Tools for Teaching Demand-Side Management

Jörg Bremer, Barbara Rapp, Frank Jellinghaus, Michael Sonnenschein
Carl von Ossietzky Universität Oldenburg, Department für Informatik
Ammerländer Heerstraße 114-118, D-26111 Oldenburg
{ joerg.bremer|barbara.rapp| frank.jellinghaus|sonnenschein}@informatik.uni-oldenburg.de

Abstract

Problem solving in demand-side management from an information technological point of view demands for a variety of interdisciplinary capabilities. Additionally to basic optimisation techniques a deeper knowledge of operating modes of electrical devices and modelling of their energy flows are required for building up proper models for optimisation. In the course of a lecture oriented cooperation of three universities two software tools have been developed in order to support students in getting acquainted with complex lecture topics on their own and by using examples with practical contexts. In addition, these tools double as a demonstration means for the lecturers. This paper describes how these tools cover the aspects of optimisation techniques as well as the impacts of mathematically modelling electrical devices and their application to teaching scenarios.

1. Introduction

Since autumn 2007 the e-learning course decentralised energy systems (German: Dezentrale Energiesysteme) has been offered by a lecture oriented cooperation of the Technical University at Brunswick, the Clausthal University of Technology and the Carl von Ossietzky University of Oldenburg (Kurrat et al. 2009). The course is part of the eLearning Academic Network (ELAN) of Lower Saxony (German: ELAN III-Initiative des Landes Niedersachsen, http://www.elan-niedersachsen.de/).

The blended course decentralised energy systems deals with the operation of electric power systems. It covers the processes of electricity generation, electric power transmission and electricity distribution. Build on these basics, changes in generating electricity, regulatory framework and business environment are covered. Among others, the course focuses on increased use of information technology for the enhancement of business processes in energy economics and its surrounding conditions. From a technical point of view, increasing decentralisation leads to new concepts for managing the operation of distributed devices in order to maintain grid-stability. Usually, demand-side management (DSM) is concerned with managing the consumption of energy by shifting load in time (Stadler 2005). But, increasing decentralisation in production, e.g. co-generation at home (Jungbluth 2006; Becker 2006), leads to a more holistic management of consumption and production at the same time. These management topics are contributed to the lecture oriented cooperation by the environmental informatics department in Oldenburg.

2. Problem Statement

The audiences of the lecture are students from different fields of science. At the University of Oldenburg the course is part of the curriculum in physics and computing science, whereas in Brunswick it is part of electrical or industrial engineering and in Clausthal this lecture is offered to students in energy systems engineering and also (industrial) engineering. The contribution to DSM from the computing science department in Oldenburg deals among others with different mathematical models for optimisation and simulation. In earlier terms exercises have been presented and students had to solve tasks by giving a solution as an implementation in any programming language or pseudo-code in order to further their skills and

In: V. Wohlgemuth, B. Page, K. Voigt (Eds.): Environmental Informatics and Industrial Environmental Protection - 23rd International Conference on Informatics for Environmental Protection. Shaker Verlag, Vol. 1, ISBN 978-3-8322-8397-1, pp. 455-463.

knowledge. Since not all students take classes in computing science, they often fail to get to the core of the matter, because low programming skills lead to time consuming attempts in getting a solution articulated as a formal description. Experience has shown a lack of practicing such skills from students outside computing science. The same seems to hold true for practicing mathematics in real-world problems outside engineering study courses. Therefore, two software tools have been developed to support students in focussing on the issues of DSM rather than on programming skills. In this way, students from a wide variety of study topics can compensate their previous knowledge and different skills. These tools address the related but different topics of optimisation and device modelling and are discussed in greater detail in the following sections.

