

# Optimizing Energy Costs for Offices Connected to the Smart Grid

Ilche Georgievski, *Student Member, IEEE*, Viktoriya Degeler, *Student Member, IEEE*,  
Giuliano Andrea Pagani, *Student Member, IEEE*, Tuan Anh Nguyen, Alexander Lazovik, Marco Aiello, *Senior Member, IEEE*

**Abstract**—In addition to providing for a more reliable distribution infrastructure, the Smart Grid promises to give the end users better pricing and usage information. It is thus interesting for them to be ready to take advantage of features such as dynamic energy pricing and real-time choice of operators. In this work, we propose a system to monitor and control an office environment and to couple it with the Smart Grid. The idea is to schedule the operation of devices according to policies defined by the users, in order to minimize the cost of operation while leaving unaffected user comfort and productivity. The implementation of the system and its testing in a living lab environment show interesting economic savings of an average of about 35% and in some cases even overall energy savings in the order of 10% for a building equipped with renewable generation plants, and economic and energy savings of 20% and 10%, respectively, for a building without local renewable installations.

**Index Terms**—Demand-Response, Intelligent Building, Energy Monitoring, Pervasive Systems, Ubiquitous Computing.

## I. THE SMART GRID FOR THE OFFICE

THE SMART GRID promises to bring two-way communication, digital metering, inclusion of renewables, and dynamic pricing to the world of energy management and distribution. The office of tomorrow will take advantage of these factors by adapting its energy consumption patterns to the price and availability of energy. In particular, the possibility of having a dynamic price structure (real-time pricing) and to be equipped with renewable energy generation facilities will change the way offices are controlled.

The idea of dynamic pricing is in line with the current trend in most countries, where we have moved from a single provider/single tariff system to models with competing providers and, basically, two prices over long-term contracts (usually in the term of months). For example, an energy provider can offer distinct tariffs for daytime and night-time or weekdays and weekend, the so-called *peak* and *off-peak* tariffs. The goal of the energy provider is to incentivize the users to balance the supply of energy (generation) with an adjustment in their required demand. This is due to the fact that the costs of increasing the energy supply do not increase linearly with the demand; rather, they follow a convex function that is composed of linear intervals with increasing slope as

the energy use increases [1]. The situation promises to be even more delicate with the increase of renewable sources of energy [2], as these imply a greater uncertainty of supply.

The renewables are and will be increasingly present not only in a medium-large scale on the Grid, but are also increasingly available at the level of the single building as solar panels, wind and combined heat-power generators. The *Intelligent Building* has to be aware of the energy generated locally in order to decide the proper policies to adopt: either use the energy produced for its local needs or feed the energy into the Power Grid and receive a payment for it. Therefore, the intelligent elements inside the building have to be able to know the energy produced on-site (or energy production forecast) in order to eventually adapt their operations.

Consider the point of view of the chief financial officer or his delegated building manager whose goal is that of saving money on the energy bill, while keeping an adequate level of comfort for and productivity of the employees working in the building. Honeywell claims that, in a typical commercial building, the energy bill accounts for 25% of the operating costs, which are mainly fixed ones. This monetary goal translates into three practical objectives: reducing the overall consumption of energy, adding attractive forms of local energy generation, and buying energy at the lowest possible price.

Here we present an approach to controlling offices to save energy and overall energy bill costs; this assumes the availability of a Smart Grid that offers dynamic prices from competing providers. The approach is based on (1) monitoring the energy consumption at the device level, (2) monitoring energy production of small-scale generating units, (3) associating policies for the devices that conform with user requirements for comfort and productivity, (4) controlling in an optimal way the energy consumption patterns of devices following the usage policies, and (5) being able to acquire dynamically the prices of energy from different providers and closing contracts for short-term time intervals.

Notice that the leading hypothesis of the present approach is that *the office managers will be incentivized to reduce energy consumption by attractive real-time pricing, thus balancing globally the Power Grid* rather than being forced to follow governmental policies or, even worse, being forced to give up control of their building energy consumption equipment as advocated by several approaches. For instance, dynamicDemand<sup>1</sup> promotes the integration in the appliances of technologies

All authors are with the Department of Computer Science, University of Groningen, Groningen, The Netherlands (e-mails: {i.georgievski, v.degeler, g.a.pagani, t.a.nguyen, a.lazovik, m.aiello}@rug.nl).

The work is supported by the EU FP7 Project GreenerBuildings, contract no. 258888, and the Dutch National Research Council under the NWO Smart Energy Systems program, contract no. 647.000.004.

<sup>1</sup><http://www.dynamicdemand.co.uk>

that can automatically enable them to respond to the Grid's imbalance situations without user notification. Another example is the explanation given by Iowa State Office of Energy Independence for Smart Grid in complementing renewable energy; here a scenario is envisioned in which, to face the lack of power, customers' air conditioning is automatically turned off<sup>2</sup>. A key finding by a customer-interview study [3] is that the users must be economically incentivized and anyway given the possibility to reverse utility decisions of turning on/off appliances.

The proposed approach has been implemented in our own offices at the University of Groningen and tested over a short period as a proof of concept. This initial investigation shows that automatic control of devices can save the overall energy consumption and, if coupled with dynamic pricing from the Smart Grid, can provide considerable financial savings from the end user perspective; this considers both the case of a building equipped with renewable-based small scale energy sources and the case without such installation that provides a baseline for the study. We do not investigate the provider's point of view, but we conjecture that also the provider will experience significant financial benefit if most of the end users would be price driven in their energy use.

The remainder of the paper is organized as follows. Section II discusses the related work. In Section III, we present the model of our system. Section IV shows its general architecture and describes each component in details. Section V specifies the technologies we used during actual implementation of our system and living lab setting. Our experiments are described and evaluated in Section VI. Finally, conclusions are given in Section VII.

## II. RELATED WORK

The Smart Grid is a broad term used in several communities and there exist many investigations on the interactions between the Smart Grid and Smart Buildings. Here we present the main works concerning residential and commercial building environments where interoperation between a building and the Smart Grid appears. A general distinction in the literature is between the works focusing on the house environment, the majority [1], [4], [5], [6], [7], [8], [9], [10], and the office environment which has received little attention so far [11], [12], [13]. In general, most of the works tend to be quite generic in presenting high-level architectures [5], [6], [14] and usually lack an experimental part where simulation of the energy consumption of devices is considered. On the other hand, other approaches are related to the very low-level concepts of protocol integration [4] or sensor network communication used for information exchange of energy consumed by devices [15]. Control at the device level is generally based on price aspects that are a good proxy for the energy availability in the Power Grid: these tend to consider agent-based systems that negotiate in a virtual energy market [9], sometimes using strategies coming from game-theory concepts [1], [16]. Usually actions on real devices are only simulated [1], [7], [9] and only few approaches report on actual control and actuation

of appliances [8]. In addition, in the range of equipment considered, the tendency is generally to take into account only the electrical appliances in the home/office environment, while heating, ventilation, and air conditioning (HVAC) systems are generally ignored. An exception is the work by Ma *et al.* [12], as is also our study.

A very significant test bed referring to office environment is realized by Han *et al.* [13], where a commercial building of seven floors is equipped with temperature, lighting, quality of air, and occupancy sensors to retrieve information and provide it to the Building Energy Management System (BEMS) whose aim is to extract the context of the building. The system is based on an ontology for the building description; this defines the optimal situation of the building, in which sensors are used to acquire the context and build inference rules used to actuate on the environment in order to achieve an optimal state of the building (e.g., optimal air quality, optimal lightning). Although the system seems one of the most well-described and includes a highly advanced approach to understand the building context, the paper by Han *et al.* [13] misses a quantitative evaluation of the benefits achieved through the system. In addition, unlike our approach, there is no mention of possible interactions with the Smart Grid.

With respect to the state of the art, the novelty of the approach presented here is to combine all these elements together in the less explored scenario of the office environment. Features of the Smart Grid such as Demand-Response functionalities in a real environment, with prices that come from real energy market conditions, real renewable source energy production, and appliances actuation are new, to the best of our knowledge. In addition, the device consumption and the energy savings achieved are neither estimated nor simulated as in many other works, but come from our living lab setting. To achieve the goals of energy cost reduction and energy cost usage, it is essential to monitor not only energy usage [11], [10], but also the control and actuation on real equipment, and this is missing in the literature analyzed.

## III. SYSTEM MODEL

The system we design for saving energy in buildings is based on a likely future evolution of the Smart Grid and on the possibility of associating policies with energy consuming devices. We assume that each building (or part of a building) is equipped with an interface with the Smart Grid that offers information on the price of energy proposed by different providers per time interval and possible maximum amount available at that price. The time intervals are discrete and last one hour. Thus, contracts are electronically signed on an hourly basis, as each hour the price and amounts can be different.

