Optimizing Offices for the Smart Grid

Ilče Georgievski, Viktoriya Degeler, Giuliano Andrea Pagani,
Tuan Anh Nguyen, Alexander Lazovik, Marco Aiello

Distributed Systems Group
Institute for Mathematics and Computer Science
University of Groningen

JBI preprint 2011-12-01

November, 2011

Abstract

The Smart Grid promises to not only provide for a more reliable distribution infrastructure, but also give the end-users better pricing and 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 shows interesting economic saving of an average of about 35% and in some cases even overall energy savings in the order of 10%.

This 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.

This is a serie of preprints of the Johann Bernoulli Institute for Mathematics and Computing Science of the University of Groningen.

Contents

List of Figures

List of Tables

1 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, we have moved from a single provider/single tariff system, to models with competing providers and two prices over long term contracts (usually in the term of months). E.g., an energy provider can offer distinct tariffs for daytime and night-time or weekdays and week-end, 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, but they are rather a convex function which is composed by linear intervals with increasing slope as the energy used increase [1]. The situation promises to be even more delicate with the increase of renewable source of energy [2] as these imply a greater uncertainty of supply.

The renewables are and will increasingly be present not only at the medium-large scale on the Grid, but are also more and more 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 (CFO) or his delegated building manager, his goal is that of saving money on the energy bill, while keeping an adequate level of comfort and productivity from the employees working in the building. Honeywell claims that in a typical commercial building 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, buying energy at the lowest possible price.

Here we present an approach to controlling offices to save energy and overall energy bill costs assuming the availability of a Smart Grid which 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 which 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 forced to give up control of their building energy consumption equipment as advocated by several approaches. For instance dynamicDemand[1] promotes the integration in the appliances of technologies that can automatically enable them to respond to 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 given where a scenario is envisioned in which to face the lack of power, customers' air conditioning is automatically turned off[2]. A key finding by an customer-interview study [3] found 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. Such 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. We don't investigate the provider's point of view, but we conjecture that also the provider will experience significant financial benefit if most the end-users would be price driven in their energy use.

The remainder of the paper is organized as follows. In Section 2 we present the model of our system. Section 3 shows its general architecture and describes each component in details. Section 4 specifies the technologies we used during actual implementation of our system and living lab setting. Our experiments are described and evaluated in Section 5. Section 6 discusses the related work, and Section 7 concludes the paper.

2 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 (portion 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 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, beamer, can be measured in its electrical energy consumption in kWh and can be controlled. Each device

[1] `http://www.dynamicdemand.co.uk`

[2] `http://energy.iowa.gov/SmartGrid/SmartGrid.html`

has associated a 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 full access to reading the state of a device and can trigger a state transition. Data about energy consumption levels is 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, which 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, 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.

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	Beamer	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 1: Device policies

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 1 and defined next.

REPEAT. The device requires to operate cyclically by entering the state *sid* repeatedly with a certain periodicity. E.g., a fridge that should operate for 15 minutes each hour is specified via 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* a device should be put in a state *sid*. An example is a discharged laptop that needs to be charged for two hours, but the exact time, when it is going to happen, doesn't 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 bit 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 3 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 a 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 beamer 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 the way the device consume energy. Instead of offering the possibility of controlling the device, it provides information on expected energy usage that can help schedule other devices. For example, a microwave always stays turned on, but historical data shows the higher level of energy consumption is expected during lunch-time.

SLEEP. For a device for which there is no demand for the work during a given period, the $SLEEP$ policy can be used. This policy cannot be combined with any other policy for the same device. 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.

3 System Architecture

To take advantage of the dynamic pricing on the Smart Grid and the controllability of the devices, we design an architecture which 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.

