

ecoControl: Entwurf und Implementierung einer Software zur Optimierung heterogener Energiesysteme in Mehrfamilienhäusern

Eva-Maria Herbst, Fabian Maschler, Fabio Niephaus,
Max Reimann, Julia Steier, Tim Felgentreff, Jens Lincke,
Marcel Taeumel, Robert Hirschfeld, Carsten Witt

Technische Berichte Nr. 93

des Hasso-Plattner-Instituts für
Softwaresystemtechnik
an der Universität Potsdam



Technische Berichte des Hasso-Plattner-Instituts für
Softwaresystemtechnik an der Universität Potsdam

Eva-Maria Herbst | Fabian Maschler | Fabio Niephaus | Max Reimann |
Julia Steier | Tim Felgentreff | Jens Lincke | Marcel Taeumel |
Robert Hirschfeld | Carsten Witt

ecoControl

Entwurf und Implementierung einer Software
zur Optimierung heterogener Energiesysteme in Mehrfamilienhäusern

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.dnb.de/> abrufbar.

Universitätsverlag Potsdam 2015

<http://verlag.ub.uni-potsdam.de/>

Am Neuen Palais 10, 14469 Potsdam

Tel.: +49 (0)331 977 2533 / Fax: 2292

E-Mail: verlag@uni-potsdam.de

Die Schriftenreihe **Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam** wird herausgegeben von den Professoren des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam.

ISSN (print) 1613-5652

ISSN (online) 2191-1665

Das Manuskript ist urheberrechtlich geschützt.

Online veröffentlicht auf dem Publikationsserver der Universität Potsdam

URL <http://publishup.uni-potsdam.de/opus4-ubp/frontdoor/index/index/docId/7214>

URN <urn:nbn:de:kobv:517-opus4-72147>

<http://nbn-resolving.de/urn:nbn:de:kobv:517-opus4-72147>

Zugleich gedruckt erschienen im Universitätsverlag Potsdam:

ISBN 978-3-86956-318-3

Vorwort

Eine dezentrale Energieversorgung ist ein erster Schritt in Richtung Energiewende. Dabei werden auch in Mehrfamilienhäusern vermehrt verschiedene Strom- und Wärmeerzeuger eingesetzt. Besonders in Deutschland werden in diesem Zusammenhang Blockheizkraftwerke immer häufiger genutzt, weil sie Gas sehr effizient in Strom und Wärme umwandeln können. Außerdem ermöglichen sie, im Zusammenspiel mit anderen Energiesystemen wie beispielsweise Photovoltaik-Anlagen, eine kontinuierliche und dezentrale Energieversorgung.

Bei dem Betrieb von unterschiedlichen Energiesystemen ist es wünschenswert, dass die Systeme aufeinander abgestimmt arbeiten. Allerdings ist es bisher schwierig, heterogene Energiesysteme effizient miteinander zu betreiben. Dadurch bleiben Einsparungspotentiale ungenutzt. Eine zentrale Steuerung kann deshalb die Effizienz des Gesamtsystems verbessern.

Mit `ecoControl` stellen wir einen erweiterbaren Prototypen vor, der die Kooperation von Energiesystemen optimiert und Umweltfaktoren miteinbezieht. Dazu stellt die Software eine einheitliche Bedienungsoberfläche zur Konfiguration aller Systeme zur Verfügung. Außerdem bietet sie die Möglichkeit, Optimierungsalgorithmen mit Hilfe einer Programmierschnittstelle zu entwickeln, zu testen und auszuführen. Innerhalb solcher Algorithmen können von `ecoControl` bereitgestellte Vorhersagen genutzt werden. Diese Vorhersagen basieren auf dem individuellen Verhalten von jedem Energiesystem, Wettervorhersagen und auf Prognosen des Energieverbrauchs. Mithilfe einer Simulation können Techniker unterschiedliche Konfigurationen und Optimierungen sofort ausprobieren, ohne diese über einen langen Zeitraum an realen Geräten testen zu müssen. `ecoControl` hilft darüber hinaus auch Hausverwaltungen und Vermietern bei der Verwaltung und Analyse der Energiekosten.

Wir haben anhand von Fallbeispielen gezeigt, dass Optimierungsalgorithmen, welche die Nutzung von Wärmespeichern verbessern, die Effizienz des Gesamtsystems erheblich verbessern können. Schließlich kommen wir zu dem Schluss, dass `ecoControl` in einem nächsten Schritt unter echten Bedingungen getestet werden muss, sobald eine geeignete Hardwarekomponente verfügbar ist. Über diese Schnittstelle werden die Messwerte an `ecoControl` gesendet und Steuersignale an die Geräte weitergeleitet.

Januar 2015

Eva-Maria Herbst
Fabian Maschler
Fabio Niephaus
Max Reimann
Julia Steier

Marcel Taeumel
Jens Lincke
Tim Felgentreff
Robert Hirschfeld
Carsten Witt

Inhaltsverzeichnis

Vorwort	v
1 Simulation von Energiesystemen	1
1.1 Motivation	1
1.2 Simulationsframework	2
1.3 Energiesystem-Szenario	4
1.4 Simulation der Geräte	5
1.5 Validierung und Verifikation	19
1.6 Zusammenfassung	25
2 Nutzerstudien für die Entwicklung von ecoControl	27
2.1 Zielsetzung	27
2.2 Methoden	28
2.3 Vorgehensweise	32
2.4 Ergebnisse	35
2.5 Qualität der Ergebnisse	54
2.6 Zusammenfassung	55
3 Entwicklung von ecoControl	57
3.1 Anwendungsgebiet von ecoControl	57
3.2 Architektur von ecoControl	58
3.3 Die Kommunikation zwischen Server und Client	68
3.4 ecoControl als Open Source Software	70
3.5 Nächste Schritte	70
3.6 Prozess	72
3.7 Zusammenfassung	77
4 Prognose und Optimierung von Energiesystemen	79
4.1 Einleitung	79
4.2 Analyse	80
4.3 Prognose des elektrischen Lastverhaltens	83
4.4 Automatisierte Prognosen	89
4.5 Kostenoptimierung mittels Prognosen	95
4.6 Evaluation	99
4.7 Zusammenfassung	105

5	Evaluation und Diskussion von Optimierungsalgorithmen	109
5.1	Motivation	109
5.2	Fallbeispiel	110
5.3	Verallgemeinerung des Ergebnisses	120
5.4	ecoControl zur Optimierung von Energiesystemen	124
5.5	Zusammenfassung	134
	Schlussbetrachtung	137

1 Funktionsweise und Simulation von heterogenen Energiesystemen in Mehrfamilienhäusern

Die Entscheidung zur Energiewende in Deutschland macht viele unterschiedliche Formen der Energiebereitstellung notwendig. Blockheizkraftwerke und dazugehörige Geräte wie Spitzenlastkessel und Wärmespeicher bieten dafür eine effiziente Möglichkeit. Diese komplexeren werdenden Energiesysteme erfordern eine effiziente Steuerung, damit die Energie zur richtigen Zeit verfügbar ist. Abhängigkeiten zwischen verschiedenen Energiesystemen können ausgenutzt werden, um Energie zu sparen und Überproduktion zu vermeiden. Es ist jedoch schwer verschiedene Steuerungskonfigurationen zu testen, da die Auswirkungen erst nach einiger Zeit sichtbar werden und nicht im Voraus analysiert werden können. Eine Simulation der Energiesysteme kann dieses Problem lösen.

Wir haben eine Simulation für das Verhalten der Energiesysteme und den Energieverbrauch eines Mehrfamilienhauses erstellt. Die Abhängigkeiten eines solchen Systems haben wir iterativ analysiert und anschließend vereinfacht in Python simuliert. Die zukünftigen Auswirkungen einer geänderten Steuerung können wir mit der Simulation, in unter einer Sekunde, visualisieren. Dadurch kann sie für Vorhersagen genutzt werden, welche den Betreiber der Energiesysteme bei der Optimierung unterstützen. Mit `ecoControl` können wir so die beste Konfiguration finden, bevor reale Geräte entsprechend angesteuert werden. Die effiziente Steuerung der immer komplexeren Energiesysteme wird dadurch vereinfacht.

1.1 Motivation

Bei der Entwicklung von `ecoControl` mussten wir sicherstellen, dass die Veränderungen des zu steuernden Systems in der Realität auch die gewünschten Ergebnisse erzielt. Problematisch war, dass wir kein reales System zur Verfügung hatten, um die Auswirkungen der Steuerung von `ecoControl` zu untersuchen. Zur Lösung dieses Problems haben wir anstelle des echten Energiesystems eine Simulation genutzt, um die Software zu testen. Dadurch haben wir den Vorteil, dass die Ergebnisse der Software zeitunabhängig und damit schneller überprüft werden können. Für die Simulation der Energiesysteme entwickelten wir zunächst ein Modell, das die Wirklichkeit hinreichend genau abbildet. Den Grad der Abstraktion haben wir entsprechend der Genauigkeit der Simulationsergebnisse fortlaufend angepasst.

In Abschnitt 1.2 wird zunächst das allgemeine Prinzip der Simulation mit Hilfe des Frameworks `SimPy` erläutert. Anschließend erfolgt in Abschnitt 1.3 eine generelle und in Abschnitt 1.4 eine detaillierte Betrachtung des realen Systems und der entwickelten Simulation. Schließlich wird in Abschnitt 1.5 diskutiert, ob die Simulation

die Realität adäquat abbildet. Abschnitt 1.6 bietet eine abschließende Beurteilung der Herangehensweise und der Simulation.

1.1.1 Informationsbeschaffung

Am Anfang mussten wir die Domäne kennenlernen. Dazu haben wir im Internet recherchiert, um Wissen über Heizungssysteme und verschiedene Geräte zur Energiebereitstellung zu erlangen. Besonders hilfreich zum Verständnis eines Blockheizkraftwerks (BHKW) erwies sich eine Planungsbeschreibung [18]. Auch die Betriebsanleitung eines BHKWs hat uns geholfen, einen Einblick in die Parameter eines solchen Geräts zu bekommen [33]. Weiterhin bietet die BHKW-Infothek eine detaillierte Beschreibung zu verschiedenen BHKWs [30].

Die Aufgaben der einzelnen Geräte konnten wir so im Wesentlichen erfassen, es blieben jedoch einige Fragen weiterhin offen. Wir wollten zum Beispiel klären, welche Abhängigkeiten zwischen den verschiedenen Geräten vorliegen. Weiterhin war unklar, welche Informationen zwischen den Geräten ausgetauscht werden müssen und welche Auswirkungen die Änderung verschiedener Parameter haben. In einem komplexen System, wie dem der Energiesysteme, ist dies nicht einfach zu überschauen.

1.1.2 Simulation zum besseren Verständnis

Nachdem wir die grundlegenden Funktionen verstanden hatten, fingen wir an das gesamte Energiesystem zu simulieren. Zunächst haben wir das Verhalten stark vereinfacht. Mit der Zeit haben wir mehr Wissen über die Domäne gesammelt und konnten die Simulation der Geräte verbessern. Als Vergleich dienten zum Beispiel Echtdaten eines ähnlichen Systems, mit dem auch in Abschnitt 1.5 die Qualität der Simulation evaluiert wird.

Bei der Recherche nach dem Verhalten eines Gerätes traten immer wieder Unklarheiten zu einzelnen Funktionalitäten der Geräte auf. Hierzu fehlte uns das Verständnis des Gesamtsystems. Als wir die Aufgabe jedes einzelnen Geräts für das Gesamtsystem verstanden hatten, konnten wir die einzelnen Geräte klar trennen. Dies erforderte Schnittstellen, die Auskunft geben, welche Informationen zwischen den Geräten ausgetauscht werden müssen. Bei der Untersuchung des internen Verhaltens eines Gerätes wurden die Informationen ersichtlich. Diese konnten wir dann bei der Entwicklung der Simulation des Geräts nutzen. In welcher Form sie angefragt oder zur Verfügung gestellt werden, wird in Abschnitt 1.2.2 beschrieben. Vorher unklare Funktionen verstanden wir besser in Verbindung mit einer anderen Funktionalität eines zweiten Gerätes.

1.2 Simulationsframework

Zunächst haben wir ohne die Hilfe eines Frameworks versucht eine Simulation zu erstellen. Dabei sind einige Probleme aufgetreten. Beispielfhaft seien hier die konsistenten Berechnungen der Zeit in der Simulation oder der Kommunikation ver-

schiedener Simulationsobjekte genannt. Einfacher wurde es, als wir das Simulationsframework *SimPy*¹ für *Python* nutzten. Nach kurzer Einarbeitungszeit haben wir durch die Struktur des Frameworks ein besseres Verständnis des Simulationsaufbaus bekommen. Zudem nutzten wir fertige Schnittstellen, statt zum Beispiel den Zeitverlauf innerhalb der Simulation selbst zu berechnen.

1.2.1 SimPy

Die grundlegende Komponente der Simulation ist die *Environment* Klasse. Objekte dieser Klasse enthalten die zu simulierenden Entitäten. Für jede zu simulierende Entität (BHKW, SLK, ...) muss lediglich ein Python-Objekt erstellt werden. Diese implementieren in einer Methode ihr Verhalten über den Zeitverlauf. Diese *step()*-Methode wird von einem *Process*, einem vordefinierten Objekt von SimPy, wiederholt aufgerufen. Diese wichtigste Methode der Simulation wird während der gesamten Laufzeit immer wieder aufgerufen und bildet den Verlauf der Zeit nach. Das ist möglich, da als Parameter die virtuell vergangene Zeit mitgeteilt wird und die Geräte darüber ihr Verhalten entsprechend steuern. Sie berechnen in der *step()*-Methode ihren neuen Zustand und damit das Verhalten über den Verlauf der Zeit.

Wenn die Simulation gestartet wird, beginnt die *Environment* Komponente die *step()*-Methoden in bestimmten Abständen für alle enthaltenen Entitäten aufzurufen. Damit wird der Zustand der Entitäten geändert. Den Zugriff auf den Zustand erlaubt die Komponente *Environment* ebenfalls pro Entität und speichert darüber hinaus den zeitlichen Verlauf der einzelnen Parameter jedes Objekts.

Das Zeitverhalten exakt zu beschreiben, ist entscheidend bei der dynamischen Simulation. Das mitgegebene Zeitintervall der *step()*-Methoden (*step_size*) ist in unserer Simulation standardmäßig 120 s. Dies kann als Genauigkeit bezeichnet werden. Je kleiner das Intervall ist, desto häufiger und damit genauer wird die Zustandsberechnung durchgeführt. Gleichzeitig erfordern die häufigeren Aufrufe mehr Rechenleistung.

In diesem Zeitintervall wird der kontinuierliche Verlauf eines echten Systems diskretisiert. Der Verlauf der Daten über die Zeit wird pro Zeitschritt integriert. In einem Beispiel der Domäne benötigt ein elektrisches Gerät (Verbraucher) 3000 W Leistung. Dies wird über die Zeit von 1 h zu 3 kWh integriert. Aus der benötigten Leistung wird so die Energiemenge für einen Zeitschritt errechnet.

$$\int_0^{1 \text{ h}} 3000 \text{ W dt} = 3000 \text{ Wh} = 3 \text{ kWh}$$

Weiterhin kann in Bezug auf das Zeitverhalten eingestellt werden, wie häufig die *step()*-Methoden aufgerufen wird. Damit lässt sich das simulierte Verhalten im Verhältnis zur Realität verzerren. Wird die *step()*-Methode im gleichen Zeitabstand aufgerufen, wie die Schrittgröße, wird das Verhalten gleichschnell wie ein echtes System simuliert. Häufig ist das Intervall jedoch kleiner, da so das Verhalten im

¹SimPy, <http://simpy.readthedocs.org/> (abgerufen am 19.11.2014).

Zeitraffer simuliert wird. Dies vermeidet, wie gewollt, lange Wartezeiten bei der Analyse geänderter Systemkonfigurationen.

Für die Kommunikation verschiedener Objekte einer Simulation implementiert die Komponente *Environment* in SimPy spezielle *Ressourcen*. Diese sind ebenfalls Entitäten in der Simulation und dienen dem Austausch von Informationen zwischen Geräten.

1.2.2 Anpassungen

Die vorgegebene Struktur von SimPy hat uns anfangs bei der Erstellung der Simulation geholfen. Zum Beispiel haben wir keine eigene Zeitberechnungen für Zustandsänderungen mehr benötigt, sondern die einheitliche Schnittstelle dafür genutzt. Als wir das Prinzip der Simulation verstanden hatten und das Verhalten der Geräte immer komplexer wurde, haben wir einige Anpassungen vorgenommen.

Um Aussagen über das Verhalten der Energiesysteme zu treffen, mussten wir häufig die Daten der Simulation eines längeren Zeitraums, zum Beispiel eines Jahres, speichern. Dazu haben wir den Speichermechanismus verändert, der den Zustand der Geräte speichert. Das Verhalten der Geräte wird nicht mehr nur in einer flüchtigen Datenstruktur im Hauptspeicher gehalten, sondern auch in einer Datenbank gesichert. Die neue Datenstruktur speichert die Daten zunächst zwischen und dann gesammelt, in regelmäßigen Abständen in der Datenbank, womit sie für weitere Analysen verfügbar sind.

An anderer Stelle konnten wir die Performance in den *step()*-Methoden verbessern. Für jeden Aufruf bei jedem Gerät wurden zusätzliche Objekte erstellt, um beispielsweise das Zeitintervall zu übermitteln. Da die Konfiguration von Zeitschritt und Geräten jedoch fest war, konnten wir die *step()*-Methoden problemlos vereinfachen. Alle Geräte wurden einheitlich durch nur noch ein Simulationsobjekt verwaltet. Auch die Kommunikation der Geräte haben wir auf gegenseitige Funktionsaufrufe beschränkt. Die *Ressourcen* von SimPy haben wir dabei nicht verwendet.

Mit der Zeit haben wir immer weniger Funktionalitäten des Frameworks genutzt. Schließlich haben wir die Abhängigkeit von der Bibliothek beseitigt und basierend auf den Prinzipien von SimPy eine eigene Simulation erstellt.

