

## SIMULATING THE EFFECT ON THE ENERGY EFFICIENCY OF SMART GRID TECHNOLOGIES

Albert Molderink  
Maurice G.C. Bosman  
Vincent Bakker  
Johann L. Hurink  
Gerard J.M. Smit

University of Twente, Department of EEMCS  
P.O. Box 217  
Enschede, 7500AE, The Netherlands

### ABSTRACT

The awareness of the greenhouse gas effect and rising energy prices lead to initiatives to improve energy efficiency. These initiatives range from micro-generation, energy storage and efficient appliances to controllers with optimization objectives. Although these technologies are promising, their introduction may rise further questions. The implementation of such initiatives may have a severe impact on the electricity infrastructure. If several of these initiatives are introduced in a combined way, it is difficult to analyse their overall impact.

In this paper a model is defined and a developed simulator is described to analyse the impact of different combinations of micro-generators, energy buffers, appliances and control algorithms on the energy efficiency, both within the house and on larger scale. The simulator is easily adaptable to new types of micro-generators, controllers and other supported devices. Simulation of two case studies with the simulator shows that the achieved results are promising.

### 1 INTRODUCTION

In this paper a simulator is described that simulates the heat and electricity streams within and between houses. More and more energy efficiency increasing devices and control methodologies are developed. Especially a combination of these devices and a large penetration rate can have a significant impact on the electricity grid. Therefore, a model of domestic energy streams is deduced and based on this model a simulator is built.

Today, most residential used electricity is generated in central power plants. However, the efficiency of central generation is at most 55% due to inefficient generation (transport losses not taken into account). This low efficiency is mainly caused by wasting heat produced as byproduct and high fluctuations in demand (de Jong et al. 2006, Molderink et al. 2008). The growing awareness of the greenhouse gas effect and increasing energy prices require efficiency improvements of electricity production, distribution and consumption.

In the last years, a lot of technologies have been developed to improve the electricity usage and supply. One of the most eye catching technologies is generation based on renewable sources like large windturbine- and photovoltaic (PV) parks. Also on domestic level a lot of technologies are developed. These technologies range from PV on roofs and micro Combined Heat and Power (microCHP) (United States Department of Energy 2003) up to controllable appliances (Hammerstrom et al. 2007). Roughly, these technologies can be subdivided into 1) Distributed Generation (DG), 2) Distributed Electricity Storage and 3) Demand Side Load Management.

*Distributed Generation* - A growing share of the total generated electricity is produced in smaller, geographically distributed generators. These generators often have a higher efficiency or are based on renewable sources (de Jong et al. 2006, United States Department of Energy 2003). Two types of DG can be distinguished, (1) small generation sites on a megawatt level (e.g. Combined Heat and Power, Wind turbine parks) and (2) domestic micro-generators on a kilowatt level (de Jong et al. 2006).

*Distributed Electricity Storage* - Energy storage is considered having a high potential in addition to the energy infrastructure. Some houses already have a hot water store to optimize the runtime of an inhome heater. Distributed electricity storage capacity in the houses is seen as a solution to reduce peaks in electricity demand and cope with less manageable renewable sources (e.g. windmills or PV installations).

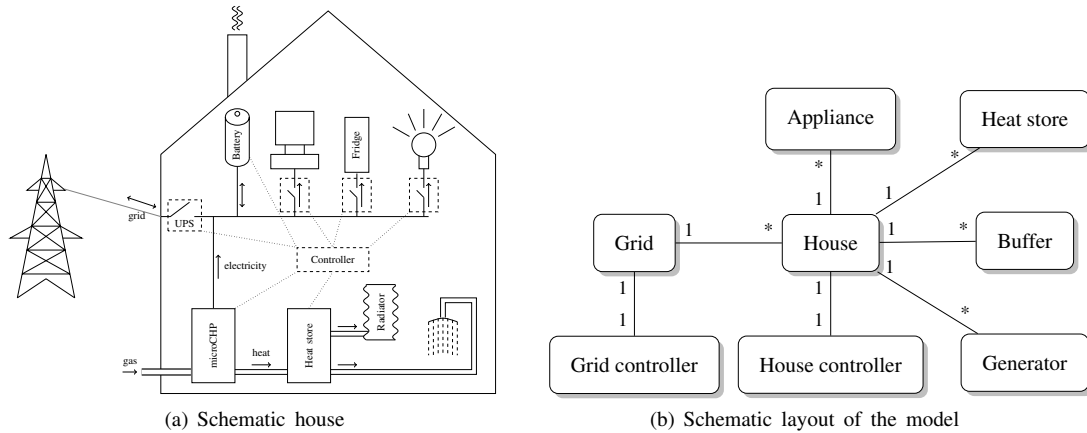


Figure 1: Proposed house

*Demand Side Load Management* - Demand side load management can increase the generation efficiency by peak shaving and by shifting load to more beneficial periods (Peacock and Newborough 2007). Parts of an appliance can be temporarily switched off (e.g. the heating element of a dryer) or the start of an appliance can be postponed. In extreme cases the appliance can even be switched off temporarily during its runtime (preemption). About 50% of the load in houses is dedicated to appliances that can be switched off or postponed without much discomfort; examples are refrigerators and washing machines (Block et al. 2008). Moreover, energy efficiency improvement can also be reached by using more efficient appliances, for example by using led-lights instead of conventional bulbs.

Distributed generation, distributed electricity storage and demand side load management already increase the energy efficiency. However, cooperation between these technologies and the existing infrastructure is often neglected. With an overall control algorithm optimizing the behavior of the domestic generation, storage and load management technologies even more efficiency can be gained. Such a control algorithm can have a local and a global scope. Within the local scope a single house is managed. Within a global scope multiple houses in a neighborhood are managed. Objectives on a local and global scope can be equal, for example peak shaving.