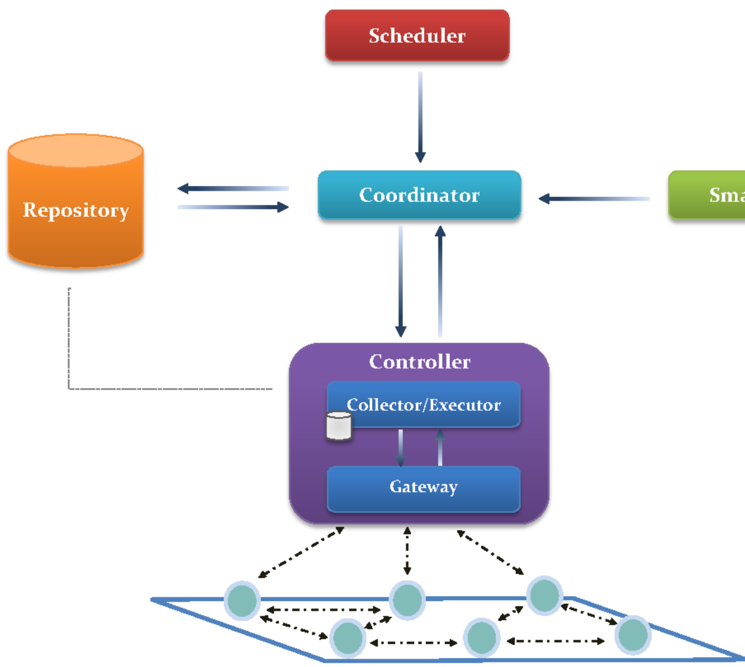

Figure 1: Architecture Design

3.1 Smart Meter

A Smart Meter is a physical device that is able to measure consumed and produced energy, provide this information to the energy company, and change electricity tariffs according to the signals received. In the 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 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 are 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, it 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.

3.2 Device Network

The Device Network, most usually realized as a Wireless Sensor Network, provides the basic infrastructure for gathering the information on device's power consumption, device's state and controlling appliances. Typically, this type of energy monitoring equipment are plugged into power sockets instead of running on battery. In addition they have embedded wireless chip that is sufficient to form a wireless mesh network around the gateway, providing a cost effective and dynamic high-bandwidth network, 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 [4]).

3.3 Controller

Controller consists of Collector and Executor (CE) subcomponent and a Gateway between 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 CE subcomponent for gathering information as well as controlling the devices. The Collector and Executor 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 Gateway subcomponent. In order to access lower-level tools of the Gateway in a more intuitive fashion, 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.

3.4 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 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: *WS Interface*, *DAO* and *Database*.

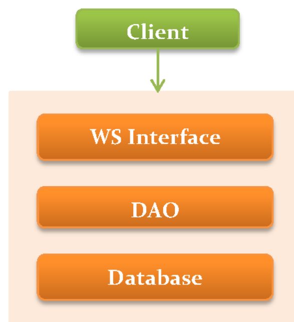

Figure 2: Repository Internal Architecture

The *Web Service (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 *Data Access Object* (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.

3.5 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. But 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 among the varying prices over the course of the day, not schedule too many devices at the same time so not to buy the more expensive energy, but at the same time keeping 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 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 than 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

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

s.t.

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

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.

An example of cost calculation for the energy providers in Table 2 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.

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,

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 2: Example of energy providers and prices.

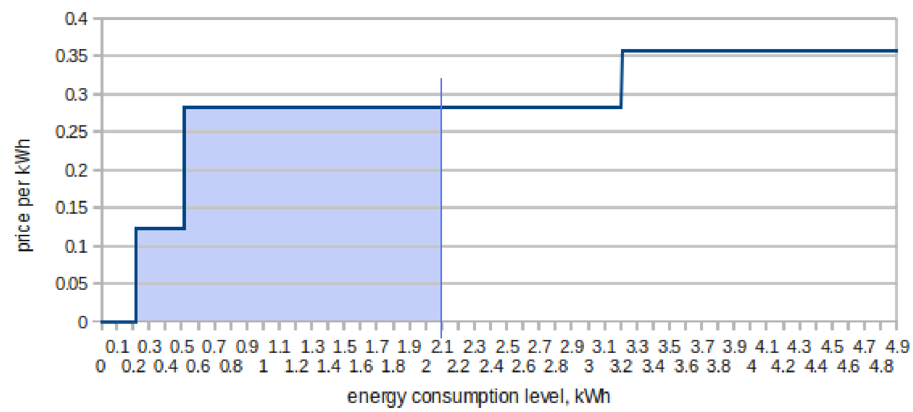

Figure 3: Price per kWh given the energy consumption. Total price paid equals to the area under the graph.

