Published in: D. Möst, W. Fichtner, M. Ragwitz, D. Veit (Eds.), 2008:
Proceedings of the 1st European Workshop on Energy Market Modelling using Agent Based Computational Economics.
Universitätsverlag Karlsruhe, ISBN 978-3-86644-238-2, S. 77-92

A modelling tool for interaction and correlation in demand-side market behaviour

Jörg Bremer, Stefan Andreßen, Barbara Rapp, M. Sonnenschein and M. Stadler

Carl von Ossietzky Universität Oldenburg
and OFFIS Institute for Information Technology, Oldenburg
Joerg.Bremer@infomatik.uni-oldenburg.de

Summary. We present an agent-based software environment for modeling and simulation of adaptive consumers responding to dynamic electricity pricing. It has been specially designed for scenarios involving household customers. Households can be modeled down to the layer of single appliances, even taking into account presence and price awareness of inhabitants. Modeled utilities can calculate prices from different factors using different methods. The focus of investigations conducted is the analysis of household load shifting potential under different tariffs and different negotiation strategies.
Keywords: agent-based modeling, electrical load shifting, household consumers

Acknowledgement

We thank EWE AG, a regional utility in northern Germany for funding large portions of the work described herein in conjunction with the subproject 'Adaptive Consumers' embedded in the project DEMS for exploring management of decentralised energy systems.

1 Introduction

It is important to keep the balance between electricity consumption and electricity production at all times. This is usually done by controlling generators. This approach does not take into account the necessity of minimising the marginal costs of electricity generation and electricity distribution at all times. The major factors responsible for high marginal costs are peak loads that usually are met using expensive peak load generation facilities, and grids dimensioned to support peak loads. Thus, it is necessary to minimise peaks while at the same time optimally utilise base load plants.

Peak load reduction can be achieved either by power conservation methods or via load shifting on the consumer side which is also suited to address the problem of optimal utilisation of base load plants. Load shifting can be achieved by methods of demand side management involving direct control of appliances or indirect control by addressing the inhabitants of households through time varying tariffs reflecting the marginal costs of electricity production. The maximum load shifting potential of households has still got to be examined and also the percentage of load shifting potential activated by different tariff signals or other control signals is currently not well known.

This calls for a model of power consumption reflecting electricity demand down to the device level also encompassing human price awareness, presence profiles, as well as seasonal influences and thereby the elasticity of demand. Furthermore, if different negotiation strategies shall be examined w.r.t. to the achievable matching between load and production, the models must also support federations of consumer agents negotiating with utilities.

In recent years, a lot of research has been done in the field of optimizing electrical load through demand side management methods. We limit a short overview here on attempts related to multi-agent technologies. The HomeBots approach [Ygge and Akkermans 1996, Ygge 1998] is based on the idea of an electronic market place where utilities and devices (e.g. electrical heaters) are trading small amounts of energy to match the current demand with the supply. Prices are fixed by a market based approach. A similar approach is proposed by [Wedde et al. 2006] focussing more on the technical aspects of implementing a distributed market place. Both attempts aim for a completely new, distributed method of demand-supply matching as well as electricity pricing that is not oriented at our current system of energy supply. They do not incorporate planning agents scheduling energy demanding tasks to optimally match between supply and demand.

In [Penya 2006] also a retail market place for electricity is investigated, where a reverse combinatorial auction is used as a method for electricity pricing and load shifting. This model consists of two different agent-based systems: one system for price fixing between utilities and consumers, and another system for scheduling of demand tasks, This approach results in very high communication costs - it has not been tested for ‘real world’ scenarios.

For these reasons we decided to develop our own agent-based modelling and simulation platform ACDC (adaptive consumers for dynamic cost models) aiming at dynamic pricing for consumers which is presented in this paper. A preliminary version of the ACDC frameworks has been described in [Sonnenschein et al. 2006].

This paper is organised as follows: After a short introduction to the domain of load management in households we explain the architecture and some basic concepts of our agent based simulator ACDC in chapter two. In chapter three different hypothetical scenarios for negotiation between consumers and a utility are introduced. These scenarios are implemented in the ACDC framework and demonstrate its flexibility.

1.1 Load shifting within households

Load shifting is a basic mechanism for adapting load curves so as to reduce marginal costs of electric power supply. Load shifting within households is based on their inhabitants behavior and in their equipment. Thus, we have to account for both.