1.3 Energiesystem-Szenario

In Abbildung 1.1 ist vereinfacht dargestellt, wie die verschiedenen Energiesysteme zusammenarbeiten. Das Basis-Szenario besteht aus einem Blockheizkraftwerk, dem dazugehörigen Wärmespeicher und Spitzenlastkessel (SLK). Diese Geräte sind für den Betrieb eines BHKWs nötig und wurden durch den Projektpartner als zentrales Szenario festgelegt. Wenn in einem Mehrfamilienhaus ein BHKW installiert wird ist dies meist die Konstellation der Geräte. Einzig die Leistungsparameter müssen an das entsprechende Haus angepasst werden.

Durch gelbe Linien ist die Primärenergiequelle (Gas) angegeben. Rot bezeichnet den Fluss des warmen Wassers (Vorlauf) und blau den Rücklauf des kühleren Wassers. Die elektrische Energie wird durch schwarze Linien dargestellt.

1.4 Simulation der Geräte

Die wichtigste Abstraktion der Simulation ist die Energieeinheit. Jede Energiemenge wird in Kilowattstunden (kWh) angegeben. Dadurch wird die Kommunikation zwischen den Geräten erleichtert, da keine weitere Information zur Form der Energie (thermal, elektrisch, chemisch) mitgeteilt werden muss. Jedes Gerät kann intern die Form der Energie in die äquivalente Menge in kWh umrechnen.

Weiterhin werden beim Transfer von Energie keine Leitungsverluste berücksichtigt. In der Simulation ist nicht relevant, ob das Heizungsrohr die Wärme in den ersten oder fünften Stock transportiert oder wie lang das Stromkabel ist. Im Empfängergerät kommt die selbe Energiemenge an, wie das Sendergerät geschickt hat. Der Informationsaustausch zwischen den Geräten ist immer bilateral. Benötigte Werte werden angefragt oder an das entsprechende Gerät gesendet. Es gibt keine Instanz, die über alle Parameter verfügt und als Schnittstelle dient.

1.4.1 Blockheizkraftwerk

Das Blockheizkraftwerk ist im Basis-Szenario das wichtigste Gerät um Wärme bereitzustellen. Gleichzeitig wird Strom geliefert, der im Haus genutzt werden kann.

Funktionsweise

Blockheizkraftwerke arbeiten nach dem Prinzip der Kraft-Wärme-Kopplung. Diese bezeichnet die Nutzung der abfallenden Wärmeenergie, bei der Gewinnung von elektrischer Energie. Der Nutzungsgrad des Brennstoffs wird somit erhöht. Kernelement ist ein Generator, der meist durch die Verbrennung von Erdgas angetrieben wird. Der durch den Generator zur Verfügung gestellte elektrische Strom kann sowohl im Haus genutzt, als auch in das öffentliche Stromnetz eingespeist werden. Gemessen wird dies durch den Zweizegähler (Abschnitt 1.4.4). Durch den Antrieb des Generators entsteht im Verbrennungsmotor thermale Energie. Diese muss durch eine Kühlung abgeführt werden. Die Energie im Kühlmittel bleibt nicht ungenutzt, sondern wird weiter verwendet. Dies macht die Kraft-Wärme-Kopplung aus. Abwärme dient dazu den Wärmespeicher mit heißem Wasser zu befüllen. Daraus wird wiederum das Heizungssystem mit Energie versorgt und Warmwasser bereitgestellt.

Die Vorlauftemperatur des BHKWs muss immer im Bereich 55 °C – 60 °C liegen, um eine ausreichende Kühlung des Motors zu gewährleisten [33]. Um dies sicherzustellen wird häufig ein eigener Kreislauf zwischen BHKW und Wärmespeicher eingesetzt. Über einen Mischer oder Wärmetauscher wird der Rücklauf der Heizung (20 °C – 60 °C) mit wärmerem Wasser des Wärmespeichers gemischt und so die passende Vorlauftemperatur des BHKWs bereitgestellt [18]. Die anfallende thermische

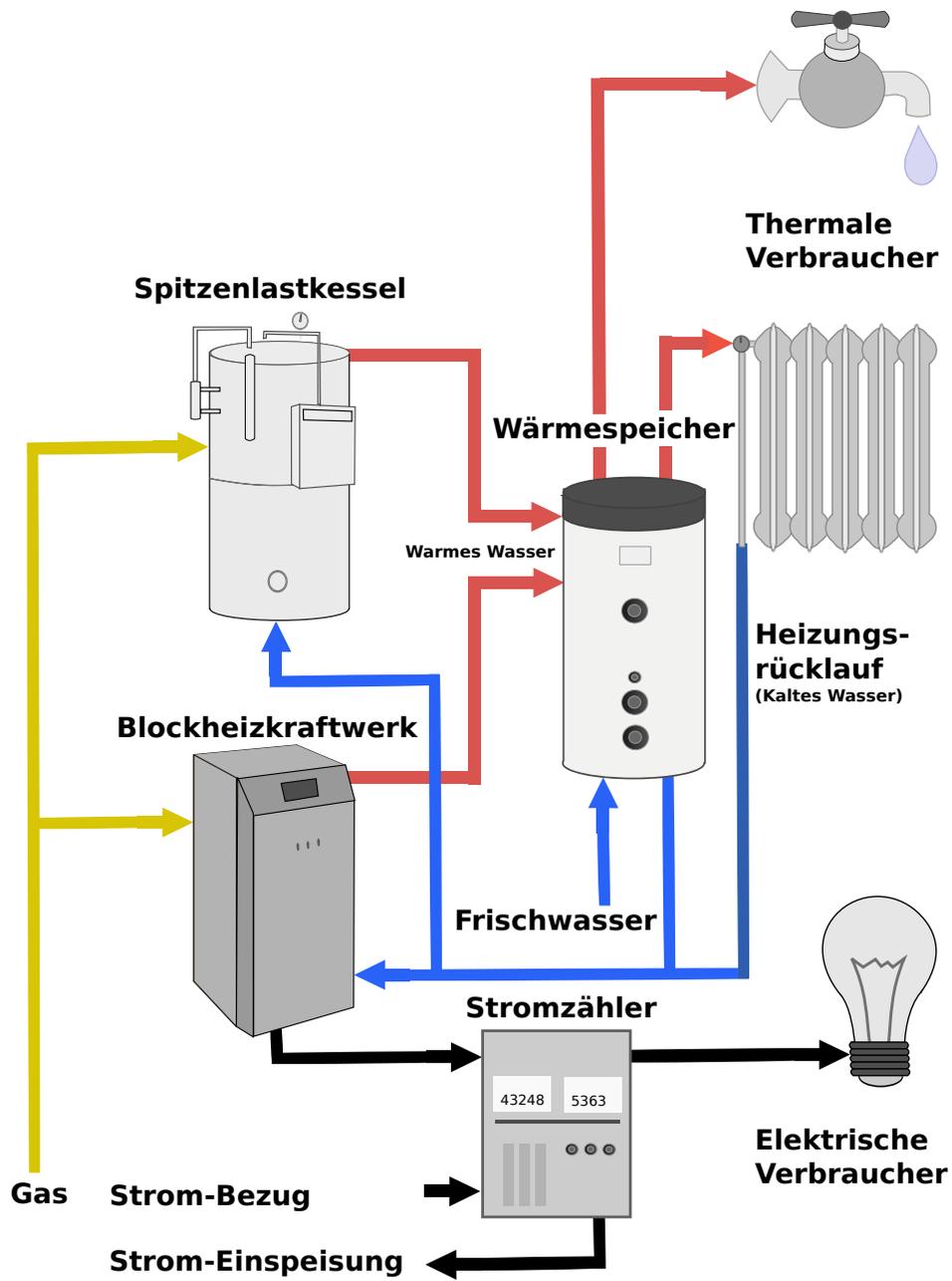


Abbildung 1.1: Basis-Szenario der Energiesysteme

Energiemenge ist deutlich höher als die Menge an gleichzeitig bereitgestelltem elektrischen Strom. Der Wirkungsgrad, mit dem elektrische Energie aus dem Brennstoff umgewandelt wird, liegt bei circa 30 %. Thermische Energie kann zu über 50 % aus der Eingangsenergie umgewandelt werden. Daraus resultiert ein Nutzungsgrad von 80 %–95 %. Herkömmliche Kohlekraftwerke ohne Kraft-Wärme-Kopplung erreichen einen Wirkungsgrad von 38 % und auch Gaskraftwerke können den Brennstoff nur zu maximal 60 % nutzen [30].

Einige BHKWs verfügen über einen modulierenden Betriebsmodus. Damit ist es möglich, die Drehzahl des Motors zu variieren und somit die Auslastung zu steuern. Dabei gibt es eine minimale Auslastung, unter der kein Betrieb möglich ist und das BHKW ausgeschaltet wird. Über der Minimalauslastung kann die Drehzahl entweder stufenlos oder in festen Schritten eingestellt werden. Bei geringerer Drehzahl wird weniger Brennstoff benötigt und somit auch weniger thermische und elektrische Energie abgegeben. Im Allgemeinen nimmt mit der Auslastung auch der Wirkungsgrad um bis zu 6 %² ab. Es wird also nicht nur in absoluten Zahlen, sondern auch in Relation zur Eingangsenergie weniger Energie abgegeben.

Gesteuert wird das BHKW über den Energiebedarf. Unterschieden wird zwischen der *wärmegeführten* und *stromgeführten* Fahrweise. Standardmäßig wird versucht, den Bedarf an thermaler Energie zu decken (wärmegeführt). Ist die Temperatur im Wärmespeicher zu niedrig, wird das BHKW angeschaltet. Wenn das BHKW modulieren kann, wird die Auslastung an die Höhe des thermalen Bedarfs angepasst. Der anfallende Strom wird bei Bedarf vor Ort genutzt oder andernfalls ins öffentliche Stromnetz eingespeist. Der Strombedarf wirkt sich jedoch nicht auf das Verhalten des BHKWs aus. Bei der *stromgeführten* Fahrweise wird die Auslastung des BHKWs nach dem Strombedarf berechnet. Dies ist nicht bei allen BHKWs möglich. Wenn elektrische Energie benötigt wird, ist das BHKW angeschaltet und moduliert entsprechend des Bedarfs. Begrenzt wird die Laufzeit jedoch weiterhin durch den Wärmespeicher. Dieser nimmt die abfallende thermale Energie auf. Ist keine weitere Aufnahme thermischer Energie möglich, schaltet sich das BHKW aus Sicherheitsgründen aus (Abschnitt 1.4.3). Auch bei weiter bestehendem Strombedarf muss es zu dieser Notabschaltung kommen.

Das BHKW muss regelmäßig gewartet werden, wobei zum Beispiel das Motorenöl gewechselt wird und einige andere Verschleißteile geprüft werden. Der Zeitpunkt der Wartung richtet sich dabei meist nach den Betriebsstunden und liegt im Bereich von 1400 h–10 000 h Laufzeit [30].

Simulation

Die Simulation des BHKWs basiert auf verschiedenen Parametern. Die maximale Eingangsleistung gibt an, wie viel Brennstoff aufgenommen werden kann. Schon an dieser Stelle wird in der Simulation jedoch abstrahiert, um welchen Brennstoff es sich handelt. Wir geben die äquivalente Leistung des Energieträgers in kW an.

²Interview mit Herrn Gailfuß am 24.06.2014, http://www.bhkw-consult.de/en/bhkw-consult_kontakt.html (abgerufen am 25.08.2014).

Eine Betrachtung der Güte des Erdgases (Brennwert, Heizwert) muss somit vorher geschehen. Im Allgemeinen gehen wir von 10 kWh pro 1 m³ Erdgas aus. Um zu berechnen, wie viel thermische und elektrische Energie das BHKW daraus gewinnt, müssen zwei Wirkungsgrade (*thermale_effizienz* und *elektrische_effizienz*) angegeben werden. Je ein Faktor gibt an, wie viel aus der Eingangsenergie in thermische bzw. elektrische Energie umgewandelt werden kann.

Diese Faktoren können leicht aus der maximalen Leistung des BHKWs errechnet werden:

$$\eta_{\text{ele max}} = \frac{P_{\text{ele max}}}{P_{\text{zu max}}}$$

Da einige BHKWs modulieren können, haben wir weiterhin die Möglichkeit geschaffen, die geringste Auslastung anzugeben (*minimale_auslastung*). Würde die Auslastung unter die Minimalauslastung sinken, ginge das BHKW aus, über dem Minimum kann die Auslastung frei gewählt werden. Es gibt jedoch keine Möglichkeit Leistungsstufen einzustellen. Ein nicht-modulierendes BHKW kann durch Setzen des Parameters *minimale_auslastung* = 100 simuliert werden.

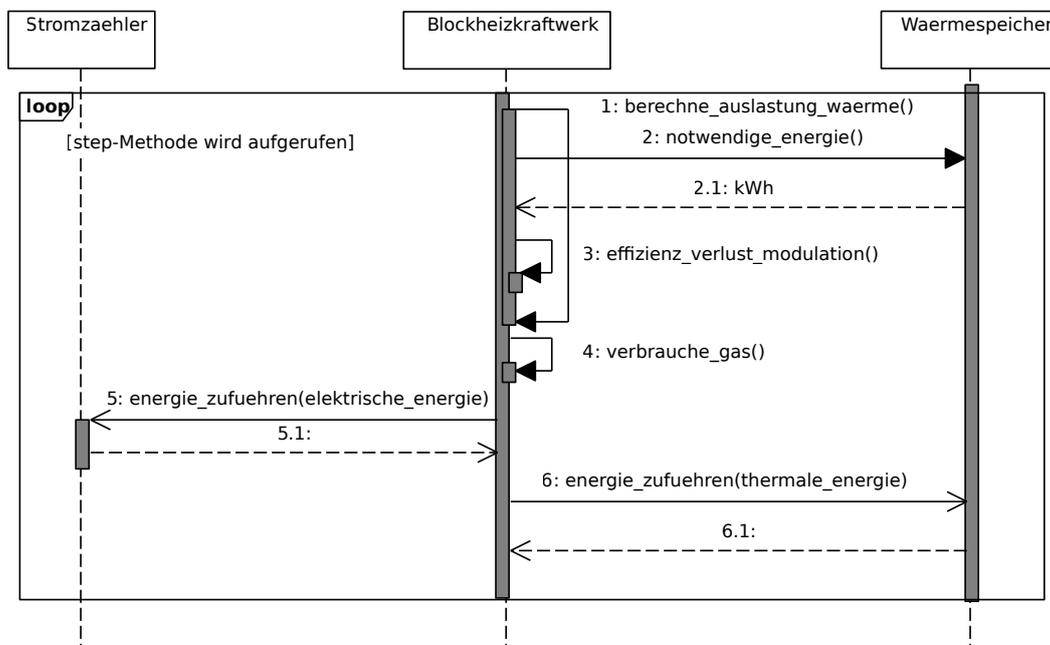


Abbildung 1.2: *step()*-Methode des simulierten BHKWs

Abbildung 1.2 zeigt, welche Methoden in jedem Zeitschritt aufgerufen werden. Dargestellt ist die wärmegeführte Fahrweise. Das BHKW berechnet die Auslastung mit Hilfe des thermalen Energiebedarfs, stellt die entsprechende Menge elektrische und thermale Energie zur Verfügung und bezieht die Eingangsenergie. Diese wird entsprechend der momentanen Auslastung aus der Eingangsleistung berechnet (Methode *verbrauche_gas()*). Mit dieser Eingangsleistung und den Wirkungsgra-

den kann dann die umgewandelte thermische und elektrische Energiemenge bestimmt werden. Die beiden Fahrweisen des BHKWs werden durch die Methoden `berechne_auslastung_strom()` und `berechne_auslastung_waerme()` implementiert. Sie berechnen die nötige Auslastung des BHKWs für den nächsten Zeitschritt in der jeweiligen Fahrweise. Dazu wird zunächst in der Methode `notwendige_energie()` der thermale beziehungsweise elektrische Bedarf ermittelt. Damit wird die Energieabgabe zwischen der maximal möglichen (Volllast) und der minimal möglichen (Variable `minimale_auslastung`) Auslastung interpoliert. Bei der Stromführung muss zusätzlich sichergestellt werden, dass nicht zu viel thermische Energie in den Wärmespeicher geleitet wird (Abschnitt 1.4.3).

Um das Abfallen der Wirkungsgrade bei der Modulation nachzuempfinden, muss ein weiterer Faktor angegeben werden (`wirkungsgrad_verlust_max`). Dieser gibt an, um wie viel der Wirkungsgrad bei geringster Leistung abnimmt. Die Methode `effizienz_verlust_modulation()` ist in Quelltext 1.1 gezeigt. Sie berechnet die Höhe der Abnahme, indem der maximale Wirkungsgrad mit dem genannten Faktor zwischen der Minimalauslastung und Volllast interpoliert wird.

Die Betriebszeit des BHKWs wird ebenfalls gemessen, um rechtzeitig eine Wartung durchzuführen. Dazu werden auch die Schaltzyklen (Ein-, Ausschaltvorgänge) registriert, da diese große Auswirkungen auf den Verschleiß und damit das Wartungsintervall haben.

Schließlich sei noch zu erwähnen, dass von dem internen Kühlungskreislauf des BHKWs abstrahiert wurde. Da die Kommunikation zwischen den Geräten allgemein über Energie gelöst ist, entfällt die Betrachtung von Vor- und Rücklauftemperaturen. Es gibt keinen Rücklauf vom Wärmespeicher zum BHKW, weshalb die Mindesttemperatur nicht überprüft werden kann. Es gilt die Annahme, dass genügend Wasser zur Kühlung des BHKWs vorhanden ist, solange der Wärmespeicher nicht voll ist.

1.4.2 Spitzenlastkessel

Neben dem BHKW wird ein weiterer Wärmelieferant benötigt, um die ausreichende Versorgung mit thermaler Energie zu garantieren.