Algorithm 1 Cost depending on energy consumed

function getCost($time, energy$):$Double$
  $provs$ ←getProvidersAt ($time$)
  $sortedprovs$ ← Sort $provs$ by $provs(i).cost$
  $energyleft \leftarrow energy$
  $totalcost \leftarrow 0$
while $energyleft > 0$ **do**
  $prov \leftarrow sortedprovs.next$
  $totalcost \leftarrow totalcost + min(energyleft, prov.energy) * prov.cost$
  $energyleft \leftarrow energyleft - prov.energy$
end while
return $totalcost$

"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

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

where $e_t = \sum_{d \in D} x_{td}.energy$

For solving the problem we implement a priority queue with BFS algorithm. To decrease the search space we extensively use domain knowledge (per policy). For example, if a device has the policy TOTAL 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, as 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 it will not satisfy our policy. Another example is the policy MULTIPLE, 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 printer is turned on for 45 minutes, as it means the printer will definitely be idle for 15 minutes and unnecessarily consume energy.

3.6 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 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 servers 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 CE component web service to process changes deeper into the physical layer.

4 Implementation

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

4.1 Interfacing with the Smart Grid

Since the Smart Grid is not yet deployed and implemented for the end user, but just as proof-of-concepts [5], simulations of Smart Grid customer behaviour [6] or small scale pilot projects [7] and there are not yet agreed general available standards (though initiatives are underway from IEEE, NIST and others), we simulate the dynamic pricing. To make the simulation realistic, we use data and services coming 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[3] which is a regional transmission organization (RTO) 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 (LMP) which are the prices of energy negotiated in the wholesale market for the following day by energy companies at specific location where energy is delivered or received. Data contain the energy price for each energy unit (dollars per Megawatt-hour) for each hour of the day agreed for the next day at 20 location of delivery. The prices of the wholesale market are adapted by a 10 factor multiplication 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.

[3] `http://www.pjm.com/`

with an installation of 2.4 kW of power. This is actually a real PV installation in New York at Dalton School in Manhattan[4] which has a PV array installation and whose real-time data can be accessed through the School Power Naturally data portal[5].

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 the anemometer data obtained from the set of sensor measuring the environmental conditions on top of the Dalton School. We simulate the presence on site of a Proven 2.5 wind turbine[6] 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 [8]: $P = \frac{1}{2}\rho A U^3 C_p$ where ρ 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 meters 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, we consider the wind turbine 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 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)[7]. Second, 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. Third, the investment cost is based on the results of Wise *et al.* [9] that investigate 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 dollar for each installed watt of power.

4.2 Implementation of the Wireless Device Network

We use Plugwise[8] 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

[4] `http://www.dalton.org/`
[5] `http://sunviewer.net/portals/NYSERDA/index.php`
[6] `http://www.windandsun.co.uk/Wind/wind_proven.htm`
[7] `http://atlantasolar.com/pdf/Astropower/ap-100.pdf`
[8] `http://www.plugwise.com`

(called 'Stick'). One typical Plugwise network is illustrated in the bottom part of Figure 4.

4.3 Implementation of 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[9]. In the subcomponent, illustrated in

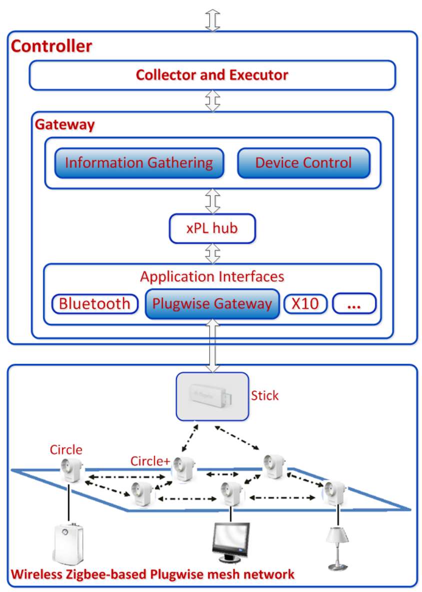

Figure 4: Subcomponents of Controller and Wireless Device Network

Figure 4, 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 instruction that need to be forwarded to the device network.