There are different types of appliances. Controlling appliances control an environmental factor within the household. Examples of this appliance type are fridges, air conditions, heaters and boilers. Appliances for spontaneous use are switched on and off by persons, following their needs of daily living. TV sets, lighting and cooking equipment are examples for this type. Finally, programmed appliances after having been switched on follow a given program and have a predictable power consumption sequence. It is important to note that the mechanisms of load shifting differ substantially for the appliance types.

The load shifting potential of appliances for spontaneous use is only dictated by the needs of persons that vary due to many factors as circadian rhythms and social events. Usage patterns for this appliance type are influenced by the presence of persons within a household and circadian rhythms but can be modified through social events or other causes. The only way of exploiting a load shifting potential is by modifying people's behavior, for example by varying electricity costs and thereby exploiting price awareness.

Load shifting potential for programmed appliances is also driven by human behavior, but to a lesser degree. Usage of dishwashers, washing machines and alike devices is not always spontaneous and often planned. For instance, people might refrain from using their washing machine late in the evening due to noise. Also usage of programmed appliances is mostly delayed until after a certain payload has been reached. A very interesting fact concerning those appliances is, that they are equipped with controllers allowing to delay operation after starting the device. This can be interpreted as a user wanting the appliance to terminate its program at a latest given point in time. Taking this as a hook for integrating a scheduler allows for load to be shifted to a moment when electricity consumption is most desirable due to sufficient resources or low prices. Note, that the load can be shifted forward in time.

Controlling appliances control a physical property (e.g. temperature) to stay within bounds defined by the persons living in a household. For that purpose they do not depend from human interaction. By intelligently exploiting the range between the bounds, load can be shifted. For instance, the temperature within a fridge's cooling compartment may be allowed to vary between 5°C and 8°C. Whenever the upper bound is reached, the fridge's cooling aggregate is started. It stays on as long as the lower temperature bound is not under-run. Using an intelligent fridge controller, a fridge's thermal storage capacity can be exploited as to prepone or postpone the cooling aggregate's activity and thereby shifting load for time spans of about an hour [Stadler et al. 2007].

Single persons are reflected in our consumer models by their presence profiles. The usage patterns for household appliances derived from human behaviour are modeled through usage profiles.

2 ACDC – a modelling and simulation framework for cost adaptive electricity consumption scenarios

A model in our ACDC framework consists of an electricity supplier (power company) calculating real-time prices for electricity, different classes of electricity consumers (households) modeled in essential by their scheduling mechanism of energy consuming tasks, and a communication protocol between consumers and the electricity supplier. Power plants and electric generators as e.g. wind turbines are modelled only as sources for time series of available electrical energy.

2.1 System architecture

The simulation and modeling environment ACDC consists of the basic building blocks depicted in figure 1. A purpose-built modeling tool allows graphical construction of models and parametrisation of models and their components. Aspects that cannot be graphically defined (e.g. interaction protocols and rule sets for defining tariff calculation) may be composed from within a simple built-in text editor. Once defined, parametrised models are stored as XML-files within the system's scenario repository.

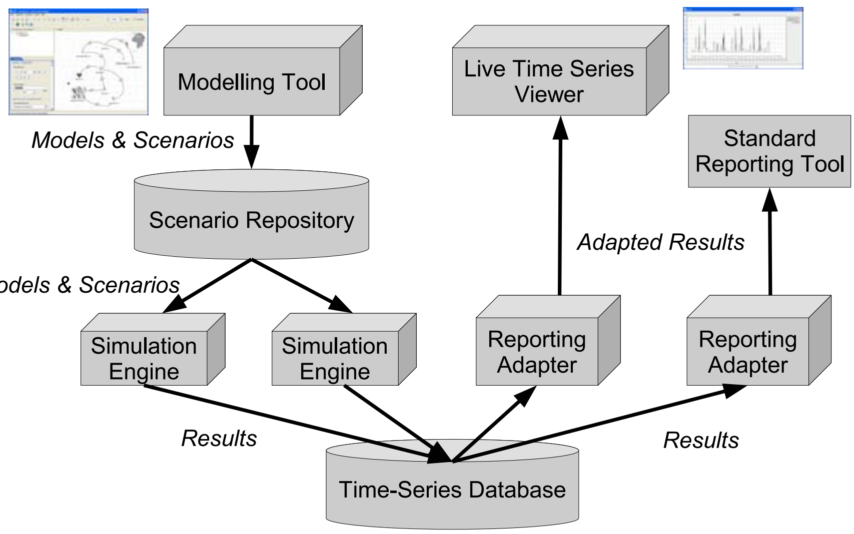

Figure 1

In standalone mode, these scenarios are read by a single simulation engine as soon as the 'start simulation' button is clicked. In distributed mode, multiple simulation clients may fetch scenarios from the scenario server as soon as they run out of a job. The generation time series, tariff time series and load timeseries are stored within a relational database at simulation time. From there, they can be retrieved via reporting adapters, either by standard reporting tools or by a live time series display and analysis tool currently under construction.