Quelltext 1.1: Berechnung der Abnahme des Wirkungsgrads bei Modulation

```
def effizienz_verlust_modulation(BHKW):
    if BHKW.auslastung() <= BHKW.minimale_auslastung:
        return BHKW.wirkungsgrad_verlust_max
    auslastung_bereich = 1.0 - BHKW.minimale_auslastung
    auslastung_relativ = (BHKW.auslastung() - \
        BHKW.minimale_auslastung) / auslastung_bereich
    auslastung_invers = 1.0 - auslastung_relativ
    return BHKW.wirkungsgrad_verlust_max * auslastung_invers
```

Funktionsweise

Der Spitzenlastkessel arbeitet wie ein herkömmlicher Heizkessel. Bei der Verbrennung von Gas wird Wasser erhitzt, welches danach in den Wärmespeicher geleitet wird. Als Energieträger kann neben Erdgas auch Erdöl oder Rapsöl genutzt werden. Da der SLK nicht nach dem Prinzip der Kraft-Wärme-Kopplung arbeitet, kann er sich nicht über zusätzliche Einnahmen aus dem Stromverkauf refinanzieren.

Der SLK wird benötigt, weil die Leistung des BHKWs meist zu gering angesetzt wird. Damit das BHKW möglichst lange Laufzeiten erzielt, wird die thermale Leistung auf ungefähr 50 %–70 %³ des eigentlichen maximalen Bedarfs bemessen. Zu Spitzenlasten kann das BHKW den Wärmebedarf daher nicht abdecken. Dies ist ab circa $-5\text{ }^{\circ}\text{C}$ Außentemperatur⁴ der Fall. Dann kann die Temperatur des Wärmespeichers ohne Betrieb des SLKs unter die Mindesttemperatur fallen. Der SLK sorgt dann für ausreichende Wärmeversorgung. Um die Zieltemperatur möglichst schnell wieder herzustellen, verfügt der SLK meist über eine weitaus höhere Leistung als das BHKW. Die Wartung des SLKs richtet sich analog zum BHKW entweder nach der Laufzeit oder den Schaltzyklen.⁵

Simulation

Das Modell des SLKs in der Simulation geht von einem sehr simplen Brennkessel aus. Der von uns simulierte SLK verfügt über keinen modulierenden Betrieb wie das BHKW, da dies für die Abdeckung von Leistungsspitzen nicht nötig ist. Der Zustand ist demnach binär:

- Ausgeschaltet wird kein Gas verbraucht und keine Energie abgegeben.
- Eingeschaltet wird der maximale Gasverbrauch angefordert und entsprechend die maximale Energiemenge abgegeben.

Durch Verschleiß, Gasgüte oder Umgebungstemperaturen könnte die abgegebene Leistung bei konstantem Gasverbrauch schwanken. Die Simulation abstrahiert jedoch von einem schwankenden Wirkungsgrad. Dieser ist als Konstante festgelegt (*thermale_effizienz*). Mit zusätzlicher Angabe des maximalen Gasverbrauchs kann dann die maximal abzugebende Energie errechnet werden. In jedem Simulationsschritt wird mittels des Wärmespeichers bestimmt, ob zusätzliche Energie benötigt wird. Wenn dies nicht der Fall ist, bleibt der SLK ausgeschaltet. Andernfalls wird die Auslastung auf das Maximum gesetzt und somit die Wärmeproduktion gestartet. Latenzzeiten während des Startvorgangs oder Aufwärmphasen, bis das heiße Wasser den Wärmespeicher erreicht, werden nicht berücksichtigt. Wegen der hohen Leistung des SLKs sind diese sehr kurz.

In Abbildung 1.3 ist das Verhalten des SLKs in der *step()*-Methode dargestellt. Der ausgeschaltete SLK wird zunächst wegen des erhöhten thermalen Bedarfs eingeschaltet. In diesem Zustand wird dem Wärmespeicher Energie hinzugefügt und Gas

³Interview mit KRAFTLAND GmbH am 07.02.2014, <http://www.kraftland.de/> (abgerufen am 25.08.2014).

⁴Erfahrung des BHKW-Betreibers Schmidt.

⁵Auskunft des Projektpartners.

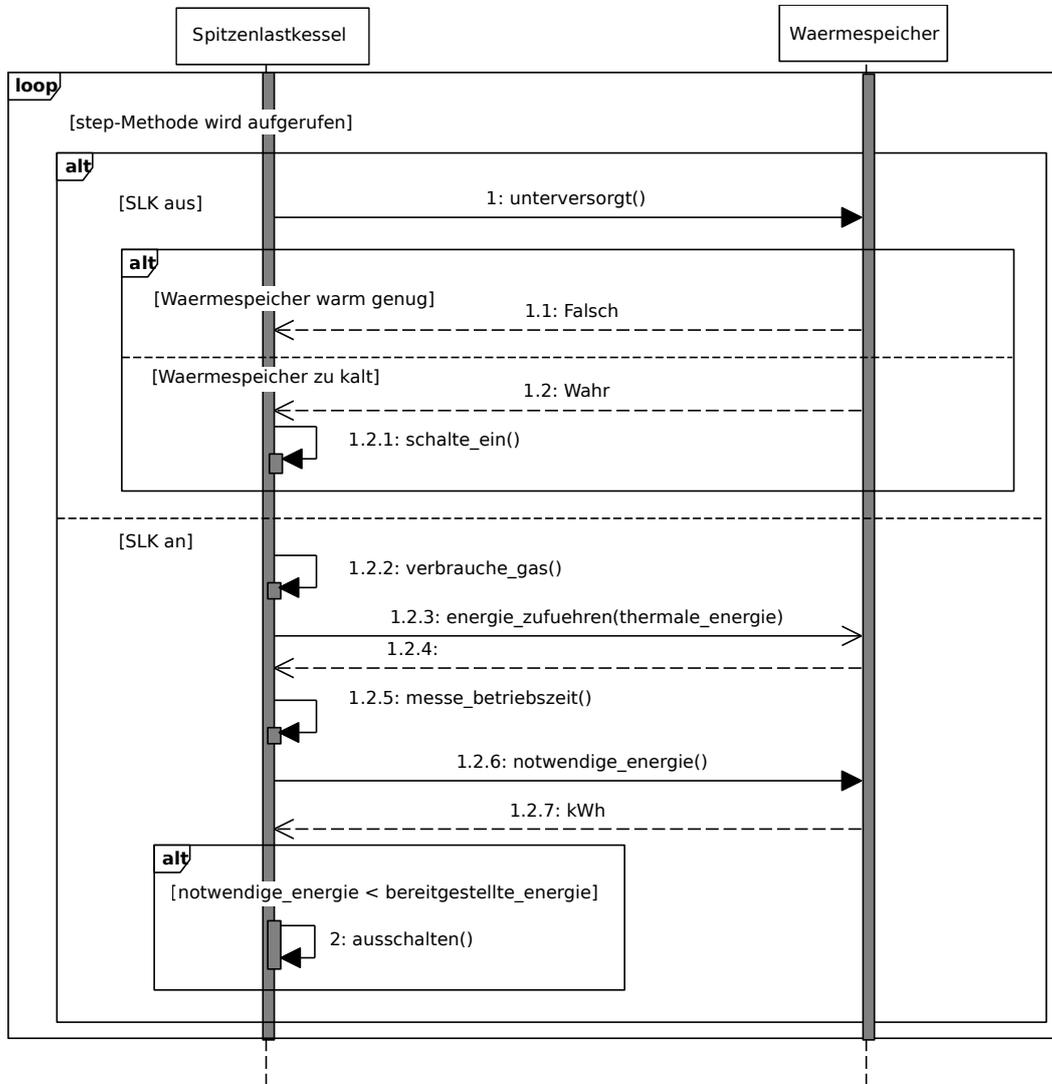


Abbildung 1.3: *step()*-Methode des simulierten SLKs

verbraucht. Dies geschieht wie folgt:

$$\text{Gasverbrauch} = \frac{\text{Auslastung}}{100} \cdot \text{Gasverbrauch}_{\text{max.}}$$

Wenn die thermale Energie des SLKs höher ist als die benötigte Energie des Wärmespeichers, wird der SLK abgeschaltet (*ausschalten()*). Analog zum BHKW werden Betriebszeit und Einschaltvorgänge mitgezählt (*messe_betriebszeit()*), da zahlreiche Schaltzyklen den Verschleiß erhöhen und somit ein Indiz für den Systemzustand sind. Um häufige Zyklen zu unterbinden, gibt es die Möglichkeit, eine Zeit festzulegen, die der SLK nach dem Abschalten ausgeschaltet bleibt.

1.4.3 Wärmespeicher

Der Wärmespeicher wird benötigt, um die Versorgung mit thermaler Energie sicherzustellen, da eine direkte Verbindung von Energielieferanten und Verbrauchern nicht möglich ist.

Funktionsweise

Der Wärmespeicher ist prinzipiell ein isolierter Wassertank, der heißes Wasser für die Heizung und Warmwasserversorgung beinhaltet. Wegen der langen Aufwärmzeit von Wasser ist eine Verbindung von Wärmeerzeuger und Heizung nur durch diesen Puffer möglich. Der stark schwankende Bedarf an thermaler Energie für Heizung und Warmwasser kann ohne diesen nicht umgehend vom BHKW oder SLK ausgeglichen werden. Oft wird die thermale Energie für Heizung und Warmwasser in getrennten Speichern vorgehalten. Der sogenannte Pufferspeicher stellt heißes Wasser für die Heizung und der Warmwasserspeicher das erwärmte Trinkwasser zur Verfügung. In Kombinationsspeichern ist dies auch mittels getrennter Kreisläufe in einem Speicher möglich. Im Folgenden werden solche allgemein als Wärmespeicher bezeichnet.

Im Wärmespeicher wird erhitztes Wasser vorgehalten, damit zu jeder Zeit die gewünschte Heizleistung abgegeben und Warmwasser zur Verfügung gestellt werden kann. Meist geschieht dies über sogenannte Mischer. Diese mischen das kühlere Rücklaufwasser der Heizung mit dem heißen Wasser des Wärmespeichers auf eine eingestellte Temperatur. Für Warmwasser wird statt des Rücklaufwassers Frischwasser verwendet. Gleichzeitig kann dem Wärmespeicher durch beispielsweise BHKW oder SLK Energie in Form von heißem Wasser hinzugefügt werden. Dies ist nötig, wenn die Temperatur im Wärmespeicher unter eine eingestellte Zieltemperatur sinkt. Die Speichertemperatur wird dabei an mehreren Stellen (normalerweise oben, mittig und unten) gemessen und differiert um bis zu 30 °C. Die Temperaturdifferenz wird Schichtung genannt. Oft kann das Wasser in verschiedenen Höhen entnommen werden, was die Mischung von gewünschten Wassertemperaturen erleichtert und die Energie effizienter nutzt. Wenn die eingestellte Maximaltemperatur erreicht wird, muss die Energiezufuhr gestoppt werden, um im schlimmsten Fall eine Explosion des Wärmespeichers bei siedendem Wasser zu vermeiden.

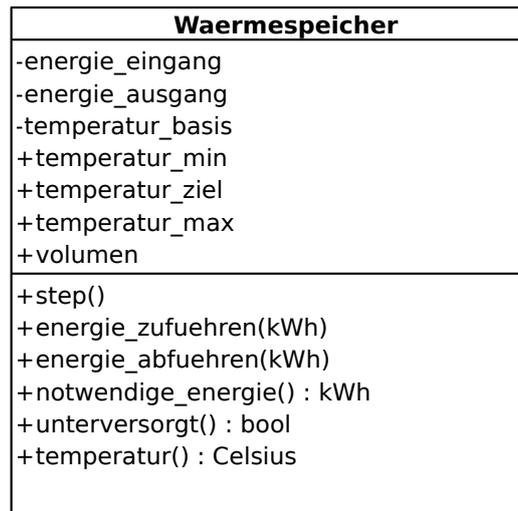


Abbildung 1.4: Klassendiagramm des simulierten Wärmespeichers

Die Größe des Wärmespeichers hängt von der Leistung angeschlossener Geräte und der Varianz des Energiebedarfs ab. Im Allgemeinen ist eine Größe von mindestens 500 l gefordert [33]. Ein unterdimensionierter Wärmespeicher würde zu häufigen Schaltzyklen führen, da die Maximaltemperatur schnell erreicht wird. Ein zu großer Wärmespeicher hingegen verliert unnötig viel Energie durch Abwärme. Die optimale Größe des Wärmespeichers wird durch folgende Regel abgeschätzt: Das BHKW sollte mehr als 30 min benötigen den Wärmespeicher mit Energie zu füllen, ohne dass Energie entnommen wird. Gleichzeitig soll der gefüllte Speicher den thermalen Bedarf, ohne Energiezufuhr, über 5 h ausgleichen können⁶.

Da der Wärmespeicher nicht perfekt isoliert werden kann, strahlt dieser kontinuierlich Wärme ab. Auch wenn kein thermaler Energiebedarf besteht, nimmt die Temperatur stetig ab. Wie viel Energie dabei entweicht ist stark von der Dämmung des Wärmespeichers abhängig.

Simulation

In Abbildung 1.4 ist die Klasse des Wärmespeichers mit den wichtigsten Attributen und Methoden dargestellt. In der Simulation wird nicht zwischen Puffer- und Warmwasserspeicher unterschieden. Sowohl die Energie für die Heizung, als auch für warmes Trinkwasser werden aus einem zentralen Wärmespeicher entnommen.

Von dem im Wärmespeicher enthaltenen Wärmeträger Wasser wurde bei der Simulation komplett abstrahiert. Die von BHKW und SLK abgegebene Energiemenge wird im Wärmespeicher in der Variable *energie_eingang* gespeichert. Dies geschieht über die Methode *energie_zufuehren()*. Mit Hilfe des Volumens und der

⁶Interview mit Herrn Gailfuß am 24.06.2014, http://www.bhkw-consult.de/en/bhkw-consult_kontakt.html (abgerufen am 25.08.2014).

spezifischen Wärmekapazität von Wasser (4,19 kWh/l/K), kann die Temperatur des Wärmespeichers errechnet werden (*temperatur()*). Wie in Gleichung 1.1 dargestellt, gehen wir von einer Basistemperatur aus, die nicht unterschritten werden kann (*temperatur_basis* = 20 °C). Bei Zugabe von Energie wird eine Temperaturdifferenz berechnet, um die sich der Wärmespeicher erwärmt. Dies führt zur Berechnung einer durchschnittlichen Temperatur, welche die realen Temperaturschichten im Wärmespeicher nicht berücksichtigt. Die Energieabgabe funktioniert analog zur Energiezufuhr. Auf Grundlage der von den Verbrauchern angefragten Leistung wird die resultierende Energiemenge pro Zeiteinheit von der vorhandenen Energie im Wärmespeicher abgezogen. Gespeichert wird dies in der Variable *energie_ausgang* über die Methode *energie_abfuehren()*.

$$\text{Temperatur} = \text{temperatur_basis} + \frac{\text{energie_eingang} - \text{energie_ausgang}}{\text{Volumen} \cdot \text{Wärmekapazität}} \quad (1.1)$$

Generell wird versucht, die eingestellte Zieltemperatur zu erreichen. Die Methode *notwendige_energie()* gibt die Energiemenge in kWh zurück, die nötig ist, um die Zieltemperatur zu erreichen. Neben der Ziel- und der Maximaltemperatur kann eine Minimaltemperatur eingestellt werden. Diese dient dazu den Wärmespeicher besser auszunutzen. Der SLK wird erst eingeschaltet, wenn die Minimaltemperatur unterschritten ist (*unterversorgt()*). Andernfalls wird versucht die Energie nur vom BHKW zu beziehen.

Den Temperaturverlust des Speichers simulieren wir, indem pro Zeitschritt eine bestimmte Energiemenge abgezogen wird. Dazu nehmen wir standardmäßig einen Verlust von 3 °C pro Tag⁷ an und rechnen diesen in Energie pro Zeitschritt um.

1.4.4 Stromzähler

Ein Stromzähler bestimmt die benötigte elektrische Energie eines Hauses. Die Energiemenge kann mit dem Stromversorger abgerechnet werden.

Funktionsweise

Ein herkömmlicher Stromzähler misst die vom öffentlichen Stromnetz bezogene elektrische Energie. Wenn jedoch eigene elektrische Energie generiert wird, muss neben dem extern bezogenen Strom auch der Verbrauch des Eigenstroms und die Einspeisung gemessen werden. Dies ist mit Zweiwegezählern möglich. Diese zeigen aufsummiert den vom externen Strombetreiber bezogenen Strom und den eingespeisten Strom in kWh an. Für den Bezug fällt nach jeder Abrechnungsperiode (quartalsweise oder jährlich) ein Entgelt von 0,29 €/kWh⁸ an. Dieser Strompreis beinhaltet zahlreiche Steuern und Abgaben, wie die beispielsweise die EEG-Umlage

⁷Aussage des Projektpartners.

⁸Daten zur Energiepreisentwicklung, <https://www.destatis.de/DE/Publikationen/Thematisch/Preise/Energiepreise/Energiepreisentwicklung.html> (abgerufen am 20.06.2014).

oder Netzentgelte. Im Gegenzug wird die eingespeiste Energiemenge nur mit circa 0,05 €/kWh⁹ vom Staat vergütet.

Simulation

Der simulierte Stromzähler spiegelt den Hauptzähler des Hauses wider. Es können somit keine Rückschlüsse auf mieterspezifische Stromverbräuche geschlossen werden, da nur der gesamte Stromverbrauch des Hauses simuliert wird.

In jedem Zeitschritt meldet das BHKW die zur Verfügung gestellte Energiemenge an den Zähler. Gleichzeitig wird durch den Stromverbraucher der Bedarf an elektrischer Energie ermittelt. Aus der Differenz kann entschieden werden, ob überschüssiger Strom eingespeist oder zusätzlicher Strom bezogen werden muss. Neben den Strommengen von Bezug und Einspeisung berechnet der Zähler zusätzlich die insgesamt anfallenden Stromkosten und den Gewinn aus der Stromeinspeisung. Die dazu nötigen Kosten pro kWh können eingestellt werden.

Nicht zu vergessen ist, dass grundlegend von Leitungsverlusten abstrahiert wird. Sowohl beim Transfer von der Energiebereitstellung zum Verbraucher, als auch bei Bezug und Einspeisung berechnen wir in der Simulation keine Energieverluste.

1.4.5 Verbraucher

Im Folgenden wird beschrieben, wie die von den Energielieferanten bereitgestellte Energie genutzt wird. In dem Energiesystem muss der Bedarf an thermaler und elektrischer Energie bestimmt werden der von den Verbrauchern angefordert wird.