4.4 Repository, Collector and Executor

The Repository and the Collector and Executor components are implemented as Web server that can be accessed with a simple standard protocol, namely, the Jetty[10], HTTP Java-based server and Representational State Transfer (REST) [10]

[9] `http://xplproject.org.uk/`
[10] `http://eclipse.org/jetty/`

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 getting the devices 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 appropriate operation from 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[11] (JAXB) as a technique for mapping model objects to an XML representation or vice versa. DAO achieves data persistence by using Hibernate framework [11] that enables transparent and automatic mapping of the system domain object model into a (relational) database. We use MySQL[12] as a relational database management system for all databases.

4.5 Implementation of the Scheduler

The Scheduler is a standalone program module written in the Scala programming language[13] 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.

4.6 The Coordinator

The Coordinator plays a role of a client to the Repository and to the Controller through the Collector and Executor 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.

5 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 10.000+ m^2 recently erected building.[14]

[11] `http://jaxb.java.net/`
[12] `http://www.mysql.com/`
[13] `http://www.scala-lang.org/`
[14] `http://nl.wikipedia.org/wiki/Bernoulliborg`

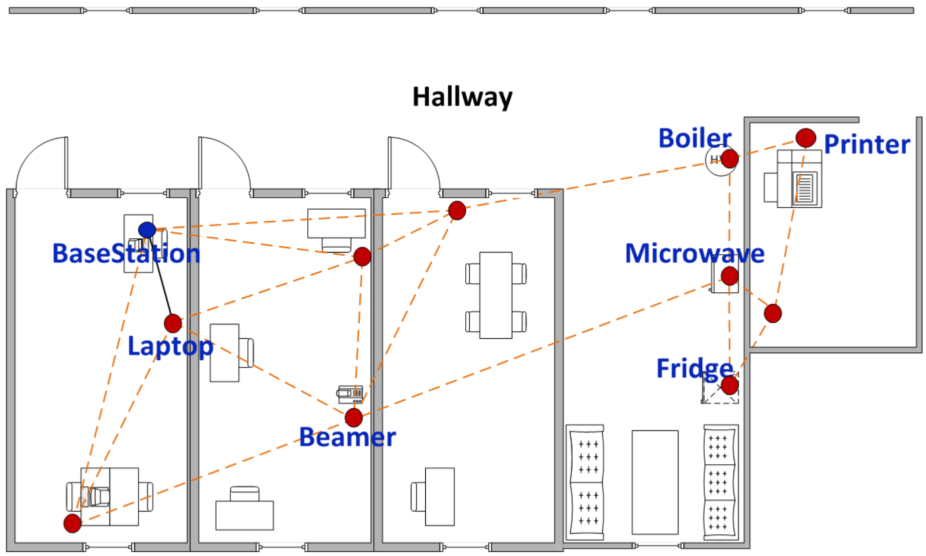

Figure 5: Living lab setup

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 beamer, a microwave, and a water boiler). The power consumptions of the fridge and the laptop are 70 Watts and 90 Watts respectively, while the one for the printer is 100 Watts. The beamer consumes 252 Watts when working while the microwave 1500 Watts. The water boiler consumes when heating up to 2200 Watts. 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 performing measurements from Monday to 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), we let the scheduling component control the environment in order to measure the actual savings. 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).