The simulation engine is built around the Repast framework [North et al. 2006] offering a comprehensive infrastructure for agent-based modelling and simulations including mechanisms for communication and synchronisation. Agents in this framework do not pursue their tasks in real-time but execute concurrently based on a common global simulation time. Developing the ideas from [Rölke 2004] this allows for partitioning a complex process of interaction into independent, loosely coupled software entities.

Inter-agent communication is based upon the Repast Framework's event handling mechanism. To guarantee for extensibility, the flow of communication is specified within replaceable interaction protocols implemented through a control unit and a monitoring unit. Communication channels are configurable by allowing for specification of bandwidth or message-delay by the modeller and they are separated from the communication methods used (e.g. messages or handshakes).

Aside from messages, the behavior of agents may be influenced by events. Whenever an event occurs, an agent may react to it or choose to ignore it. The simulator allows for specification of contents, purpose and other properties of messages and moreover it offers a broadcasting functionality that can be used for agents sending a message to multiple receivers.

2.1.1 *Agent types and agent structure*

Each agent within a simulation represents an entity within our scenario. The type specific functionality of an agent is structured in a number of type specific modules. Helper classes needed within modules or agents are represented by items which are not limited to representing data but may also have functionality assigned to them.

We need three types of agents for modeling the scenarios required for our analyses. Utility agents represent utilities calculating electricity prices from different inputs comprising wind power predictions, electricity stock market prices at the European Energy Exchange, and also electricity consumption predictions. All of these inputs may be calculated from outputs of consumer agents or from utility agent modules providing interfaces to Excel files and comma separated value files.

Electricity prices can be negotiated between utility agents and consumer agents in several rounds. The information exchange is performed by sending tentative pricing time series from the utility to its consumers, and by sending predicted consumption time series calculated for a given pricing time series from consumers to their utility.

Consumer agents represent households that consume electricity taking into account given pricing information. Consumption is aggregated from the set of appliances present within a household.

Both, the activities of programmed appliances and controlling appliances are scheduled by per-agent schedulers using different strategies.

As an example, figure 2 depicts the internal structure of an agent representing electricity consumers. Device models, scheduler, metering and control unit are modules, while human behavior and consumption statistics are delivered by items.

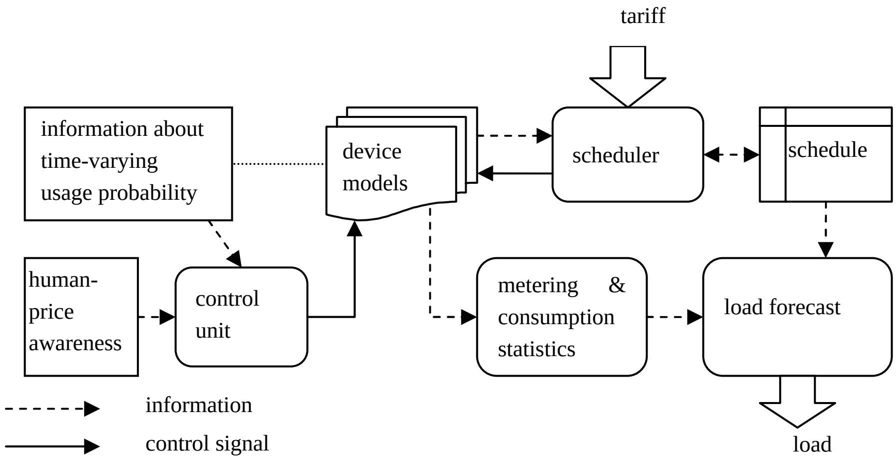

Figure 2

2.2 Configurability

The ACDC tool consists of a simulator engine and an optional graphical user interface. Both are highly configurable in favor of adjustability to specific needs. The simulator engine was implemented as a standalone desktop application based on the Spring Framework [Johnson 2007]. This layered Java/J2EE application framework includes a non-invasive lightweight container that is able to link loosely-coupled components to a complex system.