The introduction of (a large fleet of) micro-generators, energy buffers, appliances consuming both heat and electricity and control algorithms can have a significant impact on the efficiency and reliability of the current generation and transportation infrastructure. To study the effects of the introduction of these elements (devices and controllers) a simulator is developed, which can simulate a single house to analyse the local improvements as well as multiple houses to calculate global improvements. Furthermore, the effect on the grid (e.g. electricity import/export per neighborhood/city or peak reductions) resulting from the introduction of (combinations of) micro-generators, buffers, efficient appliances and control algorithms can be studied.

In the current version of the simulator it is assumed that electricity can be exchanged between houses via the grid, but that heat stays within the house. The next four sections describe the requirements, related work, the used model for the simulator and the simulator itself. In Section 6 possible control algorithms are discussed and two case studies and their results are presented. Finally, the last section presents the conclusions.

## 2 REQUIREMENTS

The goal of the simulator is to create a tool to analyze different combinations of micro-generators, energy buffers, appliances and both local and global control algorithms. This tool should be easily adaptable to new types of micro-generators, controllers and other supported elements. Furthermore, it should be easy to simulate different scenarios and combinations of elements and houses.

The simulated situation should be an accurate model of the actual situation. The modelled house should be able to represent different types of normal houses (family/single-person houses, big/small houses, etc.). Multiple houses grouped together should form a grid, e.g. a city with a realistic mix of houses.

Figure 1(a) gives an overview of the infrastructure of a single house. Within the house micro-generator(s), energy buffers and appliances are installed. In addition to the current situation all devices within the house (micro-generators, buffers

and appliances) are managed (whenever possible) by a local controller. The buffers and micro-generators are observed and managed directly, just like the current situation regarding central heating and the thermostat. The local controller should be able to query information from the local buffers about the fill rate and from the micro-generator about the current generation quantity. Furthermore, it should be able to send start/stop requests to the micro-generator. However, not every micro-generator may be able to follow this request since not every device is controllable (e.g. micro-wind) nor can every micro-generator immediately respond to such a request.

Today, many appliances do not (yet) have a communication interface. Therefore, we assume the appliances are controlled by an overruling switch connected to the controller that can switch off the supply to the appliance. However, the design of the simulator should be such, that it is possible to add more intelligent appliances with a communication interface and management possibilities. All appliances in the house should be separately manageable.

For the simulation of a massive introduction of micro-generation, multiple houses are to be combined into a grid. Each house should be individually addressed because every house has its own characteristics and internal state. The grid can optionally be extended with a global controller communicating with the local controllers, like the situation concerning a Virtual Power Plant (VPP) described in (Scott et al. 2008). Due to the different local controllers in the houses, it may be possible that not all houses respond to the global controller because not every type of local controller reacts on the signals in the same way (which is a realistic scenario).

The grid has to keep track of the total import/export of electricity within the grid. This data can be used to determine the success of the controllers. Furthermore, the grid should be capable of limiting the maximum amount of electricity that can be imported from or exported to the grid.

Since the purpose of the simulator is to simulate various new scenarios and elements, the simulator has to be flexible. Adding new types of micro-generators, buffers and appliances has to be easy, just as changing control algorithms. Some elements act upon both heat and electricity, for example the earlier mentioned hot-fill washing machine and microCHP devices. Therefore, for all devices the behaviour concerning both heat and electricity has to be defined. The combination of heat and electricity is referred to by "the energy profile" in the rest of this paper.

The simulator should be capable of simulating both the results of local control algorithms within a single house in detail, and the overall effect of a global control algorithm on a large number of houses. For the latter, not all details of individual houses are necessary. However, within a simulation of a single house these details are essential. So, the level of detail in stored data needs to be adjustable.

The last requirements concern the speed and memory usage of the simulator. It has to simulate a large fleet of houses in detail for the VPP simulations. An average windmill park produces around 50 MW. In order to have an applicable VPP that is comparable to such a windmill park, a generation potential of 50 MW is necessary. Therefore, the number of houses that can be simulated should be at least 50.000 within a reasonable time (hours) on a normal PC, leading to requirements on CPU and memory usage. Since it is a discrete simulation (due to modeling choices), the length of the time intervals is the choice which mainly influences the amount of data. It is a tradeoff between precision and data usage. The minimum possible time interval length may be a second, but this has to be adjustable. A five minute time interval is a good tradeoff between precision and data usage (Wright and Firth 2007). The precision of the data itself does not form a major issue, since almost all data can be stored as integers representing Watts.

The requirements described above are summarized in six requirements for the simulator:

1. Simulation of **realistic** settings and devices
2. Combined **heat and electricity**
3. Both **single house** and a grid with a **large amount of houses**
4. **Flexible**, both in adding new elements and in the supported scenarios
5. Minimal **50.000 houses**
6. **Adjustable logging/precision in time interval length**

### **3 RELATED WORK**

In this chapter, first we describe a few existing simulation platforms from literature and their general properties. Based on this, we then argue why we have developed a simulator ourselves.

The simulator needs to be capable of handling discrete time intervals and has to be flexible. As far as we know, no other generic simulator has been developed yet that focuses on simulating energy usage on domestic level, including new technologies and algorithms. However, both free and commercial flexible simulation platforms exist that handle discrete time intervals. Below, three of them are listed.

**Tortuga** (Weatherly and Page 2004) is a framework for discrete event simulation. It takes care of the event handling of up to several thousands of predefined simulation entities. The simulation has to be written in Java, since Tortuga uses the Java-based tool Ant.

**SimPy** (Simulation in Python - <http://simpy.sourceforge.net/>) is an object oriented, discrete event simulation language that is based on Python. It uses *processes*, entities which evolve in time, and *resources*, used by the processes. Variables can be used to store data during simulation. Furthermore, SimPy has its own GUI (Graphical User Interface) and plotting packages.

**AnyLogic**(<http://www.xjtek.com/>) is commercial simulation software, based on Java, which uses an interactive GUI as the starting point of building a model. A model comprises of predefined entities, whose functionality can be adapted. At the moment several millions of entities can be incorporated in one simulation. AnyLogic has its own data analysis and plotting packages.