3. DesylX – A Tool for Experiments with Optimisation in DSM

In order to give students a better understanding of optimisation problems, their practical application to real-world DSM problems and their solution, a specialised experimentation platform has been developed. This software has been named DesyIX. The main purpose is to provide a means for executing experiments in optimising the operation of a set of electricity consuming and/or producing devices in order to match a given target load profile. Here, the teaching goal is to give students not only the opportunity to experiment with parameterisation of an already implemented algorithm but also to alter or even re-implement essential parts of the latter. Because the audience not only consists of computing science students our tool provides the complete framework for parameterisation and running an optimisation algorithm for a given scenario as well as for scenario specification and result presentation. Additionally, an editor and interpreter for script based implementations of e.g. new objective functions are integrated. Currently, the system supports tabu search as optimisation technique, but the architecture allows for integrating further algorithms.

3.1 Model and Architecture

Optimisation in DSM usually aims at planning or altering the operation of electrical devices in a way that consumption and/or production of electricity is shifted in time. In practice this shifting always involves storage, e.g. a charged storage allows for a longer delay, an empty one for preponed operation. Since this tool primarily focuses on teaching discrete optimisation techniques, storages are involved only implicitly and are not explicitly modelled. For this reason the model and the problem are defined as follows:

Each scenario contains at least a target load profile, a not empty set of jobs (devices in conjunction with operating characteristics) and specific scenario parameters. Internally, each job j is a 4-tuple and defined as $j = (P(t), O(t), C, W)$. $P(t)$ represents the power at time t after turning-on a device. $P(t) < 0$ denotes negative power (generation of electricity) and $P(t) >= 0$ characterises positive power (energy consumption) respectively. The operational mode $O(t)$ can be zero or one, which represents the modes off and on. Because time in our model is discrete, the operations are represented by one binary sequence.

The constraints that restrict the number of respective switching actions are given by the set C, representing the minimum and maximum respectively of operation time after turning-on a device, non-operation time after switching-off a device as well as the number of turn-on actions. W represents a time slot for operations to take place. In this context, the slot characterises the earliest point in time a turn-on action and the latest point in time a switch-off action can occur.

Once the scenario is defined, the objective is to find a sequence of operational modes for each job such that the similarity between the preset target load profile and the resulting load profile is maximised subject to the satisfaction of each constraint. The system currently supports solving the problem by tabu search (for a good introduction see Glover and Laguna 2001). Tabu search is a local search based technique which incrementally improves a given best solution by exploring the neighbourhood for better solutions.

The pre-implemented neighbourhood definition makes use of the hamming distance, i.e. every solution with one job altered in operation is said to be in the neighbourhood of the previous solution, if the new binary operation string of this job has a hamming distance less or equal to a given limit.

The software DesyIX is written completely in Java in order to facilitate scripting features for more complex parameterisations. Moreover, Java is part of the computing science curriculum. In this way, it eases further student work on this tool. Figure 1 shows the architecture of optimisation scenarios and the options for individualisation.

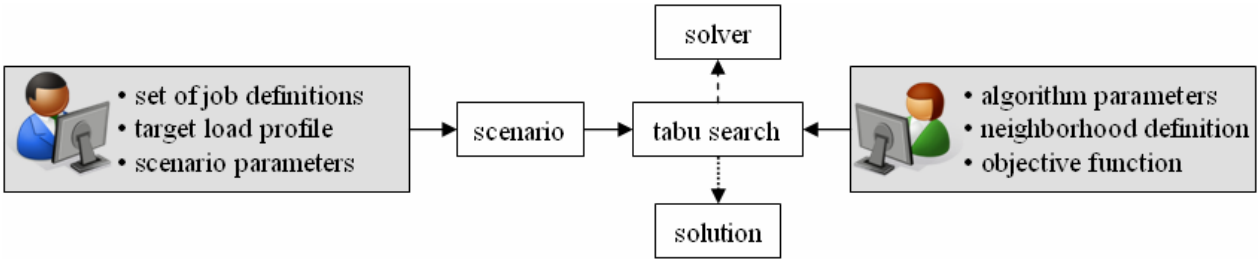

Figure 1: Options for parameterisation of optimisation scenarios in DesyIX.

The components listed in the outer boxes in figure 1 can be specified by the user. Before running an optimisation, the model has to be parameterised. This preparation might optionally include setting global parameters like time horizon and resolution, but first and foremost a target load profile and a set of jobs have to be specified.