The simulator engine's configuration is done in terms of XML. The Spring Inversion of Control (IoC) processes configuration options, such as the prefered persistence technology, the list of databases to access, or the option of declarative transaction management, and so on, given in a simple and intuitive XML format. At runtime the configuration file is processed by the container. In case of a distributed simulation on more than one machine and hence different simulator engines the underlying configuration can be shared across the network. This allows for configuring different decentral simulator engines either identically or independently of each other.

Configuring scenarios for simulation mostly takes place in terms of XML. Each scenario configuration is divided into two separate parts: Firstly, all agents participating in a simulation, their associated functionality, all required communication channels as well as desired choreographies are listed. All these entities are required to achieve the target-settings by simulating a scenario. This first brief part of configuration is passed to and evaluated by our custom-developed parser. The second part of the configuration file contains a series of Java object definitions that are required for simulation as well. Syntactically this part corresponds to a format processable by the IoC container out-of-the-box. At runtime the container performs instantiating or sourcing application objects (e.g. agents, modules, items etc.), configuring these objects, and assembling the dependencies between them.

Most of the participating agents require different external definitions to satisfy their functions. Such external definitions like profiles of presence, price awareness or rule sets have to be specified with external tools and can be used for simulation via referencing such files in the scenario configuration. In addition, individual inhabitant types can be modeled with individual behaviour in using certain appliances wich is also be influenced by their individual price awareness. Hence a correlated behaviour can be achieved.

Individual modelling of each single agent in large scenarios is a tedious and error-prone work. Thus, our system exploits a concept for factory driven generation of heterogeneous agent populations for large scenarios. Individual agents of such a population with mutually distinct traits and behaviour as well as diverse equipment may be described by means of specialized XML-based definitions. Specialized, so called factory agents take use of these descriptions and generate the actual definitions of a large number of agents according to the given stencil.

In connection with developing appropriate structures for such descriptions of distinct model parts and agents for use in factories the following problems had to be taken into account:

Prefabricated model components must be available from different libraries in order to avoid double definitions, for reuse, and ease of modelling.

Modelling of well defined varieties within model components must be possible in order to support heterogeneous agent populations; e. g. the definition of a refrigerator with an arbitrary but well defined distributed power input of its compressor.

Dependencies between model parts must be part of their definitions. E. g. supposed a specific class of household agents possesses appliance A with a given probability P1. Now it should be possible to define that if a specific instance of these agents actually possesses appliance A than it also possesses appliance B with probability P2.

In order to support a flexible integration of functional elements into the structure of our XML-based model description, we adapted the principles of custom elements. Custom elements are a commonly used build-in programming means within JavaServer Pages (JSP) wich allow for program generated parts within fixed template HTML-code.

So, we adapted this idea and enriched our XML-based factory definitions with similar structures for dynamic content. That means that our parsers for XML factory definitions are capable of identifying dynamic content by means of namespace, interpreting such elements and replacing them with the generated output of the corresponding implementation. Figure 3 schematically shows the chosen approach.

In JSPs action elements represent dynamic actions that are executed at runtime when a JSP is requestet [Bergsten 2002]. In analogy to this, our dynamic XML elements are executed at parse-time. It is the factory parsers task to add a given number of agent definitions to a given scenario definition. In this way an agent description in a factory serves as a fuzzy stencil for the definitions to be inserted into the scenario. For each inserted instance the dynamic elements' action classes are executed. In order to find the approriate implementation, a mapping between element names and implementing class existes. The individually generated output of these action classes then replaces the respective element within the stencil resulting in distinguishing agent definitions derived from a single prototype.