An existing simulator framework should offer an environment in which the six requirements, explained in Section 2, could be met. For the choice of a simulation environment, this comes down to high speed, low memory usage and flexibility in modeling.

Speed and memory usage of the intended simulator are of importance, since large sized fleets are simulated. The simulator does not only have to process discrete events, but it uses time consuming local and/or global algorithms/solvers too. The access to data, which is required by these algorithms/solvers, needs to be fast. Since especially this part of the model is hard to incorporate in an existing simulation framework, we investigate the advantages of the development of an own simulator. The proposed, self-made simulator incorporates the algorithms and solvers, rather than that it communicates with them via other programs. This incorporation increases simulation speed compared to existing simulation platforms with slower data exchange.

A further advantage of a tailored self made simulator is that its overhead is reduced to necessary features imposed by the stated requirements. Although the simulator is composed of generic parts, it is specifically tailored for the energy infrastructure. The disk usage is efficient, since no unnecessary elements are created or stored.

Finally, the simulator needs flexibility in adding, for example, new algorithms or other types of generators. The generic modelling approach, presented in the next section, allows adaptations in the model without directly asking for a large change in the simulation environment. This prevents that such adaptations suddenly lead to a huge increase in memory usage or a large decrease in speed. In existing simulation platforms, it is difficult, for example, to add algorithms efficiently or to create a grid of very large size.

For these reasons, we choose to develop our own simulator, instead of using an existing framework.

## 4 MODEL

The model of a house is deduced from the situation shown in Figure 1(a) and the requirements described in the previous section. The setup of this model is schematically shown in Figure 1(b). Every house consists of (several) micro-generators, heat and electricity buffers, appliances and a local controller. Multiple of these houses are combined into a grid, exchanging electricity and information. The electricity generation of power plants (coal, gas, windturbines) is left outside of the model in first instance; only the total electricity consumption of all houses is registered in the grid. The generation of all large power plants in the grid is assumed to be equal to the total accumulated import of the house (where export is negative import). This import can be steered by the global controller, for example to match the uncontrollable generation of wind turbines. In the current situation the power plants continuously have to adapt to the demand.

Micro-generators can produce heat and electricity. All available micro-generators are modelled in this way, considering that the generation can be zero or even negative. A microCHP ([www.whispergen.com](http://www.whispergen.com)) produces electricity and heat, a Photo Voltaic (PV) produces electricity and no heat where a conventional electrical heater generates heat with negative electricity production. Next to the production of heat and electricity, the import of "fuel" for the micro-generators is considered. This can be natural gas (e.g. for a microCHP device) but also sun or wind. The model keeps track of the amount of natural gas consumed.

All appliances in the house are modelled as combined electricity and heat consumers. All consumers are defined as appliances, from a fridge and a coffeemaker to central heating and tap water. For appliances it also holds that consumption (of electricity and/or heat) can be zero. Next to their runtimes, appliances have two more parameters which (1) indicate whether or not the appliance is preemptable and (2) give the priority of the appliance. These parameters can be used by the control algorithms to decide which appliances to supply and which to switch off. In the simulator, we assume that the demand of an appliance can be queried. This is realistic since switching off appliances requires smart appliances or smart outlets.

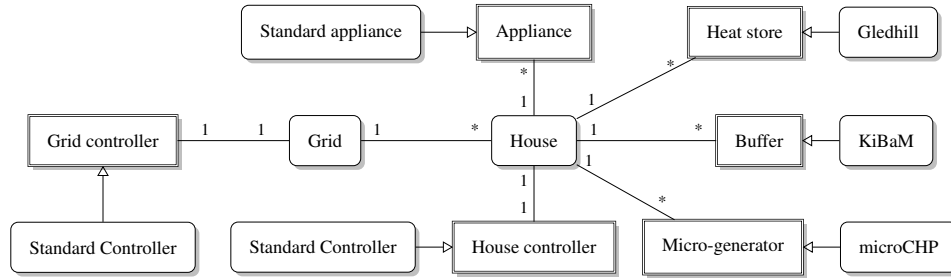


Figure 2: Simulator classes

Buffers and heatstores are temporary electricity and heat storages. When there is more energy production than consumption, this surplus flows into (one of) the buffers. When there is less energy production than consumption, this shortage flows out of the buffers. The surplus and shortage are separately calculated for heat and electricity.

Within the model the planning horizon is discretized resulting in a set of consecutive time intervals. The number of intervals depends on the length of the planning horizon and the length of the intervals. In general we use an interval length of six minutes and a planning horizon of one day, resulting in 240 time intervals.

## 5 IMPLEMENTATION DETAILS

The simulator is built in an object oriented manner in C++. A simulation consists of a grid. A grid consists of a controller and houses, etc. In this way we can follow the structure given in Figure 1(b). The controller, micro-generator, appliance and buffer classes are abstract classes with the minimum functionality implemented and the required functionality defined in an abstract way. An implementation of an actual element consists of a class that extends the abstract class and implements the abstract defined functionality. Optionally, standard behaviour can be adapted by overriding the corresponding functions. In this way the class implements the specific behaviour of the element. Because of this construction, a device (micro-generator, appliance or buffer) can be added by implementing only the necessary functionality and without changes to the house or the controller. The house and grid class are not abstract classes but normal classes. In Figure 2 the current structure of the implementation is given. Upto now, only a Whispergen microCHP is implemented as micro-generator, a Gledhill as heatstore ([www.gledhill.net](http://www.gledhill.net)), the KiBaM battery model as electricity buffer ([Stevens and Corey 1996](#)) and a standard appliance consuming heat and electricity. With this standard appliance class a group of often used appliances in a house is modelled. The start- and runtimes of an appliance in a certain house are defined in that house. The following subsection describes the details of the structure of the implementation and the configuration. Subsection 5.2 gives the details of the initialisation and simulation itself.