Funktionsweise

Der Stromverbrauch einer Wohnung kann über den jeweiligen Stromzähler abgelesen werden. Über sogenannte *Smart-Meter* kann dies sogar digital und automatisiert geschehen. Die angeforderte Leistung von allen Stromverbrauchern der Wohnung wird über die Zeit als Energie in kWh abgerechnet.

Die aufgewendete Energie der Heizung wird oft über Verdunstungsröhrchen an jedem Heizkörper abgerechnet. Kleine Glasröhren mit langsam verdunstender Flüssigkeit sind fest an jedem Heizkörper montiert. Mit einer Skala wird nach einem Jahr ermittelt, wie viel der Flüssigkeit verdunstet ist. Wurde viel geheizt, ist entsprechend mehr Flüssigkeit verdunstet. Ein Rückschluss auf die abgegebene Wärmeenergie ist jedoch nicht möglich. Entsprechend der ermittelten Heizeinheiten werden die Gesamtkosten der Heizung zwischen allen Mietparteien aufgeteilt. Die Gesamtkosten beinhalten Gaskosten, Wartung und Betriebsstrom. Diese werden durch die gesamten Heizeinheiten geteilt und, je nach Anteil der Mietparteien, in Rechnung gestellt.

Nach ähnlichem Prinzip arbeiten die moderneren elektronischen Heizkostenverteiler. Sie sind ebenfalls an jedem Heizkörper montiert und messen die abgegebene Wärmeenergie. Dazu wird die Differenz von Raum- und Heizungstemperatur ermittelt und integriert. Genaue Angaben über die abgegebene Energiemenge sind jedoch auch hier nicht möglich. Die Aufteilung der Kosten erfolgt analog zu den

⁹BHKW-Betreiber Schmidt.

Verdunsterzählern. Der Vorteil der elektronischen Zähler besteht in der genaueren Messung und der Möglichkeit einer automatisierten Auswertung über Funksignale.

Wie viel Wärme der Heizkörper abgibt, wird bei jedem Heizkörper über das Ventil eingestellt. Dieses steuert den Durchfluss des heißen Wassers von den Heizungsrohren in den Heizkörper. Wird das Ventil weiter geöffnet, kann mehr heißes Wasser in den Heizkörper fließen und dieser strahlt mehr Wärme ab. Fließt weniger Wasser vom Heizungsrohr in den Heizkörper, kühlt dieser aus und gibt weniger Wärme ab. Wenn das heiße Wasser oben in den Heizkörper fließt, erwärmt dieser die Raumluft. Das kühlere Wasser fließt unten wieder als Rücklauf in die Rohre des Heizungssystems. Von dort gelangt es in den Wärmespeicher oder das BHKW und wird erneut erwärmt. Die thermale Energie des heißen Wassers wurde damit in den Wohnraum übertragen. Dieser kühlt jedoch kontinuierlich aus, da die thermale Energie über Außenwände und Fenster in die Umwelt abgegeben wird.

Bei der zentralen Bereitstellung von Warmwasser muss auch dieser Verbrauch gemessen und berechnet werden. Dies funktioniert über einen Warmwasserzähler, der die Menge an genutztem Warmwasser in Kubikmetern misst. Dennoch kann der aktuelle Verbrauch meist nicht automatisch ausgelesen werden. Zur Abrechnung werden die Gesamtkosten zur Bereitstellung des Warmwassers anteilig je nach Verbrauch auf die Mieter verteilt. Um die Gesamtkosten zu berechnen, ist ab 31.12.2013 ein Wärmehzähler einzusetzen [9, §9 II]. Dieser berechnet aus der Temperatur und Durchflussgeschwindigkeit die benötigte Energiemenge.

Simulation

In der Simulation wird nur der Gesamtverbrauch des Hauses von Strom und Wärme ermittelt. Es ist nur der summierte Bedarf aller Mieter verfügbar.

Der Strombedarf wird nicht von der Simulation ermittelt. Dies wäre nur durch eine Vorhersage des Nutzerverhaltens möglich. Eine solche Vorhersage wird in Kapitel 4 näher betrachtet. Der Strombedarf wird hier als gegeben angenommen und muss als Eingabe der Simulation zur Verfügung gestellt werden.

Der Bedarf an thermaler Energie für die Heizung und Warmwasser wird wie in Abbildung 1.5 dargestellt berechnet. Dieser Vorgang kann grob in die drei Schritte aufgeteilt werden: Aufwärmen des Raumes durch den Heizkörper, Veränderung der Wärmeleistung des Heizkörpers und entweichen von Wärme in die Umwelt. So wird der Wärmekreislauf eines Hauses simuliert. Damit können wir die starke Abhängigkeit des Verbrauches von der Außentemperatur berücksichtigen und Parameter verschiedener Häuser einbeziehen. Alternativ könnte der Bedarf auch durch eine Formel errechnet werden. Diese würde die Parameter kapseln und damit das Verhalten schlechter nachvollziehen lassen. Außerdem könnte der Bedarf unterschiedlich beschaffener Häuser nicht variiert werden. Die durch unseren Ansatz entstehenden Abstraktionen von komplexen physikalischen Vorgängen könnten, genau wie eine Formel, zu einem von der realen Welt abweichenden Bedarf führen. Für ein spezielles Szenario mit Echtdateen wird dies in Abschnitt 1.5 untersucht.

Die Übertragung der Wärmeenergie des Heizkörpers in den Raum wird ähnlich wie im Wärmespeicher simuliert. Der Heizkörper verfügt über eine gewisse Leistung mit der thermale Energie an die Luft des Raumes übertragen wird. Diese Leistung

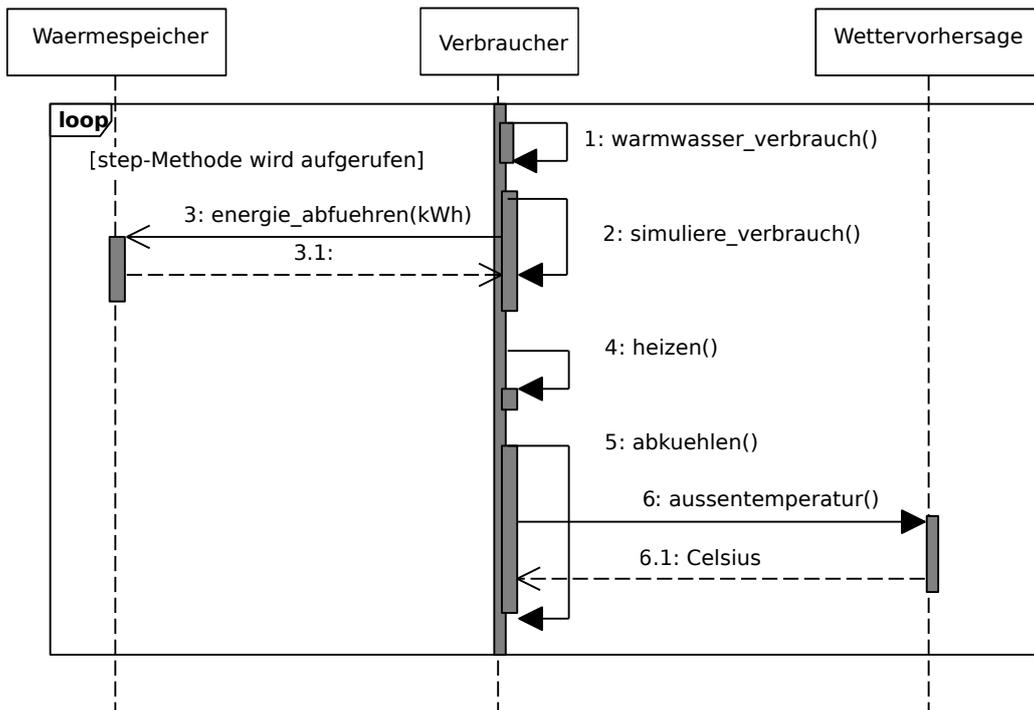


Abbildung 1.5: *step()*-Methode des simulierten thermalen Verbrauchers

kann entsprechend des Typs und der Größe des Heizkörpers bestimmt werden [43]. Allgemein geht man von 70 kW–100 kW pro Quadratmeter zu beheizender Fläche aus. Hierbei sei angemerkt, dass zahlreiche Informationen aus den Heizungssimulationen von [36] und [45] übernommen wurden.

Über die aktuelle Leistung des Heizkörpers wird pro Zeitschritt die Energiemenge berechnet, die in den Raum abgegeben wird. Da der Heizkörper nicht konstant die gleiche Leistung bereitstellt, wird diese fortlaufend angepasst (*simuliere_verbrauch()*): Ist die Zieltemperatur des Raumes noch nicht erreicht, erhöht sich die Leistung schrittweise bis zu ihrem Maximalwert. Ist die Temperatur erreicht, beginnt die Leistung abzunehmen. Der Anstieg der Leistungsveränderung wurde so gewählt, dass die volle Leistung in 60 min erreicht wird und der Heizkörper auch in dieser Zeit komplett auskühlt. Dabei wird nicht berücksichtigt, dass das Erwärmen der Heizung deutlich schneller möglich ist, als das Abkühlen. Aus der so berechneten momentanen Leistung der Heizung wird die angeforderte Energiemenge pro Zeitschritt berechnet und an den Wärmespeicher gesendet (*energie_abfuehren()*). Wieder werden Verluste durch Leitungen und sonstige Abstrahlung vernachlässigt. Aus der so berechneten Leistung kann mit der spezifischen Wärmekapazität der Luft (0,27 kWh/m³K) und dem Volumen des Raumes eine Temperaturdifferenz errechnet werden. Um diese Temperaturdifferenz erwärmt sich die Raumluft bei Hinzugabe der berechneten Energie. Die entsprechende Methode *heizen()* ist in Quelltext 1.2 dargestellt.

Als letztes wird in jedem Zeitschritt berechnet, wie viel Energie durch Fenster und Außenwände an die Umwelt abgegeben wird (*abkuehlen()*). Der Wärmeübergangskoeffizient gibt an, wie viel thermale Energie von einem Stoff übertragen wird. Entscheidend ist darüberhinaus die Fläche der Außenwände und Temperaturdifferenz nach Draußen. Die Fläche von Außenwänden und Fenstern wird mit den eingestellten Parametern des Hauses berechnet. Die Temperaturdifferenz kann aus Außentemperatur und errechneter Raumtemperatur gebildet werden. Ist die Differenz zwischen Außentemperatur und Raumtemperatur hoch, wird mehr Energie in die Umwelt abgegeben, als wenn die Differenz klein ist. Die Außentemperatur ist dabei eine Eingabe der Simulation und wird als gegeben angenommen (*aussentemperatur()*). Aus Fläche, Wärmeübergangskoeffizient und Temperaturdifferenz wird die Leistung bestimmt, mit der das gesamte Haus Wärme abgibt. Dadurch nimmt die im Haus vorhandene thermale Energie ab.

Eine weitere Annahme setzt voraus, dass die Außentemperatur immer geringer ist als die Raumtemperatur. So wird in jedem Fall Energie in die Umwelt abgegeben. Die Leistung der Wärmeabstrahlung kann dann bei der Berechnung der Raumtemperatur als „negative“ Leistung mit einbezogen werden. Sie trägt als Gegenstück des Heizkörpers zum Auskühlen des Raumes bei.

Schließlich haben wir die Berechnung wie folgt vereinfacht: Die Berechnung der Raumtemperatur, wie sie beschrieben wurde, erfolgt für einen durchschnittlichen Raum. In einem Haus haben einige Räume gar keine Außenwände, andere jedoch mehrere. Um das zu berücksichtigen werden Durchschnittswerte für Außenwände und der Fläche an Fenstern approximiert. Diese starken Abstraktionen wurden gemacht, um den Verwaltungsaufwand gering zu halten. Andernfalls müssten für jeden Raum die Außenwandfläche, Fenstergröße und Heizkörperleistung eingegeben werden. Auch für die Ermittlung des Wärmebedarfs müsste dann entschieden werden, welcher Raum als Referenz gewählt wird. Durch die Vereinfachung ist nur eine Raumtemperatur gegeben und für die Parameter des Hauses müssen nur Durchschnittswerte angegeben werden.

Weiterhin wird nicht beachtet, dass zur Erwärmung des Raumes nicht nur die Raumluft, sondern auch Gegenstände und Wände gehören. Durch die unterschiedlichen spezifischen Wärmekapazitäten kann auch verschieden viel Energie aufgenommen werden.

Die Berechnung des Verbrauchs von Warmwasser basiert auf den Werten von [31]. Pro Stunde ist der durchschnittliche Bedarf an Warmwasser bestimmt worden. Unterschieden wird zwischen Werktag und Wochenende. Je nach Tag wird zwischen den

Quelltext 1.2: Zunahme der Temperatur eines Raumes bei Energiezufuhr

```
def heizen(RAUM):  
    waerme_kapazitaet_luft = 0.27  
    raum_energie = RAUM.leistung * SIMULATION.step_size  
    temp_delta = raum_energie / RAUM.volumen * waerme_kapazitaet_luft  
    RAUM.temperatur += temp_delta
```

simulierten Stundenwerten interpoliert und die benötigte Menge an Warmwasser ermittelt. Mit der eingestellten Temperatur von 40 °C und der spezifischen Wärmekapazität von Wasser, kann so die erforderliche Energiemenge berechnet werden. Dies geschieht analog zum Wärmespeicher nach Gleichung 1.1 auf Seite 14. Die Energiemenge wird für alle Mieter summiert und wie die Heizenergie vom Wärmespeicher angefordert (*warmwasser_verbrauch()*).

1.5 Validierung und Verifikation

Wie aussagekräftig die Ergebnisse einer Simulation sind, muss für jedes Modell analysiert werden. Sind die Abweichungen zu dem Verhalten des realen Systems zu groß, ist die Simulation unbrauchbar. Um dies zu prüfen sind zwei verschiedene Fragestellungen zu beachten. Bei der Validierung wird das Simulationsmodell mit dem entsprechenden realen System verglichen. Die Ergebnisse des Verhaltens sollten, unter Berücksichtigung der Abstraktionen, ähnlich sein. Verifikation hingegen prüft, ob das Simulationsmodell in sich korrekt ist. Die Vorgänge in der Simulation sollten schlüssig sein und keine logischen Fehler enthalten.

Für die vorliegende Simulation wurden sogenannte Realitätstests [10] durchgeführt, um das simulierte Verhalten mit einem vergleichbaren realen System in Verbindung zu setzen (Abschnitt 1.5.1). Weiterhin wurden Plausibilitätstests [10] genutzt, um Widersprüche, der physikalischen Korrektheit aufzudecken. Diese wurde vor allem bei der Validierung der Simulation genutzt, da physikalische Widersprüche bei der Analyse des Verhaltens erkennbar wurden. Beispiele für Widersprüche sind Wärmespeichertemperaturen außerhalb 0 °C–100 °C oder eine Auslastung größer 100 %.

Zur Verifikation sollten Korrektheitsbeweise für die erstellten Programme genutzt werden [10]. Diese wurden nicht aufgestellt, stattdessen jedoch Unittests geschrieben, die das gewünschte Verhalten überprüfen. Die Testabdeckung der Simulation beträgt 71,7 %¹⁰.

Oftmals wird Kreuzvalidierung genutzt, um ein Modell zu validieren. Dies ist gerade bei einer kleinen Menge von Ausgangsdaten vorteilhaft [25]. Diese Herangehensweise wurde jedoch nicht gewählt, da ein Training des Modells mit realen Daten bei unserer Simulation nicht möglich ist.

1.5.1 Realitätstest

Eine Möglichkeit zur Validierung einer Simulation ist der Vergleich mit einem realen Referenzsystem. Die Echtdaten können auf verschiedene Weise mit den Daten der Simulation in Relation gesetzt werden.

¹⁰Ermittelt durch Travis-CI am 25.06.2014.

Datenbeschaffung

Um die Validierung mit Realitätstests durchzuführen, werden zunächst Echt Daten benötigt, welche dann dem simulierten Verhalten der Energiesysteme verglichen werden. Die Beschaffung von Echt Daten war für uns eine große Herausforderung. Zunächst muss ein ähnliches System mit ähnlich funktionierenden Geräten gefunden werden. Dazu muss ein möglichst umfangreicher Datensatz vorhanden sein. Damit ist gemeint, dass der Datensatz sowohl über einen langen Beobachtungszeitraum, als auch über eine Vielzahl an gemessenen Parametern verfügt.

Der von uns genutzte Datensatz¹¹ umfasst zahlreiche Parameter über den Zeitraum von Juni 2012 bis Mai 2014. Neben BHKW, SLK und Wärmespeicher sind auch zwei Photovoltaikanlagen Teil des erfassten Systems. Die für die Validierung genutzten Parameter mit entsprechendem Messintervall sind in Tabelle 1.1 aufgezählt. Die Funktionsweise dieses Energiesystems diente schon bei der Entwicklung der Simulation als Referenz. Viele Verständnisfragen konnten durch die Analyse der Geräte gelöst werden.

Eine wichtige Frage in Bezug auf die Datensätze ist die Vergleichbarkeit mit der Simulation. Die Zusammensetzung der realen Geräte sollte ähnlich zu dem Basis-Szenario aus Abschnitt 1.3 sein. Eventuell vorhandene Zusatzgeräte wie die Photovoltaikanlagen müssen herausgerechnet werden. Dies mindert wiederum die Qualität des Datensatzes.

Des Weiteren muss die Simulation auf das korrespondierende System der Echt Daten eingestellt werden. Neben den Parametern wie Leistung oder Gasverbrauch sind die Werte des Hauses zu ermitteln. Dazu gehören zum Beispiel die Größe der Wohnungen oder die Mieteranzahl. Bei der anschließenden Simulation des realen Szenarios, werden die Eingangswerte der Simulation, die Außentemperatur und der Strombedarf, entsprechend aus den Echt Daten entnommen. Damit ist sichergestellt, dass für die Simulation und das reale System die gleichen Ausgangsbedingungen gelten. Die restlichen Parameter können anschließend verglichen werden. Die Analyse von Fehlerwerten wie in Abschnitt 1.5.2 hilft dabei, die Aussagekraft der Simulation zu bestimmen.