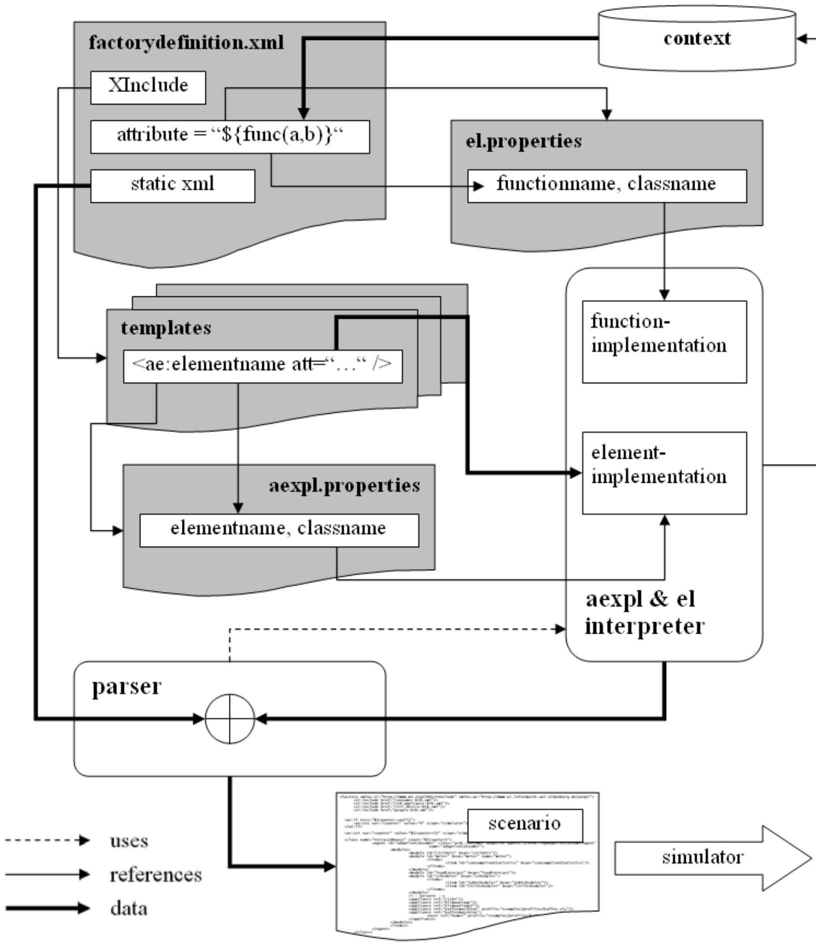

Figure 3

As yet, we use these dynamic elements to realize control structures, for an integration of numerically derived content, i. e. specifically distributed values, and script generated content by external script-languages like for instance java script. In order to pass information between different executions of actions as well as between different simulation runs, two different scopes for variable values are provided: *scenario* and *simulation.* Data in the scenario scope is only valid for a single scenario simulation run whereas the simulation scope allows for passing data from one simulation run to another; listing 1 shows an example of modelling a batch simulation with a growing number of agents for each run.

```
<ae:if test="${empty counter}">
  <ae:set var="counter" value="0" scope="simulation"/>
</ae:if>
<ae:set var="counter" value="${counter+5}" scope="simulator"/>
<class name="growingExample" count="${counter}">[...]</class>
```

Listing 1

In addition, we integrated support for dynamic attribute values. The Expression Language (EL) is defined by the Java Standard Tag Library (JSTL) specification [Delisle 2002]. EL expressions can be used to set attributes to dynamic values. Whereas the JSTL implementations only allow for a use with action elements our implementation allows for setting attributes of arbitrary (including non-dynamic) elements to dynamically generated values. A shared context allows for exchanging variable values between dynamic elements and EL expressions, or rather, their implementations. Another advantage resulting from our implementation is an easy way of extending the original standard language by integrating own implementations of additional functions as shown in listing 2 where the non EL function *weighted_pick* is used within an EL expression to choose from the given choices with respective probabilities.

```
<appliance ref=${weighted_pic(fridgeA, 20, fridgeB, 80)} />
```

Listing 2

Factory driven generation of scenarios in conjunction with dynamic definitions and commonly used external profile definitions enables us to model and simulate scenarios with a well defined correlation between different actors within a simulated scenario as well as dependencies between different simulations runs. For example, in a scenario with multiple different agents representing varying instances of different classes of households generated by factories, it is still possible to model a correlation in the usage time of the refrigators by providing them with the same usage profile. In this way, all households will show a certain similarity in behaviour concerning the usage of the refrigator (i. e. opening the door, filling in warm food, etc.) while each of these appliances is separatedly treated, or rather, optimized according to the settings of the simulation.

3 Definition of three scenarios for negotiating tariffs between a utility and its customers

Negotiation in each scenario presented in this chapter is based on feedback loops between a utility and its customers. The feedback to a tariff given by a utility is a load forecast. We view tariffs as a control signal leading to a load shift resulting in a better balance of electricity consumption and electricity production. This approach differs from the approach described in [Wedde et al 2006] where the feedback of a customer contains pricing information instead of information on load.

3.1 Independent customers negotiating with a utility

The standard scenario used for conducting analyses of tariff induced load shifting of household customers is based on the assumption that customers act independently of each other, each reacting on tariff prognoses. Figure 4 shows a simplified structure of the scenario with an emphasis on agent communication.