From the point of view of the office devices, we assume that any energy consuming apparatus, e.g., heater, fridge, printer, projector, can be measured in its electrical energy consumption in kWh and can be controlled. Each device has an associated state machine and an energy consumption level for each state. For example, a fridge consumes about  $10^{-3}$  kWh when idle, but about 0.63 kWh when actively cooling. The system has

<sup>2</sup><http://energy.iowa.gov/SmartGrid/SmartGrid.html>

| Policy type | Associated device | Description  |
|-------------|-------------------|--|
| REPEAT      | Fridge, Boiler    | Device should be put to a specified state repeatedly with a certain periodicity.       |
| TOTAL       | Laptop            | Device should operate for at least a certain amount of time.                           |
| MULTIPLE    | Printer           | Device should operate for the time that allows for all scheduled jobs to be performed. |
| STRICT      | Projector         | A strict schedule is given in advance.   |
| PATTERN     | Microwave         | An expected pattern of device operations.  |
| SLEEP       | Any device        | No demand for device during the scheduling period.                                     |

TABLE I: Device policies

full access to reading the state of a device and can trigger a state transition. Data about energy consumption levels are obtained by analysis of historical data for that type of device. To avoid changing states of devices too often, we propose the notion of the minimum time unit. The minimum time unit is an adjustable parameter that tells the system how rapidly the devices can be forced to change states. In our implementation, we used 15 minutes.

For each device, there is an associated *policy*. A policy is a set of consistent rules that hold for device operations. For example, “a fridge must work at least 15 minutes per hour” to be able to maintain its internal temperature below a certain threshold temperature level. Policies can have different parameters, a few of which are common to all:  $(tBegin, tEnd)$  – time period, when the policy is active; and *sid* – state ID that the policy is applied to. State IDs are unique per device. In general, we assume several possible states per device, together with associated actions to move a device to these states. In the presented setting, each device has two states: “on” and “off,” and two associated actions: “turn on” and “turn off.”

In this work, we define and use five types of policies, which represent common rules for widely deployed devices. The five policies are summarized in Table I and defined next.

**REPEAT.** The device must be operated cyclically by entering the state *sid* repeatedly with a certain periodicity. For example, a fridge that should operate for 15 minutes each hour is specified using this policy. Parameters specific to this policy are:  $tCycle$  – a total cycle time; and  $tOn$  – a time during this cycle, when the device should be in a state *sid*.

**TOTAL.** Specifies a total amount of time  $tOn$  that a device should be put in a state *sid*. An example is a laptop that needs recharging for two hours; however the exact time when it is going to happen does not matter, as long as it stays within  $(tBegin, tEnd)$  bounds. This policy also assumes that the time when a device is in the state *sid* can be split into several parts. For example, we can charge a laptop for half an hour, then for another hour a little later, and for another half an hour even later.

**MULTIPLE.** Devices that schedule a number of jobs over a certain period of time use the *MULTIPLE* policy. It has two specific parameters:  $nJobs$  – a total number of jobs to

be scheduled; and  $tDuration$  – a time needed to complete a single job. An example is a printer that processes large batch jobs (e.g., printing a book): each job needs 15 minutes to be completed, and a total of three jobs are required to be performed. With such a policy it does not matter when a particular job is scheduled, but it is important that the device is not turned off in the middle of performing a job.

**STRICT.** To enforce a state *sid* to be active from  $tBegin$  to time  $tEnd$ , the *STRICT* policy is used. An example is a projector that should be turned on at the beginning of a meeting and turned off when a meeting ends. The policy firmly defines the schedule for this device, as times are strict, so the scheduler has no possibility to change the energy consumption time of the device.

**PATTERN.** The *PATTERN* policy provides information about a way the device consumes energy. Instead of offering the possibility of controlling the device, it provides information on expected energy usage that can help to schedule other devices. For example, while a microwave is never completely turned off, the energy consumption in stand by mode is much lower than the energy consumption when it is actively in use. Historical data show that a higher level of energy consumption is expected during lunchtime, so the scheduler takes this into account when scheduling other devices.

**SLEEP.** For a device for which there is no demand for operation during a given period, the *SLEEP* policy can be used. The policy is used mostly at night, when there is no activity in the office and many devices can be turned off in order to save energy. There are no additional parameters for this policy.

#### IV. SYSTEM ARCHITECTURE

To take advantage of the dynamic pricing on the Smart Grid and the controllability of the devices, we design an architecture that goes from the hardware level of energy measurement and control up to the scheduling logic. The overall architecture is shown in Figure 1. On the right is the Smart Meter, intended as the interface to the Smart Grid and responsible for the two-way communication. At the bottom sits the hardware responsible for the monitoring and control of energy use, above which there is the Controller acting as a bridge between the controller and the hardware. On the left side, the Repository contains historical data of energy use and the policies for the devices. This information is essential for the Scheduler (on top), who needs to plan, based also on the information from the Smart Grid, optimal control strategies for the office. The Coordinator component at the center of the figure acts as a facilitator between the devices, the Smart Grid, and the Repository.

##### A. Smart Meter

A Smart Meter is a physical device that is able to measure consumed and produced energy, provide this information to the energy metering companies, and change electricity tariffs according to the signals received by the energy companies participating in the Smart Grid real-time tariff service. In the

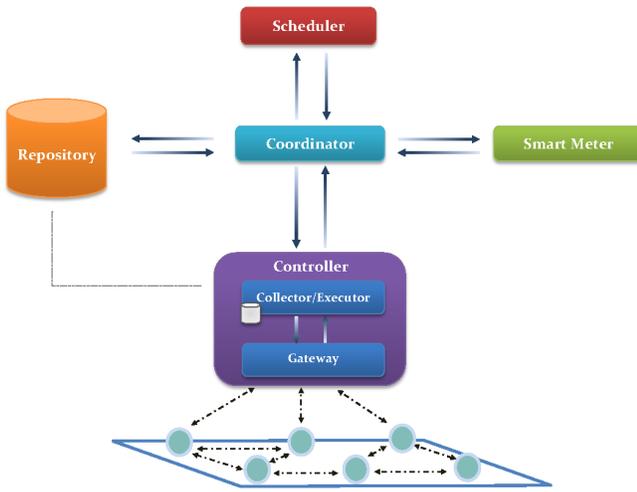


Fig. 1: Architecture Design

proposed architecture, the Smart Meter is seen as the component that interacts with the Smart Grid in order to receive the energy prices that are applied by the different energy providers for the same hourly time interval. We envision a service, either from the Smart Grid itself or from energy providers, to provide through the Internet the changing energy prices. Once the information is received or retrieved (it can be both a pull-based or push-based communication), the Smart Meter component stores the data in the Repository through the interaction with the Coordinator component. The energy measuring functionalities are also provided by the Smart Meter component; however, these are obtained by aggregating the single device energy consumption that is available in the repository once measured by the device controller. For the measurement of produced energy, the Smart Meter component can interact with the sensing equipment (either directly or once the production data are published on the Internet or on the *Intelligent Building LAN*) available in the local energy production units such as photovoltaic panels. In this way, the meter collects the overall amount of produced energy and stores it in the Repository through the Coordinator component. Once the generation and consumption data are available, it is then easy for the Smart Meter to provide this information either periodically or upon request to the energy provider for accounting/billing purposes. In the current implementation, we consider one Smart Meter in the office environment. However, the proposed architecture supports, with minimal changes, more Smart Meters, for instance, a Smart Meter per office floor, or per section of the building, or even per working business unit.

### B. Device Network

The Device Network, most usually realized as a Wireless Sensor Network, provides the basic infrastructure for gathering the information on a device's power consumption, the device's state, and controlling appliances. Typically, this type of energy monitoring equipment is plugged into power sockets instead of running on battery. In addition, it has embedded wireless

chip that is sufficient to form a wireless mesh network around the gateway, providing a cost effective and dynamic high-bandwidth network, with a relatively stable topology. One can also envision these functionalities to be directly available at the appliance level (e.g., a laptop that offers external control and energy consumption values as system calls that can be remotely invoked [17]).

### C. Controller

The Controller consists of a Collector and Executor (CE) subcomponent and a Gateway between the Device Network and the above layers, illustrated in Figure 4. The Gateway is in charge of managing the network. It runs in the background, providing basic tools to the CE subcomponent for gathering information as well as controlling the devices. The CE subcomponent, in turn, is responsible for the collection and storage of the office information. On a regular basis, the CE collects the devices' data gathered through the Gateway subcomponent. In order to access lower-level tools of the Gateway in a more intuitive fashion, the CE contains a wrapper that provides a standard interface for interaction. The information received is then stored into a database. Another responsibility of the CE subcomponent is the execution of the actions over devices. It uses its wrapper to interact with the Gateway in order to send the execution commands to the physical layer.

### D. Repository

The Repository component comprises two basic functionalities: (i) storage of information provided by devices, policy manager and energy providers; and (ii) retrieval of data queries issued by other components, namely by the Coordinator component. Communication between the Repository and Coordinator is enabled by exchanging an agreed format of messages. In Figure 2, we schematize the internal architecture: its configuration is abstracted into three subcomponents: *Web Service (WS) Interface*, *Data Access Object (DAO)*, and *Database*.

