In: K. Greve, A. B. Cremers (Eds.): Integration of Environmental Information in Europe - 24th International Conference on Informatics for Environmental Protection. Shaker Verlag, ISBN 978-3-8322-9458-8, pp. 734-743

A pooling based load shift strategy for household appliances

Ontje Lünsdorf[1], Michael Sonnenschein[2]
[1]OFFIS Institute for Information Technology, Oldenburg,
[2]Carl von Ossietzky University of Oldenburg, Department of Computing Science
D-26111 Oldenburg
Ontje.Luensdorf@offis.de
Michael.Sonnenschein@uni-oldenburg.de

Abstract

An increasing utilization of renewable energy sources like wind or sunlight demands for balancing energy to overcome (potentially unforeseen) fluctuations in the power production. Instead of adapting the power supply, as it is currently done, it is also possible to control the power demand. As household appliances account for a large fraction of the overall load and are a major contributor to the daily load peaks, we investigate how a pool of household appliances can be used for load shifting. The outlined strategy has the potential to integrate different appliance types into demand side management measures. A pooling algorithm is used to group appliances into virtual devices which can fulfil given stochastic constraints. Due to permanent reorganization of virtual devices, the system is able to adapt itself to long-term changes. Demand side management measures like these may be used for example to compensate errors in wind power prognosis, or to smooth the startup-ramps of conventional power plants generating balance energy.

1. Introduction

Today's power generation output is adapted to match the current consumption. The declining availability of fossil energy resources and the environmental damage caused by their use give reason for an increasing utilization of renewable energy resources. However, their availability depends on environmental conditions whose predictions are burdened by uncertainty. Therefore balancing energy is required to overcome (potentially unforeseen) fluctuations in the power production. The reasons mentioned above argue against the use of conventional power plants for this purpose.
Instead of adapting the power supply, it is also possible to control the power demand. This strategy has two benefits: By shifting power consumption load peaks can be dampened effectively and the overall load fluctuation can be reduced. Moreover, fluctuations in the power generation of renewable energy resources may be levelled by an intelligent scheduling of such load shifts.
As household appliances account for a large fraction of the overall load and are a major contributor to the daily load peaks, we are investigating how a pool of household appliances can be used for load shifting.

2. Related work

Many schemes have been developed to integrate household appliances into the power balancing processes. (Short et al. 2007) presents a method to modulate the consumption of refrigerators based on a grid frequency control loop. In (Kupzog 2008) this concept is improved by a more sophisticated control loop. Control schemes based on the grid frequency are charming because of their simple yet effective principle. However, they act only reactively to support the grid. It is not possible to directly compose load shift schedules in advance. (Koch et al. 2009) presents a control scheme which allows direct control of thermostat-controlled household appliances. By connecting all appliances to a central controller a set-point trajectory may be specified which is tracked by the appliances.
So far, all these approaches are restricted to appliances with thermal storages. As shown in (Stamminger 2008), other types of appliances may also be used for demand side management. Appliances with deferrable operation as dishwashers, tumble dryers, and washing machines are of particular interest. The chal-

lenge for the integration of these appliance types into demand side management schemes lies within their unpredictable operation times. Deterministic control schemes developed in previous work can't be applied to these appliances. Instead a probabilistic control scheme is devised to circumvent the unpredictability in their operation.

3. A pooling-based strategy

Appliances without implicit or explicit energy storage are more difficult to integrate into load shift techniques as their activity and consumption highly depends on unpredictable user interaction. For example, the exact timeframe of a dishwasher's activity can't be reliably forecasted. Therefore, any possible load shift techniques implicating these appliances have to be applicable under uncertain conditions as well. Other aspects to be considered in a load-shifting strategy are long-term changes in appliance usage. The most intuitive examples are air conditioners whose activities are subject to environmental influences. Moreover, long-term changes may also be affected by user interaction. For example, many appliances won't be used if the user is on vacation and thus can't participate in load shift measures.
As reported in (Stamminger 2008) appliances usage and operation times underlie a probability distribution. By aggregating many appliances with similar consumption and load shift characteristics (whose are among other aspects governed by usage) into a pool, a so called virtual device, the individual probabilistic effects diminish. Instead of controlling each appliance individually, load shift actions are only scheduled for virtual devices. Because assumptions about the internal mode of operations are not made, the approach at hand is not limited to appliances with thermal storages and allows scheduling of basically any type of (household) appliance as long as it is possible to control its operation with respect to energy consumption.