### 5.1 Implementation

As mentioned above, there are abstract classes for the controllers, micro-generator, buffers and the consumers. The implementation of a actual element defines the behaviour of the corresponding element. It can have an internal state to be able to implement realistic behavior, for example to implement the start- and stop behavior of a micro-generator and losses in the buffers.

Every time interval every element receives a signal so it can update its internal state concerning the previous state and the input signals. These input signals are on/off switch requests, the energy flowing into or from the buffers, etc. For all elements within the house the control input signals are generated by the local controller. However, the element might not always respond (immediately) to a request: e.g. a microCHP device only switches on when the minimum cooldown period has elapsed.

Within an implementation of an element, parameters can be defined. For the current microCHP implementation for example, the electricity generation level and the ratio between electricity and heat are parameters. In this way multiple versions of an element can be defined with one single implementation. This can be used to optimize the parameters for an element or for a quick exploration of different possibilities for some features. A concrete version of an implementation, with

values defined for the parameters, is called a configuration. The parameters of a configuration can be defined at run-time and multiple configurations of the same implementation can be used during one simulation.

The energy buffers, both for heat and electricity, can absorb and supply energy. The controller calculates the energy flow from and into the buffers every time interval. However, there is a maximum flow and a maximum level. When more energy flows to a buffer than the buffer can absorb in one time interval or when the buffer is full, the energy is lost. When more energy is demanded from a buffer than it can deliver, there is an energy shortage. These problems should be avoided by correctly functioning controllers.

The current standard appliance implementation defines a profile and information whether the appliance is preemptable during runtime. A profile is the shortest repetitive pattern in the energy usage, e.g. the hourly on-off behavior of a fridge. The runtime itself and the priority are defined in the parameters of the corresponding house object. So, a coffeemaker configuration can be used in multiple houses with different runtimes. However, for every house an object of the coffeemaker appliance is created because an appliance keeps track of its own runtime. Based on the elapsed runtime and the profile the demand for the next time interval can be determined. Figure 4(a) shows an example of the total electricity demand of a house for a complete day, based on the electricity profiles of around 25 appliances. The local controller in a house can query every appliance how much energy it requires the next time interval and then decides whether the appliance is supplied.

The local controller decides in one house which micro-generators and appliances are to be switched on, based on the current state of all elements in the house and optionally the signals of the global controller. Next, the energy streams between these elements are calculated by the controller. A local controller can steer all devices in the house and thus exploit the potential of the different devices. It can optimize the runtime of the generators, use the buffers optimally and shift load from appliances (within the boundaries of comfort). The basic version of the controller only regulates the energy streams between the elements and starts a micro-generator when necessary (only heat-led), without optimizations in runtimes of appliances or battery usage. A more sophisticated controller is described in subsection 6.1.

A house configuration combines the above described elements to one house. It defines which micro-generator configurations are installed, just like which buffer configurations, appliance configurations and local controller configuration are available. For all appliances the runtimes and the priorities are also defined.

Finally, the grid configuration combines house configurations and the global controller configuration. For every house configuration the multiplicity of particular configurations can be defined within the grid. The global controller can query the status of the houses from their local controllers and give requests for the next state. However, the local controllers decide whether they follow the requests.

Every element logs the status of its variables every time interval. Which values are logged depends on the implementation of the element. After the simulation all data is stored in a result file. Logging can be disabled per element in order to decrease the memory usage. When the logging of an element is disabled, the logging of lower level elements, which are part of this element, is also disabled.

The length of the time intervals used in the simulation can differ, with a minimum stepsize of one second. Since every configuration of an element is based on a time interval length this has to be synchronized. For example, the energy profile of an appliance is based on the length of the time intervals and has to be recalculated for different time interval lengths. In the simulation configuration the simulated time interval length is specified, during initialization the time intervals of all elements are synchronized to this length.

Since a limited set of configurations for individual houses is used to form a grid, stochastic variations are useful in order to get a more realistic reflection of real world behavior. To introduce stochastic variations, random variates are generated and applied to the data in the configurations. Several distributions are available (Uniform, Exponential, Weibull, Normal and Poisson) which gives enough space to create realistic variations. For each appliance the start- and runtime can be varied. Next, a single stochastic variation per house is applied to all values of the energy profiles resulting in variations in the overall energy consumption of houses (high/low overall energy usage). On top of this variation, a second variation can be applied on the energy profile with a different random variate for each appliance and each time period.

## **5.2 Simulation**

The configurations are saved in configuration files. At the start, a simulation is initiated based on these files. The grid configuration defines which house configurations are present. The corresponding house configuration files define which micro-generator configurations are installed, etc. Within a large scale simulation, in general only a few different house configurations are used, but with the stochastic variations of appliances we still get unique houses. For every house, multiple configuration files have to be accessed (approximately 25). Since file access is very slow, the objects of equal house

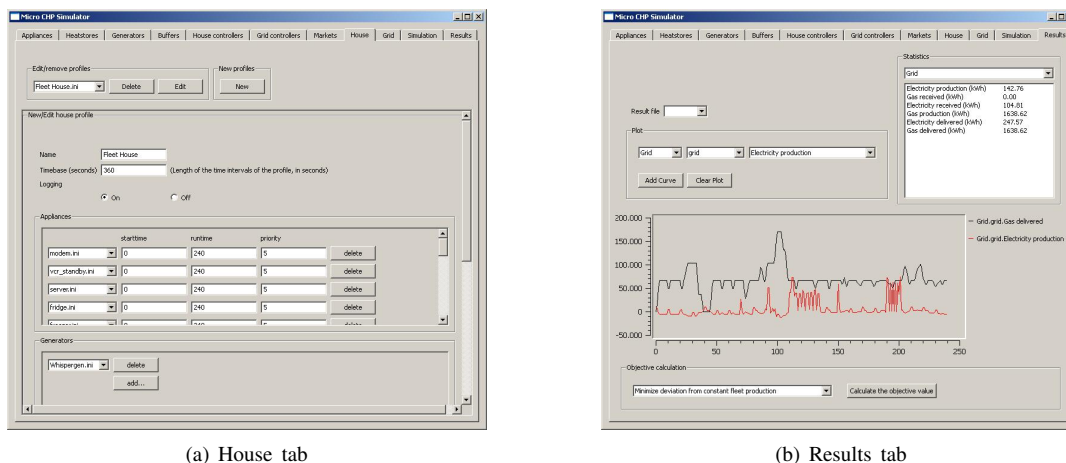


Figure 3: GUI of the simulator

configurations are copied, resulting in only one initiation with configuration files per configuration. The variation is added after copying.

