm will be executed.

In Table 3, *consumption* is the total office’s consumption, *flexibility* is the total consumption of the controllable resources – the maximum consumption that can be decreased –, *generation* is the total photovoltaic generation and the *lastExecution* represents the algorithm’s last execution timestamp.

Table 3. Rules for Algorithm Execution

| Type | Rule |
|-------------|--|
| Restriction | $consumption - flexibility < generation \times 0.95$ |
| Restriction | $lastExecution \geq 4 \text{ seconds}$ |
| Condition | $consumption > generation$ |
| Condition | $consumption < generation \times 0.8$ |

The rules of Table 3 are verified each second. Therefore, the reaction of the algorithm to sudden generation and/or consumption variations is not immediate. The algorithm can take until 2.4 seconds to be executed – 1 second representing the energy data reading period, 0.4 seconds representing the timeout used in the energy data reading communications, and 1 second for the agent’s verification of Table 3 rules. The execution of the algorithm also takes 2 seconds and then the communication to resources’ must be done – with a timeout of 0.4 seconds. Making a maximum reaction time of 4.8 between the generation and/or consumption variation and the office’s energy matching to be concluded.

During deployment, the users requested the inclusion of a new rule for the heater to turn off automaticity if its priority is zero. In this case, the heater will be turned off when temperature is equal or bigger than 24° C; regardless the office context.

VI. RESULTS AND DISCUSSION

This section will present the results of the deployment described in the previous section. Two days were chosen to be presented in this paper. The complete data of these two days, as well as an additional cloudy day, are publicly publish in Zenodo platform under record 2565289 [36]. Figure 6 and Figure 7 show the results of total consumption (blue line), total

generation (green line), energy bought from the grid (blue area), energy sold to the grid (green area) and the temperature of Zone 3 (yellow line); where the working desks are located – see Figure 5. The days were chosen having in mind the difference in the photovoltaic generation.

The winter day has three distinct periods that stand out from the flat daily profile: from 10:42 to 12:32, from 12:32 to 13:33, and from 13:33 to 13:55. The first and third periods happen because of the heater. As can be seen in Table 1, the heater has a significant consumption when compared with the remaining loads. The second period, that stands out, happens because the system executes the balancing algorithm proposed in Section III. During this period the system identifies the possibility to balance the consumption and generation and therefore starts executing the algorithm of Section III; turning off controllable loads that will not harm the users’ comfort. As previously stated, in the first restriction of Table 3, the balancing algorithm will only be executed in periods where is possible to have the total consumption lower than the generation; otherwise, no control over the loads are executed.

During the summer day, presented in Figure 7, the system is able to keep the consumption balanced with the generation through the majority of the day; from 8:30 to 19:07. Around 16:50 is possible to see the ability of the balancing algorithm to decrease and increase the office’s consumption, following the generation pattern.

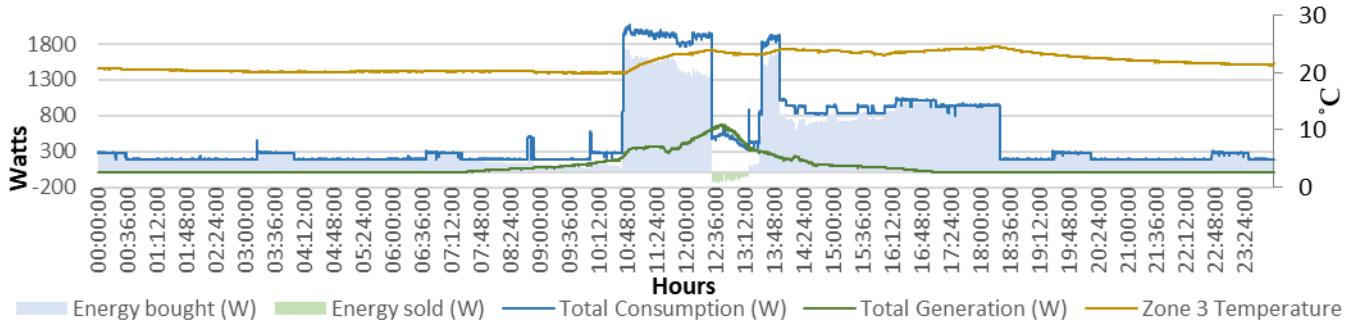


Figure 6. Winter day (12th of December of 2018)