3.1 Device representation

For such a pooling approach, first of all we need a strategy that is independent from concrete appliance's characteristics. This is achieved by abstraction: No explicit knowledge of appliance types or their specific properties is incorporated into the pooling process. Instead, any appliance is represented only by its load shift properties within a selected scheduling horizon. These are expressed in a list of so called 'actions'.

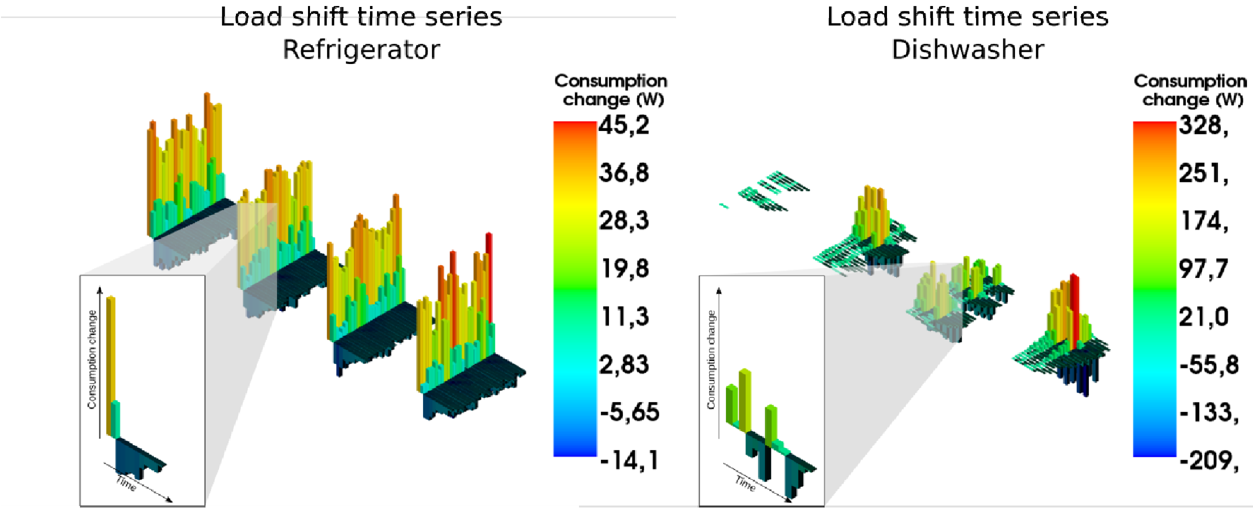

Figure 1: Load shift time series (refrigerator left, dishwasher right)

An action is composed of time series of the expected load shift with respect to the average consumption and its variance. Because the consumption may vary over time, distinct time series of load change need to be available for each time step of the chosen scheduling horizon. The controller of an appliance updates

these data for every scheduling horizon (e.g. once a day) and publishes it along with the appliance average consumption. These data are used to aggregate appliance into a pool which will be described later on. Schedules computed for the pool are mapped to individual appliance actions and transmitted to the appliance controller. These will initiate the given actions at the scheduled points in time in order to produce a superposition of load profiles which matches the pool schedule.
For the following examples, a scheduling horizon of 24 hours and a 15 minute resolution (i.e. 96 time steps in total) has been selected. As outlined in (Stadler et al. 2009), a refrigerator's consumption can be shifted, for example by a premature cooling process. This load shift effect can be represented by an action. Based on the average daily consumption the controller of the refrigerator is able to calculate the expected time series of consumption change for each of the 96 time steps. The time series for the premature cooling process are shown in figure 1 on the left hand side. A premature cooling process will result in a short consumption peak above average followed by a period of consumption below average. User interactions are only of minor effect to the refrigerator's operation. Therefore, the expected consumption changes look similar for all time-steps of the scheduling horizon.
The abstraction of appliances into actions also allows the specification of load shifts of other appliance types, like for example dishwashers. A dishwasher equipped with a start time delay function may be started by an externally scheduled control action at any time in the given delay timeframe. Based on the delay timeframe the dishwasher's controller may also update its load shift time series for each time step of the scheduling horizon. The time series for a force-start signal are shown in figure 1 on the right hand side. These have been generated by simulating the dishwasher's usage based on the operation probabilities described in (Stamminger 2008). Note how the effect of the action varies over the scheduling horizon.