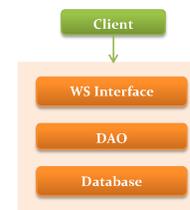


Fig. 2: Repository Internal Architecture

The *WS Interface* is a thin layer over the Repository that offers its capabilities across the network in form of Web services. By implementing such an interface, we simplify the overall system architecture and the visibility of interactions is improved. We view each Web service as a resource on which a set of actions can be performed. Furthermore, such an action is mapped onto an operation of the lower-level *DAO* component. *DAO* encapsulates and implements all of the functionalities required to work with the data source. It persists the requests

and information provided by the client calls into the *Database*. Naturally, the back-end database can be freely chosen.

### E. Scheduler

The Scheduler component is where the logic of the system resides. The Scheduler receives the information from the Grid energy providers about the available supply and price of energy. Also, the Scheduler receives the information about controllable devices, their levels of energy consumption, and their policies (rules of operation). Given this information, the Scheduler then finds the optimal solution with the minimum price paid for the total energy consumed over a certain period of time.

Prices on the market change regularly, say each hour, so the Scheduler takes into account varying prices over the course of the day and tries to schedule devices to operate at times, when the price per consumed kWh is the lowest. Generally, those prices vary from provider to provider, and the system can choose a provider to buy energy from. However, since providers have a finite energy supply, if many devices are scheduled to operate at the same time, their total energy consumption will likely be bigger than the cheapest energy supplier is ready to provide. That will lead to the necessity to buy energy from a more expensive energy provider.

To summarize, the Scheduler needs to balance the varying prices over the course of the day and not to schedule too many devices at the same time; thus it can avoid purchasing the more expensive energy, but at the same time keep all device policies satisfied.

*Scheduling optimization problem:* Let  $EP(t) = \{ep_i\}$  denote a set of energy providers at the time unit  $t$ , where each *energy provider* is represented by a tuple  $ep_i = \langle cost, energy \rangle$ , *cost* is the cost of 1 kWh of energy, and *energy* is the maximum amount of energy that the current provider can provide at the time unit  $t$ .

To calculate the accumulated cost that an *Intelligent Building* needs to pay for the energy it consumes in a certain time unit, we need to sort energy providers by their price. Since we assume that a Smart Meter can choose which provider to buy energy from, it first buys energy from the cheapest providers, and then continues to more expensive providers, if the amount of energy the building needs to consume is bigger than the amount offered by the cheapest energy providers. Thus the total cost that the building pays at time unit  $t$  if it needs to consume an amount of energy  $e$  is

$$cost(t, e) = \min \left( \sum_{i=1}^{|EP(t)|} (k_i * ep_i.energy * ep_i.cost) \right) \quad (1)$$

s.t.

$$\sum_{i=1}^{|EP(t)|} (k_i * ep_i.energy) = e \quad (2)$$

where  $k_i$  is the coefficient that shows a fraction of energy bought from energy provider  $ep_i$ . In practice,  $k_i$  will be equal to 1 for the cheapest providers, then be in a region  $[0, 1]$  for one of the other providers, and be equal to 0 for all more expensive providers.

| Provider              | Energy supply | Price per kWh |
|-----------------------|---------------|---------------|
| Internal Wind Turbine | 0.214292      | 0.0           |
| Internal Solar Panel  | 0.302314      | 0.122916      |
| COMED                 | 2.755946      | 0.282973      |
| ATSI                  | 3.154828      | 0.357123      |
| AEP                   | 2.411659      | 0.360658      |
| more providers ...    |               |               |

TABLE II: Example of energy providers and prices

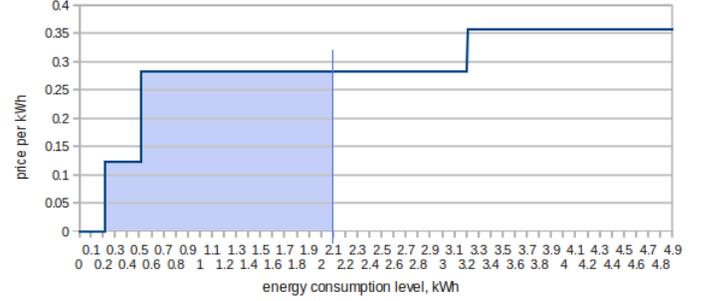


Fig. 3: Price per kWh given the energy consumption. Total price paid equals the area under the graph

An example of cost calculation for the energy providers in Table II is shown in Figure 3. For the consumption level of 2.1 kWh, the *Intelligent Building* has to use energy from internal Wind Turbine and Solar Panels, and also buy some energy from the cheapest provider COMED, resulting in a total of \$0.485217 per hour.

#### Algorithm 1 Cost depending on energy consumed

```

1: function getCost(time, energy):Double
2:   provs ← getProvidersAt (time)
3:   sortedprovs ← Sort provs by provs(i).cost
4:   energyleft ← energy
5:   totalcost ← 0
6:   while energyleft > 0 do
7:     prov ← sortedprovs.next
8:     totalcost ← totalcost +
       min(energyleft, prov.energy) * prov.cost
9:     energyleft ← energyleft - prov.energy
10:  end while
11:  return totalcost

```

The algorithm to compute the cost (shown in Algorithm 1) goes as follows. Let  $D$  denote a set of devices in the building that are connected to a Smart Meter. Each *device*  $d_i \in D$  is represented by a tuple  $d_i = \langle did, S_i \rangle$ , where *did* is the unique identifier of a device that in our case is equal to the device's MAC address, and  $S_i$  is a set of states that the device  $d_i$  can take (for example, "on" and "off"), where each state  $s_{ij} \in S_i$  is a tuple  $s_{ij} = \langle sid, energy \rangle$ , *sid* being the unique identifier of a state, and *energy* being an amount of energy that the device consumes while being in this state. Let  $P$  denote a set of policies that apply to the devices in the building. Each *policy*  $p_i \in P$  is a tuple  $p_i = \langle did, type, params \rangle$ , where *did* is the unique identifier of the device the policy is applied to, *type* is the type of policy, and *params* is a set of parameters.

Parameters differ per type of policy. Each policy has different conditions that must be fulfilled in order for it to be satisfied. Here we define a general Boolean function  $isSatisfied(p, X)$  that takes *true*, if the policy  $p$  is satisfied in the schedule  $X$ , and *false* otherwise.

Over time period  $T$ , where  $t \in T$  is a *time unit* in the time period  $T$ , the *schedule*  $X = \{x_{td}\}$  is a set of values, where each value  $x_{td} \in S_d$  represents the state that the device  $d$  takes at the time unit  $t$ . Now we can present the scheduling optimization problem:

Schedule  $X = \{x_{td}\}$ ,  $\forall t \in T, \forall d \in D$  is optimal iff

$$\sum_{t \in T} cost(t, e_t) \rightarrow min, \quad \forall p \in P : isSatisfied(p, X) \quad (3)$$

where  $e_t = \sum_{d \in D} x_{td}.energy$ . Thus the optimal schedule is the one where price paid for all consumed energy is minimal, and the constraints for the schedule are the policies of device operation, which must be fulfilled for all devices during the scheduling period.

To solve the problem, we implement a priority queue with the Breadth-First Search optimization algorithm [18]. Each search state of the algorithm is a partially fulfilled schedule. We start by creating possible solutions for the first time slot and putting them to the queue. Then, at each iteration, we expand the state with the least energy cost. With each expansion, we add only those solutions that are compliant to all policies. To decrease the search space, we extensively use domain knowledge (per policy). For example, if a device has the TOTAL policy and should be turned on for a certain period of time, we automatically restrict from the search space all schedules where this device is turned on for more or less than the required time; this is because having it turned on more than it is absolutely necessary will only increase the energy consumption and price, and having it turned on less than absolutely necessary will not satisfy our policy. Another example is the policy MULTIPLE policy, where we have multiple jobs for a certain period of time each. We remove from the search space all schedules where time of being turned on for a device is not equal to a multiple of the time it takes to complete a single job. For example, if a single job of a printer takes 30 minutes to complete, we remove from the search space all schedules where the printer is turned on for 45 minutes, as it means the printer will definitely be idle for 15 minutes and unnecessarily consume energy.

#### F. Coordinator

Finally, the Coordinator component is a software element that enables a coherent execution of the system as a whole. Firstly, it serves as a client to the Controller, more specifically to the CE and Repository component. Once the CE component collects the device information, the Coordinator instance calls the CE specific Web service to retrieve that description, and, in consequence, it sends the data to be stored into the Repository to be available for later usage. The Coordinator also serves as a client to the policy manager to provide the system with policies needed by the Scheduler. Secondly, the Coordinator invokes the Smart Meter and Scheduler components. On a

regular basis, the Coordinator asks the Smart Meter to provide the energy price information and sends gathered data to the Repository. At a point when all necessary input parameters for the Scheduler are secured, the Coordinator continues with the system execution flow by instantiating the Scheduler component. Thirdly, the received schedule of actions is controlled by this component. Each action is scheduled for one-time execution by invoking the CE component Web service to process changes deeper into the physical layer.