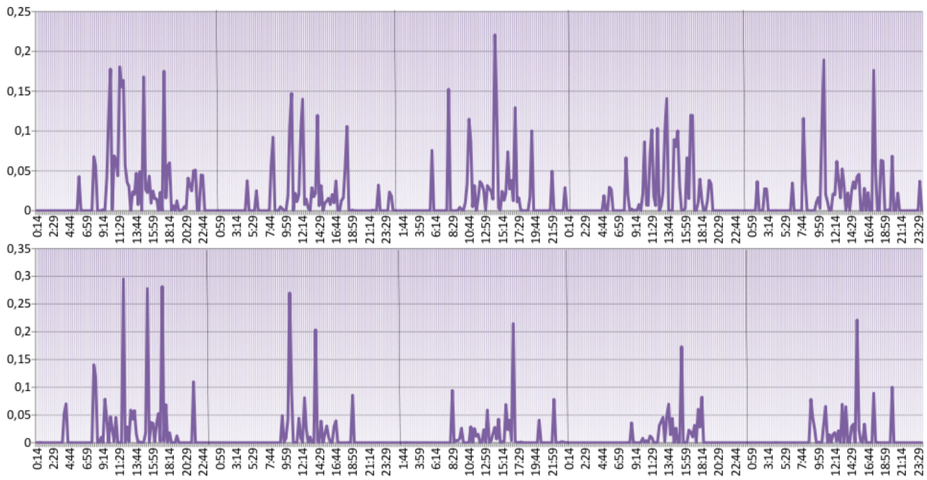

Figure 6: Average price ($ per kWh) comparison between non-scheduled (upper chart) and scheduled (lower chart) appliances for each work day.

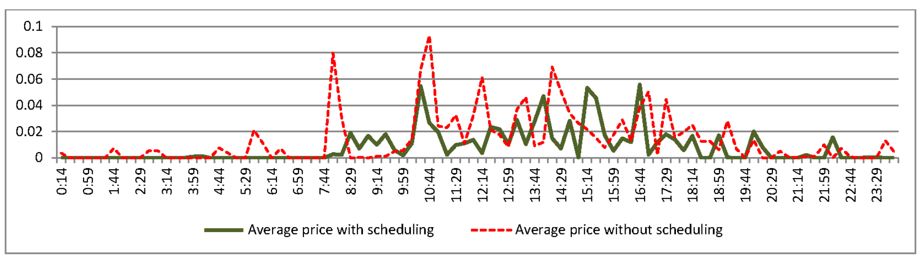

Figure 7: Average price ($ per kWh) comparison between scheduled (continuous line) and non-scheduled (dashed line) situation.

5.1 Economic savings

The goal of the system is to save money for the office 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. To make the comparison fair, we use the energy prices of the third week (W3) and apply those same retrieved prices for the energy consumed in the other two weeks. 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 $ 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 7. 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

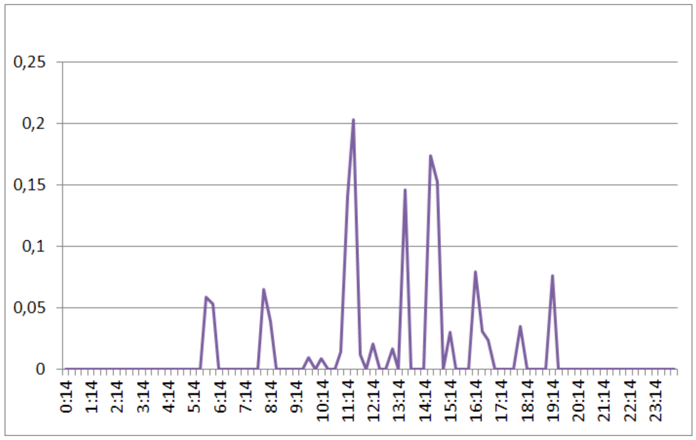

Figure 8: Price of energy ($ per kWh) during non-scheduled day October 27th.