The target load profile reflects the amount of load that is demanded at certain points in time (assumed to be in the future) by a utility. Although all algorithms in our system are supposed to work in a discrete way such a load profile has to be defined in terms of a continuous function of time. Thus it becomes possible to support different grains of discretisation. The same procedure applies for defining the characteristic load profile of a device which itself is modelled in form of a job (see figure 2). Additionally, for each job the prior mentioned constraints may be specified. The scenario parameters consist for example of details about width and number of time intervals and must be configured afore specifying the algorithm.

These optimisation scenario (problem) definitions are usually specified by the instructor and are thus part of an exercise. After completing the scenario specification and therefore setting up the problem, the student can start the algorithm to solve the problem. In order to allow for a quick start, pre-implementations for all subroutines of an algorithm are provided. For advanced learning goals it is possible to add own implementations to selected parts.

Here, the objective of optimisations is usually the minimisation of the dissimilarity between the preset target load profile and the load profile resulting from the device operations according to a given solution. It is the task of the objective function to evaluate a solution by giving a similarity measure for two load profiles. By default the system provides a least squares measure. In order to enable students to add own implementations of further measures, a built-in editor allows for writing JavaScript based objective functions. These functions are then used directly within the optimisation. Due to the interpreted code the execution time grows noticeably. But on the other hand this allows for implementing different functions with low or even without experience in programming.

New neighbourhood definitions for the representation of the optimisation problem currently can only be added as compiled Java classes. In this field the next improvements of the system are going to be made.

Finally, for each optimisation run the algorithm can be parameterised individually in order to observe the impact on the result as well as on the process itself. For tabu search, these parameters contain for example the size of the neighbourhood, whether or not to make use of the aspiration criterion or the number of iterations.

3.2 User Interface and Workflow

Figure 2 shows a screenshot of the job specification perspective of DesyIX. A subdivision of the main window in three different perspectives for parameterising, running the optimisation process and result presentation reflects the intended workflow for a student working with the tool. Starting from a new (empty) scenario, first of all a target load profile has to be chosen from a list of predefined profiles. Alternatively, a new profile might be specified by the user. There are currently two ways of defining a function (apart from using predefined ones) for load profiles. One way is to write the function directly with a scripting language within the system. At present, JavaScript is supported for this purpose. In order not to slow down the optimisation process too much by repeatingly interpreting the script, such a function is discretised once as a first step prior to starting the optimisation. The other way is to define the target load profile as an external file of character separated values. Such a file can be loaded into the system and is translated automatically into a function. For this purpose the user may currently choose between two interpolation methods: Spline and Neville interpolation. A graph depicting the interpolation results (implemented within an import wizard) together with the original data points serves as a visual feedback. Thus the student can decide which method fits best. Afterwards the data is treated as a function as described above. In an analogous manner an objective function must be added to the scenario. After specifying at least one job and adding it to the scenario, the student will be able to switch to the second perspective in order to start optimisation runs.