## V. IMPLEMENTATION

We have implemented the proposed system in a prototype that we have deployed in our own offices. Next, we detail the realization of each component.

1) *Interfacing with the Smart Grid*: The Smart Grid has not yet been deployed and implemented for the end user, but has been used just as proof-of-concepts [19], simulations of Smart Grid customer behavior [20], or small scale pilot projects [21], and no generally available standards have been agreed yet (though initiatives are underway from IEEE, NIST, and others); therefore, we simulate the dynamic pricing. To make the simulation realistic, we use data and services obtained from real markets and real energy generation installations. In particular, in order to simulate the variable energy tariffs, we use the energy prices coming from the PJM Interconnection<sup>3</sup>, which is a regional transmission organization that coordinates the movement of wholesale electricity in more than 13 states of Eastern U.S.A. The data extracted are the Day-Ahead Energy Market locational marginal pricing, which are the prices of energy negotiated in the wholesale market for the following day by energy companies at a specific location where energy is delivered or received. Data contain the energy price for each energy unit (\$ per MWh) for each hour of the day agreed for the next day at 20 locations of delivery. The prices of the wholesale market are adapted by a multiplication factor of 10 in order to make them closer to the prices paid in the end user market. We stipulate a maximum theoretical power consumption for our *Intelligent Building* of little more than 4.2 kW; we assume that each simulated energy provider can provide in an hour a quantity of energy that is equal to a random value between 0 and 4.2 kWh. It is not then granted that just one provider can satisfy the energy needs of the *Intelligent Building*, but more of them could be considered as energy providers at the same time. Though this is an approximation of possible Demand-Response implementations, it contains all the required components and price dynamics that are likely to be present in the future Smart Grid: a multitude of energy providers with different tariffs that change with high granularity (e.g., the hour). In addition, these prices are real.

Moreover, we consider the inclusion of micro-generation facilities as if they were available on the building. We simulate the presence of a photovoltaic (PV) installation and a small-scale wind turbine. Again, to make the simulation realistic, we use actual data coming from existing installations. For the PV installation, we consider the location of the building to be New York, U.S.A., with an installation of 2.4 kW of power. This is

<sup>3</sup><http://www.pjm.com/>

actually a real PV installation in New York at Dalton School in Manhattan<sup>4</sup>, which has a PV array installation and whose real-time data can be accessed through the School Power Naturally data portal<sup>5</sup>.

We simulate the presence of a small-scale wind turbine on top of the same building considering the average annual wind speed experienced in New York and the anemometer data obtained from the set of sensors measuring the environmental conditions on top of Dalton School. We simulate the presence on site of a Proven 2.5 wind turbine<sup>6</sup> which has a rating of 2.5 kW with a 12 m/s wind speed. To compute the actual power extracted from a wind turbine, a cubic relation applies [22]:  $P = \frac{1}{2}\rho AU^3 C_p$ , where  $\rho$  is the air density,  $A$  is the rotor swept area,  $U$  is the wind speed, and  $C_p$  is the power coefficient representing the efficiency of the turbine rotor. Once we have chosen the turbine, the parameters are known, in particular:  $A = \pi(\frac{3.5}{2})^2$  (the turbine blades have a 3.5 meter diameter),  $\rho = 1.225$  (typical air density value), and  $C_p = 0.35$  (a typical value of rotor efficiency for wind turbines). We assume to have the wind data every hour and constant during the whole hour.

Regarding the pricing of the energy produced locally, firstly, we consider the wind turbine as a *sunk cost*, that is, the energy produced is for free, as its investment has been already amortized. On the other hand, for the PV we assume a price of \$0.12 per kWh by considering the investment cost and the energy produced over the expected lifetime of the PV array. More precisely,  $EC_{PV} = C_{Inv}/En_L$ , where  $C_{Inv}$  is the initial total investment cost for the PV array, and  $En_L$  is the estimated overall energy to be produced during the lifetime (supposed to be 40 years, i.e., the double the amount certified by the producer of the specific panels installed at Dalton School)<sup>7</sup>. This increased lifetime, compared to the panel specifications used at the Dalton site, represents the fact that most often the lifetime of solar panels is longer than the minimum guaranteed by the producing companies, and new industry norms will soon likely to be at least 30 years [23]. Secondly, we estimate a production of energy during the 40 years that is on average the same as the one produced in the previous years since the installation. Thirdly, the investment cost is based on the results of Wise *et al.* [24], who investigated the cost of PV panels in the U.S.A. The cost that emerges from their analysis, considering the cost for PV panels, inverters, and installation once the incentives applied by the U.S. government are subtracted, is \$5.1 for each installed watt of power.

2) *Implementation of the Wireless Device Network:* We use Plugwise<sup>8</sup> adapters consisting of plug-in adapters that fit between a device and the power socket. The adapters can turn the plugged mains device on and off, and can at the same time measure the power consumption of the device that is attached. The plugs are called “Circles” and they form a wireless ZigBee mesh network around a coordinator (called “Circle+”). The network communicates with the Controller

through a link provided by a USB stick device (called “Stick”). One typical Plugwise network is illustrated in the bottom part of Figure 4.

3) *Implementation of the Gateway:* The Gateway is a process running in the background, providing two functionalities: (i) Information Gathering, reporting power consumption and state of controlled devices; (ii) Device Control, used to turn the devices on and off. It is written in Perl using xPL Protocol<sup>9</sup>. In the subcomponent, illustrated in Figure 4,

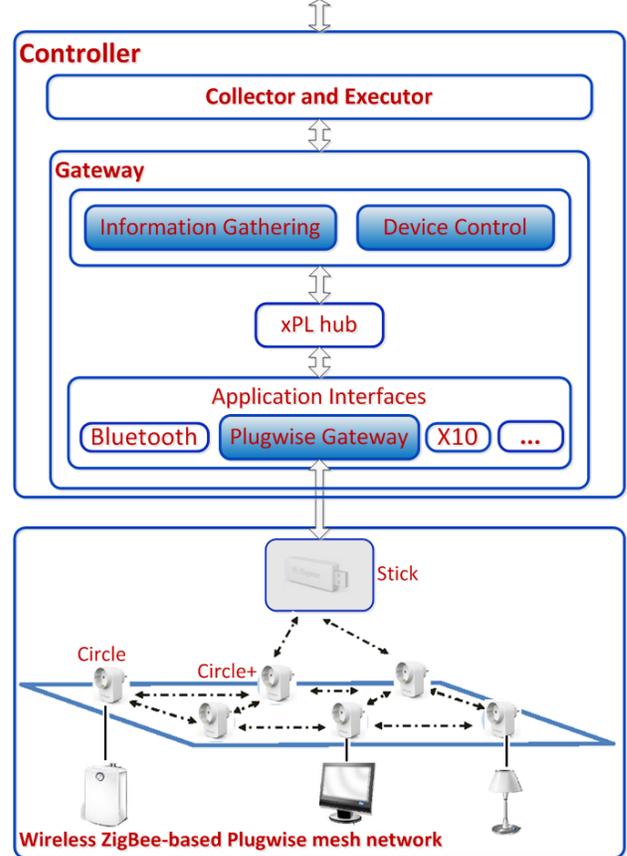


Fig. 4: Subcomponents of the Controller and Wireless Device Network

the *Application Interfaces* allow the interoperation of devices (based on possibly different protocols such as ZigBee, X10, Bluetooth, Infrared) and the xPL Protocol. The *xPL Hub* can bridge various application interfaces and is responsible for passing on the message to the application level for information gathering. It also collects back device control instructions that need to be forwarded to the device network.

4) *Repository, Collector and Executor:* The Repository and the CE components are implemented as a Web server that can be accessed with a simple standard protocol, namely, the Jetty<sup>10</sup>, HTTP Java-based server, and Representational State Transfer (REST) [25] for the communication. Each resource is mapped to a certain resource identifier, usually a Uniform Resource Identifier (URI). For example, assuming the Repository web server is installed on a local host, the web service for

<sup>4</sup><http://www.dalton.org/>

<sup>5</sup><http://sunviewer.net/portals/NYSERDA/index.php>

<sup>6</sup>[http://www.windandsun.co.uk/Wind/wind\\_proven.htm](http://www.windandsun.co.uk/Wind/wind_proven.htm)

<sup>7</sup><http://atlantasolar.com/pdf/Astropower/ap-100.pdf>

<sup>8</sup><http://www.plugwise.com>

<sup>9</sup><http://xplproject.org.uk/>

<sup>10</sup><http://eclipse.org/jetty/>

getting the device’s information can be accessed by calling the URI <http://localhost:8080/repository/services/devices>. A client can access these resources and transfer the content using methods that describe the actions to be performed on the resource. The methods are analogous to typical HTTP methods such as GET and POST that describe read and update actions. Each method from the *WS Interface* component calls an appropriate operation from the DAO component, see Figure 2. The DAO implements operations that store and retrieve information. It also forms appropriate XML data representation needed for other components in the architecture. We use Java Architecture for XML Binding<sup>11</sup> as a technique to map model objects to an XML representation or vice versa. DAO achieves data persistence by using Hibernate framework [26] that enables transparent and automatic mapping of the system domain object model into a database. We use MySQL<sup>12</sup> as a relational database management system for all databases.