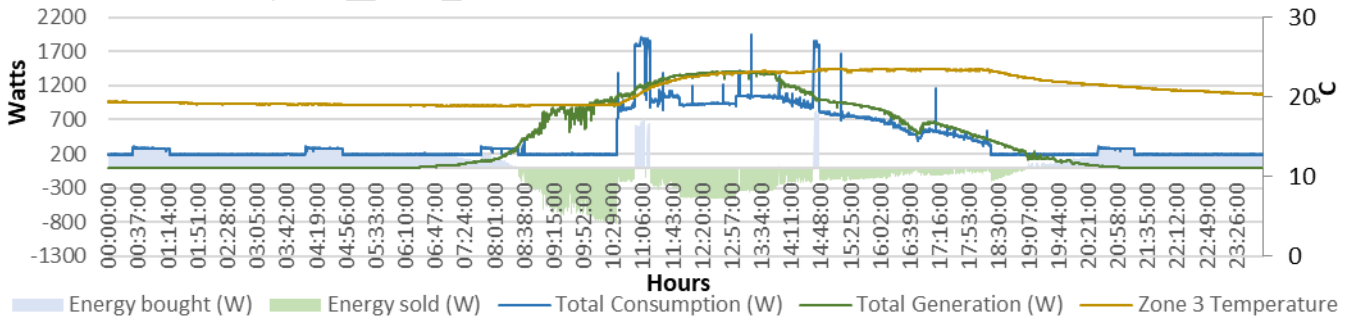


Figure 7. Summer day (9th of July of 2018)

The real-time resource optimization was a success, especially for days where the generation was enough to support the office consumption demands – i.e. where generation was higher than consumption less flexibility. However, looking at the presented days, there is a strange situation with the temperature. Both days starts with temperatures near 20 °C and both of them turn on the heater when the users arrive to the office. However, in the winter day the heater stays on for 50 minutes, achieving a

23.9 °C temperature, and in the summer day the heater stays on for 15 minutes, achieving a 21.2 °C temperature. The heater is turned off in three possible conditions: manually by the user (this did not happen), automatically because the temperature achieves 24 °C (this only happens in the winter day at 13:55), and automatically because the balancing algorithm decides to turn off the heater – if and only if the priority value of the heater

allows this action. The reason the heater worked for 50 minutes, in the winter day, was because none of the previous conditions have been fulfilled; the users did not turn off the heater, the temperature did not achieve 24 °C and the algorithm was not executed because of the low generation value. And the reason the heater worked for 15 minutes, in the summer day, was because the algorithm was executed, and the heater's priority value allowed the turned off of the heater.

The overall results of this case study can be seen in Table 4 where for each kWh bought from the grid was applied a 0.20 EUR cost and for each kWh sold to the grid was applied a 0.10 EUR profit; reflecting the current Portuguese consumer market. Is also possible to see that in the summer day the algorithm was executed 4,929 times; 3201% increase when comparing to the winter day. In the winter day – where the algorithm was executed 154 times – the energy cost for the building was 2.10 EUR, while in the summer day the energy cost dropped to 0.27 EUR. This was in fact a result incited by the balancing algorithm implemented in the proposed agent-based architecture using real-time contextual resources' priorities.