Das Zeitintervall, in dem die Echt Daten gemessen wurden, sollte in der Simulation gleich groß gewählt werden. In Übereinstimmung mit herkömmlicher Messtechnik wurden die meisten Werte in einem Abstand von 15 min gemessen. Einige Parameter des Referenzdatensatzes wurden jedoch nicht automatisiert gemessen und sind somit nur als aggregierter Tageswert gegeben. Für die Raumtemperatur, welche in der Simulation berechnet wird, sind keine Echt Daten vorhanden. Eine Übersicht findet sich in Tabelle 1.1.

In Bezug auf die Qualität der Echt Daten sind Messfehler zu berücksichtigen. Bei der automatisierten Messung sind über einige Zeiträume keine Daten erfasst worden. Dies ist auf technische Probleme der Messgeräte oder auch Störungen der Energiesysteme zurückzuführen. Darüber hinaus ergeben Störungen der Sensoren sporadisch falsche Werte. Erkennbare Beispiele sind negative Temperaturen oder extrem hohe

¹¹BHKW-Betreiber Schmidt.

Tabelle 1.1: Sensoren der Simulation mit vergleichbaren Echtdate

Gerät	Sensor	Einheit	Messintervall der Echtdate
BHKW	Auslastung	%	15 min
	Gasverbrauch	kWh	1 d
SLK	Auslastung	%	15 min
	Gasverbrauch	kWh	1 d
Wärmespeicher	Temperatur	°C	15 min
Stromzähler	Einspeisung	kWh	1 d
	Bezug	kWh	1 d
Verbraucher	Stromverbrauch (Eingabe)	kWh	1 d
	Heizenergie Verbrauch	kWh	1 d
	Energie Warmwasser	kWh	1 d
	Raumtemperatur	°C	keine Echtdate
	Außentemperatur (Eingabe)	°C	1 h

Differenz zwischen aufeinander folgenden Werten. Durch eine Analyse der Echtdate wurde versucht die Daten zu bereinigen, indem hohe Ausschläge gefiltert und fehlerhafte Werte bei der Fehlerberechnung nicht eingezogen wurden.

Schließlich ist festzulegen, welche Parameter des Systems für die Evaluation ausschlaggebend sind. Entscheidend für die Einsparung von Energie ist die Temperatur des Wärmespeichers, da mit einer Vorhersage dieser der Wärmespeicher besser genutzt werden kann. Auch die Auslastung von BHKW und SLK hat einen großen Einfluss auf die anderen Parameter des Systems.

Grafischer Vergleich

Sind Echtdate zu einem simulierten System gegeben, wird häufig auch über den grafischen Vergleich der Datensätze oder Augenscheinvalidierung versucht, die Simulation subjektiv zu validieren [12, 37, 38]. Zum Einen können bei der Entwicklung die Ergebnisse der Simulation visuell schnell eingeschätzt werden. Zum Anderen ist durch Domänenexperten eine subjektive Beurteilung der Graphen möglich.

Beide Methoden der grafischen Validierung wurden angewendet. Während der Entwicklung gab vor allem das durch Liniendiagramme visualisierte Verhalten des Systems Anhaltspunkte, ob Unstimmigkeiten auftraten. Zu hohe Temperaturen im Wärmespeicher oder Auslastungen über 100 % konnten so entdeckt und korrigiert werden. In regelmäßigen Abständen wurde das visualisierte Systemverhalten auch dem Projektpartner und anderen Fachleuten vorgestellt.

Quadratische Abweichung

Um den Unterschied zwischen simulierten und realen Daten genauer zu ermitteln, wird die mittlere quadratische Abweichung (MSD) berechnet. Dies ist ein häufig ge-

nutztes Kriterium zur Bestimmung der Abweichung zwischen zwei Messreihen [26, 50].

$$\text{MSD} = \frac{1}{n} \sum_{t=1}^n (y_t^{\text{real}} - y_t^{\text{simuliert}})^2$$

Dabei bezeichnet n die Anzahl der Messdaten, y_t^{real} einen jeweiligen Sensorwert der Echtdaten und $y_t^{\text{simuliert}}$ ist der jeweilige Sensorwert der Simulation zur gleichen Zeit. Durch das Quadrieren der Differenz wird verhindert, dass sich positive und negative Abweichungen ausgleichen. Weiterhin werden große Abweichungen stärker gewichtet. Für die weitere Evaluation wird die Wurzel der quadratischen Abweichung gebildet, da diese die Fehlergrößen in die Dimension der realen Abweichung versetzt und daher intuitiver zu bewerten ist.

Um die Abweichungen besser interpretieren zu können, kann weiterhin der Anteil an ausreichend genau berechneten Werten bestimmt werden. Dazu muss zunächst für jeden Parameter eine Maximaldifferenz festgelegt werden, ab welcher der simulierte Wert als fehlerhaft angesehen wird. Ist die Differenz von realem und simuliertem Wert geringer als diese Differenz, gilt der Wert als ausreichend genau.

1.5.2 Ergebnisse

Für den Vergleich von Simulation und Echtdaten wurden Daten des Jahres 2013 des in Abschnitt 1.5.1 beschriebenen Datensatzes analysiert. So kann das unterschiedliche Verhalten der Simulation zu jeder Jahreszeit berücksichtigt werden.

Gasverbrauch von Blockheizkraftwerk und Spitzenlastkessel

Der Vergleich des Gasverbrauchs ist in Tabelle 1.2 zu sehen. Daraus kann abgeschätzt werden, ob der Bedarf an Energie in der Simulation ungefähr in der gleichen Dimension wie der des realen Mehrfamilienhauses ist. Weiterhin wird der Gasverbrauch für die Ermittlung der Kosten benötigt.

Bei der Summierung der Gesamtkosten des Jahres fällt auf, dass die realen Kosten für beide Geräte höher ausfallen. Der Verbrauch des SLKs beträgt im Referenzsystem mehr als das fünffache des simulierten Verbrauchs. Die simulierten Messwerte des Verbrauchs des SLKs lagen ausschließlich unter denen des realen SLKs (99,2%). Die Werte des BHKWs hatten eine größere Streuung. Jeder zweite Wert (51,0%) der simulierten Verbrauchsdaten war höher als in der Realität gemessen.

Allgemein ist der Gesamtgasverbrauch des simulierten BHKWs in der Größenordnung des realen BHKWs. Darauf aufbauend wird das Verhalten untersucht, da es der eigentliche Zweck der Simulation ist, den Zustand der Geräte für kurze Zeit vorherzusagen. Anhaltspunkte dafür ist die Auslastung, die im Folgenden evaluiert wird. Auch die Abweichung der einzelnen Messwerte kann besser durch die Betrachtung der jeweiligen Auslastung eingeschätzt werden.

Im Gegensatz zum BHKW ist der Verbrauch des SLKs wesentlich schlechter simuliert worden. Dies könnte an falschen Annahmen in der Simulation liegen, was in Abschnitt 1.5.3 näher erläutert wird.

Tabelle 1.2: Vergleich der Verbrauchswerte von Gas

Gerät	Wurzel quadratischer Abweichung	Gesamtverbrauch Real (kWh)	Gesamtverbrauch Simuliert (kWh)
BHKW	105,7	160 170,0	149 756,5
SLK	295,2	159 439,9	30 453,8

Auslastung von Blockheizkraftwerk und Spitzenlastkessel

Die Werte zur Auslastung von BHKW und SLK wurden wie aus Tabelle 1.1 ersichtlich in 15 min Abständen gemessen. Messfehler können bei diesen Daten nicht bestimmt werden. Im Gegensatz zu der Temperatur des Wärmespeichers, bei dem 0 °C auf einen Messfehler schließen lässt, spiegelt 0 % Auslastung den Zustand ausgeschalteter Geräte wider. Anzunehmen ist ein ähnlicher Anteil an fehlerhaften Messwerten wie bei der Temperatur des Wärmespeichers in Abschnitt 1.5.2.

Die Wurzel der quadratischen Abweichung der Auslastung des BHKWs ist relativ hoch (Tabelle 1.3). Das deutet darauf hin, dass die simulierten Werte oft sehr große Abweichungen haben. Bei der Betrachtung der prozentual fehlerhaften Werte wird dies bestätigt: Jeder zweite Wert (49,6 %) der simulierten Auslastung des BHKWs liegt innerhalb eines Intervalls von 10 % um den gemessenen Wert. Bei einer Maximaldifferenz von 1 % sind jedoch nur 5 % mehr Simulationswerte außerhalb des Intervalls.

Die Auslastung des SLKs kann über die Abweichung der Werte nicht gut eingeschätzt werden, da diese nur die Werte 0 % und 100 % annehmen kann. Bei den binären Werten ist der Anteil an fehlerhaften Werten somit deutlich aussagekräftiger als die quadratische Abweichung. Unabhängig von dem akzeptierten Intervall hat circa jeder vierte Wert (27,1 %) der simulierten Auslastung einen größeren Abstand zum gemessenen Wert.

Temperatur des Wärmespeichers

Der grafische Vergleich der Temperatur des Wärmespeichers ist in Abbildung 1.6 zu sehen. Dazu wurde für jeden Tag eine Durchschnittstemperatur errechnet. Wie in Abschnitt 1.5.1 beschrieben, wurden dabei fehlerhafte Werte nicht berücksichtigt. Die daraus resultierende Abweichung im Monat Juni, in dem kaum Echt Daten gemessen wurden, ist deutlich erkennbar. Im Allgemeinen ist die jährliche Veränderung der Temperatur sehr gut nachempfunden worden. Die Simulation weicht jedoch von kurzfristigen Schwankungen der realen Temperatur ab.

Vergleicht man Echtwert mit simuliertem Wert der Wärmespeichertemperatur, beträgt die Wurzel der quadratischen Abweichung 9,3 (Tabelle 1.3). Das bedeutet, dass die simulierte Temperatur im Durchschnitt pro Messung um 9,3 °C abweicht. Ein Ausgleich des Fehlers durch wechselnd zu hoch- oder zu niedrig-simulierter Werte, ist durch das Quadrieren gefiltert. Wählt man als akzeptable Temperaturdifferenz den Wert der quadratischen Abweichung sind 37,5 % der Werte außerhalb des Intervalls. Dies erhöht sich jedoch auf 76,8 %, wenn die annehmbare Abweichung nur

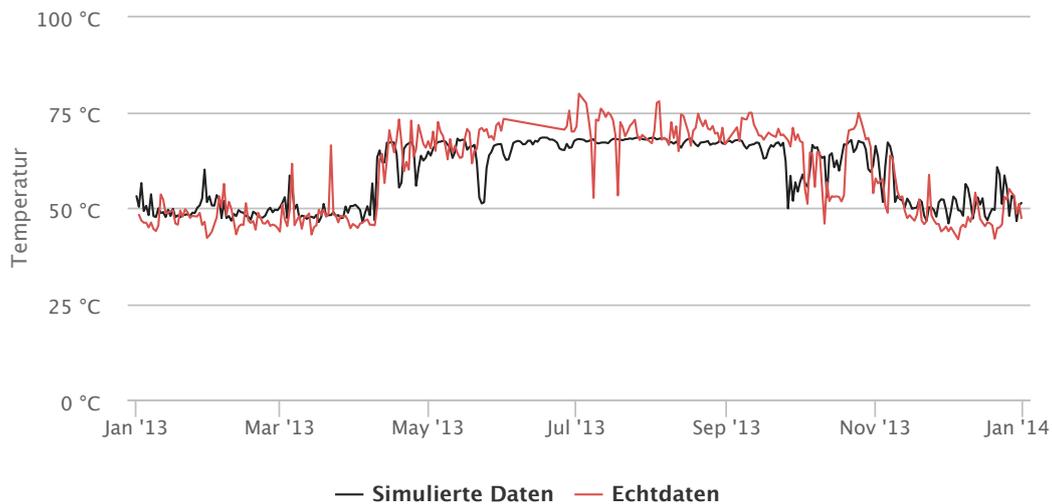


Abbildung 1.6: Vergleich der Wärmespeichertemperatur für das Jahr 2013

Tabelle 1.3: Wurzel der quadratischen Abweichung

Sensor	Abweichung
BHKW Auslastung (%)	47,5
SLK Auslastung (%)	51,6
Wärmespeichertemperatur (°C)	9,3
Strom Einspeisung (kWh)	33,4
Strom Bezug (kWh)	17,4
Thermaler Bedarf (kWh)	97,2
Warmwasser Bedarf (kWh)	32,7

3,5 °C beträgt. Der Anteil unberücksichtigten Daten wegen Messfehlern ist 18,5 % und summiert sich damit auf circa zwei Monate.

1.5.3 Diskussion

Bei der Interpretation der Ergebnisse muss bedacht werden, dass wir die Simulation nur mit einem Datensatz realer Werte vergleichen konnten. Dieses reale System ist zwar mit dem Basis-Szenario vergleichbar, jedoch werden von dem Betreiber der Geräte schon zahlreiche Änderungen vorgenommen. Das Verhalten der Geräte entspricht somit nicht dem eines geschlossenen Systems ohne äußere Einflüsse wie die externe Steuerung. Weiterhin müssten mehrere unterschiedliche Datensätze vorliegen, um aussagekräftige Ergebnisse zu erzielen.

Die großen Unterschiede bei der Simulation des Gasverbrauchs könnten in der vereinfachten Betrachtung der Eingangsenergie liegen. In der Realität schwankt der Gasbezug bezüglich unterschiedlicher Brennwerte. Diese werden bei der wirtschaft-

lichen Berechnung von Gaskosten einbezogen. Simuliert wird jedoch ein konstanter Bezug von $0,1 \text{ m}^3$ Gas pro kWh Eingangsenergie. Weiterhin werden die angegebenen Wirkungsgrade vom BHKW oft „nur unter Laborbedingungen bei den Herstellern“¹² erreicht. Somit sollten diese, gerade bei älteren BHKWs, eventuell geringer eingegeben werden, als vom Hersteller bescheinigt.

Die hohen Abstraktionen bei der Berechnung des Wärmebedarfs spiegeln sich auch in der Genauigkeit wider. Durchschnittswerte für Fenster und Raumgrößen lassen sich für das gesamte Haus schwer ermitteln und führen zu großen Abweichungen. Auch die Vereinfachung der Leitungsverluste macht sich durch den insgesamt geringeren thermalen Bedarf bemerkbar. Dies lässt sich aus dem deutlich geringeren Gasverbrauch folgern.

Die Gründe für die ungenaue Simulation des SLKs liegen an einem falsch angenommenen Verhalten. Durch die Simulation wird bereits ein effizient betriebener SLK angenommen. Im Gegensatz dazu wird der SLK in der Realität schlechter genutzt. Der simulierte SLK wird, wie in Abschnitt 1.4.2 beschrieben, nur angeschaltet, wenn die Wärmeversorgung des BHKWs unzureichend ist. Dies entspricht dem Idealfall des Basis-Szenarios. Bei den Echtdateen treten durch unzureichende Steuerung Effekte auf, die das BHKW gegenüber dem SLK benachteiligen. Beispielsweise benötigt das BHKW eine längere Zeit zum Erreichen der Zieltemperatur, weshalb der SLK vorher eingeschaltet wird. Diese Fälle wurden bei der Simulation bereits optimiert und wirken sich so negativ auf die Realitätstreue der Simulation aus. Es müsste untersucht werden, ob ein geändertes Verhalten des SLKs zu einem realitätsgetreueren Verhalten führen kann. Zum Beispiel kann die Minimaltemperatur des Wärmespeichers entsprechend angehoben werden, wodurch der SLK dann schon bei höheren Temperaturen angeschaltet werden würde.

Die Wärmespeichertemperatur kann, über einen größeren Zeitraum betrachtet, mit akzeptabler Genauigkeit simuliert werden (Abbildung 1.6). Wird der Zeitraum verkleinert werden größere Abweichungen sichtbar. Gerade diese kurzfristigen Änderungen der Temperatur werden jedoch bei der effizienteren Aussteuerung benötigt. Die Ursache könnte darin liegen, dass die Schichtung des Wärmespeichers nicht simuliert wird. Durch die teils großen Temperaturunterschiede innerhalb des Speichers kann der zur Messung der Echtdateen ausgewählte Sensor stark von der Durchschnittstemperatur des Wärmespeichers abweichen. In der Simulation wird jedoch nur eine Durchschnittstemperatur bestimmt.

1.6 Zusammenfassung

Abschließend können wir festhalten, dass die Zielstellung zur Entwicklung der Simulation teilweise erfüllt wurden. Wir haben unser Verständnis der Domäne vertieft und konnten die Zusammenhänge einzelner Systeme dadurch besser verstehen. Auch sind wir im Stande die Auswirkungen von ecoControl zu testen, da kein

¹²BHKW-Betreiber Schmidt.

reales System mehr benötigt wird. Durch eine beschleunigte Simulation konnten wir zudem die Ergebnisse schneller analysieren. Zusätzlich lässt sich das simulierte System einfacher steuern, da keine Hardwareschnittstelle erforderlich ist.

Die Schlüsse, die wir aus dem Verhalten der Simulation ziehen, sollten jedoch genau betrachtet werden. Die zum Teil größeren Abweichungen zu dem betrachteten Referenzsystem könnten zu anderen Ergebnissen führen, als es ein echtes System tun würde. Gerade in kleinen Zeitintervallen betrachtet kann sich das Verhalten des simulierten Systems stark von dem des realen Systems unterscheiden.

In den Kapiteln 4 und 5 wird beschrieben, wie wir die Simulation für Vorhersagen in einem realen System nutzen. Bei den Vorhersagen der Optimierungsalgorithmen werden Simulationen kurzer Zeitintervalle genutzt und der Zustand der simulierten Geräte regelmäßig mit dem der realen Geräte abgeglichen. Dadurch könnten große Abweichungen verhindert werden, da die Simulation statt auf anders simulierten, auf den korrekten Echtdateien fortgesetzt wird. Ob dies Auswirkungen auf die Genauigkeit hat, bedarf weiterer Analysen.

2 Nutzerstudien für die Entwicklung von ecoControl

Neben der Dezentralisierung der Energieversorgung ist auch die nachhaltige Nutzung von Energie ein wichtiger Schritt in Richtung Energiewende. In Mehrfamilienhäusern mit mehreren heterogenen Energiesystemen besteht die Herausforderung für die Betreiber darin, die Systeme auf den Energiebedarf abzustimmen und so die Energie effizient zu nutzen. Derzeit gibt es wenig Hilfestellung für die Optimierung von Energiesystemen.