5) *Implementation of the Scheduler*: The Scheduler is a standalone program module written in the Scala programming language<sup>13</sup> that is called by the Coordinator whenever there is a need to create a schedule for the following time period. The Scheduler obtains the information about the energy supply and prices from the Smart Grid via the Controller in XML format. Also, it uses the information about the devices and their policies, presented in this format as well. The schedule, created as an XML object, is returned to the Controller, and contains a set of actions that should be performed with each device during the next time period.

6) *The Coordinator*: The Coordinator plays a role of a client to the Repository and to the Controller through the CE subcomponent. We use the same technology as for the Repository and CE, that is, a Jersey-based client to consume HTTP-based REST web services requests.

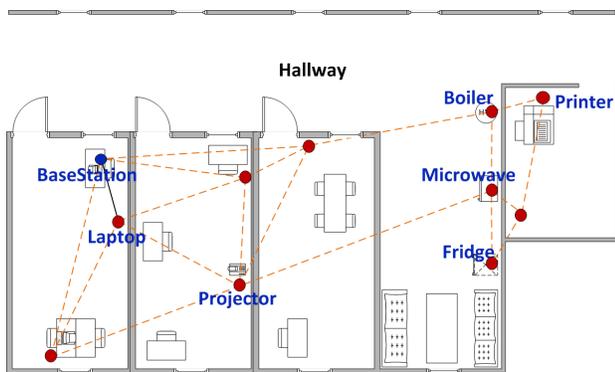


Fig. 5: Living lab setup

### Discussion

The wireless network of plug-in adapters presents a relatively stable topology. We experienced in general a quite good stability during system performance of data collection and command execution. Having a ZigBee network deployed

in our building environment, we faced some communication issues due to the radio disturbed environment. In particular, we observed that the microwave, while in working mode, affects the transmission of data through the frequency band of the ZigBee network. In most such cases, the data delivery ratio is lower than 100% (e.g., from 166720 collections, we expected to collect 1000320 measures, but we received 977724 measures), i.e., the information for a particular device or devices is lost. We did not try to solve this issue because the system collects data fairly often so that it does not lose the records of any important state changes. However, one possible solution for the transmission loss would be to displace the microwave far enough not to interfere with the wireless network of Plugwise devices. Unfortunately, relocating the microwave in our environment was not possible due to space limitations. Another way to improve data transmissions would be to use an acknowledgement process included within the communication [27].

Similarly, we noticed another inconvenience when at times the system would not execute the controlling commands for the devices. In fact, there were two reasons for this behavior. The first relates to the above-described radio disturbances. The other corresponds to the responsiveness of the Plugwise devices themselves. In particular, as the system is collecting data continuously, the execution of a command performed at the same moment as the collection of data was not successful. To resolve the responsiveness issue, we employed programmatically employed a simple form of reliable messaging with message acknowledgement. In this way, the system re-executes the command until the plug-in adapter is turned into the desired state.

## VI. EVALUATION

We have deployed the system in our own offices at the University of Groningen in order to assess the possible savings obtainable with such a system. Our offices are located on the fifth and last floor of a more than 10000 m<sup>2</sup> recently erected building.<sup>14</sup> The test site consists of three offices occupied by permanent and PhD staff, a coffee corner/social area, and a printer area. The layout is illustrated together with the ZigBee network and the electrical appliances in Figure 5. In particular, we include in our testing six available devices (a fridge, a laptop, a printer, a projector, a microwave, and a water boiler). The rated power plate consumptions of the fridge and the laptop are 70 W and 90 W respectively, while that for the printer is 100 W. The projector consumes 252 W when working, while the microwave 1500 W. The water boiler consumes when heating up to 2200 W. Four other sensor nodes are also comprised in the network to strengthen the mesh network connections. We use a set of Plugwise plugs forming a wireless ZigBee mesh network around a coordinator (called “Circle+”). The network communicates with the BaseStation through a link provided by a USB stick device (called “Stick”).

We have used the system over three weeks in the months of October and November 2011, and one week in the month of March 2012, performing measurements from Monday to

<sup>11</sup><http://jaxb.java.net/>

<sup>12</sup><http://www.mysql.com/>

<sup>13</sup><http://www.scala-lang.org/>

<sup>14</sup><http://nl.wikipedia.org/wiki/Bernoulliborg>

Friday (as in the weekend there is irregular presence). In particular, in the first 2 weeks (W1-W2) we measured energy use in order to define a baseline. The third week (W3) in 2011 and the fourth week (W4) in 2012, we let the scheduling component control the environment in order to measure the actual savings. We used the REPEAT policy for the fridge (turn on for 15 minutes each hour) and the boiler (turn on for 15 minutes each two hours). The printer used the MULTIPLE policy, and was assigned three jobs over the course of four hours. The microwave used the PATTERN policy, so we used the statistical information from the previously collected data to calculate the expected level of microwave consumption at each hour of the day. The laptop used the TOTAL policy, so it had to be charged for a total one hour during four hours scheduled slots. During week W3, we used the laptop each day. During week W4, we introduced variability of policies usage, so the laptop was used during Tuesday and Thursday. Projector used the STRICT policy to strictly follow the agenda of presentations. During week W3, presentations were given each day from 2 p.m. to 3 p.m. During week W4 presentations were given on Tuesday and Wednesday from 2 p.m. to 4 p.m., thus two hours each.

Next, we present the results in terms of economic savings (due to the varying prices of the Smart Grid) and of energy savings (due to the introduction of device policies).

#### A. Economic savings

The goal of the system is to save money for the office by taking advantage of the Smart Grid. Therefore, the first evaluation we make is based on taking the energy bill for a week using the system versus a week without it. We have considered two situations for office environments to evaluate the economic benefits of the proposed device scheduling policy: (1) an intelligent office building that interacts with the Smart Grid Demand-Response tariff service and has small scale renewable installations in its premises that provide power (W3 simulation), and (2) a more ordinary office that has no renewable-based power installation that provide power (W4 simulation) and that benefits only from the tariff differentiation of the Smart Grid. To obtain a fair comparison in the two simulations, we use the energy prices of the third week (W3) and fourth week (W4) and apply those same retrieved prices for the energy consumed in the other two weeks (W1-W2).

In the first set of simulations (office with on site small-scale renewable sources), the situation between each working day of the two weeks (average) without scheduling policies and the week where the policy has been applied is shown in Figure 6, where the price of energy (\$ per kWh) is shown versus the time of the day (from Monday to Friday). It is interesting to notice the difference in the average price paid for each kWh of energy in the situation without device scheduling and, on the other hand, considering scheduling. The chart is shown in Figure 9 (top chart). On average, the price in \$ per kWh drops by more than 27% in the two situations. An interesting day where the savings on energy expenses are particularly significant is between the three consecutive Thursdays monitored (October 20th, 27th, and November 3rd).

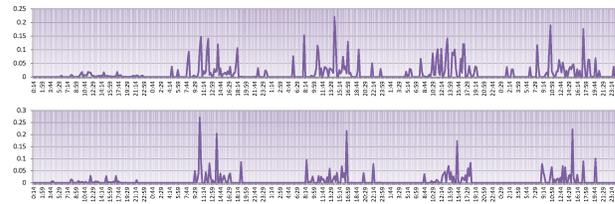


Fig. 6: Average price (\$ per kWh) comparison between non-scheduled (upper chart) and scheduled (lower chart) appliances for each work day (W3 experiment)

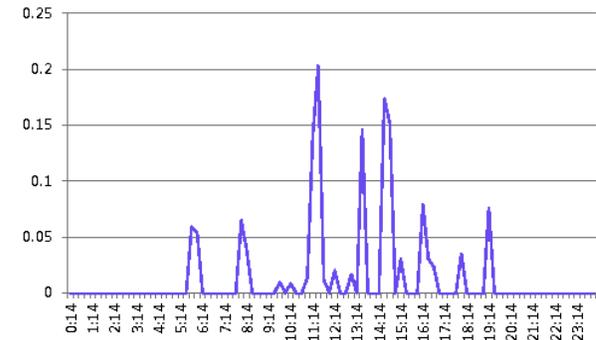


Fig. 7: Price of energy (\$ per kWh) during non-scheduled day October 27th

Comparing these three days, the money savings are on average more than 50%. A comparison between the price paid for energy in each hour between the situation in October 27th and November 3rd is shown in Figures 7 and 8, respectively. In particular, one can see the cut-off of unnecessary energy expenses related to those consumptions that happen during non-working time (late evening or during the night) by devices that are not strictly necessary (most notably the hot water boiler). Another optimization the system achieves is the most efficient schedule of devices, when the energy generated by photovoltaic panel is more intense and whose cost is generally smaller than energy provisioning on the market.