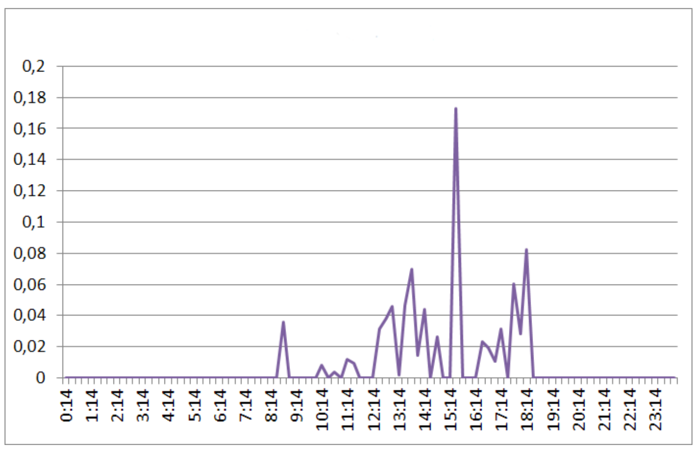

Figure 9: Price of energy ($ per kWh) during scheduled day November 3rd.

is between the three consecutive Thursdays monitored (October 20th, 27th and November 3rd). 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 8 and 9. In particular, one can see the cut of unnecessary energy expenses related to those consumptions that happen during non-working time (late evening or during the night) by non strictly necessary devices (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.

5.2 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 shows the average energy consumption (kWh) considering the use and the absence of the scheduling system. One can see that the scheduling reduces the consumption of devices that are not used during non working hours and that do not impact the habits of the user (e.g., keeping hot water boiler working at night); in addition the scheduler tries to use at best the cheap electricity coming from the solar panels during day-light hours. Figure 11 visually reinforces the idea of reducing loads when unnecessary among the normal (upper chart) and the scheduled solution (lower chart): one notices a more compact chart in which energy is mostly used during daytime (8 a.m. 6.30 p.m.) in each day of the week. On average the savings in energy consumed between the situation without the scheduling policy and the situation considering it, is more than 15%.

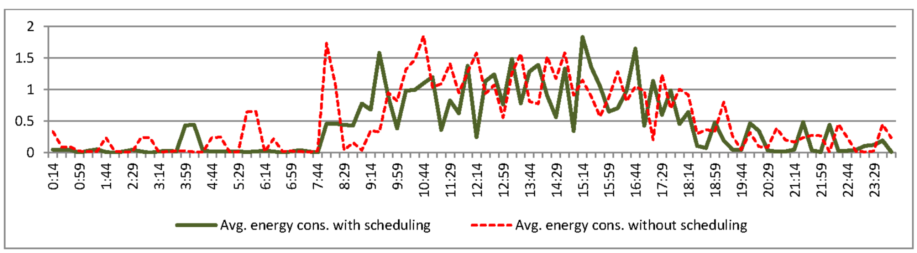

Figure 10: Average energy usage (kWh) comparison between scheduled (continuous line) and non-scheduled (dashed line) situations.

5.3 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 [12], 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 only has to be computed 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

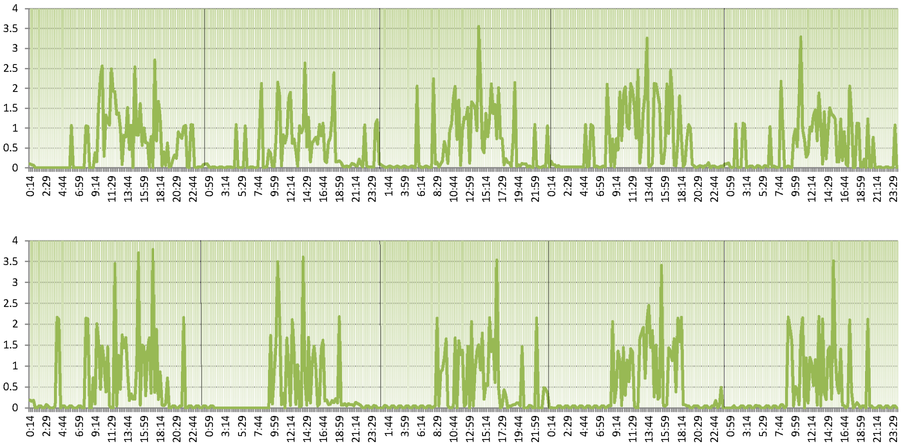

Figure 11: Energy (per kWh) comparison between non-scheduled (upper chart) and scheduled (lower chart) appliances for each work day.