Um eine Software zu entwickeln, die dabei helfen kann, muss die Funktionalität genau auf die Nutzer abgestimmt werden. Verschiedene Methoden für Nutzerstudien haben bei der Entwicklung von der Software ecoControl geholfen, die Anforderungen und Wünsche der Zielgruppen zu definieren. Die Software hilft, wie von den Nutzern gewünscht, sowohl bei der Optimierung als auch bei der Verwaltung von Energiesystemen. Techniker können mit Hilfe von ecoControl optimierende Einstellungen an den Energiesystemen vornehmen. Für verwaltende Personen bietet die Software eine Übersicht über die Bilanzen verschiedener Jahre. Die Betreiber mehrerer heterogener Energiesysteme können so in Zukunft einfacher energiesparende Maßnahmen ergreifen.

2.1 Zielsetzung

Das Interesse am Thema Energiesparen ist in den letzten Jahren stark gestiegen. Inzwischen gibt es viele Softwarelösungen, die dabei helfen sollen, Energie effizienter zu nutzen und Kosten einzusparen.

Die von uns vorgestellte Softwarelösung ecoControl konzentriert sich besonders auf die Eigenproduktion von Strom und Wärme mit Fokus auf Blockheizkraftwerke (BHKW). Allein durch den Einsatz eines BHKWs im eigenen Haus kann Energie effizienter genutzt werden. Mehr hierzu befindet sich in Kapitel 5. Bei dem Einsatz von BHKWs gibt es weitere Einsparungspotentiale. Folgende These war Ausgangspunkt für unser Projekt:

Es kann Energie eingespart werden, wenn die Energieproduktion genau auf den Bedarf abgestimmt ist. Die verschiedenen heterogenen Geräte in einem Haus arbeiten nicht effizient zusammen. Dadurch bleiben Energieeinsparpotentiale ungenutzt. Durch eine softwaregestützte zentrale Steuerung können diese Potentiale ausgeschöpft und Energiekosten gespart werden.

Als Spezifikation für die Entwicklung einer Software ist dies natürlich nicht ausreichend. Bevor wir mit der Entwicklung von ecoControl beginnen konnten, mussten wir viel über BHKWs und andere Energiesysteme wie Photovoltaikanlagen lernen.

Danach haben wir versucht, die These zu bestätigen, Einsparpotentiale zu finden und diese auszunutzen.

ecoControl ist eine Softwarelösung, die ihren Nutzern dabei hilft, die Eigenproduktion von Strom und Wärme effizienter zu gestalten. Daher musste vor der Entwicklung zuerst die Zielgruppe definiert werden. Die Anforderungen an unsere Software konnten dann aus dem Verhalten, den Bedürfnissen und den Wünschen der Zielgruppen extrahiert werden.

Damit Energie und Kosten gespart werden können, muss ecoControl die Funktionalität bereitstellen, die der Nutzer benötigt, um eine Optimierung durchzuführen. Nutzerstudien haben uns dabei geholfen, die Anforderungen der Nutzer zu erfüllen und die Benutzeroberfläche intuitiv bedienbar zu machen.

Im Folgenden wird dargelegt, wie wir bei der Durchführung von Nutzerstudien vorgegangen sind und welche Methoden wir genutzt haben. Danach werden die Ergebnisse dargestellt und die gefundenen Anforderungen in Beziehung zu den einzelnen Komponenten der Software gesetzt.

Die Zielgruppe unserer Software war zu Anfang unklar. Bei der Durchführung von Nutzerstudien stellte besonders das Recruiting der Teilnehmer eine Herausforderung dar. Es gibt zwei Gruppen, die an einer Software für die effiziente Steuerung von Energiesystemen interessiert sind. Techniker und verwaltende Personen. Bisher kann oder will keine der involvierten Personen eine Optimierung der Energiesysteme vornehmen.

ecoControl ermöglicht es, durch Vorhersagen die Hemmschwelle für Techniker zu senken, optimierende Einstellungen am BHKW vorzunehmen. Auch die Beobachtung und die Analyse der verschiedenen Energiesysteme und die Erstellung von Abrechnungen werden für die Verwaltung vereinfacht.

2.2 Methoden

Es gibt eine Vielzahl von Methoden für Nutzerstudien. Meist bringt die Kombination mehrerer Methoden die größten Vorteile [35]. Je nach Zielgruppe, Zeit und Budget sind verschiedene Methoden für Nutzerstudien anwendbar und hilfreich. Bedingt durch einige Herausforderungen bei der Art und Anzahl der Studien-Teilnehmer (siehe Abschnitt 2.3.1) waren für uns nicht alle gängigen Methoden anwendbar.

In diesem Abschnitt sollen die von uns genutzten Methoden für Nutzerstudien vorgestellt werden. Außerdem soll erläutert werden, welche Rolle diese in unserem Projekt gespielt haben. Der Fokus lag bei uns auf der Interaktion mit dem Nutzer, also den qualitativen Methoden [35].

Die genutzten Methoden lassen sich dabei in drei Phasen untergliedern. Zuerst müssen Anforderungen gesammelt werden. Danach müssen die Ergebnisse ausgewertet werden. Im dritten Schritt können erste Ideen oder auch das Endprodukt getestet werden.

2.2.1 Anforderungen sammeln

Am Anfang des Projektes müssen zuerst die Domäne erforscht und die Zielgruppen definiert werden. Wichtig war es hier, viele Informationen durch die Nutzerstudien zu sammeln.

Interviews

Interviews sind persönliche Gespräche mit Teilnehmern der Nutzergruppen. Sie können auch über das Telefon durchgeführt werden. Der Interviewer bereitet im Voraus einige Fragen vor und stellt sie dem Teilnehmer. Das Interview wird hier von dem Interviewer geleitet, es kann aber trotzdem ein Gespräch zustande kommen. In Interviews kann in einer kurzen Zeit sehr viel über die Nutzer herausgefunden werden. Sie sind eine sehr effektive Methode, um Anforderungen zu identifizieren und die Domäne näher kennenzulernen [34].

Interviews waren unsere meist genutzte Methode für Nutzerstudien. Durch das Gespräch mit den Teilnehmern ergaben sich immer weit mehr Erkenntnisse als erwartet. Am effizientesten waren die Interviews, die persönlich stattfanden. Aber auch die spontanen telefonischen Interviews gaben Antworten auf gerade aktuelle Fragen und konnten in einer sehr kurzen Zeit großen Nutzen bringen.

Contextual-Inquiry

Contextual-Inquirys basieren auf dem Master-Apprentice Modell des Lernens [11]. Der Interviewer lernt dadurch, dass eine Person der Zielgruppe gleichzeitig befragt und bei der Ausführung ihrer Aufgaben beobachtet wird. Während des Contextual-Inquirys erläutert der Teilnehmer seine Arbeitsabläufe in seiner gewohnten Arbeitsumgebung. Der Interviewer bekommt Informationen darüber, wann, wie oft, wie und wo die Person diese Aufgaben ausführt und kann ebenfalls Emotionen beobachten. So bietet das Contextual-Inquiry tiefe Einblicke in den Arbeitsalltag der Zielgruppe [11].

Aus Contextual-Inquirys konnten wir die meisten nützlichen Informationen ziehen. Leider ist diese Methode für beide Seiten sehr zeitintensiv.

2.2.2 Anforderungen mit Personas auswerten

Die Ergebnisse aus Interviews und Contextual-Inquirys wurden von uns meist in Form von Notizen festgehalten. Umso mehr Informationen hinzukommen, desto schwieriger wird es, alle Ergebnisse im Blick zu behalten.

Der beste Weg, es vielen verschiedenen Nutzern recht zu machen ist es, für einen spezifischen Personentyp mit spezifischen Bedürfnissen zu designen. (Übersetzt nach [11]).

Personas sind detaillierte, zusammengesetzte Nutzer-Stereotypen, die unterschiedliche Gruppen von Verhaltensweisen, Einstellungen, Neigungen, Zielen und Motiven repräsentieren, die in der Nutzerstudie beobachtet und identifiziert wurden (übersetzt nach [11]). Personas helfen dabei, von vielen Einzelpersonen zu abstrahieren.

Während des Entwicklungsprozesses ist es so möglich, für eine konkrete Person zu entwickeln, ohne die gesamte Zielgruppe aus dem Auge zu verlieren.

Die Entwicklung von Personas hat uns geholfen, die gefundenen Anforderungen zu den richtigen Zielgruppen zuzuordnen. Schwierig war für uns danach, unsere entwickelten Personas konsequent zu verwenden.

2.2.3 Ideen testen

Mit den ausgewerteten Informationen konnten erste Prototypen entwickelt und getestet werden. So wird das Risiko minimiert, später im Entwicklungsprozess, große Änderungen durchführen zu müssen. Darüber hinaus kann schon früh geprüft werden, ob erste Ideen zur Umsetzung der Anforderungen richtig umgesetzt werden.

Mock-Ups und Paper-Prototypes

Mock-Ups sind eine einfache aber effektive Methode. Carolyn Snyder fasste dies so zusammen:

Natürlich ist es simpel, aber diese Usability-Testing Methode kann helfen, Probleme zu vermeiden, bevor der Code geschrieben wird. (Übersetzt nach [42]).

Mit Mock-Ups, also gezeichneten Entwürfen der Benutzeroberfläche, können schnell und mit einfachen Mitteln erste Ideen zum Design bereit gestellt werden. Sie können dabei helfen, einen Überblick über die geplante Funktionalität der Software zu geben.

Paper-Prototypes gehen einen Schritt weiter. Zuerst werden die zu testenden Komponenten der Benutzeroberfläche gezeichnet. Danach wird dem Testteilnehmer eine Aufgabe gegeben, die er bewältigen soll. Eine weitere Person simuliert währenddessen das Verhalten des Systems.

So kann schon vor der Programmierung ausprobiert werden, wie der Nutzer mit der geplanten Software interagieren wird. Diese beiden Usability-Testing Methoden können schon zu Anfang der Entwicklung angewendet werden. Design-Probleme können so behoben werden, bevor sie kostenintensiv werden [41].

Die Mock-Ups haben uns geholfen, erste Ideen festzuhalten und diese unseren Zielgruppen zu vermitteln. So konnten wir ungeeignete Ideen schon vor der Umsetzung verwerfen. Leider waren für uns die Paper-Prototypes weniger relevant, da es schwierig war, Teilnehmer für diese Art des Testings zu finden.

High-Fidelity-Prototypes

Der High-Fidelity-Prototype ist ein Prototyp, der dem Endprodukt schon sehr ähnlich ist. Die Testteilnehmer interagieren mit einem High-Fidelity-Prototype fast wie mit dem richtigen Endprodukt. Daher kann er sehr gut für das Usability-Testing verwendet werden. Diese Art von Prototyp wird vor allem später im Entwicklungsprozess verwendet, wenn die meisten Design-Entscheidungen schon getroffen sind. Nachteil ist hier, dass diese Art von Prototyp aufwendiger zu erstellen ist, als Paper-Prototypes oder Mock-Ups [41].

Bei uns war der High-Fidelity-Prototyp eine Möglichkeit, zu prüfen, ob unsere Pläne für die Entwicklung von ecoControl sinnvoll waren. Der Prototyp fokussiert

sich auf die Benutzeroberfläche von ecoControl. Genutzt wurden hier die gleichen Technologien (HTML und JavaScript) wie das Endprodukt auch nutzen wird. Ein Vorteil für uns war dabei, dass unsere Software, also auch der High-Fidelity-Prototyp, in jedem Browser läuft. So konnten wir per Screen-Sharing schon vor der Entwicklung des Endprodukts Feedback auch von Nutzern bekommen, die wir nicht besuchen konnten.

Usability-Testing mit Thinking-Aloud-Method

Während eines Usability-Tests wird das Endprodukt oder ein Prototyp von Personen aus der Zielgruppe getestet. Der Teilnehmer muss durch Interaktion mit der Software verschiedene Aufgaben erfüllen. Der Interviewer beobachtet währenddessen, mit welchen Klicks und Navigationspfaden die Aufgaben erfüllt wurden und achtet gleichzeitig auch auf Gesichtsausdrücke, um Emotionen ablesen zu können. Geholfen hat dabei auch die „Thinking Aloud Method“ bei welcher der Teilnehmer seine Gedanken laut ausspricht [11]. So kann nachvollzogen werden, warum der Nutzer wie mit der Software interagiert und wo Missverständnisse vorliegen.

Da bei uns das Usability-Testing oft per Screen-Sharing vorgenommen wurde, war für uns die Thinking-Aloud-Method sehr hilfreich. Getestet wurden so der High-Fidelity-Prototyp sowie das Endprodukt. Schließlich sind wir dazu übergegangen, auch Teilnehmer, die wir besucht haben, aufzufordern, ihre Vorgehensweise zu erläutern.

Hallway-Testing

Eine weitere Form des Usability-Testings ist das Hallway-Testing. Hier werden statt Personen, die zu der Zielgruppe gehören, zufällig ausgesuchte Personen als Testperson gewählt [14]. Besonders wenn die Zielgruppe schwer erreichbar ist (siehe Abschnitt 2.3.1), ist dies eine Möglichkeit, die Bedienbarkeit der Software festzustellen. So kann das Produkt mit einer größeren Varietät von Teilnehmern getestet werden und zu ganz neuen Erkenntnissen führen.

Für uns war das Hallway-Testing ein zuverlässiger Weg, Fehler in der Software festzustellen und Uneindeutigkeiten der Benutzeroberfläche zu verbessern. Über die Praxisrelevanz unserer Anwendung konnten wir allerdings nur begrenzt viel herausfinden.

Umfragen

Umfragen gehören eigentlich zu den quantitativen Methoden, da sie keine direkte Interaktion mit dem Nutzer erfordern. Es geht darum, von möglichst vielen Nutzern mit einer beliebigen örtlichen Distanz und begrenzter Zeit, Antworten auf Fragen zu bekommen. Umfragen werden oft dazu genutzt, die Ziele und Ressourcen des Nutzers kennenzulernen [34].

Durch Umfragen konnten wir auch Testteilnehmer, die nicht zu uns kommen konnten oder wenig Zeit hatten, in das Usability-Testing involvieren. Dazu haben wir die eigentliche Idee der Umfragen etwas abgewandelt. In unserem Fall bestand die Umfrage aus Aufgaben, die der Nutzer mit unserer Software bewältigen sollte. Danach wurden Fragen zu ihrem Vorgehen gestellt. Für Teilnehmer, die wir nicht

besuchen konnten, hat die Umfrage das traditionelle Usability-Testing mit Thinking-Aloud-Method ersetzt.

2.3 Vorgehensweise

Im Folgenden soll unsere Vorgehensweise bei der Durchführung von Nutzerstudien erläutert werden. Außerdem gehen wir darauf ein, welche Herausforderungen wir zu bewältigen hatten und wen wir für unsere Nutzerstudien gewinnen konnten.

2.3.1 Herausforderungen bei dem Recruiting

Die größte Herausforderung bestand bei uns in dem Recruiting von Teilnehmern. Normalerweise besteht das Recruiting aus drei Schritten [17]:

1. Zielgruppe finden und definieren (siehe Abschnitt 2.4.3).
2. Repräsentative Mitglieder der Zielgruppe finden.
3. Repräsentative Mitglieder der Zielgruppe zu einer Teilnahme an der Nutzerstudie überzeugen.

Wir hatten bei den verschiedenen Personengruppen Schwierigkeiten, repräsentative Teilnehmer zu finden und diese zur Teilnahme zu motivieren:

BHKW-Eigentümer

Es gibt derzeit noch nicht viele Mehrfamilienhäuser, die BHKWs einsetzen. Die Personengruppe mit dem größten Interesse an ecoControl sind die BHKW-Eigentümer. Mehr dazu kann in Abschnitt 2.4.3 nachgelesen werden. Die Anzahl der Eigentümer von Mini-KWK-Anlagen, die bei dem Bundesamt für Wirtschaft und Ausfuhrkontrolle gemeldet sind, liegt deutschlandweit bei bisher 4270 Mini-KWK-Anlagen, 89 davon in Berlin [8].

Es gibt also schon eine große Anzahl von Personen, die sich mit BHKWs auskennen und uns Auskunft geben könnten. Die Schwierigkeit liegt allerdings darin, diese Eigentümer zu identifizieren und für eine Teilnahme an Interviews oder anderen Methoden zu überzeugen. Das BHKW-Diskussionsforum [5] hat uns dabei geholfen, einige dieser Personen ausfindig zu machen.

Hauseigentümer und Hausverwalter

ecoControl soll auch die Anforderungen der Hauseigentümer erfüllen, die über den Einsatz von KWK-Anlagen nachdenken. Diese Nutzergruppe ist allerdings noch schwerer zu finden, da wir nicht wussten, welche Hauseigentümer Interesse haben könnten. Da ecoControl auch die Beobachtung der Geräte und Abrechnungen vereinfachen soll, war es für uns wichtig, Informationen von Hausverwaltern zu bekommen.

Beide Rollen, Hauseigentümer und Verwalter, scheinen im Arbeitsalltag so viel zu tun zu haben, dass Sie sich nicht oft Zeit für die Teilnahme an einer Nutzerstudie genommen haben. Dadurch haben wir auf unsere Anfragen oft keine Antworten oder Absagen erhalten.

Planungsfirmen und BHKW-Hersteller

Hauptnutzer von `ecoControl` sind Techniker (siehe Abschnitt 2.4.3). Potentielle Teilnehmer waren daher Planungsfirmen und BHKW-Hersteller. Um es dem Techniker zu erleichtern, Optimierungen vorzunehmen, mussten wir auch die Anforderungen und den Arbeitsalltag dieser Personengruppen näher kennenlernen.

Planungsfirmen und BHKW-Hersteller sind telefonisch zu erreichen, das Interesse an einer Optimierung ist jedoch gering. Bisher verkaufen sich BHKWs auch ohne eine effizienzsteigernde Software. Daher war das Interesse dieser Personengruppen, an unseren Nutzerstudien teilzunehmen, eher mäßig. Teilweise entstand der Eindruck, dass die Angst vor einer Konkurrenz-Software besteht.