Table 4. Overall results

| | Winter day | Summer day |
|---------------------|------------|------------|
| Generation (kWh) | 1.910 | 10.053 |
| Consumption (kWh) | 12.364 | 9.945 |
| Algorithm execution | 154 | 4,929 |
| Outcome (EUR) | 2.11 | 0.56 |
| Income (EUR) | 0.01 | 0.29 |
| Daily energy (EUR) | 2.10 | 0.27 |

VII. CONCLUSIONS

The application of demand side management and demand response programs in energy grids will enable the participation of small and medium players in the smart grid and microgrid contexts. However, this active participation will demand from end-consumers the ability to manage their own consumptions in an efficient, smart and autonomous way. This paper proposes three key components that enables small and medium player participation: a real-time contextual resources' priority mechanism enabling the prioritization of individual resources in a building, a balance algorithm for generation and consumption enabling the resources optimization and reducing the energy bought from the grid while promoting auto-consumption, and an agent-based architecture for demand side management enabling the necessary means for an actual participation of end-consumers.

The proposed real-time contextual resources' prioritization mechanism was implemented together with the proposed balance algorithm in the Energy Management Layer of the proposed agent-based architecture. The proposed architecture was then deployed in a small office in Portugal. This paper presents the main overall results of this deployment for a winter day and a summer day.

The proposed real-time contextual resources' prioritizations mechanism together with the proposed balance algorithm were able to be executed in real-time using a Raspberry Pi 3 Model B. No issues were detected in the contextual resources' prioritizations mechanism, but they can occur if sensor data is missed or unable to be read. During the execution of the case study was noticed that the balance algorithm sometimes fails

when not a feasible solution was found. Yet, this algorithm failure is handled by the software agent.

The algorithm had a good performance, as shown in the results. However, one issue found in the real-time algorithm makes the execution of the algorithm unfeasible during sunny days with heavy passage of clouds. These days have a big impact in the photovoltaic generation, producing high fluctuations that will be reflected in the consumption, making the office to go from full consumption to lower consumption, and vice-versa, in few moments. This issue makes it unfeasible the use of the proposed algorithm during sunny days with heavy passage of clouds.

Because the proposed agent-based architecture was successfully implemented in an office, new studies should follow to create an agent community representing a microgrid and/or a smart grid. This paper presents an individual deployment in an office, but the agent-based architecture supports community implementations where several agents communicate with each other in a cooperative or competitive mode.

ACKNOWLEDGEMENTS

The present work has been developed under the Project GREEDI (ANI/P2020 17822) co-funded by Portugal 2020 "Fundo Europeu de Desenvolvimento Regional" (FEDER) through PO CI, and Project SIMOCE (ANI/P2020 17690), and has received funding from FEDER Funds through COMPETE program and from National Funds through FCT under the project UID/EEA/00760/2019 and SFRH/BD/109248/2015.

REFERENCES

- [1] D. Kolokotsa, "The role of smart grids in the building sector," *Energy and Buildings*, vol. 116, pp. 703-708, Mar. 2016.
- [2] T. Zhao and Z. Ding, "Distributed Agent Consensus-Based Optimal Resource Management for Microgrids," in *IEEE Transactions on Sustainable Energy*, vol. PP, no. 99, pp. 1-1.
- [3] L. Gomes, T. Pinto, P. Faria, and Z. Vale, "Distributed intelligent management of microgrids using a multi-agent simulation platform," in *Proc. 2014 IEEE Symposium on Intelligent Agents (IA)*, pp. 1-7.
- [4] P. Faria, J. Spínola, and Z. Vale, "Aggregation and Remuneration of Electricity Consumers and Producers for the Definition of Demand-Response Programs," *IEEE Transactions on Industrial Informatics*, vol. 12, pp. 952-961, June 2016.
- [5] L. Martirano, E. Habib, G. Parise, G. Greco, M. Manganelli, F. Massarella, and L. Parise, "Demand Side Management in Microgrids for Load Control in Nearly Zero Energy Buildings," *IEEE Transactions on Industry Applications*, vol. PP, pp.1-1, Feb. 2017.
- [6] J. Spínola, P. Faria, Z. Vale, "Photovoltaic Inverter Scheduler with the Support of Storage Unit to Minimize Electricity Bill", *Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection - 15th International Conference*, pp. 63-71, 16 July 2017.
- [7] X. Liu, L. Ivanescu, R. Kang and M. Maier, "Real-time household load priority scheduling algorithm based on prediction of renewable source availability," in *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 318-326, May 2012.
- [8] Z. Yang, W. Xu, J. Shi, H. Xu and M. Chen, "Association and Load Optimization With User Priorities in Load-Coupled Heterogeneous Networks," in *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 324-338, Jan. 2018. doi: 10.1109/TWC.2017.2765322
- [9] I. Joo and D. Choi, "Distributed Optimization Framework for Energy Management of Multiple Smart Homes With Distributed Energy Resources," in *IEEE Access*, vol. 5, pp. 15551-15560, 2017. doi: 10.1109/ACCESS.2017.2734911