3.2 Pooling measure

Load shifting properties of appliances differ depending on the type of the appliance, its usage and the environment. Therefore it's necessary to develop a measure for evaluation and comparison of appliance pools in virtual devices.
The measure is calculated for each timestep of the scheduling horizon (e.g. 96 15-minute intervals) individually. The mean of these values denotes the final pool rating.

The most important aspect of a virtual device is the shift characteristics. Only appliances may be aggregated whose load shift effects match to each other. Refrigerators will respond with a short load increase to a force-cooling action followed by a longer phase of reduced consumption (compare figure 1). The force-start action of a dishwasher will immediately increase the consumption and reduce the load only at a latter point in time. If these two appliances would be added to the same virtual device, the resulting load shift would be sub-optimal because the dishwasher will shift it's consumption into the reduced load phase of the refrigerator.

Based on the load shift time series a_e of each appliance a (which describe the change in energy consumption) in the virtual device pool A the energy ratio r_e is calculated as follows:

$$r_e = \frac{\sum abs(\sum_{a \in A} \overrightarrow{a_e})}{\sum \sum_{a \in A} abs(\overrightarrow{a_e})}$$

Another aspect necessary to be considered in pooling of devices is the duration of a load shift action. As an appliance is only able to carry out one action at a time, an action of the virtual device is required to last as long as the longest action of all appliances in its pool. The durations of actions must be kept short because no other actions may be scheduled concurrently. Therefore, only appliances with actions of similar duration should be added to a virtual device. This need is expressed in the duration ratio r_d:

$$r_d = \frac{mean_{a \in A}(\mathrm{len}(\overrightarrow{a_e}))}{max_{a \in A}(\mathrm{len}(\overrightarrow{a_e}))}$$

As mentioned before, the actions describe the expected load shifts. Because of stochastic influences the actual load shift differs by a certain variance from the expected ratio. By adding more and more appliances to a virtual device the stochastic influences will reduce relatively to the expected overall load shift. The variance of the load shift time series is used to describe the effect of stochastic influences. The operations of different appliances are assumed to be stochastically independent of each other. This allows calculating the variance of an action of a virtual device. Based on the overall load shift and the individual appliance load shift variances a_σ the relative standard deviation r_σ is calculated as follows:

$$r_\sigma = \frac{\sqrt{\sum_{a \in A} a_\sigma}}{\sum_{a \in A} \overrightarrow{a_e} \Big/ mean_{a \in A}(\mathrm{len}(\overrightarrow{a_e}))}$$

As can be seen, r_σ depends on the amount of shifted energy per timestep and describes the error which is to be expected by invoking the load shift action.

These three factors can be used to evaluate the appliance pool of a virtual device. However, none of these factors may be used to limit the integration of new appliances to a virtual device. Because of this, a relative standard deviation t_σ must be given as a target. If the relative standard deviation of an appliance pool diverges from this threshold, appliances from the pool are distributed to other pools.

The resulting measure r is calculated as follows:

$$r = \frac{r_e \cdot r_d}{abs(t_\sigma - r_\sigma) + 1}$$

Note that r is limited in a range between 0 and 1, whereas 1 denotes a perfect appliance pool.

3.3 Architecture

The pooling measure ensures that only appliances with similar load shift properties are assigned to the same pool. Using this mechanism, it is possible to pool appliances into virtual devices whose aggregated load shift actions meet a given constraint on its standard deviation.

Changes in usage or in the environment will influence the load shift properties of an appliance. Furthermore, new appliances may be installed, old appliances replaced (possibly by new models with reduced energy usage or new capabilities) or even completely removed from a household. All changes may happen concurrently and each virtual device must react appropriately.

Multiagent systems have been successfully applied in many different domains to cope with inherent system dynamics (Wooldridge 2009). Therefore the system at hand has been implemented as a simulation model based on a multiagent system. Agents are arranged in a network and communicate with connected siblings. The simulation model also includes a network layer model which realizes the transmission of messages between the agents. The purpose of the network layer is used to simulate communication latencies which are the cause for asynchronous message delivery. This approach has been chosen to closely model reality and validate the behavior of the pooling strategy under these circumstances. However the network layer model is not described here for reasons of space.