After the initialisation, the simulation starts. All objects receive a signal (called a tick) to update their internal state every time interval. But, the order in which the elements update their internal state is very important. First, if present, the global controller queries all houses about their status (import/export of electricity, via the local controller) and calculates an optimal next status for every house based on his objectives. The preferred status for every house is passed to the local controllers as a request. In real situations, an information exchange may occur only a limited number of times during a day, in which the information is combined for several time intervals.

Next, the status of each house is calculated. Based on the status of all elements within the house, optionally on the received request from the global controller and on its own objectives, the local controller decides which appliances are supplied, which micro-generators are switched on and off, etc and passes this to every device. Then, each device of the house calculates its own next state, based on the decisions of the controller. For example, if the controller wants to switch off a micro-generator, if the micro-generator is controllable and if it is allowed concerning its current state, the next state of the micro-generator will be the beginning of the stopping state. When all houses are evaluated the next state of the grid can be calculated based on the status of each individual house.

### 5.3 GUI

On top of the simulator a GUI is built. This GUI is a graphical user interface to the functionality of the simulator. Configurations can be added, edited or removed. Every part of the simulator as shown in Figure 1(b) has its own tab in the GUI. For abstract parts, an actual implementation can be selected and a configuration for that implementation can be defined. Configurations of non-abstract parts can be defined by just setting the parameters. Within the simulation tab the parameters of the simulation can be set (time interval length, number of intervals) and the simulation can be started. The result tab visualizes the results of the simulation and the results can be stored here. Two screenshots of the simulator can be seen in Figure 3. The first screenshot shows the configuration screen for a house where all available appliances, generators, etc. can be configured. The second screenshot shows the results interface where the results can easily be studied and compared.

## 6 RESULTS AND DISCUSSION

In Section 2 the requirements of the simulator are stated. This section discusses whether these requirements are met. The first four requirements (R1-R4) are discussed in the following paragraphs based on the setup of the simulator. Then, two case studies are presented to support this discussion and to verify the last two requirements (R5, R6).

The developed simulator is based on a model deducted from an actual house and connections between houses. Parts of the model are verified using realistic data (both measured and from literature); the heatstore, microCHP and appliances

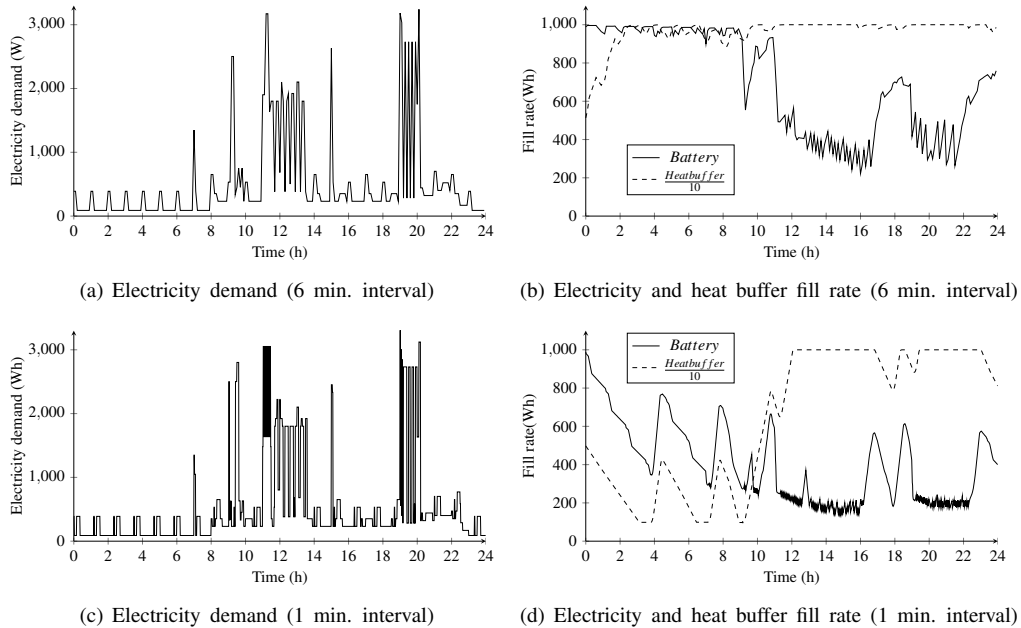


Figure 4: Results single house islanded scenario simulation

are verified with data from a prototype and measured in houses. However, the algorithms cannot be verified in real houses before they are exhaustively tested (due to residents).

Realistic data is used for the energy profiles and runtimes of appliances so the energy usage of houses is realistic (R1). The same holds for a large fleet of houses (R3). Variations on the profile increase the usefulness of the simulator since houses are often divided into a small number of standard houses. For example, electricity distributors use only five standard profiles for their electricity usage prediction. Furthermore, when good models are used for the generators and buffers, simulations of realistic scenarios are possible.

The model on which the simulator is based incorporates both heat and electricity (R2). Therefore, the simulator combines heat and electricity consumption in appliances and generators and concerns both heat and electricity within optimizations.