- [10] Recommended Light Levels (Illuminance) for Outdoor and Indoor Venues, Available online: https://www.noao.edu/education/QLTkit/ACTIVITY_Documents/Safety/LightLevels_outdoor+indoor.pdf (accessed on 31 December 2018).
- [11] J. Vasudevan and K. S. Swarup, "Price based Demand Response strategy considering load priorities," *2016 IEEE 6th International Conference on Power Systems (ICPS)*, New Delhi, 2016, pp. 1-6.
- [12] J. H. oon, R. Baldick and A. Novoselac, "Dynamic Demand Response Controller Based on Real-Time Retail Price for Residential Buildings," in *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 121-129, Jan. 2014. doi: 10.1109/TSG.2013.2264970
- [13] Mieczysław Porowski, "The optimization method of HVAC system from a holistic perspective according to energy criterion," *Energy Conversion and Management*, Volume 181, 2019, Pages 621-644, <https://doi.org/10.1016/j.enconman.2018.11.069>.
- [14] Gomes, L, Fernandes, F, Faria, P, Vale, Z, Ramos, C & Morais, H 2013, Contextual Intelligent Load Management Considering Real Time Pricing in a Smart Grid Environment. in *Proceedings of 7th International Conference on Intelligent System Applications to Power Systems*. IEEE, 17th International Conference on Intelligent Systems Application to Power Systems, Tokyo, Japan, 01/07/2013
- [15] L. Gomes, F. Fernandes, Z. Vale, P. Faria and C. Ramos, "A learning algorithm and system approach to address exceptional events in domestic consumption management," *2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG)*, Orlando, FL, 2014, pp. 1-7. doi: 10.1109/CIASG.2014.7011564
- [16] Y. Han, K. Zhang, H. Li, E. A. A. Coelho and J. M. Guerrero, "MAS-Based Distributed Coordinated Control and Optimization in Microgrid and Microgrid Clusters: A Comprehensive Overview," in *IEEE Transactions on Power Electronics*, vol. 33, no. 8, pp. 6488-6508, Aug. 2018. doi: 10.1109/TPEL.2017.2761438
- [17] Z. Li, C. Zang, P. Zeng, H. Yu and H. Li, "MAS based distributed automatic generation control for cyber-physical microgrid system," in *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 1, pp. 78-89, 10 January 2016. doi: 10.1109/JAS.2016.7373765
- [18] F. Chen, M. Chen, Q. Li, K. Meng, J. M. Guerrero and D. Abbott, "Multiagent-Based Reactive Power Sharing and Control Model for Islanded Microgrids," in *IEEE Transactions on Sustainable Energy*, vol. 7, no. 3, pp. 1232-1244, July 2016. doi: 10.1109/TSTE.2016.2539213
- [19] Farhan H. Malik, Matti Lehtonen, "A review: Agents in smart grids," *Electric Power Systems Research*, Volume 131, 2016, Pages 71-79, <https://doi.org/10.1016/j.epsr.2015.10.004>
- [20] Ipakchi, A.; Albuyeh, F., "Grid of the future," *Power and Energy Magazine*, IEEE, vol.7, no.2, pp.52,62, March-April 2009
- [21] S. Paul, A. Parajuli, M. R. Barzegaran and A. Rahman, "Cyber physical renewable energy microgrid: A novel approach to make the powersystem reliable, resilient and secure," *2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, Melbourne, VIC, 2016, pp. 659-664.
- [22] E. Vinagre, L. Gomes and Z. Vale, "Electrical Energy Consumption Forecast Using External Facility Data," *2015 IEEE Symposium Series on Computational Intelligence*, Cape Town, 2015, pp. 659-664.