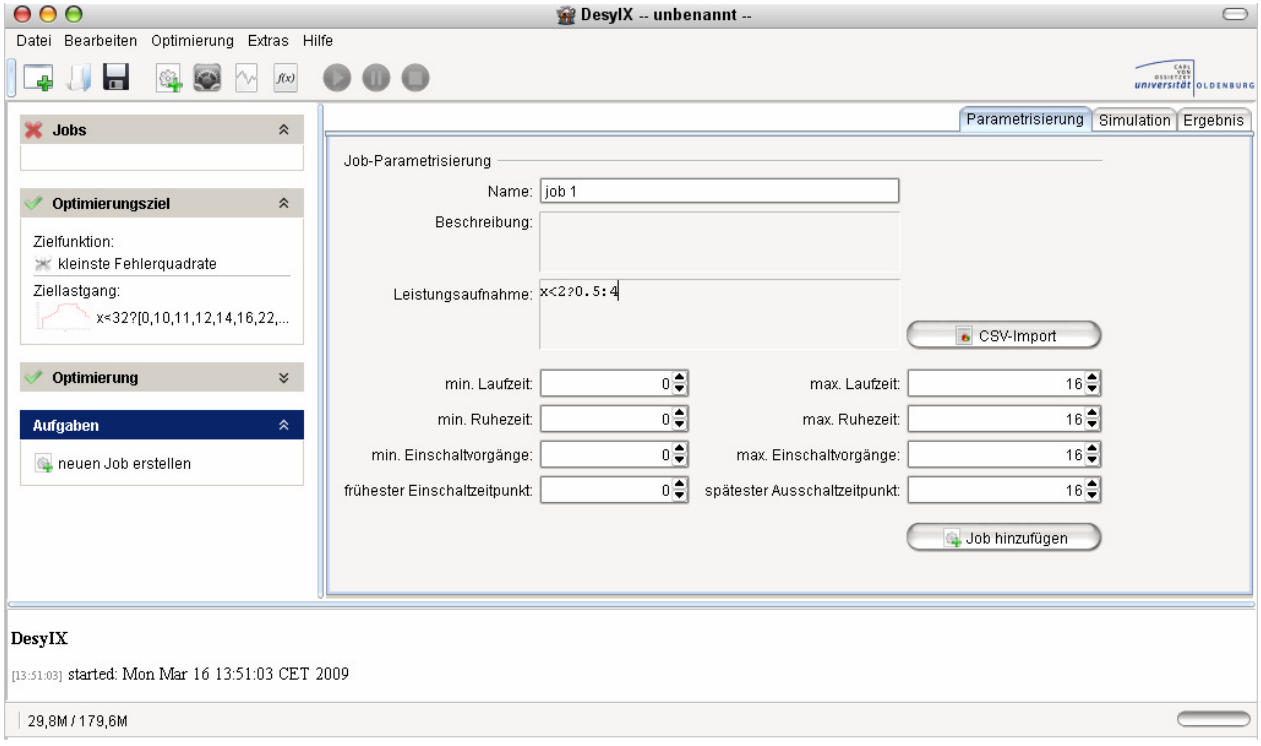

Figure 2: Job definition with DesyIX.

While running the incremental optimisation process the user is concurrently informed not only about progress, e.g. how many iterations to go: In the course of the optimisation process the system gives information about the currently achieved solution quality and the load profile resulting from the current solution as well. The progression of the improving solution quality is shown as a graph for the best solutions

compared with the quality of the currently scrutinised solution. Thus the student is able to recognise how tabu search temporarily comprises deteriorating solutions in order to escape local minima. At the same time the resulting load profiles of the solutions are presented together with the earlier best solutions, with the latter fainting in colour over time in order to maintain clarity of the chart.