The underlying model is a generic description of appliances, generators, etc. Due to the implementation of this model with abstract classes, it is easy to add new types of appliances, buffers and generators (R4). Furthermore, multiple objectives and scenarios can be simulated with different controllers and configurations (without changing source code). Thus, the simulator has enough flexibility.

In order to verify whether the simulator meets the other requirements, two case studies are simulated, one of a single house and one of a large fleet of houses. The single house simulation is used to verify the simulation in detail and to verify time interval length calculations. The fleet simulation is used to verify the operation of global control: how many houses can be simulated and what is the speed of these simulations.

## 6.1 Single house simulation

For the single house simulation an Islanded House scenario is chosen. Islanded House means that the house is disconnected from the grid, so no import/export of electricity to the grid is possible. The simulation uses a normal electricity and heat demand profile and it tries to meet all demand instead of only supplying important appliances during a power cut. Although this might not be a realistic scenario, it is very useful to test the simulator. In this scenario, the generator is used for electricity generation, the battery for storage and peak supply and appliances can be switched off during runtime and shifted in time. An optimization algorithm manages the generation, storage, etc. so all possible optimization techniques are used. Thus, this is a good test for the simulator. Within the scenario it is now allowed to dump heat (it is assumed this is possible). This Islanded scenario and the setup of the house is described in (Molderink et al. 2008). The results in (Molderink et al. 2008) are produced with Matlab scripts and are used to verify the results of the simulator described in this paper.



Two different controllers are implemented, one tailored to Islanded scenario and one generic controller. The first controller is described in (Molderink et al. 2008). The generic controller is applicable in many scenarios, both for in house optimizations and for fleet optimizations (see next subsection). This controller is a realtime scheduling algorithm, it is executed every time interval and only based on the current status of all elements in the house (no usage of predictions). Starting point of the algorithm is the demand during a time interval which has to be matched. The demand is defined as the sum of the heat and electricity demand of all consuming appliances. This demand is given as an input parameter and can be matched with 1) import from the grid, 2) production by generators, 3) the buffers and 4) switching off consumers (not providing them). Switching off a device is seen as matching its demand (the *demand* is an input parameter and thus should not change). Every matching has a certain cost associated with it. Delivering energy to the grid or a buffer is seen as negative matching and leads to negative costs. Using negative import (= export) and buffer charging gives enough freedom for the four possibilities to choose the values such that all demand is matched. The best solution is the matching with the lowest costs. Steering signals from a global controller are encapsulated in the electricity import/export costs. A detailed description of this controller can be found in (Molderink et al. 2009).

The above described scenario is simulated two times, first with a six minute time interval and secondly on a one minute time interval. Since the demand profiles are defined in a six minute timebase, the profiles have to be recalculated for the one minute time interval simulation. Therefore, we can verify whether the recalculations work correctly. Furthermore, the difference in the decision of the local controllers between one and six minute interval can be analysed. The costs for electricity import/export are set to infinite during Islanded operation.

Figure 4 shows four plots of the results of the simulations. The results of the simulation are equal to the results of the Matlab implementation. Results of the simulation are very detailed; the logging of every element in every time interval gives a lot of insight in the status of the elements. Figure 4(a) and 4(c) show the electricity demand. There are some small differences because of rounding errors between the six and one minute simulations, but the recalculation of the profiles was correct (R6). Figure 4(b) and 4(d) show details of the buffers. The differences in the buffer levels are much larger than the differences between the recalculated demands. This is caused by other decisions of the local controller, when to switch on the microCHP. The decisions of the local controller are based on costs of electricity and heat supply. Since in a one minute time interval the demand/supply is six times lower than in a six minute time interval, the costs (and therefore decisions) are different for different time intervals. The cost functions are not adjusted for other time intervals, the only difference between the simulations is the time interval length. However, the overall results (amount of supply, number of appliances switched off, etc.) of both simulations are equal.

## 6.2 (Large) Fleet simulation

A simulation framework for the problem considered in this paper has to offer an environment in which next to detailed simulation of single house settings, the simulation of a large fleet of houses is an important goal. A large fleet consisting of tens of thousands of houses should be considered. This amount of houses corresponds to a relatively large city, with a production potential that can have a significant impact on the electricity grid.

When running a simulation of this size, simulation speed and memory usage are important. On top of this, the interaction between global and local controllers needs to work properly. In order to verify these properties, the second case study is split into two parts. In the first part, a global controller, which creates an optimal schedule for the fleet, is used to create the steering signals for the local controller mentioned in the previous sub section. The original optimal global schedule is then compared to the results of the simulation. This gives a measure of quality of the interaction between global and local control. Due to solving limitations of the optimal global scheduler, only fleets of small size can be compared. The second part deals with execution time and memory. For computational reasons, a heuristic is now used to create the fleet schedule offline. This schedule is again used as input for the local controllers. Furthermore, a modified version of the islanded local controller is used since the current version of the generic local controller uses an ILP (instead of heuristics). The ILP calculation uses too much memory and computational time for a large scale simulation. Fleets of varying sizes are compared for their simulation speed and memory usage.

A global controller does not switch micro-generators on or off. It only gathers data from the houses in its grid and makes a good schedule for the fleet, based on some fleet objectives. This schedule works as an advice for the individual houses. However, each local controller makes its own on/off decision, even if it is conflicting with the global advice. It is of course the intention to follow the global controller advice, whenever this is possible.

In (Bosman et al. 2009) a description of the global control problem, which leads to an optimal offline schedule, is given. Since this offline schedule is created on before hand, the schedule is based on heat usage predictions. Due to the large computational time needed to get this solution, the fleet size for the first part of the case study is limited to ten houses.