In the given scenario several types of agents are composed into a hierarchical network:

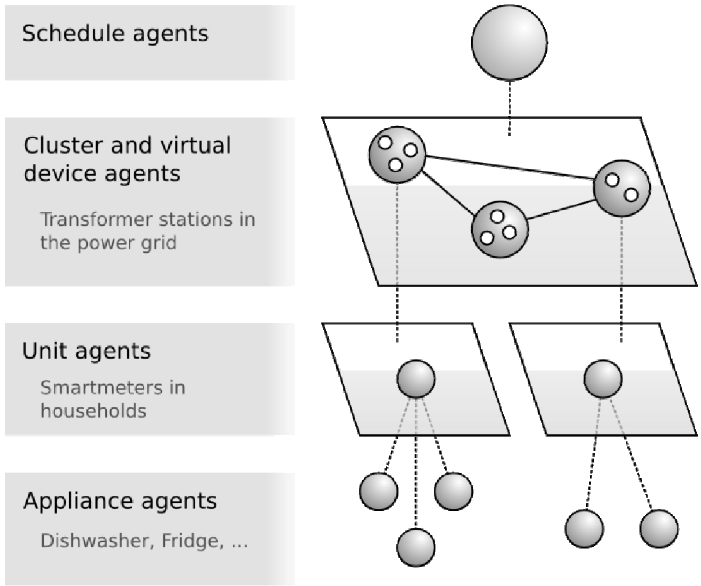

Figure 2: Architecture

- Appliance agents represent household appliances. Their responsibility is to forecast the resulting time series of possible load shift actions. The time series will be published in certain intervals (e.g. once a day). Appliance agents will also process action schedules and initiate load shift actions at the appropriate time.
- Unit agents are used to model a household with multiple attached appliance agents. These agents may run on smart meters for example. They act as a communication interface to the household and will store appliance schedules and forward appliance information.
- Cluster agents embody the structure of the power grid. Each unit agent is connected to exactly one cluster agent. Cluster agents receive notifications of newly installed or removed appliances and updates to load shift time series of existing appliances. Furthermore cluster agents manage communications between appliance agents and virtual device agents. Additionally, they maintain a neighborhood of other cluster agents.
- Virtual device agents represent a pool of appliances. A virtual device agent is bound to a cluster agent and is connected by a communication network to those virtual device agents which are managed by all cluster agents in the neighborhood. The motivation of a virtual device agent is to optimize its pool of associated physical appliances with respect to the pooling metric described above. An altruistic self-organization strategy has been developed for that purpose, which will be detailed later on.
- A scheduler agent groups several cluster agents and schedules load shift actions for appliances which are grouped in the pools of virtual devices.

3.4 Pooling strategy

The strategy developed for clustering appliances into virtual device pools is altruistic in that each virtual device agent v_i will always transfer the control of an appliance a to a neighbor $v_n \in N_{v_i}$ if v_n will get a higher pooling measure value $r(v_n + a) > r(v_i)$ by this transfer. This appliance distribution process is triggered either by time series updates from appliances or appliance transfers between virtual device agents. This allows the system to immediately respond to changes in the appliance population. The distribution of appliances from a pool v_i to a pool v_n is repeated until there are no more appliances left in v_i or no neighbor v_n will gain a pooling metric value higher thanv_i. A virtual device agent ceases to exist once all its appliances have been distributed. The repeated execution of the distribution process results in quick removals of virtual devices with small pool sizes. This approach was chosen because it reduces the amount of virtual device agents and consequently the communication overhead.

Fehler! Verweisquelle konnte nicht gefunden werden. outlines the strategy in detail. The most important aspect of this strategy is the so called revision list. It consists of an entry for each neighbor v_n containing the timestamp t_{mod} of its most recent modification. With this information an agent v_i is able to determine whether it is its turn to distribute appliances or to wait for incoming rating requests. The reason for synchronizing the distribution processes is to reduce the amount of unfortunate appliance transfers. An asynchronous distribution process would yield discrepancies between reported ratings and actual ratings. The query of neighbor ratings is completed after all neighbors have responded. In an asynchronous distribution process, a fast responding neighbor would handle concurrent requests and might have already received control over appliances while a former query request is still active. With these newly received appliances the reported rating of the former query request would have become invalid. In the worst case an appliance would be transferred to a virtual device agent v_n and cause a decrease of $r(v_n)$ because its pool has changed in the meantime.