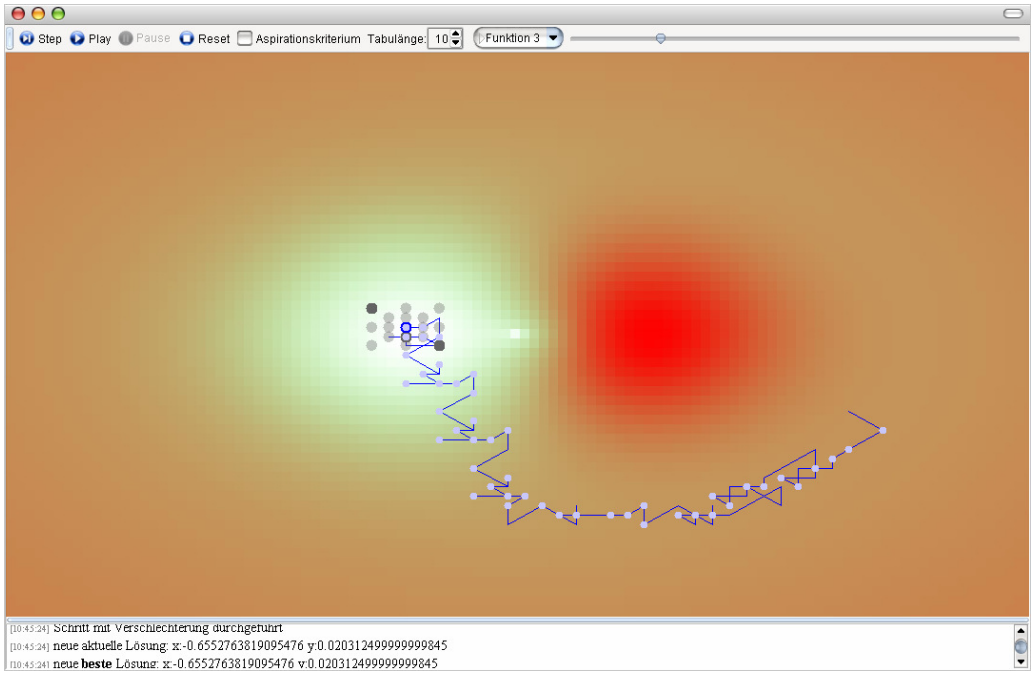

Figure 3: Demonstrating tabu search on a simple two dimensional error function.

While running the optimisation the evolution of the solutions is presented. After termination a third perspective for result presentation is provided. Here the student may compare the results of different runs as well as retrace the iterative improvement of the solutions. Apart from demand-side management scenarios, a simple tabu search demonstration widget (shown in figure 3) has been integrated. For easier visualisation it demonstrates the operation of tabu search while searching for a minima on a two dimensional function.

4. ELVisDSMP

ELVisDSMP (**e-l**earning platform for **vis**ualising **d**emand-**s**ide **m**anagement **p**otentials) is an e-learning environment for modelling the operational modes of several electrical devices. The educational goal of this platform is to demonstrate how changing device specific parameters results in changes of storage capacities over time. The definition of storage capacity is device dependant; for a fridge it is determined by the time until an item inside warms up to a specific temperature while the device is unplugged from electric power (Stadler 2005; Stadler et al. 2009). Time varying storage charging is determined by several device specific parameters. By charging when electric power is highly available the device can be used for demand-side management, because it can be unplugged at times when there is a shortage of electricity.

4.1 Model and Architecture

The architecture of ELVisDSMP, as shown in figure 4, is separated into consumer and storage models. The devices are categorised by the main purpose they are used for. Due to the fact that a fridge and a ventilation system are typically used as consumption devices and not because of their storage characteristics, a thermal buffer storage is used for storing hot water and thus for storing energy. After starting the platform, a simulation window can be opened for each integrated model. Additionally, a help window is accessible for support on the basic platform usage.

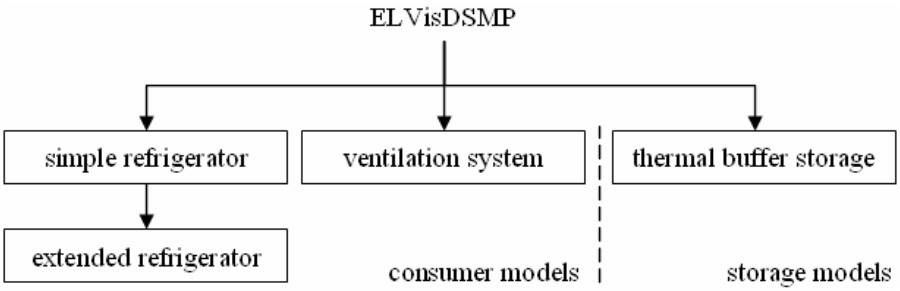

Figure 4: ELVisDSMP Architecture.

The simulation environments for different models are decoupled from the other. Therefore, no platform-wide functionality has to be implemented and new models can easily be added to the platform. For integrating a simulation environment for a new model a reference implementation has been designed, which gears especially to a didactical concept derived from Niegemann et al. (2004) and Schulmeister (1997). This concept includes a specific arrangement of input fields on the user interface, giving a formal description of storage capacities and highlighting a parameter whenever an input field is selected. Specifying a reference model to enable direct simulation facility is also part of the concept. Further parts of this concept demand enhancements of visual elements with textual descriptions and feedback while a simulation is performed. On the other hand, the amount of feedback has to be appropriate in order not to overstrain the student. As the tool is intended for follow-up course work at home, it has to be as self-explanatory as possible. If any given feedback is not easily understandable, it will be of no avail.