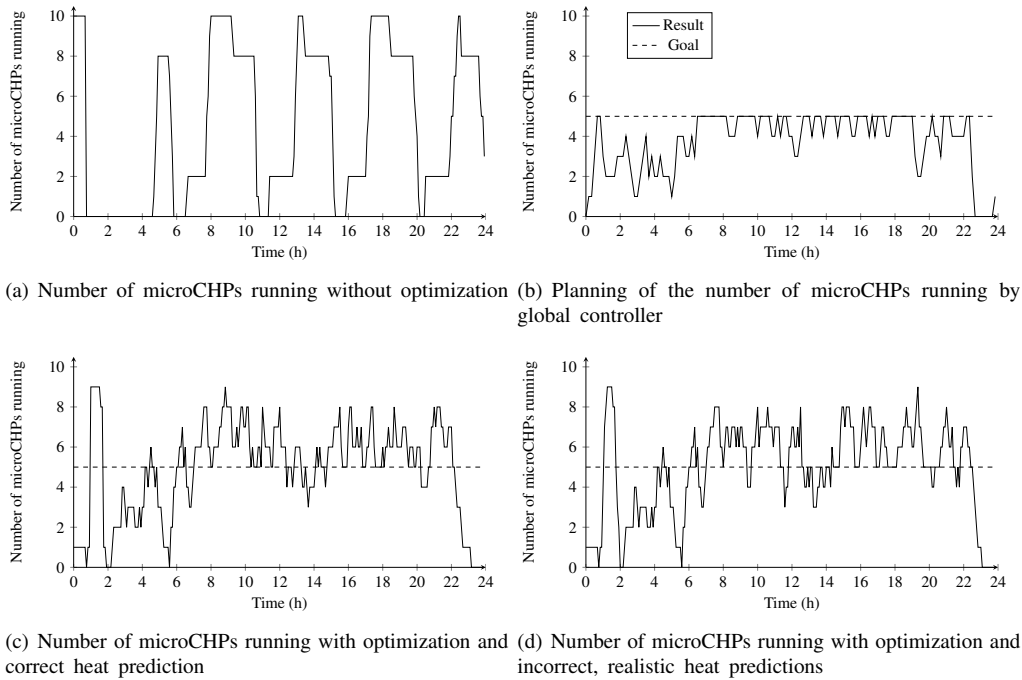


Figure 5: Results simulation interaction local and global controller

All houses have the same heat demand profile with a total heat demand of 53 kWh per day. In each house the generic controller is used; costs for import are increased when the global controller prefers a microCHP running. Since all houses are connected to the grid, switching off in home appliances is no longer allowed (by assigning infinite costs). Furthermore, no electricity buffer is used.

Two different scenarios are simulated. In the first case, the actual heat usage is equal to the predicted amounts. In this case, the actual schedule should be similar to the offline schedule. In the second case, a random variation is added to the heat profile to create realistic heat demands, not equal to the predictions and different for every house.

Figure 5 shows the results of both the optimal offline schedule and the results of the simulation in real time with and without steering signals.

The results show a clear difference between the results with and without steering signals. The simulated results are not equal to the planning of the global controller; the costs within the local controller have to be tweaked. However, the communication between the local and global controller works fine. Another interesting point is the small difference between the simulations with correct and incorrect heat usage predictions. The variations on the profiles are built such that the expected profile is equal to the original profile. However, individual profiles can have large deviations. A large group levels out these individual deviations. This is why energy distributors use a limited number of average profiles for their predictions. It seems that ten houses already level out deviations up to a certain level.

For large fleet simulations the only changes to the settings of the previous subsection are the use of a faster global controller using heuristics instead of having the problem formulation solved to optimality and another local controller. We simulated five different simulator configurations to get a good feeling about the behavior: 1) Limited logging - only logging of the grid [reference case], 2) No logging, 3) Exhaustive logging - logging of the grid and all houses, 4) Concurrency - the state transitions of the houses are concurrently executed and 5) Variations - variations on the profiles of the appliances are added. The first configuration is the reference case; the other configurations change only one parameter with respect to this case. For every configuration the memory usage and execution time are determined for both the initialisation and the simulation and for different numbers of houses. The simulations are executed on a Core2Duo T7200 with 2Gb RAM.

The results of the total memory usage and execution time (initialisation and simulation) of the reference case are given in Figure 6. It can be seen that 50.000 houses with logging of the grid can be simulated (R5) on a normal PC within a reasonable time. Both execution time (solid line) and memory usage (dashed line) scale linear.

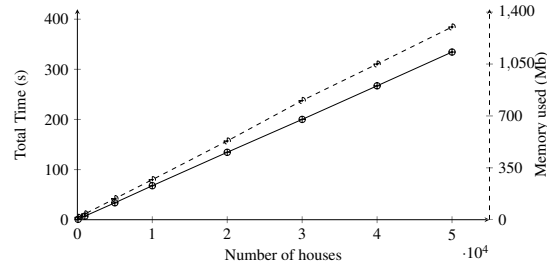


Figure 6: Execution time and memory usage reference case

No logging at all gives the same memory usage and initialisation time, but the simulation itself goes twice as fast. During the initialisation the memory for the first log values is pre-allocated (enough for only grid logging), therefore the initialisation time and memory usage do not decrease.

Using exhaustive logging, a maximum of 1000 houses can be simulated (797Mb, 17 seconds). When more houses are simulated the system has to swap memory.

Concurrency gives the same initialisation time and memory usage. However, the simulation time decreases with 23% resulting in an overall speedup of 15%. Variations on the profiles of appliances do not influence the simulation time or memory usage.

## 7 CONCLUSIONS

A new simulator, for simulating the effect on the energy efficiency of smart grid technologies, is proposed. The simulator can be used to analyse the effect of (combinations of) new technologies and algorithms on the energy efficiency and reliability of the grid. First simulations of a fleet of houses already showed an increased efficiency by lowering peaks in usage. In general, the following conclusions can be drawn about the simulator:

*Ease of use* - The simulator incorporates a generic model of the energy infrastructure. This gives enough flexibility in extending the model with new technologies of heat and/or electricity generation. The use of configurations leads to a clear construction of scenarios. This enables the user to create and run a simulation scenario easily, possibly based on available information from grid operators, energy distributors and households.