The pool of a virtual device v_i must at least contain one appliance; otherwise the virtual device agent

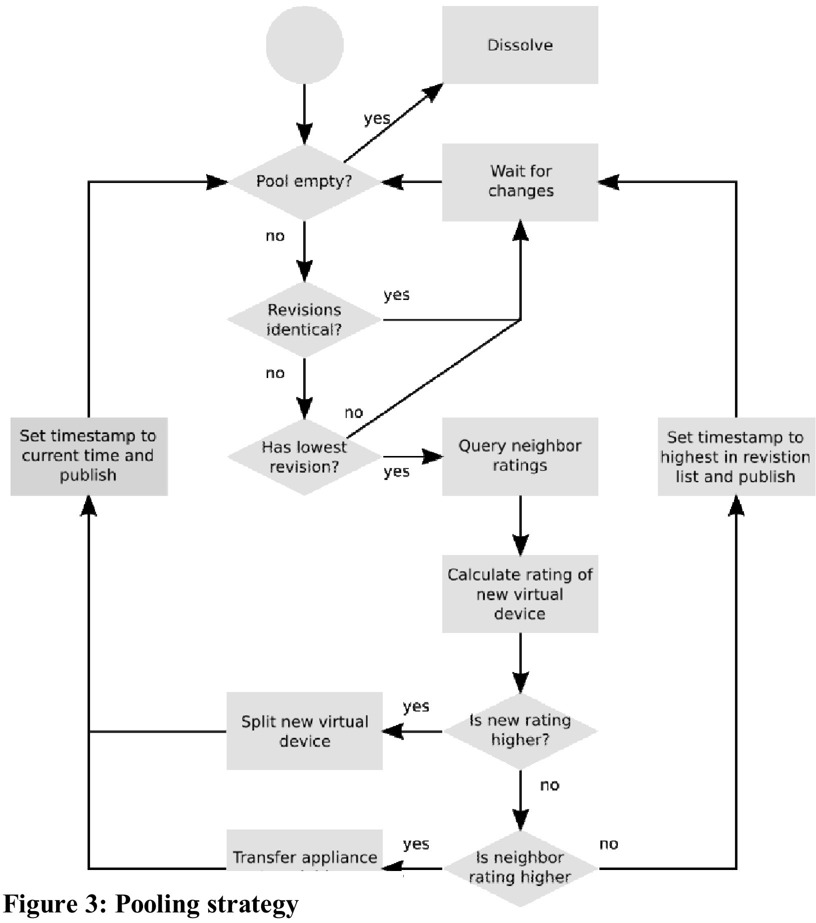

Figure 3: Pooling strategy

will terminate and dissolve. If there are appliances a_i in the pool, the agent checks its copy of the local revision list. If all neighbors v_n have reported an identical modification timestamp so that $t_{mod}(v_i) = t_{mod}(v_n)\ \forall\ v_n \in N_{v_i}$, it simply waits. In this case all virtual device agents $v_n$in the neighborhood tried to distribute an appliance a but could not find a neighbor v_m with $r(v_m + a) > r(v_m)$.

If the virtual device agent v_i has the lowest revision timestamp so that $t_{mod}(v_i) < t_{mod}(v_n)\ \forall\ v_n \in N_{v_i}$, it is v_i's turn to distribute appliances. The agent will first select the appliance a_r whose removal would yield the highest improvement of the rating, i.e. $r(v_i - a_r) > r(v_i - a_k)\ \forall\ a_k \in v_i,\ a_r \neq a_k$. Every neighbor v_n is now queried for its rating if a_r would be merged into its pool. Equally the rating of a virtual device v_r containing only a_r is computed. If the latter is the highest rating of all, i.e. $r(v_r) >$

$r(v_n + a) \; \forall \; v_n \in N_{v_i}$, a new virtual device v_r will be created. If this is not the case but there's a neighbor v_n with a higher rating, i.e. $r(v_n + a) > r(v_r)$, the control of the appliance will be transferred tov_n. In both cases the revision timestamp of the agent is set to the current time and is published to all neighbors. Afterwards, the process repeats by checking the pool size again. If no virtual device agent with a higher rating was found at all, i.e. $r(v_n + a) < r(v_i) \; \forall \; v_n \in N_{v_i}$, the appliance is kept in v_i. However the revision timestamp is only updated (and also published) to the highest timestamp found in the revision list instead of updating it to the current time.