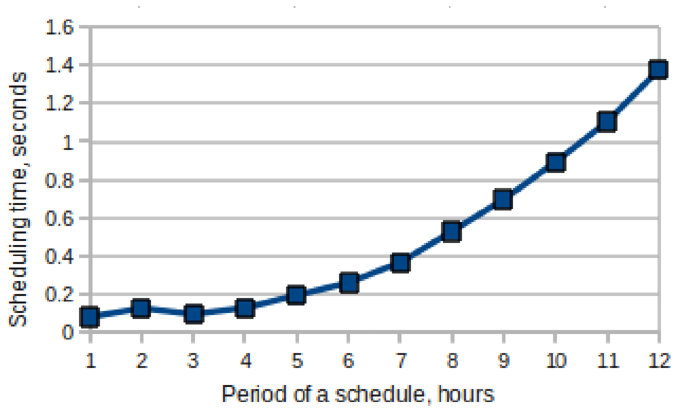

Figure 12: Scheduler performance dependence on the schedule time period.

of the Scheduler running time is shown in the Figure 12. 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 only had 6 devices, to test the Scheduler with a bigger number of them, we simulated devices by creating multiple copies of the available devices. We determined that a search for the optimal schedule can take impractically large amount of time for large buildings centrally controlled. 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 cluster them into groups. 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 provided schedule 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, which will prevent the Scheduler from placing more devices from the second group to 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 the Figure 13 we show the averaged running time of the Scheduler for different number of devices.

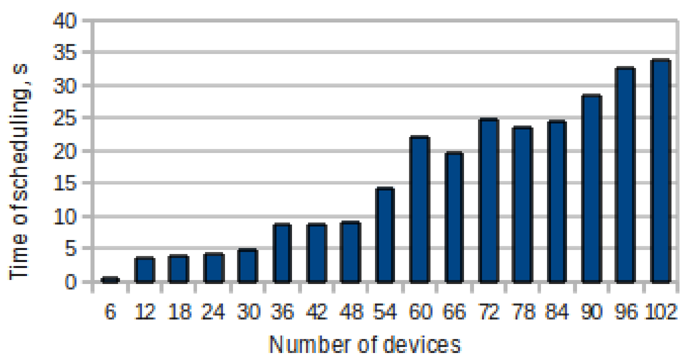

Figure 13: Scheduler performance dependence on the number of devices.

6 Related Work

The Smart Grid is a broad term used in several communities and there exists many investigations on the interactions between the Smart Grid and Smart Buildings. Here we present the main works concerning residential and commercial buildings environments where building and interoperation with Smart Grid appear. A general distinction in the literature is between the works focusing on the house environment, the majority [13, 14, 15, 16, 17, 18, 1, 19], and the office one which has received small attention so far [20, 21, 22]. As a general remark most of the works tend to be quite generic in presenting high level architectures [23, 14, 15] and usually lack in the experimental part where simulation of devices' energy consumption is considered. On the other hand, other approaches are related to the very low level concepts of protocol integration [13] or sensor network communication used for information exchange of energy consumed by devices [24]. 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 which negotiate on a virtual energy market [18], sometimes using strategies coming from game-theory concepts [1, 25]. Usually actions on real devices are only simulated [16, 1, 18] and only few approaches report on actual control and actuation of appliances [17]. In addition, in the panorama of the 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.* [21], as it is also our case.

A very significant test bed referring to office environment is realized by Han *et al.* [22] where a commercial building of seven floors is equipped with temperature, lighting, quality of air and occupancy sensors to retrieve information and provide them to the Building Energy Management System whose aim is to extract the context of the building. The system is based on an ontology for the building description which defines the optimal situation of the building, sensor are used to acquire the context and build inference rules to be 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 with a highly advanced approach to understand the building context, the paper misses a quantitative evaluation of the benefits achieved through the system. In addition, unlike our approach, there is no mentioning 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 fewer explored scenario of the office environment. Features of Smart Grid such as Demand-Response functionalities in a real environment, with prices that come from real energy market conditions, real renewable sources energy production and appliances actuation are new, to the best of our knowledge. In addition the device consumption and the energy savings achieved are not estimated or 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 not only the monitoring of energy usage [20, 19], but also the control and actuation on real equipment which is missing in the literature analyzed.