4.2 User Interface and Workflow

When starting the application, the main window is shown giving access to open a separate simulation window for each implemented model. With opening such a simulation window, a standard configuration of the respective model is loaded. In this way, a simulation run can be performed immediately without having to concern about specifying a parameterisation first. Thus the student is given a sense of achievement which eases and enforces further working with the tool.

In order to give an example for the common work flow, figure 5 shows the user interface for modelling a fridge. In the top right corner, the formal representation of such a device is displayed using a HTML visualisation. The input form in the top left corner serves for altering parameters in the formal model. Each input field is linked with the respective part of the formula. Thus a case-sensitive colour highlighting is used within the HTML representation whenever the student uses the respective input field.

Several device parameters can be adapted by the student while working on given exercises, e.g. the size of the wall and heat conductivity as well as the parameters of reefer cargo, like their mass, surface area or

heat-transfer coefficient. By giving the formula and highlighting the specified parameters their impacts on the results are demonstrated.

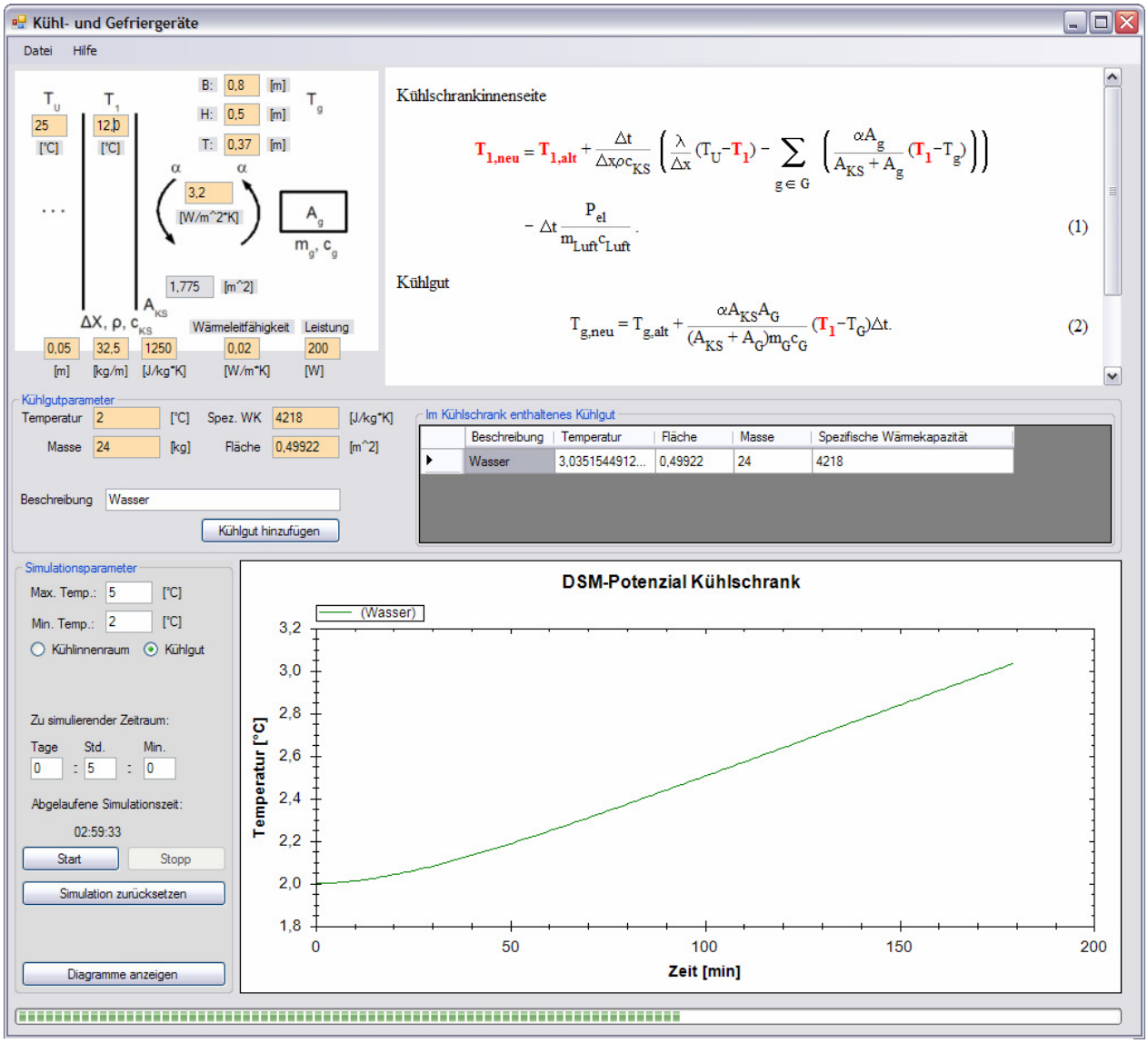

Figure 5: View for parameterisation of a fridge model.

After finishing the formal specification of the model, the simulation parameters are to be specified in the bottom left part of the user interface. With starting a simulation, results like the temperature curve of reefer cargo are printed concurrently to a diagram at the bottom right part of the interface. During a simulation run, further results are calculated and logged to the file system. They can be analysed in separate diagrams afterwards, but might also serve as input for further calculations. For example, at each single simulation step, the times for charging a device and for possible delays are calculated. The advantage of splitting up output visualisation is not to overstrain a learner with giving too much information at the same time.

The fridge described above is a model at a rather high level of abstraction. It is extended by a more complex one, which additionally includes heat exchange while opening the door. The complexity of ex-

tending a model is not predetermined. This task might vary from simply just alter the basic calculation algorithm for a model considering just a few more device parameters, up to involving a complete redesign of the user interface.