2.3.2 Teilnehmer

Wir haben versucht, jede Personengruppe, die zu unserer Zielgruppe gehört, bei unseren Nutzerstudien zu berücksichtigen. Die Zielgruppe wird in Abschnitt 2.4.3 definiert. Im technischen Bereich hatten wir Kontakt zu zwei BHKW-Herstellern über Telefon-Interviews. Darüber hinaus konnten wir in der Anfangsphase eine Planungsfirma im persönlichen Interview erreichen und viel über ihre Arbeit lernen. Über weitere Telefon-Interviews mit BHKW-Betrieben konnten zwischendurch wichtige Fragen geklärt werden. Besonders hilfreich war das Feedback von einem BHKW-Berater ganz am Ende unseres Projektes. Hier konnten wir die Idee unserer Software von einem BHKW-Experten auf Anwendbarkeit prüfen lassen.

Einer unserer Teilnehmer war sowohl mit technischen Aufgaben als auch mit verwaltenden Tätigkeiten vertraut. Von ihm konnten wir viel über Optimierungsansätze für BHKWs erfahren. Bei einem Besuch konnten wir hier seine Arbeitsumgebung näher kennenlernen und ihm viele Fragen stellen. Er stand immer als Interviewpartner bereit und konnte uns sehr viele hilfreiche Daten zur Verfügung stellen.

Im Bereich der Verwaltung konnten wir ein persönliches Gespräch mit einer Mitarbeiterin einer Wohnungsbaugesellschaft führen. Auch sie konnte uns gute Hinweise während des Usability-Testings geben. Darüber hinaus konnten wir hier einem weiteren Hausverwalter Fragen in Form von E-Mails stellen und ein Usability-Testing durchführen.

Sehr hilfreich war es weiterhin, Mitarbeiter unseres Fachgebiets, Kollegen oder Freunde in das Usability-Testing miteinzubeziehen. Fehler konnten so schnell gefunden werden. Sie haben geholfen, `ecoControl` intuitiver zu gestalten.

2.3.3 Abfolge

In Softwareprojekten wird oft ein iterativer Entwicklungsprozess gewählt. Um ein Produkt zu entwickeln, das den Anforderungen des Nutzers entspricht, müssen Versionen des Produkts immer wieder von Nutzern getestet und das Feedback eingearbeitet werden [34]. Mehr zu unserem Entwicklungsprozess findet sich in Kapitel 3.

In Abbildung 2.3.3 wird in BPMN-Syntax dargestellt, wie wir bei der Durchführung unserer Nutzerstudien vorgegangen sind. Das Vorgehen soll mit Hilfe der Nummerierung, die sich im Diagramm wieder findet, näher erläutert werden. Unser iteratives Vorgehen wird im Diagramm durch die vielen Rückpfeile deutlich. Dabei

muss beachtet werden, dass einzelne beschriebene Schritte auch übersprungen werden können. Wenn neue Anforderungen vom Projektpartner hinzukommen, dann kann sogar wieder mit Schritt 1 begonnen werden. Auch wenn Unklarheiten auftreten, ist es sehr sinnvoll, sich nochmal durch Interviews mit den Teilnehmern zu informieren und so Fehlern vorzubeugen.

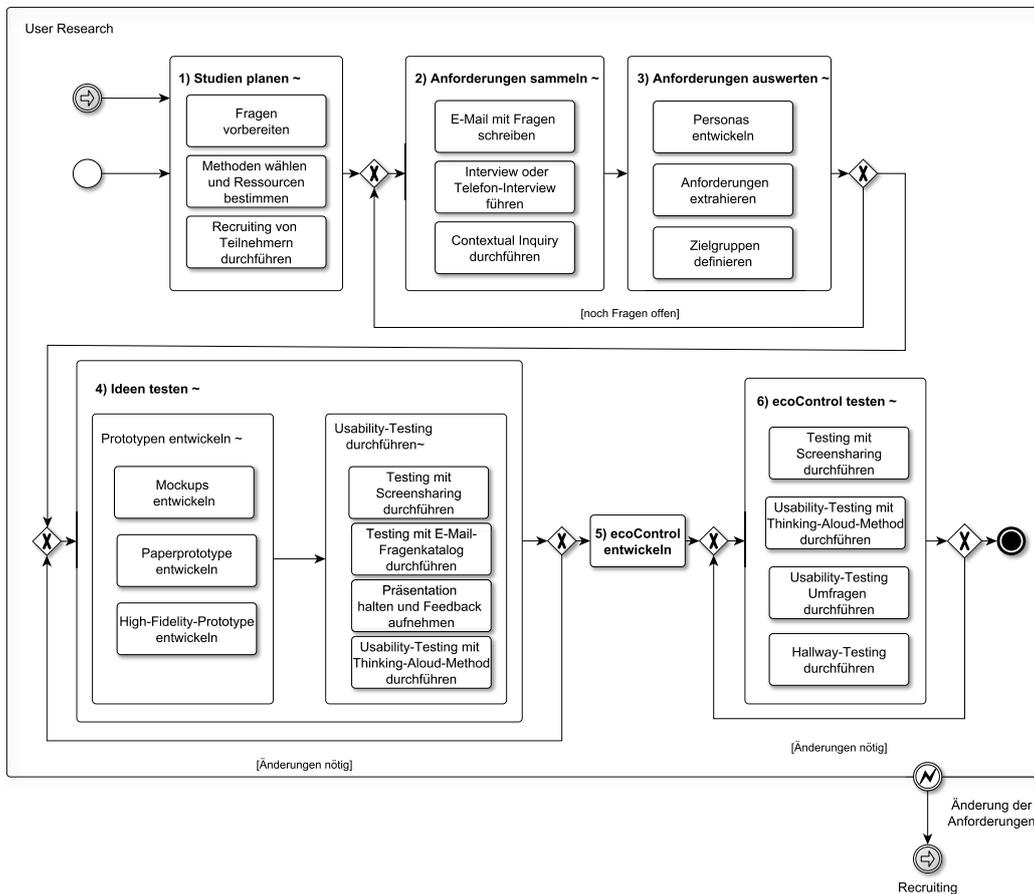


Abbildung 2.1: Darstellung der Vorgehensweise bei den Nutzerstudien

1) Studien planen Um aus den Nutzerstudien wertvolle Erkenntnisse zu gewinnen, haben wir diese geplant und vorbereitet. Zur Planung gehören [17]:

- Vorbereitung der Fragen, die durch Nutzerstudien beantwortet werden sollen. Diese können zum Beispiel in einem Fragenkatalog gesammelt werden.
- Wahl der Techniken, die für die Beantwortung der Fragen genutzt werden sollen (siehe Abschnitt 2.2). Je nach Zeit und örtlicher Distanz der Teilnehmer sind verschiedene Methoden anwendbar.

- Bestimmung der Ressourcen, die für die Nutzerstudien benötigt werden, wie Zeit, Geld, durchführende Personen, wann und wo die Research stattfinden soll.
- Recruiting der Teilnehmer (siehe Abschnitt 2.3.1).

2) Anforderungen sammeln Nach der Planung konnten wir mit den Nutzerstudien beginnen. Ziel war es, die ausgearbeiteten Fragen jetzt mit Hilfe der verschiedenen Methoden und verschiedenen Teilnehmern zu beantworten. Die genutzten Methoden wurden bereits unter Abschnitt 2.2.1 dargestellt.

3) Anforderungen auswerten Daran anschließend war es wichtig, die Ergebnisse der Nutzerstudien gut zu dokumentieren und Anforderungen zu extrahieren. Wie in Abschnitt 2.2.2 beschrieben haben Personas geholfen, die Ergebnisse zu strukturieren. Wir haben außerdem unsere Zielgruppe definiert und Anforderungen an ecoControl ausgearbeitet.

4) Ideen testen Anschließend haben wir erste Prototypen entwickelt, die von den Teilnehmern im Usability-Testing getestet wurden (siehe Abschnitt 2.2.3). Einige Ideen haben wir außerdem auch in Präsentationen vorgestellt. Danach wurden die Prototypen überarbeitet.

5) ecoControl entwickeln und 6) ecoControl testen Die Ergebnisse des Prototypings haben wir dann zur Entwicklung des Endprodukts verwendet. Die ersten Versionen von ecoControl haben wir erneut mit den Methoden des Usability-Testings (siehe Abschnitt 2.2.3) getestet. Den Ergebnissen entsprechend haben wir ecoControl überarbeitet.

2.4 Ergebnisse

Die Motivation für Nutzerstudien ist, die Bedürfnisse der Zielgruppe näher kennenzulernen und so eine Software zu entwickeln, die hilft, die in Abschnitt 2.1 vermuteten Energieeinsparungspotentiale auszunutzen. In Abschnitt 2.3 haben wir bereits die Vorgehensweise erläutert. In diesem Abschnitt sollen die Ergebnisse der Nutzerstudien zusammengefasst werden. Zuerst werden die Ansätze kurz beschrieben. Danach sollen die Ergebnisse dazu in Beziehung gesetzt werden.

2.4.1 Ansätze

Die Fragen, die durch die Nutzerstudien beantwortet werden sollen, lassen sich in unserem Fall in vier Bereiche gliedern: Erforschung des Ist-Zustandes, Definition der Zielgruppe, Anforderung der Zielgruppen und daraus abgeleitet die Funktionalität und Benutzeroberfläche von ecoControl.

1. **Erforschung des Ist-Zustandes** Bei der Erforschung des Ist-Zustandes geht es darum, den zeitlichen Ablauf und die Aufgaben bei Inbetriebnahme eines

BHKWs näher kennenzulernen. Es soll herausgefunden werden, welche Arbeitsabläufe derzeit von wem durchgeführt werden. Es sollen auch bestehende Ansätze zur Optimierung und Energieeinsparung untersucht werden.

2. **Definition der Zielgruppe** Die Definition der Zielgruppen und deren Anforderungen beruht auf den Ergebnissen aus Abschnitt 2.4.2. Es wird recherchiert, welche Nutzergruppen an einer Optimierung interessiert sind. Danach wird herausgearbeitet, wer die Optimierung mit ecoControl durchführen wird.
3. **Anforderungen der Zielgruppen** Hier werden die Wünsche und Bedürfnisse der Zielgruppen zusammengefasst. Aus den Ergebnissen lassen sich dann im nächsten Schritt die Anforderungen an die Software ableiten.
4. **Funktionalität und Benutzeroberfläche** Im nächsten Schritt kann aus den Anforderungen abgeleitet werden, welche Funktionalität die Software bereitstellen soll und welche Benutzeroberfläche sich anbietet.

2.4.2 Erforschung des Ist-Zustandes

In Abbildung 2.2 werden auf der rechten Seite die Arbeitsschritte und auf der linken Seite die zuständigen Personen dargestellt, die bei der Neuanschaffung eines BHKWs beteiligt sind. Die Aufgaben können in vier Schritte eingeteilt werden: Planung, Installation, Verwaltung und Optimierung. Der Schritt Optimierung kann derzeit noch keiner Personengruppe eindeutig zugeordnet werden.

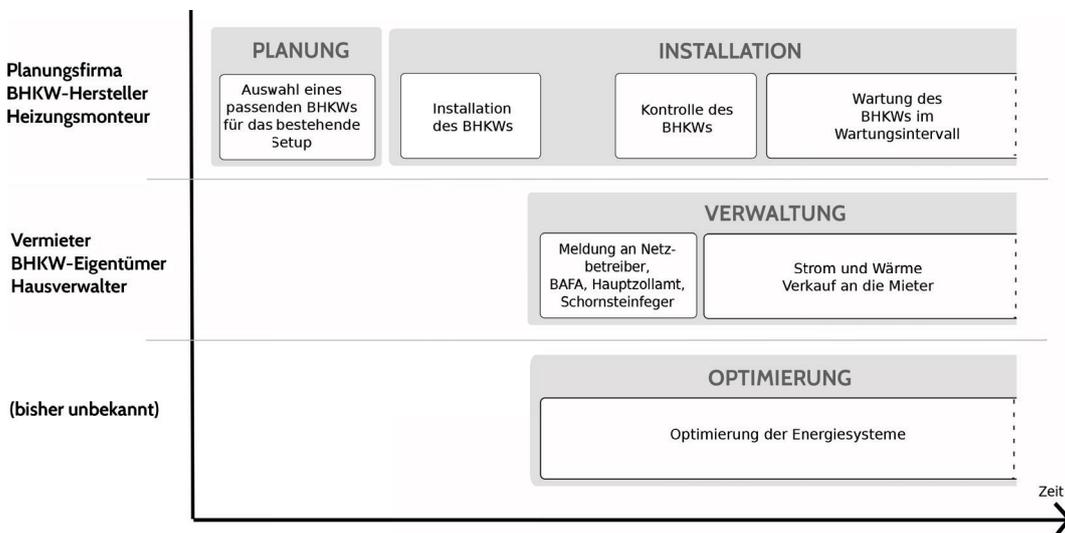


Abbildung 2.2: Aufgaben und beteiligte Personengruppen bei der Inbetriebnahme eines BHKWs

Planung

Im Aufgabenabschnitt Planung geht es darum, das richtige BHKW für das Mehrfamilienhaus auszuwählen. Die Auslegung des BHKWs erfolgt anhand der spezifischen „Wärmegrundlast und dem Strombedarf“ [7]. Oft wird dann mit Tabellen, die den Zusammenhang von Bedarf und der Dimensionierung des BHKWs herstellen, herausgefunden, welches BHKW am besten zu der Immobilie passt und wie wirtschaftlich es laufen wird.

Diese Planungsphase kann entweder von Planungsfirmen, von den BHKW-Herstellern oder Heizungsmonteuren selbst vorgenommen werden. Der Einsatz einer Planungsfirma kann sich lohnen, da sie herstellerunabhängig berät und Wirtschaftlichkeitsberechnungen durchführt [15].

Installation

Nachdem ein zu dem Szenario passendes BHKW gewählt wurde, kann es im Haus installiert werden. Nach der Installation wird die in der Verwaltung zuständige Person mit der Funktionsweise des BHKWs vertraut gemacht. Die Einweisung umfasst einfache Einstellungsmöglichkeiten und Sicherheitshinweise.

Nach einigen Wochen muss die Anlage erneut geprüft werden. Danach wird das BHKW im entsprechenden Wartungsintervall kontrolliert. In der Regel müssen Teile wegen Verschleiß ausgetauscht oder erneuert werden.

Bei Fehlern, die das BHKW beeinträchtigen können, bekommt der Installateur Benachrichtigungen von der Software des BHKWs. Daraufhin kann er den Fehler über die Fernwartungssoftware des BHKW-Herstellers einsehen und kann bei Bedarf eine Reparatur vor Ort vornehmen.

Das BHKW kann entweder über den Heizungsmonteur oder direkt über einen BHKW-Hersteller bezogen werden. Wurde das BHKW über den Heizungsmonteur gekauft, wird Installation und Wartung auch von diesem vorgenommen. Wenn das BHKW direkt über einen Hersteller bezogen wurde, übernimmt ein Installateur der Herstellerfirma diese Aufgaben.

Verwaltung

Im nächsten Schritt muss das BHKW bei dem Bundesamt für Wirtschaft und Ausführungskontrolle, dem Hauptzollamt, dem Netzbetreiber und dem Schornsteinfeger gemeldet werden [7]. Der selbst produzierte Strom und die Abwärme sollen an die Mieter verkauft werden.

Für die Abrechnungen und für die Steuererklärung müssen derzeit Zählerstände von Wärme- und Stromverbrauch oder -produktion oft manuell abgelesen werden. In einigen Fällen werden diese von verschiedenen herstellereigenen Programmen in digitaler Form zur Verfügung gestellt.

Um eine Übersicht über die Bilanz und die Wirtschaftlichkeit des BHKWs zu erhalten, werden diese Daten oft händisch zusammengeführt.

Optimierung

Durch zusätzliche Einstellungen am BHKW kann das BHKW noch effizienter betrieben werden. Mehr dazu findet sich in Kapitel 5. Ein Beispiel dafür ist die Einstellung

der Drehzahl oder des Modus entsprechend des Nutzungsverhaltens und dem Wetter.

In Einzelfällen kann die Optimierung vom BHKW-Eigentümer übernommen werden. Dies ist allerdings nur über eine Software möglich, die normalen Kunden nicht zur Verfügung gestellt wird. Darüber hinaus ist eine solche Optimierung sehr zeitintensiv. Daher wird die eigenständige Optimierung der Geräte nur in sehr seltenen Fällen von dem BHKW-Eigentümer vorgenommen.

Es gibt bereits einige wenige Planungsfirmen, BHKW-Hersteller oder Heizungsmonteur, die nach der Planung auch die Betreuung des BHKWs anbieten. Hier können auch Verträge abgeschlossen werden, die gegen eine Pauschale optimierende Maßnahmen beinhalten. Dies ist allerdings noch nicht sehr weit verbreitet.

Da weder derartige Verträge noch die eigenständige Optimierung den Normalfall darstellen, wollen wir mit ecoControl an dieser Stelle ansetzen.

2.4.3 Definition der Zielgruppe

Die oben genannten Aufgabenbereiche werden momentan von mehreren Personengruppen übernommen. Sie lassen sich in eine Personengruppe mit Schwerpunkt Verwaltung und eine Personengruppe mit Schwerpunkt Technik einteilen. Der Aufgabenbereich Optimierung wird derzeit nur in Sonderfällen abgedeckt.

Unsere Software soll beide Gruppen unterstützen. Daher sprechen wir in unserem Modell von zwei Zielgruppen: Verwaltung und Techniker. Welche Personen zu diesen beiden Zielgruppen gehören, wird im Folgenden erläutert. Darüber hinaus wird geprüft, ob sie geeignet sind, im Aufgabenbereich Optimierung aktiv zu werden.

Zielgruppe Verwaltung

Wie in Abbildung 2.2 dargestellt, kann es je nach Situation verschiedene Personen geben, die mit Verwaltungsaufgaben betraut sind:

- Der *Vermieter* vermietet die Wohnungen und ist oft gleichzeitig auch *Hausbesitzer*.
- Manchmal gibt es einen *Hausverwalter*, der für Abrechnungen und Instandhaltung der Immobilie zuständig ist.
- Der *BHKW-Eigentümer* ist derjenige, der in ein BHKW investiert hat und den produzierten Strom und die produzierte Wärme verkauft.