7 Conclusion

The Smart Grid promises not only to bring important advantages 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 not foreseeable nor desirable to do such control by hand or by enforcing policies on office users, but 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 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 10%.

Incidentally, we remark that the system itself consumes energy to operate which consists of 10 Plugwise devices and one desktop computer who respectively consume a maximum power of 1.1 W and 365 W. The value of the plugs is insignificant with respect to the overall consumption. As for the PC a few remarks are in order: 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 be scheduling 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 in our evaluation of optimization algorithms by including more devices in terms of variety and number and on the long term expand the testing to an entire building.

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*, pp. 1 –6, jan. 2010.

[2] International Eenrgy Agency, "World energy outlook," tech. rep., International Eenrgy Agency, 2009.

[3] Accenture end-consumer observatory on electricity management, "Understanding consumer preferences in energy efficiency," tech. rep., Accenture, 2010.

[4] 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* (H. Dietz, ed.), vol. 2624 of *Lecture Notes in Computer Science*, pp. 115–131, Springer Berlin / Heidelberg, 2003.

[5] 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*, pp. 143 –148, oct. 2010.

[6] W. Taqqali and N. Abdulaziz, "Smart grid and demand response technology," in *Energy Conference and Exhibition (EnergyCon), 2010 IEEE International*, pp. 710 –715, dec. 2010.

[7] 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*, p. 173, jan. 2008.

[8] L. Freris and D. Infield, *Renewable Energy in Power Systems.* Wiley, 2008.

[9] R. Wiser, G. Barbose, and C. Peterman, “Tracking the sun: the installed cost of photovoltaics in the u.s. from 1998-2007,” tech. rep., Lawrence Berkeley National Laboratory, 2009.

[10] R. T. Fielding and R. N. Taylor, “Principled design of the modern web architecture,” *ACM Trans. Internet Technol.*, vol. 2, pp. 115–150, 2002.

[11] C. Bauer and G. King, *Java Persistence with Hibernate.* Manning Publications Co., 2006.

[12] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 2nd Ed.* Englewood Cliffs, NJ: Prentice Hall, 2002.

[13] 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, pp. 8 –16, november-december 2009.

[14] 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.

[15] 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*, pp. 621 –624, june 2011.

[16] 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*, pp. 171–177, IEEE, 2009.

[17] K. Kok, S. Karnouskos, J. Ringelstein, A. Dimeas, A. Weidlich, C. Warmer, S. Drenkard, N. Hatziargyriou, 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.

[18] K. Kok, G. Venekamp, and P. Macdougall, “Market-based control in decentralized electrical power systems,” in *2First 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.

[19] A. Almeida and P. Fonseca, “Residential Monitoring to Decrease Energy Use and Carbon Emissions in Europe,” in *Int'l. Energy Efficiency Domestic Appliances and Lighting Conf.*, 2006.

[20] H. Vogt and H. Weiss, “A client architecture for market-based grid integration of smart environments,” in *PerCom Workshops*, pp. 642–647, 2010.

[21] 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*, pp. 1 –6, jan. 2011.

[22] J. Han, Y. Jeong, and I. Lee, “Efficient Building Energy Management System Based on Ontology, Inferen ce Rules, and Simulation,” in *International Conference on Intelligent Building and Management*, vol. 5, pp. 295–299, 2011.

[23] 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.

[24] C. Gezer and C. Buratti, “A zigbee smart energy implementation for energy efficient buildings,” in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp. 1 –5, may 2011.

[25] N. Capodieci, G. A. Pagani, G. Cabri, and M. Aiello, “Smart meter aware domestic energy trading agents,” in *First International E-Energy Market Challenge Workshop*, ICAC'11, 2011.