Moreover, several other models are implemented. The potentials for demand-side management of a ventilation system can be simulated as well as those of a thermal buffer storage as shown in figure 4. The user interface design of these models is similar to the one shown in figure 5. This as well as the parameter-highlighting and the way of visualising the results is part of a didactical concept based on theories about learning, like the Decision Oriented Instructional Design Model (Niegemann et al. 2004). Other parts of this concept are for example placing input boxes near to the corresponding parameter in the schematic diagram or giving reference specifications of a model. In this way, a learner is able to start a simulation immediately and is given a direct feedback by dynamically displayed results during a simulation run.

ELVisDSMP is based on the .NET-Technology and implemented in the C# programming language. For dynamically visualising the results during a simulation, the application is multithreaded so the user can still interact while a simulation is running. For depicting results in the time domain, the external library ZedGraph (http://zedgraph.org/) is used, which is also a C# implementation.

5. Application to Teaching

The goal we wanted to achieve with these tools is to teach DSM related optimisation not in a mere mathematical way. We believe that a holistic perspective is necessary for a thoroughly understanding of not only the underlying concepts but also of their application to practical problems. Creative problem solving should always involve the conscious choice of different thinking languages like mathematics or visualisation (Adams 2001). With these tools we want to enable our students to explore demand-side management problems as well as related solving techniques on their own and from different points of view. That way, our students may interactively experience the consequences of altering certain problem parameters whenever they want to reinforce their understanding during follow-up course work.

The module decentralised energy systems consists of a distant lecture and associated tutorials which have to be attended by the students in a blended learning approach. Since last winter term, the tool DesyIX is in action as a means for the students' preparation of the tutorials. Exercises which have to be solved with the aid of the tools are handed out in advance. Such exercises include for example the conversion of colloquial job descriptions into a formal model or the implementation of specialised electricity economics measures like e.g. excess supply (Born 2001).

Therefore, scrutinising the impact of different neighbourhood definitions, objective functions, etc. on result quality or converging speed together with an optional training of programming skills will be the main purpose along with providing a demonstration means for load shifting problems.