- [23] Vangelis Marinakis, Haris Doukas, John Tsapelas, Spyros Mouzakitis, Álvaro Sicilia, Leandro Madrazo, Sgouris Sgouridis, "From big data to smart energy services: An application for intelligent energy management," *Future Generation Computer Systems*, 2018, <https://doi.org/10.1016/j.future.2018.04.062>
- [24] O. Y. Al-Jarrah, Y. Al-Hammadi, P. D. Yoo and S. Muhaidat, "Multi-Layered Clustering for Power Consumption Profiling in Smart Grids," in *IEEE Access*, vol. 5, pp. 18459-18468, 2017. doi: 10.1109/ACCESS.2017.2712258
- [25] Ebby Thomas, Rahul Sharma, Yoni Nazarathy, "Towards demand side management control using household specific Markovian models," *Automatica*, Volume 101, 2019, Pages 450-457, <https://doi.org/10.1016/j.automatica.2018.11.057>
- [26] P. Palensky and D. Dietrich, "Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads," in *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381-388, Aug. 2011. doi: 10.1109/TII.2011.2158841
- [27] R. H. Byrne, T. A. Nguyen, D. A. Copp, B. R. Chalamala and I. Gyuk, "Energy Management and Optimization Methods for Grid Energy Storage Systems," in *IEEE Access*, vol. 6, pp. 13231-13260, 2018. doi: 10.1109/ACCESS.2017.2741578
- [28] M. A. A. Pedrasa, T. D. Spooner and I. F. MacGill, "Coordinated Scheduling of Residential Distributed Energy Resources to Optimize Smart Home Energy Services," in *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 134-143, Sept. 2010. doi: 10.1109/TSG.2010.2053053
- [29] Jui-Sheng Chou, Duc-Son Tran, "Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential householders," *Energy*, Volume 165, Part B, 2018, Pages 709-726, <https://doi.org/10.1016/j.energy.2018.09.144>
- [30] M. Bisadi, A. Akrami, S. Teimourzadeh, F. Aminifar, M. Kargahi and M. Shahidehpour, "IoT-Enabled Humans in the Loop for Energy Management Systems: Promoting Building Occupants' Participation in Optimizing Energy Consumption," in *IEEE Electrification Magazine*, vol. 6, no. 2, pp. 64-72, June 2018. doi: 10.1109/MELE.2018.2816844
- [31] F. G. Brundu *et al.*, "IoT Software Infrastructure for Energy Management and Simulation in Smart Cities," in *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 832-840, April 2017. doi: 10.1109/TII.2016.2627479
- [32] Min Wei, Seung Ho Hong, Musharraf Alam, "An IoT-based energy-management platform for industrial facilities," *Applied Energy*, Volume 164, 2016, Pages 607-619, <https://doi.org/10.1016/j.apenergy.2015.11.107>
- [33] H. Xu and H. Schmeck, "State-of-the-art user interfaces for building operating systems," *2017 IEEE International Conference on Smart Grid and Smart Cities (ICSGSC)*, Singapore, 2017, pp. 283-292. doi: 10.1109/ICSGSC.2017.8038592
- [34] Andrew D. Peacock, Joel Chaney, Kristin Goldbach, Guy Walker, Paul Tuohy, Salvador Santonja, David Todoli, Edward H. Owens, "Co-designing the next generation of home energy management systems with lead-users," *Applied Ergonomics*, Volume 60, 2017, Pages 194-206, <https://doi.org/10.1016/j.apergo.2016.11.016>
- [35] Gomes, L.; Sousa, F.; Vale, Z. An Intelligent Smart Plug with Shared Knowledge Capabilities. *Sensors* **2018**, *18*, 3961. <https://doi.org/10.3390/s18113961>
- [36] L. Gomes, "Dataset of uGIM deployed in an office building," Zenodo, 2019 doi: 10.5281/zenodo.2565289