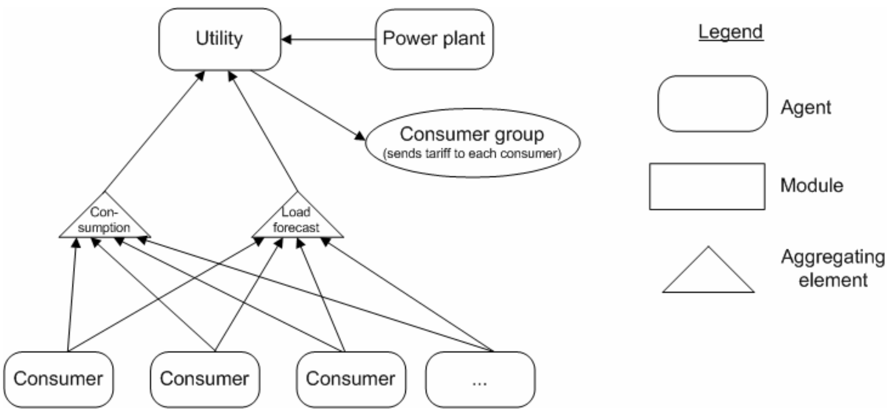

Figure 4

Each day at a specified time, the utility begins tariff negotiations. A single negotiation round is split into the following steps:

(1) The utility issues an initial tariff prognosis for the next day based upon the predicted costs of electricity, the input from wind power and a total expected load (based on prior experience).
(2) Loop until break criterion is reached

a. Consumer agents optimize the schedule of their device activities based on the predicted tariff and issue a load prognosis.
b. The utility agent calculates a new tariff based on the predicted load and the facts already known at calculation time of the initial tariff.

(3) The last predicted tariff is sent to the consumers as final tariff; monitored household consumption is measured and integrated into the utilities load expectation for future days.

The break criterion may be based upon the number of negotiation rounds or on the trend observed in the difference between desired load curves and achieved load curves. If the difference grows, it is assumed, that at least a local optimum had been reached.

This approach is well-suited to explore the maximum achievable load shifting for a given population of consumers and a given tariff calculation method. However, using this interaction protocol, tariff valleys will result in load peaks. This is due to a synchronised behaviour of agents neither having knowledge of each others reaction nor having any means of knowledge about the actions necessary to purposefully change their device activity plans with respect to the global optimization goal to reduce load peaks.

3.2 Probability based negotiation between Federated customers and their utility

An extension of the standard scenario described in section 4.1 has been designed to reduce the disadvantages of independently negotiating consumers.

Consumers negotiating following a probability based strategy are indirectly aware of other consumer's presence, i.e. they do not communicate with any other consumer, but assume the existence of other consumers scheduling their devices to minimise their electricity costs. In this negotiation type, consumers agree to the fact that independent cost optimization for all consumers result in load peaks, and they agree to contribute in a fair way to avoid this effect. For this purpose, they use a probabilistic approach to attenuate their cost minimisation, resulting in a cooperative behaviour between consumers and between consumers and the utility. This means a partial abandonment of local benefits in favour to allow a better match between electric load and production curves. The reduced load peaks lead to lower marginal costs for electricity supply and result in lower consumer prices in the long term.

The probability based negotiation is based on a modification of the signal 'strength' computed for each time slice of tariffs. This signal strength is compared with a randomly generated threshold value calculated separately by each consumer agent. The comparison of the signal strengths and the locally computed thresholds is used by consumer agents to calculate a fictitious tariff only used for device scheduling purposes but not designating electricity costs. To construct this fictitious tariff, a so called base tariff is modified by each agent during each subsequent negotiation round as follows:

If the local threshold is greater than the modification signal strength for a time slice, the base tariff's corresponding value is replaced by the time slice's value of the tariff issued by the utility agent during the current negotiation round. Other-wise the base tariff's time slice value is kept. In that way each customer agent constructs an individual tariff as input to their device activity scheduler resulting in a modified device activity plan.

The base tariff is a helper tariff which holds the costs that occurred in a time slice at the beginning of a price increase or decrease tendency of tariffs published by the utility agent during previous negotiating rounds.

In the probability based negotiation scenario the assumption of homogeneous consumers is made. All consumers behave the same way regardless of their relative contribution to electricity consumption. Shifting load from a given time slice yields different results, depending on the consumer's contribution to total consumption. This can lead to inadequate load shifting. In addition to that, the randomized threshold only guarantees for fairness of load shifting distribution between agents, if the number of consumer agents is big enough.

First simulation runs show that the probability based negotiation indeed results in a better matching between electricity consumption and electricity production, if the signal strength is increased relatively slowly between negotiating rounds. Rapid increase of signal strength often results in a customer's over-reaction and therefore can be a reason for insufficient load shifting. Besides achieving the main goal of improving the match between consumption and production, the probability based negotiating has the advantages of having both, low computational performance requirements and communication requirements.