This behavior ensures that every agent is able to react to changes in the neighborhood. In case of a change (for example an update from an appliance agent a_i), the associated virtual device agent v_i updates its revision timestamp and publishes the new value to all neighbors $v_n \in N_{v_i}$. Every neighboring virtual device agent's revision timestamp is now older than the published one, i.e. $t_{mod}(v_i) > t_{mod}(v_n) \; \forall \; v_n \in N_{v_i}$. Thus one neighbor after another will now try to distribute an appliance.

Note that although the revision list has been integrated to prevent concurrent appliance transfers, those may still happen because updates from appliance agents may arrive at any time causing new distribution processes. This results in multiple agents distributing appliances at the same time, but as the revision list is checked after each iteration this issue will be automatically resolved after some time (e.g. agents will stop distribution if a neighbor published a newer revision timestamp).

Using this strategy the virtual device agents are permanently rearranging their appliance pool.

3.5 Scheduling

With appliances grouped into virtual device in the manner proposed above, the stochastic influences of load shift actions are reduced. However the pooling strategy is causing a permanent reorganization of the appliance pools. It is necessary to take snapshots of the appliance pools to schedule load shift actions. This is the responsibility of the scheduling agents. These request snapshots of the virtual devices at fixed intervals (for example at midnight). Using these snapshots scheduling agents may for example employ an algorithm to schedule load shift actions to adapt the overall load to a given target curve. Our future work will include algorithms for this purpose.

4. Results

The pooling strategy is examined in a simulation scenario with 2000 refrigerators and 2000 dishwashers for a simulated duration of four days. The results are shown in Figure . The target relative standard deviation t_σ has been set to 20%. Because of the different load shift capabilities the different appliance types are not grouped into the same virtual device. With the targeted relative standard deviation the refrigerators are grouped into 16 virtual devices, while the dishwashers are pooled into only two virtual devices due to higher stochastic influences.

The upper graph shows the actual consumption in real and predicted consumption in 15 minute resolution. The graph in the middle shows the deviation between the scheduled consumption and the actual consumption. The distribution of appliances to virtual devices is represented in the lower graph. Each patch depicts the pool of a virtual device. The height of the patch corresponds to the amount of appliances in the pool while the color hints at the type of appliances in the pool.

Per minute two refrigerators and dishwashers are added to the system. These are integrated immediately and cause the creation of virtual device agents. Initially all appliances are publishing pre-calculated load shift time series. These do not yet match the actual appliance behavior. The small deviations are reduced at the start of the second day after the appliances have published their adapted time series. The prediction does only violate the 20% boundary a few times even though there is a virtual device of dishwashers included, whose pool size is not optimal.