*Quality of the simulator* - All requirements derived in Section 2 are met. The simulator is able to simulate realistic scenarios, both on small and on large scale. However, the simulation time does depend on the global and local optimization algorithms. Regarding speed, a large simulation speed improvement comes from the possibility to switch off the logging of elements and from concurrency. However, especially the memory usage during the simulation has to decrease.

## 8 ACKNOWLEDGEMENTS

This research is conducted within the Islanded House project supported by E.ON UK and the SFEER project (07937) supported by STW, Essent and Gasterra.

## REFERENCES

- Block, C., D. Neumann, and C. Weinhardt. 2008, Jan. A market mechanism for energy allocation in micro-chp grids. In *41st Hawaii International Conference on System Sciences*, 172–180.
- Bosman, M., V. Bakker, A. Molderink, J. Hurink, and G. Smit. 2009. The microchp scheduling problem (*accepted*). In *Second Global Conference on Power Control and Optimization: IEEE*.
- de Jong, A., E.-J. Bakker, J. Dam, and H. van Wolferen. 2006, Juli. Technisch energie- en CO<sub>2</sub>-besparingspotentieel in Nederland (2010-2030). *Platform Nieuw Gas*:45.
- Hammerstrom, D., R. Ambrosio, T. Carlon, J. DeSteeze, G. Horst, and R. Kajfasz. 2007, July. Pacific northwest gridwise testbed demonstration projects, part i and ii. Pacific Northwest National Laboratory.

- Molderink, A., V. Bakker, M. Bosman, J. Hurink, and G. Smit. 2009. Domestic energy management methodology for optimizing efficiency in smart grids (*accepted*). In *IEEE conference on Power Technology*: IEEE.
- Molderink, A., V. Bakker, J. Hurink, and G. Smit. 2008. Algorithms for balancing demand-side load and micro-generation in islanded operation. In *19th International Conference on System Engineering*, 115–120: IEEE.
- Peacock, A., and M. Newborough. 2007, July. Controlling micro-chp systems to modulate electrical load profiles. *Energy* 32 (7): 1093–1103.
- Scott, J., P. Vaessen, and F. Verheij. 2008, Apr. Reflections on smart grids for the future. Dutch Ministry of Economic Affairs.
- Stevens, J., and G. Corey. 1996, May. A study of lead-acid battery efficiency near top-of-charge and the impact on pv system design. In *Conference Record of the Twenty Fifth IEEE Photovoltaic Specialists Conference*, 1485–1488: IEEE.
- United States Department of Energy 2003, December. The micro-CHP technologies roadmap. *Results of the Micro-CHP Technologies Roadmap Workshop*.
- Weatherly, R., and E. Page. 2004, Dec. Efficient process interaction simulation in java: implementing co-routines within a single java thread. In *Proceedings of the 2004 Winter Simulation Conference*, Volume 2, 1437–1443.
- Wright, A., and S. Firth. 2007, April. The nature of domestic electricity-loads and effects of time averaging on statistics and on-site generation calculations. *Applied Energy* 84 (4): 389–403.

## AUTHOR BIOGRAPHIES

**ALBERT MOLDERINK** was born in Heerenveen (The Netherlands) in 1983. He received his B.Sc. and M.Sc. degree in Computer Science from the University of Twente, Enschede, The Netherlands, in respectively 2004 and 2007. In addition, he received an Electrical Engineering minor certificate. When he completed his study he started working towards a Ph.D. degree at the University of Twente under supervision of Prof. dr. ir. G.J.M. Smit of the Computer Architecture of Embedded Systems (CAES) group, faculty of Electrical Engineering, Mathematics and Computer Science. He is working in a research group that investigates the possibilities of increasing energy efficiency using embedded control, mainly via optimization and control algorithms. His research focus is on algorithms to optimize energy streams within a house. His research interests include energy efficiency, mathematical modelling and optimizations, algorithm development and embedded hardware.

**MAURICE G.C. BOSMAN** received his M.Sc. degree in Applied Mathematics from the University of Twente in February 2008. Currently he is a PhD student in the CAES and DMMP groups at the faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente. His research interests include energy efficiency, scheduling and online algorithms.

**VINCENT BAKKER** received his M.Sc. degree in Computer Science from the University of Twente in 2007, with a minor certificate in Entrepreneurship. Currently he is working on his Ph.D. thesis researching domestic demand prediction for in home optimizations. Currently his interest are: machine learning, optimization modeling and large scale distributed (intelligent) systems.

**JOHANN L. HURINK** received a Ph.D. degree at the University of Osnabrueck (Germany) in 1992 for a thesis on a scheduling problem occurring in the area of public transport. From 1992 until 1998 he has been an assistant professor at the same university working on local search methods and complex scheduling problems. From 1998 until 2005 he has been an assistant professor and since 2005 an associated professor in the group Discrete Mathematics and Mathematical Programming at the department of Applied Mathematics at the University of Twente. Current work includes the application of optimization techniques and scheduling models to problems from logistics, health care, and telecommunication.

**GERARD J.M. SMIT** received his M.sc. degree in electrical engineering from the University of Twente. He then worked for four years in the research and development laboratory of Océ in Venlo. He finished his Ph.D. thesis entitled "the design of Central Switch communication systems for Multimedia Applications" in 1994. He has been a visiting researcher at the Computer Lab of the Cambridge University in 1994, and a visiting researcher at Lucent Technologies Bell Labs Innovations, New Jersey in 1998. Since 1999 he works in the Chameleon project, which investigates new hardware and software architectures for battery-powered hand-held computers. Currently his interests are: low-power communication, wireless multimedia communication, and reconfigurable architectures for energy reduction. Since 2006 he is full professor in the CAES chair (Computer Architectures for Embedded Systems) at the faculty EEMCS of the University of Twente. Prof. Smit has been and still is responsible of a number of research projects sponsored by the EC, industry and Dutch government in the field of multimedia and reconfigurable systems.