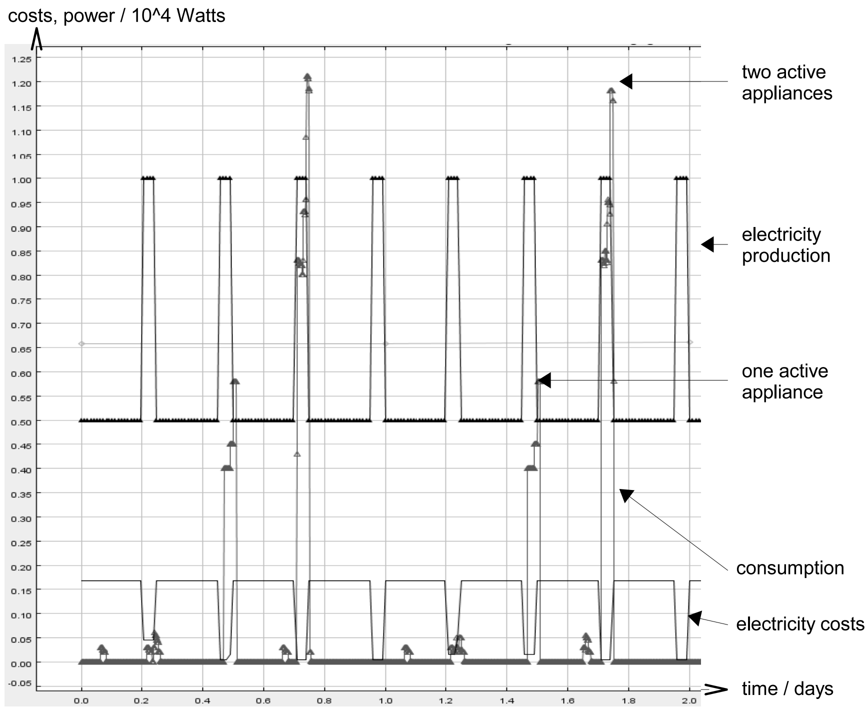

Figure 5

Figure 5 depicts the correlation between electricity costs and appliance usage for three households resulting from simulating probability based negotiation for a duration of two days. There are two appliances per household. One of those has a low power rating and is of use during morning and evening hours. The other has a high power rating and can be used throughout the day.

The diagram shows, that the appliance type with high power rating is only activated when electricity costs are low. Moreover, due to probability based scheduling of appliance activities, not all three households activate their appliances with high rating during the same time slots with low electricity costs. Instead, activity of those appliances is distributed over the available time slots. Note that the first and last time slot of each 24 hour period are not used, since appliance activity has been forbidden during the night.

The appliance type with smaller rating cannot be scheduled to be active during low cost electricity phases, because those phases do not overlap with the allowed per-day activity periods for this appliance type.

3.3 Communicative approach for federated customers negotiating with utility

We assume that an even better match between electricty consumption and electricty production can be achieved by federated planning, and by dropping the assumption of homogeneous consumers. Federated planning is conducted by a special agent named concentrator. Each concentrator is assigned to a number of consumer agents and communicates with the utility on their behalf. Communication between consumers and concentrators is bidirectional. The concentrator receives tariffs from the utility agent.

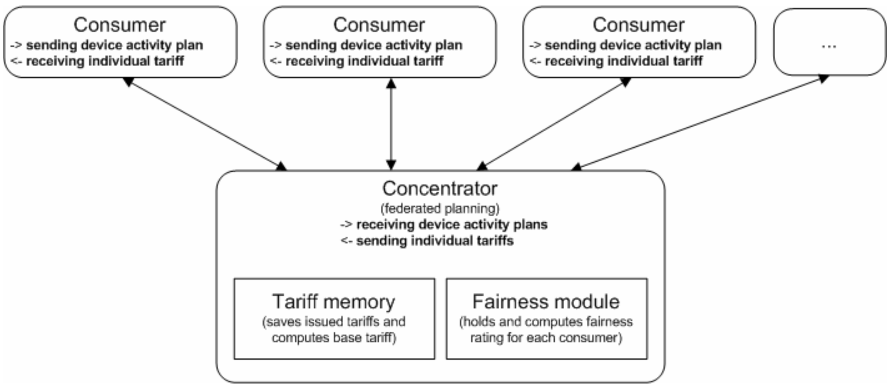

Figure 6. Integrating the concentrator

(1) Just as for the probability based approach customers agree that coordinated planning of device activity results in better overall pricing conditions and is

therefore worthwhile even if temporary disadvantages in accessibility to low priced tariff time slices can occur for single customers.