To validate the scheduling policy, in W4 we consider an office without renewable energy sources (whose price is generally cheaper than energy provision market according to the assumption made in Section V). Results comparing the

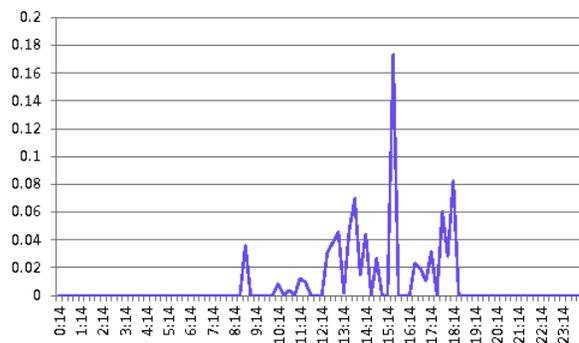


Fig. 8: Price of energy (\$ per kWh) during scheduled day November 3rd

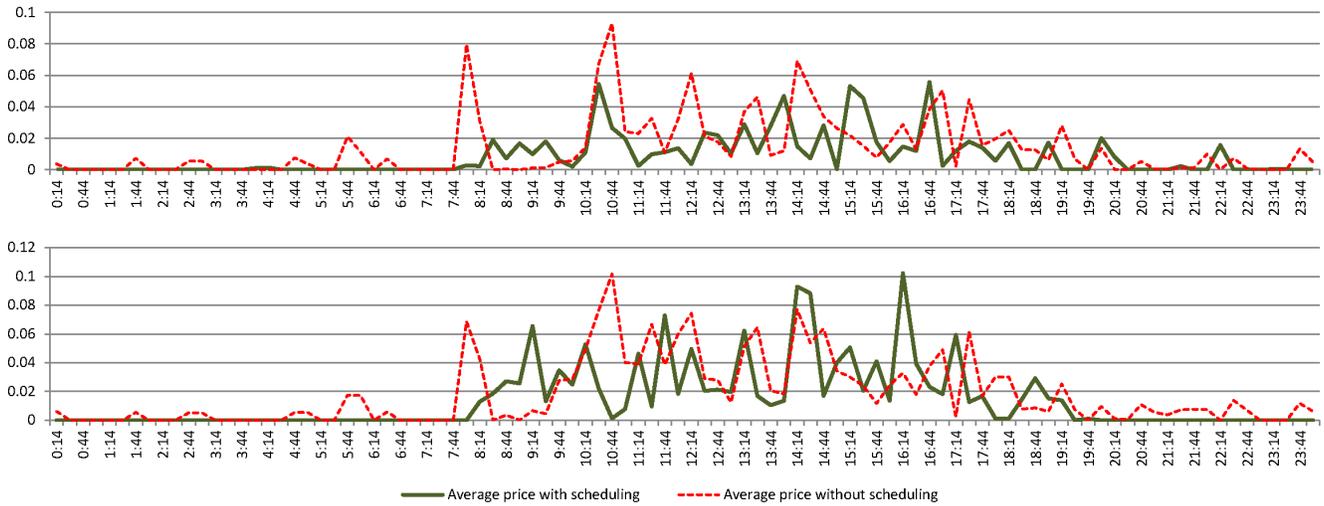


Fig. 9: Average price (\$ per kWh) comparison between scheduled (continuous line) and non-scheduled (dashed line) situations (top W3 experiment, bottom W4 experiment)

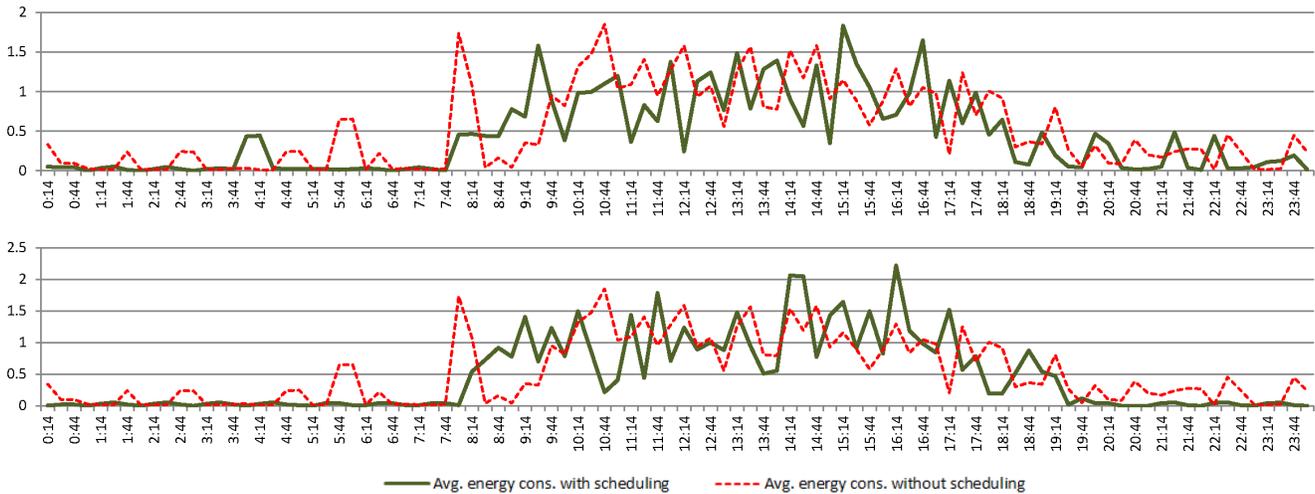


Fig. 10: Average energy usage (kWh) comparison between scheduled (continuous line) and non-scheduled (dashed line) situations (top W3 experiment, bottom W4 experiment)

day-by-day average price between the scheduling situation and the non-scheduling one are shown in Figure 11, while the daily average is shown in Figure 9 (bottom chart). One can see that the average price paid when scheduling is active is usually lower than the non-scheduled situation (cf. the continuous and dashed line in Figure 9); the overall economic savings between the situation when the schedule is implemented and when it is not is about 22%. The lower savings compared to the W3 experiment are due to the absence of renewable sources in the energy mix of the office, which we have assumed cheaper than the traditional energy market provider prices.

### B. Energy Savings

Although energy use reduction is not the primary aim of the system, but rather economic savings based on dynamic pricing, the use of policies for devices alone provides for energy saving in absolute terms. Figure 10 (top chart) shows the

average energy consumption (kWh) considering the use and the absence of the scheduling system comparing W1-W2 and W3 scenarios and Figure 10 (bottom chart) compares W1-W2 and W4 scenarios. The scheduling reduces the consumption of devices that are not used during non-working hours and that do not impact the habits of the users (e.g., keeping the hot water boiler working at night); in addition, the Scheduler tries to use at best the cheap electricity coming from the solar panels during daylight hours. Figure 12 visually reinforces the idea of reducing loads when unnecessary among the normal (first upper chart) and the scheduled solutions (the middle and bottom charts): one notices a more compact chart in which energy is used mostly during daytime (8 a.m.-6.30 p.m.) in each day of the week. The average savings of energy consumed between the situation without the scheduling policy and the situation considering it, is more than 15% (W1-2 versus W3 experiment) and about 11% (W1-2 versus W4 experiment),

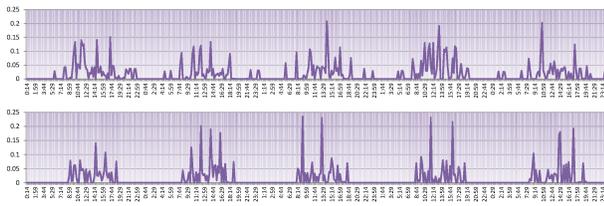


Fig. 11: Average price (\$ per kWh) comparison between non-scheduled (upper chart) and scheduled (lower chart) appliances for each work day (W4 experiment)

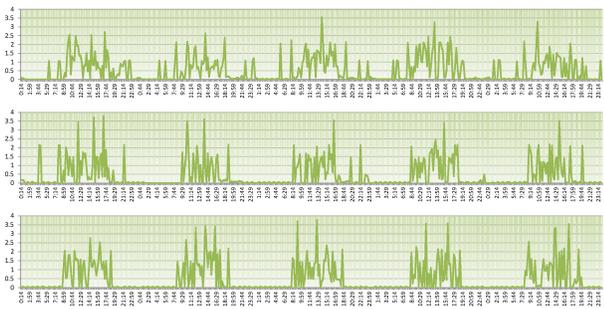


Fig. 12: Energy (per kWh) comparison between non-scheduled (upper chart) and scheduled (middle and bottom chart respectively W3 and W4) appliances for each work day

respectively. We ascribe the small difference in percentage to the unpredictable usage of equipment in the actual living lab between the two weeks (e.g., microwave use).