The second tool ELVisDSMP will be in action from the next winter term on. It is supposed to substitute for exercises like implementing models for certain devices in order to fulfil the actual task which needs an appropriate simulation of the device first. Such device models are often given in terms of differential equations. Thus such exercises often degraded to questions like how to implement such equations with a programming language in the past. Our hope is that with the new tool students will be able to focus more on working with simulated devices and exploring DSM potentials instead of mainly struggling with building up the simulation.

Both tools are realised as stand-alone applications with no further requirements. In consequence both tools are applicable not only in distant blended learning scenarios but also as auxiliary means in conventional lectures.

6. Conclusion and further work

The next step will be to intensify the use of these tools within tutorials and exercises. Whereas the tabu search tool has already been in action for one term, as yet, there is no teaching experience with the other one. Thus today it is impossible to evaluate the possible gain imposed by the tools.

While the second tool, ElVisDSMP, was realised in the course of a master thesis (Jellinghaus 2009), the development of DesyIX was made possible by project founding. The whole realisation of the latter one roughly took about four man months. Therefore it is questionable if the work would have been done merely by means of the department without knowledge of usefulness. In this sense, it is now demanded to scrutinise the advantages in learning for our students and their comprehension of complex DSM topics in order to properly balance the ratio of gain and cost.

Nevertheless we are already thinking of further extensions. For example, the job operation model allows for an integration of solvers that work with genetic algorithms. ELVisDSMP will be extended by further device models. Other modes of visualisation are applicable for this tool as well, since these are currently limited to two-dimensional models.

With these additional teaching topics the tools will probably be applicable also to further computing science courses, especially in the field of environmental informatics.

Bibliography

Adams, J. L. (2001): Conceptual Blockbusting. Perseus Books, New York, 4th ed. 2001.

Becker, R. (2006): Optimierung thermischer Systeme in dezentralen Energieversorgungsanlagen. Habilitation, Universität Dortmund, Mai 2006.

Born, F. J. (2001): Aiding Renewable Energy Integration through Complimentary Demand-Supply Matching. PhD Thesis, University of Strathclyde, Energy Systems Research Unit, 2001.

Glover, F., Laguna, M. (2001): Tabu Search. Kluwer Academic Publishers, Boston, 4th ed. 2001.

Jellinghaus, F. (2009): Entwicklung einer E-Learning-Plattform zur Veranschaulichung der Potenziale im Demand Side Management durch Visualisierung und Simulation mathematischer Modelle. Master Thesis, University of Oldenburg, Department of Computing Science, March 2009.

Jungbluth, C. H. (2006): Kraft-Wärme-Kopplung mit Brennstoffzellen in Wohngebäuden im zukünftigen Energiesystem. Schriften des Forschungszentrums Jühlich, Reihe Energietechnik, Band 59, 2006.

Kurrat, M., Deppe, B., Beck, H.-P., Mbuy, A., Wehrmann, E.-A., Sonnenschein, M., Appelrath, H.-J., Bremer, J., Rapp, B. (2009): Interdisziplinäre Wissensvermittlung am Beispiel dezentraler Energiesysteme – Ein Erfahrungsbericht, in: Appelrath, H.-J., Schulze, L.: Auf dem Weg zu exzellentem E-Learning. Vernetzung und Kooperation der Hochschullehre in Niedersachsen. Waxmann, Münster., 2009.

Niegemann, H. M., Hessel, S., Hochscheid-Mauel, D., Aslanski, K., Deimann, M., Kreuzberger, G. (2004): Kompendium E-Learning. Springer Verlag, 2004.

Schulmeister, R. (1997): Grundlagen hypermedialer Lernsysteme. R. Oldenbourg Verlag, 2. Auflage, 1997.

Stadler, I. (2005): Demand Response – Nichtelektrische Speicher für Elektrizitätsversorgungssysteme mit hohem Anteil erneuerbarer Energien. Habilitation, Fachbereich Elektrotechnik der Universität Kassel, Oktober 2005.

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 and Software 24, pp. 285–295.