Within a single negotiating round the following steps are executed:

(1) The utility issued tariff is used by each customer to schedule device activity independently of other consumers following mere personal benefits.
(2) The resulting local device activity plan is sent to the concentrator.
(3) After receiving all single device activity plans the concentrator builds an individual tariff for each consumer agent based on that device's activity plan and on the calculated signal strength.
(4) The concentrator sends the individual tariffs to its associated consumers.
(5) Finally, based on their special tariffs the customer reschedule their device activity and provide a load prognosis for feedback needed by the utility.

In detail the segment wise calculation of the individual tariff by the concentrator depends on the signal strength, the consumer's reaction strength and a rating of fairness. This can be found in [Andreßen 2007]. First results for this scenario are currently evaluated.

4 Conclusion and Further Work

We presented a flexibly configurable agent-based tool for modelling tariff based load shifting scenarios. It has already been extended to scenarios involving direct control of devices and can be easily extended to distinct market based scenarios for demand side management within the electricity domain. It still has to be tested whether the tool can also successfully be applied to modelling and simulation problems within other domains.

We showed that different inter-agent negotiation scenarios can be realised. The concept of thick agents extensible through function modules has proven particularly useful. The novel architecture of ACDC allows for introduction of completely new agents simply by coding the functionality using the tool's class framework without changing existing code and changing XML configuration files.

The downside of the flexible approach is the complexity of our tool. Together with the multiple configuration options it results in a long learning periods for both, users and developers.

The tool is currently used for carrying out analyses of load shifting potentials described in the introduction. Before the tool will be used in additional scenarios, e.g. in load shifting under market influences including settings with multiple utilities and combined producers / consumers, we will improve its runtime efficiency. Further possible improvements include the introduction of an extended model checker, providing a better guidance when modelling new scenarios, a module for time series analysis and a context-sensitive editor with auto-completion for reducing the time required to construct new orchestration protocols and agent factory definitions.

References

St. Andreßen: Entwicklung und Analyse von Kommunikationsszenarien zur Koordination der Stromnachfrage unter dynamischen Stromtarifen. Diploma Thesis, University of Oldenburg, Dept. of Computing Science, 2007

H. Bergsten : Java Server Pages, O'Reilly & Assosiates Inc. 2002, 2nd ed.

P. Delisle (ed.): JavaServer Pages Standard Tag Library, SUN Microsystems, Inc., 2002

R. Johnson et al.: The Spring Framework - Reference Documentation, Version 2.0.7

North, M.J., Collier, N.T. and Vos, J.R. (2006): Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit. ACM Transactions on Modeling and Computer Simulation, Vol. 16, Issue 1, pp. 1-25

Y. Penya: Optimal Allocation and Scheduling of Demands in Deregulated Markets, PhD Thesis, Technical University of Vienna, Austria 2006

Rölke, H. (2004): Modellierung von Agenten und Multiagentensystemen - Grundlagen und Anwendungen. In Moldt, Daniel (ed.): Agent Technology: Theory and Application, Vol 2. Logos Verlag, Berlin, Germany

M. Sonnenschein, M. Stadler, B. Rapp, J. Bremer, St. Brunhorn: A Modeling and Simultion Environment for Real-Time Pricing Scenarios in Energy Markets. In: Tochtermann, K., Scharl, A. (Eds.): Informatics for Environmental Protection - EnviroInfo Graz 2006.

M. Stadler, W. Krause, M. Sonnenschein, U. Vogel: The Adaptive Fridge – Comparing different control schemes for enhancing load shifting of electricity demand. In: Hryniewicz, O., Studzinski, J., Romaniuk, M. (Eds.): 21st Conference on Informatics for Environmental Protection - Enviroinfo Warsaw 2007

H. F. Wedde, S. Lehnho, E. Handschin, O. Krause: Autonomous Real-Time Management of Unpredictable Power Needs and Supply. ECRTS'06, IEEE Computer Society, 2006.

F. Ygge, H. Akkermans: Power Load Management as a Computational Market. Proceedings of the 2nd International Conference on Multi-Agent Systems ICMAS'96, Kyoto, Japan, 1996

F. Ygge: Market Oriented Programming and its Application to Power Load Management. PhD Thesis, Department of Computer Science, Lund University, 1998.