### C. Discussion on System Performance

Finding the optimal schedule for a set of devices is a computationally expensive problem and while there exist many tools that can solve such problems reasonably fast for practical domain sizes [18], we took a set of measures to ensure that our solution will remain within practical bounds for bigger lab settings. There are three dimensions that determine the input size of the scheduling task: number of energy providers, time period of the schedule, and number of devices.

The increase in the *number of energy providers* has negligible impact on the performance of the Scheduler. The reason for this is that the function of price levels has to be computed only once at the beginning of the scheduling task, as described in Algorithm 1. During the actual schedule search, we refer to the pre-calculated function, and the time for such a referring does not depend on the original number of energy providers.

The time it takes for the Scheduler to find the optimal schedule grows with the *number of time units* for which we are obtaining a schedule. We tried to vary the time period of the schedule from 1 hour to 12 hours; the average length of the Scheduler running time is shown in Figure 13. As can be seen in this figure, even for 12 hours period it takes only about 1.4 seconds to find the optimal schedule for our living lab setting.

The *number of devices* causes the biggest strain on the system’s performance. Since in the living lab we had only 6 devices, to test the Scheduler with a larger number of

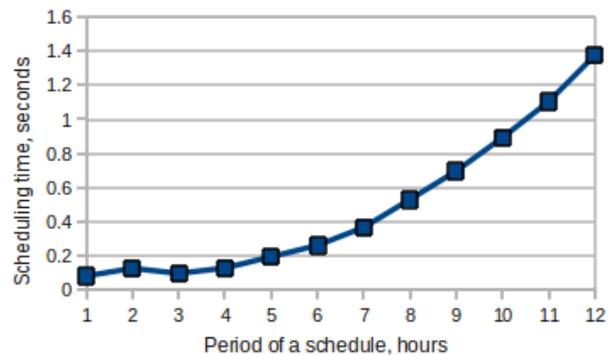


Fig. 13: Scheduler performance dependence on the schedule time period

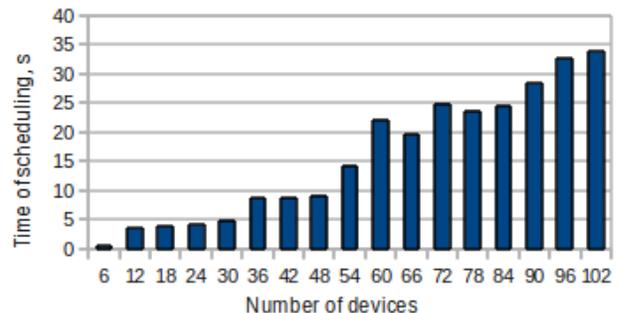


Fig. 14: Scheduler performance dependence on the number of devices

them, we simulated devices by creating multiple copies of the available devices. We determined that a search for the optimal schedule can take an impractically large amount of time for large centrally controlled buildings. This is less of a problem that it might initially appear to be. In fact, one can dynamically relax the requirement for optimality and search instead for a “good enough” schedule. For our scheduling algorithm, we implemented a gradual approach. For a large number of devices, we divide them into groups of approximately equal size. We run the Scheduler for the first group, and find the optimal solution for it. Then, given this schedule for the first group (which we do not change while scheduling the other groups), we calculate the increased amount of energy used at each time unit, and we run the Scheduler for the next group of devices, finding the optimal solution for them. After this, we recalculate the increased amount of energy again and run the Scheduler for the third group, and so on, until all devices are scheduled. Note that, while this approach follows a greedy practice, the schedule provided is still quite efficient in terms of price savings and smart distribution of devices working time. If the devices from the first group were scheduled to run at a certain time unit, the amount of energy already consumed at this time will be large; this will prevent the Scheduler from placing more devices from the second group in the same time slot. So the Scheduler is still able to distribute the working time of devices across different time units even for devices from different groups. In Figure 14, we show the averaged running time of the Scheduler for a different number of devices.

#### D. Occupant Experience Evaluation

The requirements for defining the device policies are based on the occupant needs in terms of “occupant satisfaction.” To evaluate this aspect, after experimenting in our own offices, we conducted a study by means of a questionnaire among 18 people who used scheduled facilities and were not aware of the experiments being conducted.

We consider the definition of *occupants’ satisfaction* presented in [28], where for office buildings one considers indoor workspace and environmental qualities (thermal, visual, acoustic levels, and air quality). In addition, in [28], building occupants’ satisfaction is also related to the view, control over the indoor environment, and amount of privacy as well as layout, size, cleanliness, aesthetics, and office furniture. In our context, the questionnaire evaluation and focus is on “control over the indoor environment.” In particular, we asked the people how often they used each of the scheduled facilities, and their experience with using the facilities in the period of experimentation. The goal was to determine whether, during the experiments, the scheduled facilities were perceived as functioning as usual or the scheduling was remarkable and, in that case, if it hindered the comfort and productivity.

Figure 15 summarizes the results of the questionnaires. The answers show that all people were able to use the microwave, the fridge, and the projector as they usually do. Our colleagues were able to use the microwave to warm their food and drinks when they wanted to. The fridge kept food and drinks cool enough. All people used the projector normally for presentations. For the water boiler, only one PhD staff out of 18 complained that once he did not have enough hot water, while the laptop was run off battery power one time, see Figure 15. Some people were aware of the type of research we were conducting, but they did not know when it actually would take place. The majority of the 18 people were completely unaware of the study. In summary, the results of the occupant questionnaire show that occupants’ satisfaction is confirmed. In other words, the comfort and productivity was not affected in any significant way during the running of the experiments.

#### VII. CONCLUSION

The Smart Grid promises to bring important advantages not only to the network operators, but also to the final consumers. Building managers can deploy systems to take advantage of the dynamic pricing and the availability of more providers by monitoring and controlling their devices. It is neither foreseeable nor desirable to do such control by hand or by enforcing policies on office users; rather one can think of automatic systems that work in the background and do not affect the comfort and productivity of the building inhabitants.

In this paper, we proposed a system to monitor and control electrical appliances in a building in order to save energy costs. This is achieved by coupling use with dynamic energy prices and electricity generated locally to the building with renewable sources. The system is fully implemented into a prototype system, and its deployment for a few weeks in our own offices has shown a high potential for the system, with savings of money up to 50% and of energy up to 15%.

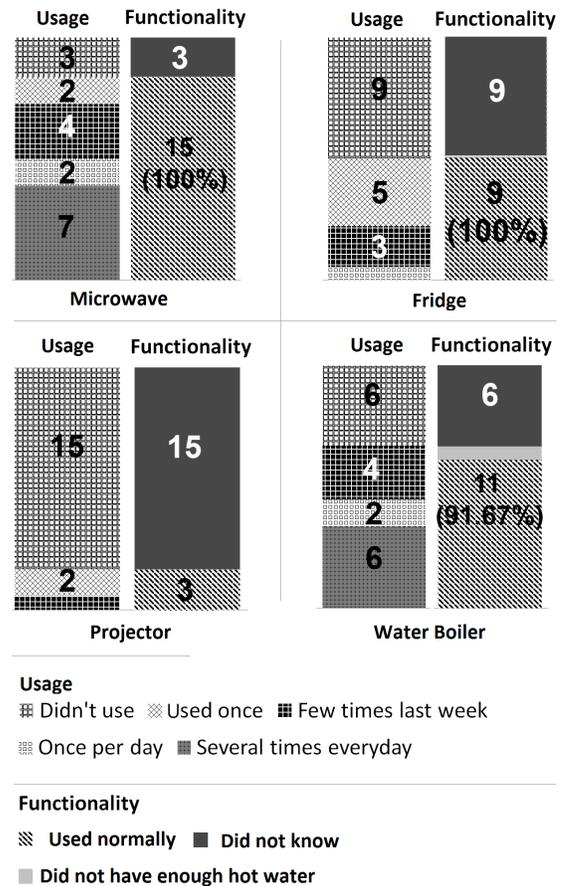


Fig. 15: Facilities’ functionality

The building heating and cooling system affects occupant’s thermal comfort while lighting equipment affects visual comfort of the occupants. Thus, in order to satisfy occupants’ comfort, the BEMS of an *Intelligent Building* mostly uses occupancy information for control strategies [29]. For this reason, policies for heating and cooling system and lighting equipment are not addressed in our current proposed solution, although they are recognized as very important contributors to the peak demand for energy. However, the study presented in [12] shows, for example, that the implementation of pre-cooling effect during the off-peak period and the discharge of autonomous cooling during the peak period can bring substantial energy and cost savings. This idea is in line with the principles of our proposed solution, thus obvious directions for our future work include taking into account occupancy information, defining policies for heating and cooling system and lighting equipment.

Incidentally, we remark that the system itself consumes energy to operate; it consists of 10 Plugwise devices and one desktop computer that respectively consume a maximum power of 1.1 W and 365 W, respectively. The value of the plugs is insignificant with respect to the overall consumption. As for the computer, a few remarks are in order: firstly, the optimization program does not need to run on a dedicated computer, so it could add little consumption to the already active computers. Secondly, in a real operational environment,

the system would schedule many more devices; thus, its energy consumption would be amortized over larger savings. For these reasons, we have not included these energy consumptions in the current evaluation. We plan to continue our evaluation of optimization algorithms by including more devices in terms of variety and number and in the long term expand the testing to an entire building. In our future efforts we have plans to include also personalized policies for devices in their access to energy supply. In this way users can set their preferences for devices or, even more challenging, the system controlling the devices can learn their policies based on the customary behavior of the user.