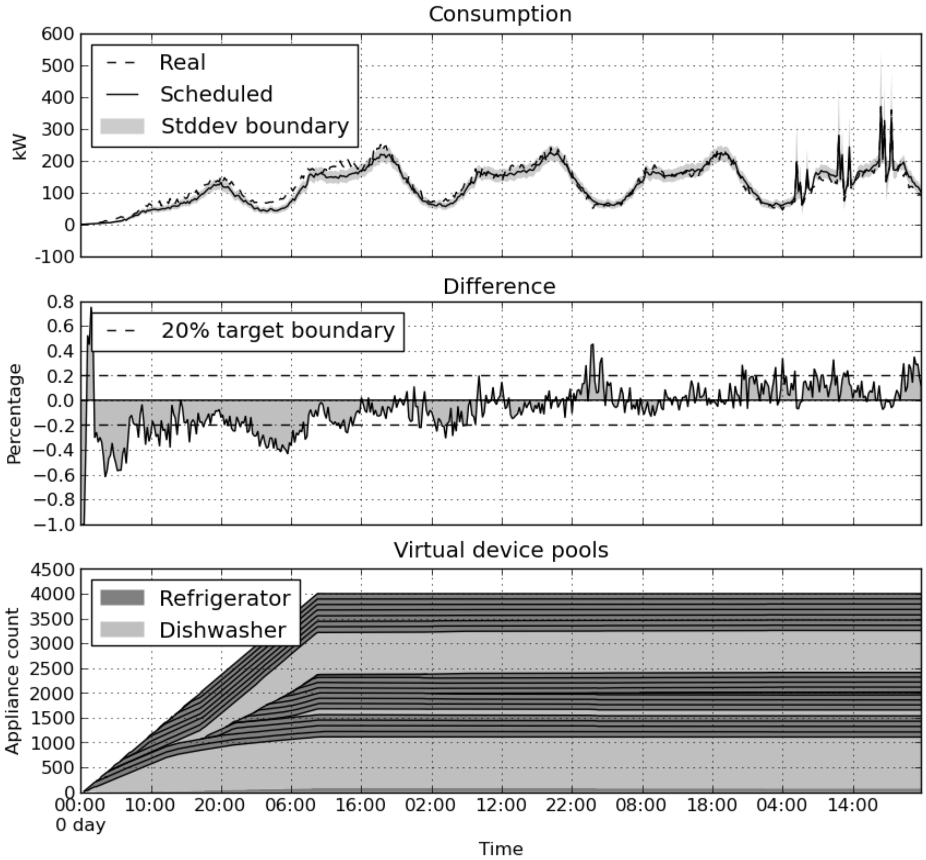

Figure 4: Results

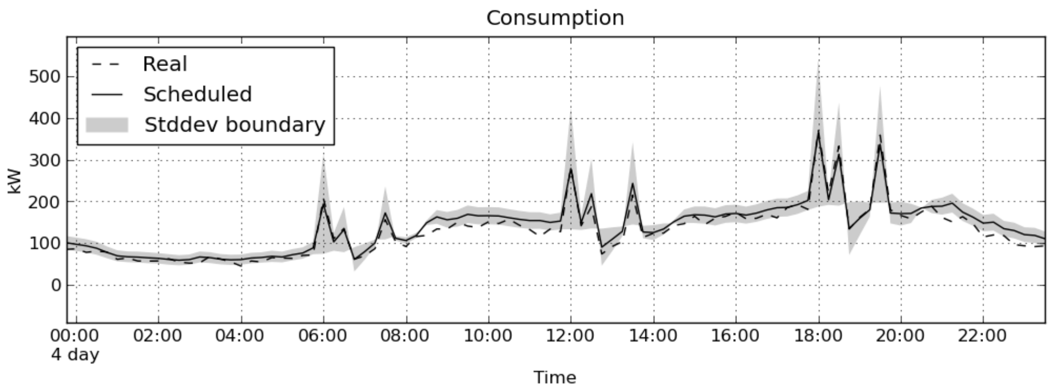

Figure 5: Scheduled load shifts

On the fourth day three load shift actions are initiated by the scheduling agent. As can be seen in the detailed view in figure 5 energy is consumed at the scheduled point in time resulting in reduced consumption in the following. In the resulting load curve the load shift time series of the dishwashers shown in **Fehler! Verweisquelle konnte nicht gefunden werden.**, although superimposed with the refrigerator load shifts, can be identified.

5. Outlook and conclusion

Household appliances are pooled into virtual devices for load control. A virtual device aggregates the load shift capabilities of its entire appliance pool and reduces stochastic effects of individual devices' behavior. The outlined strategy allows to integrate different appliance types into demand side management activities. Due to permanent reorganization of virtual devices, the system is able to adapt itself to short-time as well as to long-term changes.

In comparison to existing approaches the strategy at hand depends on the availability of many devices for effective load shift scheduling. Furthermore although stochastic influences are reduced they are not completely prevented. This is a tradeoff for generality as this approach can also be applied to appliances with deferrable operation modes. Relying on probabilistic behavior also reduces communication requirements. Persistent communication channels are not necessary. The communication with physical appliances is only necessary to publish their load shift time series once per scheduling horizon and to fetch their schedules.

Future work will further exploit this strategy and integrate additional appliance types (for example heat pumps, tumble dryers, etc). Additionally scheduling algorithms to compose load shift actions will be researched as well.

6. References

Koch S., Zima M., Andersson G. (2009): Active Coordination of Thermal Household Appliances for Load Management Purposes. IFAC Symposium on Power Plants and Power Systems Control, Tampere, Finland

Kupzog F. (2008): Frequency-responsive load management in electric power grids, Südwestdeutscher Verlag

Short J.A., Infield D.G., Freris L.L. (2007): Stabilization of grid frequency through dynamic demand control. Power Systems, IEEE Transactions on, 22:1284–1293

Stadler M., Krause W., Sonnenschein M., Vogel U. (2009): Modelling and evaluation of control schemes for enhancing load shift of electricity demand for cooling devices. Environmental Modelling & Software, 24:285–295

Stamminger R. (2008): Synergy Potential of Smart Appliances. Report of the Smart-A project.

Wooldridge M. (2009): An introduction to multiagent sytems. Wiley, 2. ed.