## REFERENCES

- [1] A.-H. Mohsenian-Rad, V. Wong, J. Jatskevich, and R. Schober, "Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid," in *Innovative Smart Grid Technologies (ISGT), 2010*, January 2010, pp. 1–6.
- [2] International Energy Agency, "World energy outlook," International Energy Agency, Tech. Rep., 2009.
- [3] Accenture end-consumer observatory on electricity management, "Understanding consumer preferences in energy efficiency," Accenture, Tech. Rep., 2010.
- [4] S. Tompros, N. Mouratidis, M. Draaijer, A. Foglar, and H. Hrasnica, "Enabling applicability of energy saving applications on the appliances of the home environment," *Network, IEEE*, vol. 23, no. 6, pp. 8–16, November–December 2009.
- [5] A. Capone, M. Barros, H. Hrasnica, and S. Tompros, "A new architecture for reduction of energy consumption of home appliances," in *TOWARDS eENVIRONMENT, European Conference of the Czech Presidency of the Council of the EU*, 2009.
- [6] C.-S. Choi, J. Han, W.-K. Park, Y.-K. Jeong, and I.-W. Lee, "Proactive energy management system architecture interworking with smart grid," in *Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on*, June 2011, pp. 621–624.
- [7] G. Giaconia, G. Fiscelli, F. Bue, A. Di Stefano, D. La Cascia, and R. Miceli, "Integration of distributed on site control actions via combined photovoltaic and solar panels system," in *Clean Electrical Power, 2009 International Conference on*. IEEE, 2009, pp. 171–177.
- [8] K. Kok, S. Karnouskos, J. Ringelstein, A. Dimeas, A. Weidlich, C. Warmer, S. Drenkard, N. Hatziaargyriou, and V. Lioliou, "Field-testing smart houses for a smart grid," in *21st International Conference and Exhibition on Electricity Distribution (CIRED 2011), Frankfurt, Germany*, 6–9 June 2011.
- [9] K. Kok, G. Venekamp, and P. Macdougall, "Market-based control in decentralized electrical power systems," in *First International Workshop on Agent Technologies for Energy Systems (ATES 2010) at the Ninth Conference on Autonomous Agents and Multiagent Systems AAMAS 2010, Toronto, Canada*, 2010.
- [10] A. Almeida and P. Fonseca, "Residential monitoring to decrease energy use and carbon emissions in Europe," in *International Energy Efficiency Domestic Appliances and Lighting Conference*, 2006.
- [11] H. Vogt and H. Weiss, "A client architecture for market-based grid integration of smart environments," in *PerCom Workshops*, 2010, pp. 642–647.
- [12] J. Ma, S. Qin, B. Li, and T. Salsbury, "Economic model predictive control for building energy systems," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, January 2011, pp. 1–6.
- [13] J. Han, Y. Jeong, and I. Lee, "Efficient building energy management system based on Ontology, inference rules, and simulation," in *International Conference on Intelligent Building and Management*, vol. 5, 2011, pp. 295–299.
- [14] R. Miceli, D. La Cascia, A. Di Stefano, G. Fiscelli, and C. Giaconia, "Impact of novel energy management actions on household appliances for money savings and co2 emissions reduction," in *Proceedings of EVER09 Conference, Principato di Monaco*, 2009.
- [15] C. Gezer and C. Buratti, "A ZigBee smart energy implementation for energy efficient buildings," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, May 2011, pp. 1–5.
- [16] N. Capodiecì, G. A. Pagani, G. Cabri, and M. Aiello, "Smart meter aware domestic energy trading agents," in *First International E-Energy Market Challenge Workshop*, ser. ICAC'11, 2011.
- [17] U. Kremer, J. Hicks, and J. Rehg, "A compilation framework for power and energy management on mobile computers," in *Languages and Compilers for Parallel Computing*, ser. Lecture Notes in Computer Science, H. Dietz, Ed. Springer, Berlin / Heidelberg, 2003, vol. 2624, pp. 115–131.
- [18] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 2nd Ed.* Englewood Cliffs, NJ: Prentice Hall, 2002.
- [19] G. Lu, D. De, and W.-Z. Song, "Smartgridlab: A laboratory-based smart grid testbed," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, October 2010, pp. 143–148.
- [20] W. Taqqali and N. Abdulaziz, "Smart grid and demand response technology," in *Energy Conference and Exhibition (EnergyCon), 2010 IEEE International*, December 2010, pp. 710–715.
- [21] K. Kok, Z. Derzsi, M. Hommelberg, C. Warmer, R. Kamphuis, and H. Akkermans, "Agent-based electricity balancing with distributed energy resources, a multiperspective case study," in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual Conference*, January 2008, p. 173.
- [22] L. Freris and D. Infield, *Renewable Energy in Power Systems*. Wiley, 2008.
- [23] K. Branker, M. Pathak, and J. Pearce, "A review of solar photovoltaic leveled cost of electricity," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 9, pp. 4470–4482, 2011.
- [24] R. Wiser, G. Barbose, and C. Peterman, "Tracking the sun: the installed cost of photovoltaics in the U.S. from 1998–2007," Lawrence Berkeley National Laboratory, Tech. Rep., 2009.
- [25] R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," *ACM Trans. Internet Technol.*, vol. 2, pp. 115–150, 2002.
- [26] C. Bauer and G. King, *Java Persistence with Hibernate*. Manning Publications Co., 2006.
- [27] M. Simek, M. Fuchs, L. Mraz, P. Moravek, and M. Botta, "Measurement of lowpan network coexistence with home microwave appliances in laboratory and home environments," *Broadband, Wireless Computing, Communication and Applications, International Conference on*, pp. 292–299, 2011.
- [28] M. Frontczak, S. Schiavon, J. Goins, E. Arens, H. Zhang, and P. Wargoeki, "Quantitative relationships between occupant satisfaction and satisfaction aspects of indoor environmental quality and building design," *Indoor Air*, vol. 22, no. 2, pp. 119–131, 2012.
- [29] T. A. Nguyen and M. Aiello, "Energy Intelligent Buildings based on User Activity: a Survey," *Energy and Buildings*, 2012, To appear.



**Ilche Georgievski** (S'12) received his engineering degree in computer science and informatics from the University of Maribor, Slovenia in 2010. He is currently enrolled as a Ph.D. candidate in computer science at the University of Groningen, The Netherlands. His research interests include automated planning, plan recognition, and smart systems.



**Viktoriya Degeler** (S'12) received her B.Sc. (2005) in Applied Mathematics from National Polytechnical University Kh.P.I., Kharkiv, Ukraine, and M.Sc. (2007) in Intelligent Systems Design from Chalmers University of Technology, Gothenburg, Sweden. She is currently a PhD candidate at the University of Groningen, The Netherlands. Her research interests include real-time optimization techniques, constraint satisfaction, and scheduling, with particular interest in intelligent buildings automation.



**Giuliano Andrea Pagani** (S'10) received the B.S. and M.S. degrees in computer engineering from University of Parma, Italy, in 2004 and 2006, respectively, and the M.S. degree in corporate management from M.I.P. Polytechnic of Milan, Italy, in 2009. He is currently working toward the Ph.D. degree in computer science at the University of Groningen, The Netherlands. His research focuses on distributed energy generation with particular interest in the fields of complex network analysis, graph theory, smart grid technology, and renewable energy.



**Tuan Anh Nguyen** received the B.S. and M.S. degrees in information technology from Hanoi University of Technology, Vietnam, in 2004 and 2006, respectively. He is currently working toward the Ph.D. degree in computer science at the University of Groningen, The Netherlands. His research interests currently focus on wireless sensor networks, activity recognition, and intelligent buildings.



**Alexander Lazovik** received his M.S. degree in applied mathematics and computer science from the Belarusian State University, Belarus, and the Ph.D. degree from the University of Trento, Italy. He is currently an assistant professor at the Johann Bernoulli Institute, University of Groningen, The Netherlands. His research interests cover the areas of service-oriented and distributed computing, particularly concerning automatic composition using AI planning and constraint propagation algorithms.



**Marco Aiello** (S'97 - M'02 - SM'12) received the engineering degree from the University of Rome La Sapienza, Italy, and the Ph.D. degree from the University of Amsterdam, The Netherlands. He was an Assistant Professor at the University of Trento, Italy, and a Lise Meitner Fellow at the Technical University of Vienna, Austria. He is currently a Professor of distributed information systems at the Johann Bernoulli Institute, University of Groningen, The Netherlands. His research interests are smart energy systems, service-oriented computing, and spatial reasoning. More information is available at <http://www.cs.rug.nl/ds>.