In vielen Fällen werden diese Rollen von einer einzigen Person übernommen. Es stellt sich die Frage, welche dieser Personengruppen an einer Optimierung interessiert sind.

Dem Vermieter ist es wichtig, dass die Attraktivität der Immobilie steigt. Dies kann er erreichen, in dem er den Mietern Strom günstiger anbietet als die öffentlichen Netzbetreiber. Durch den Einsatz eines BHKWs wird dies ermöglicht.

Der BHKW-Eigentümer verkauft den vom BHKW produzierten Strom zu einem vertraglich festgelegten Preis an die Mieter. Wenn durch Optimierungen Produktionskosten eingespart werden, wird der Differenzbetrag zwischen den Produktionskosten und den vertraglich vereinbarten Stromkosten größer. Hier profitiert der

BHKW-Eigentümer, da der Differenzbetrag seinen Gewinn darstellt. Für den Hausverwalter sind Strom- und Wärmekosten durchlaufende Kosten. Von einer Kostensenkung hat er keine Vorteile.

Daraus folgt, dass sowohl Vermieter als auch Hausverwalter nur dann an einer Geldeinsparung durch Optimierung interessiert sind, wenn ihre Rolle mit der des BHKW-Eigentümers zusammenfällt. Daher führen wir in unserem Modell diese Rollen zusammen. Diese Zielgruppe wird im Folgenden *Verwaltung* genannt.

Es stellt sich die Frage, ob die Verwaltung die Durchführung der Optimierung übernehmen kann. Derzeit gibt es einige engagierte BHKW-Eigentümer, die es sich zur Aufgabe gemacht haben, täglich optimierende Einstellungen vorzunehmen. Dies ist für den durchschnittlichen BHKW-Eigentümer nicht möglich. Ihnen fehlt das technische Wissen und die Zeit. Die Verwaltung hat daher zwar das Interesse an einer Optimierung, wird aber den Aufgabenbereich der Optimierung nicht übernehmen können.

Zielgruppe Techniker

Wie in Abschnitt 2.4.2 beschrieben, werden die technischen Aufgaben meistens von einem Installateur des BHKW-Herstellers oder von einem Heizungsmonteur vorgenommen. Zusätzlich kann eine Planungsfirma an der Planung beteiligt werden. Da die Aufgabenbereiche Planung und Installation beide von einer technisch orientierten Person vorgenommen werden, fassen wir in unserem Modell die hier involvierten Personen zusammen. Die Zielgruppe wird im Folgenden *Techniker* genannt.

Es ist die Frage zu klären, ob der Techniker an einer Optimierung interessiert ist. Im Normalfall scheint die Aufgabe der Optimierung aus den Aufgabenbereichen der Planer, Installateure und Heizungsmonteur zu fallen. Die Planung endet meist schon bei der Entscheidung für ein bestimmtes BHKW. Das primäre Ziel des Installateurs eines BHKW-Herstellers oder des Heizungsmonteurs ist es, das BHKW zu verkaufen und einzurichten. Er profitiert dabei nicht direkt von einer Optimierung, manchmal fehlt auch die Erfahrung, um optimierende Einstellungen an den echten Geräten vorzunehmen. Daher ist es für ihn von geringem Interesse, aufwendige effizienzsteigernde Maßnahmen vorzunehmen.

In wenigen Ausnahmen können mit dem Kauf des BHKWs Verträge abgeschlossen werden, die eine Optimierung der Geräte gegen eine Pauschale beinhalten. Diese Optimierung wird dann direkt vom Vertragspartner vorgenommen. Wenn die Optimierung nicht von der Personengruppe durchgeführt wird, die das BHKW installiert hat, kann es zu rechtlichen Problemen mit der Gewährleistung kommen.

Techniker übernimmt Optimierung

Weder die Zielgruppe Verwaltung noch die Zielgruppe Techniker scheint die Aufgabe der Optimierung bisher übernehmen zu wollen oder zu können. Daraus resultiert, dass der Optimierungs-Teil unserer Software für eine bisher noch nicht existierende Zielgruppe entwickelt wird.

Wir wissen jedoch, dass es von Vorteil ist, wenn die optimierende Person sowohl weiß, wie man ein BHKW richtig dimensioniert, als auch wie Einstellungen vorgenommen werden können. Ohne dieses Wissen könnten keine Optimierungen vorge-

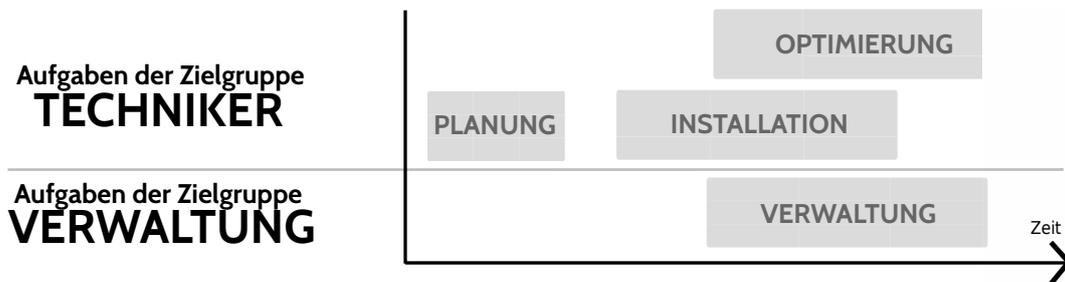


Abbildung 2.3: Zielgruppe Techniker: Zusammenlegung der Aufgaben Planung, Installation und Optimierung

nommen werden. Darüber hinaus sollte sichergestellt werden, dass die Gewährleistung bestehen bleibt. Wie in Abbildung 2.3 dargestellt, muss daher der Techniker zusätzlich zu den Aufgaben der Planung und Installation auch die Optimierung übernehmen.

Mit dieser Annahme ist es weiterhin möglich, dass eine der existierenden Personengruppen, zum Beispiel der Installateur, Optimierungen mit unserer Software vornimmt.

2.4.4 Anforderungen der Zielgruppen

Da die Rolle des Technikers die Aufgabenbereiche Planung, Installation und Optimierung beinhaltet, müssen die Anforderungen aus den drei Aufgabenbereichen abgeleitet und zusammengeführt werden. Die Anforderungen aus dem Aufgabenbereich Verwaltung können direkt durch die Nutzerstudien mit Hausverwaltern, BHKW-Betreibern oder Vermietern abgeleitet werden.

Hier sollen die Anforderungen dargestellt werden. In Abschnitt 2.4.5 erläutern wir, wie ecoControl diese Anforderungen umsetzt. Dazu werden die Anforderungen durchnummeriert. V steht bei dabei für Verwaltung, T für Techniker.

Anforderungen der Zielgruppe Verwaltung

V1: Analyse der Bilanzen

Um für Abrechnungen oder für die Steuererklärung eine Übersicht über die Bilanzen zu erstellen, müssen derzeit die Daten oft manuell zusammen geführt werden. Mindestens einmal im Quartal muss diese Aufgabe durchgeführt werden. Eine Anforderung ist daher, die Daten in aggregierter Form anzuzeigen. So wäre es dann auch möglich, jederzeit Vergleiche zwischen verschiedenen Monaten zu ziehen und die Daten zu analysieren, ohne sie vorher aufbereiten zu müssen.

V2: Fehlerfrüherkennung

Für die Wirtschaftlichkeit eines BHKWs ist es wichtig, dass es möglichst fehlerfrei läuft. Im Fehlerfall des BHKWs muss sonst Strom vom öffentlichen Netz bezogen werden. Dieser Strom ist teurer als der selbst produzierte und es entstehen so zusätz-

liche Kosten. Fehler am BHKW können derzeit immer nur nach Eintritt festgestellt und behoben werden. Gewünscht ist also eine möglichst frühe Fehlererkennung. Umso früher ein Fehler erkannt wird, desto weniger Kosten entstehen. Daher profitiert auch die Verwaltung von Werkzeugen zur Fehlerfrüherkennung.

Anforderungen der Zielgruppe Techniker

Elektrische Leistung BHKW 110 kW
 Heizleistung BHKW 215 kW
 Gasverbrauch BHKW 340 kW
 Nutzungsgrad Heizkessel 88 %

A	B	C	D	E	F	G	H	I
Monat	Gasverbrauch kWh (Hs)	Gasverbrauch kWh (Hi)	Wärmeverbr. kWh	Stunden je Monat	max. mögliche BHKW-Laufzeit	tatsächliche BHKW-Laufzeit	W.erz. BHKW kWh	W.erz. Kessel kWh
Jan	727,200	661,091	581,760	744	744	744	159,960	421,800
Feb	576,000	523,636	460,800	672	672	672	144,480	316,320
Mrz	518,400	471,273	414,720	744	744	744	159,960	254,760
Apr	345,600	314,182	276,480	720	720	720	154,800	121,680
Mai	158,400	144,000	126,720	744	589	389	83,720	43,000
Jun	72,000	65,455	57,600	720	268	268	57,600	0
Jul	79,200	72,000	63,360	744	295	295	63,360	0
Aug	57,600	52,364	46,080	744	214	214	46,080	0
Sep	172,800	157,091	138,240	720	643	443	95,240	43,000
Okt	302,400	274,909	241,920	744	744	744	159,960	81,960
Nov	684,000	621,818	547,200	720	720	720	154,800	392,400
Dez	784,800	713,455	627,840	744	744	744	159,960	467,880
Summe	4,478,400	4,071,273	3,582,720	8,760	7,097	6,697	1,439,920	2,142,800

aus Rechnung Gasversorger = B : 1,1 = C x 88 % wenn D : 215 > E dann E sonst D : 215 kW man. Korrektur (an 2 Monaten je 200 h weniger) = G x 215 kW = D - G

Abbildung 2.4: Tabelle einer Planungsfirma [6]

T1: Kontrolle der Tabellen zur Planung

Wie in Abschnitt 2.4.2 beschrieben, wird die spezifische Planung eines BHKWs derzeit oft über Tabellen vorgenommen. Abbildung 2.4 zeigt beispielhaft eine solche Tabelle. Diese Tabellen beruhen auf Erfahrungswerten. Um eine hinreichende Genauigkeit dieser Daten sicherzustellen, müssen viele Werte gesammelt werden. Die Arbeit mit den Tabellen erfordert also im Anfangsstadium viel manuelle Arbeit, ist dann aber relativ einfach. Bei Planungsfirmen ist diese Herangehensweise üblich.

Da in diesen Tabellen keine stündlichen Werte eines Szenarios, sondern nur aggregierte Werte angegeben werden können, scheint die Planungsart relativ ungenau zu sein. Eine Anforderung ist daher, die gefundene Dimension des BHKWs zu überprüfen. Wichtig könnte zum Beispiel die Überprüfung sein, wie sich die Geräte bei Spitzenlasten verhalten.

T2: Beobachtung und frühe Fehlererkennung

Der Installateur eines BHKWs bekommt im Fehlerfall Benachrichtigungen. Diese betreffen allerdings nur einige wenige systemkritische Fehler. Die genaue Analyse des Systemverhaltens kann auch noch weitere Unstimmigkeiten, wie etwa einen ungewöhnlich hohen Gasverbrauch, aufdecken. Dies erfordert aber viel Zeit und sollte vereinfacht werden. Individuelle Benachrichtigungen, die solche Fälle abde-

cken könnten, sind bisher nicht vom Hersteller vorgesehen, wären aber hilfreich für eine frühe Fehlererkennung.

T3: Einfache und schnelle Optimierungsmöglichkeiten

Ein Techniker ist im Normalfall für mehrere BHKWs in verschiedenen Häusern zuständig. Aufgrund der örtlichen Distanz zum BHKW und des Zeitmangels ist es einem Techniker natürlich nicht möglich, wöchentlich die Daten jedes seiner Objekte zu analysieren und optimierende Einstellungen vorzunehmen. Eine Anforderung an ecoControl ist daher die Möglichkeit einer einmaligen Konfiguration, die sich anschließend automatisch an das spezielle Setup anpasst.

Auch Regeln, die beispielsweise das Wetter oder das Nutzungsverhalten mit einbeziehen, können Energie einsparen. Auch diese sollten möglichst einfach erstellbar sein. Wichtig wäre auch, dass sich die vorgenommenen Einstellungen einfach validieren lassen, damit der Techniker nicht nach wenigen Wochen erneut kommen muss.

Des Weiteren haben einige Techniker trotz ihrer Ausbildung nicht genügend Fachwissen, um Einstellungen vorzunehmen. Die Hemmschwelle, Optimierungen direkt an den Geräten durchzuführen, ist daher teilweise sehr hoch. Für diese Techniker wäre es interessant, die Auswirkungen der Einstellungen vor Anwendung zu überprüfen.

2.4.5 Funktionalität und Benutzeroberfläche

Die Anforderungen finden sich in ecoControl in den Bereichen für den Techniker und die Verwaltung wieder. Wie ecoControl die verschiedenen Anforderungen erfüllt, wird anhand der Komponenten der Benutzeroberfläche erläutert. Dazu wird der direkte Bezug zu den Anforderungen aus Abschnitt 2.4.4 hergestellt.

Einige der Komponenten waren beim Usability-Testing besonders interessant. Daher werden wir in diesem Abschnitt beispielhaft einige Ergebnisse des Usability-Testings darstellen.

Bereich der Verwaltung

Für die Verwaltung stellt unsere Software ein reines Analyse- und Beobachtungswerkzeug dar. Abbildung 2.5 zeigt eine Gesamtübersicht über den Bereich der Verwaltung. Die obere Leiste stellt die Navigationsleiste unserer Software dar. Die einzelnen darunter aufgezählten Komponenten werden im Folgenden näher erläutert und den Anforderungen zugeordnet.

Erfüllung von V1: Analyse der Bilanzen

Übersicht

Der Überblick über die aktuelle Bilanz ist eine wichtige Information für die Verwaltung. Wie in Abbildung 2.6 zu sehen, stellt ecoControl eine kurze Zusammenfassung für den aktuellen Monat, das Amortisations- und Wartungsdatum zur Verfügung. Dadurch, dass ecoControl die notwendigen Daten sammelt und so aufbereitet, dass dem Nutzer eine Übersicht über seine Finanzen gegeben wird, wird der in Abschnitt 2.4.2 beschriebene manuelle Aufwand minimiert. Die Übersichtsseite

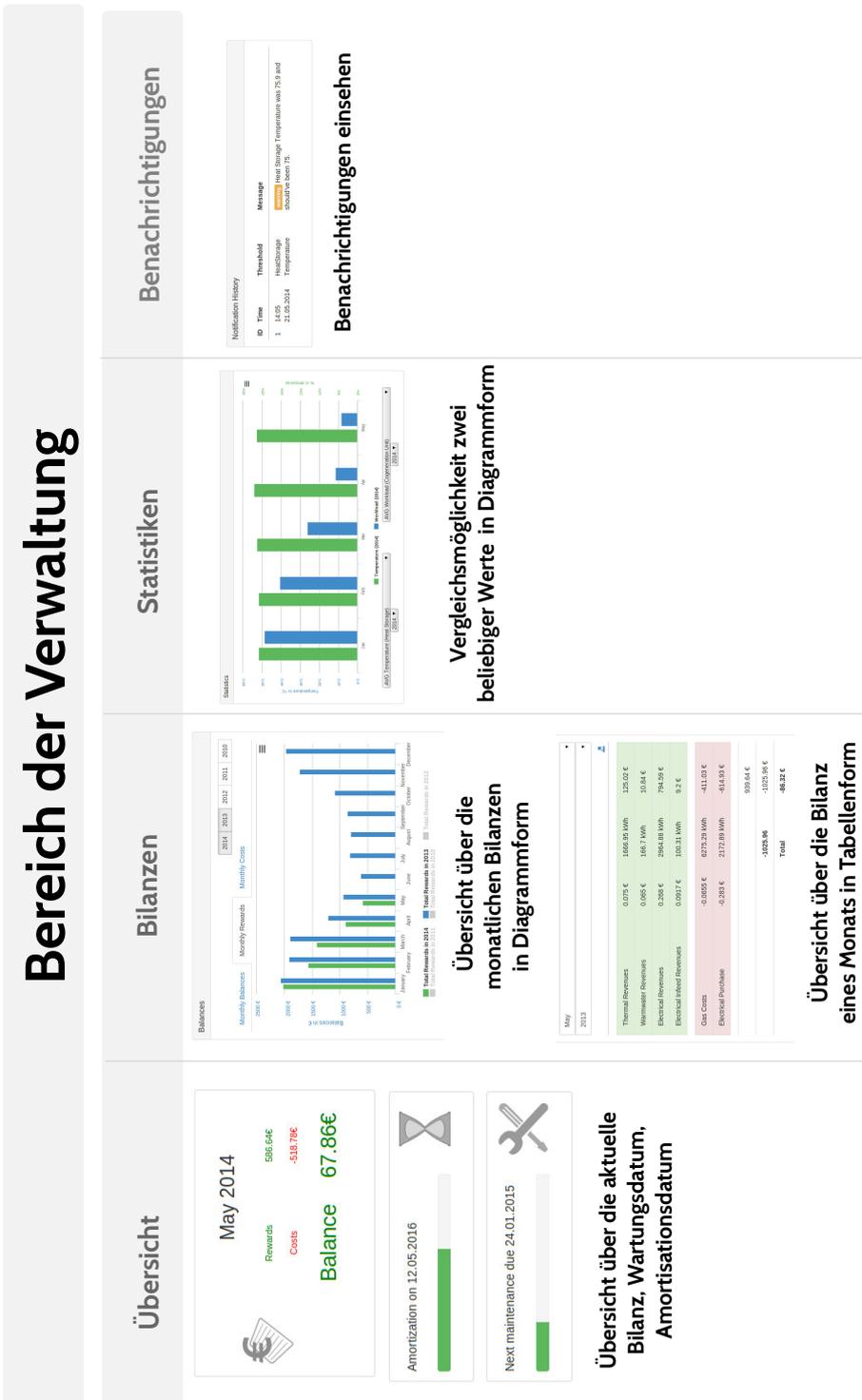


Abbildung 2.5: Übersicht über den Bereich der Verwaltung

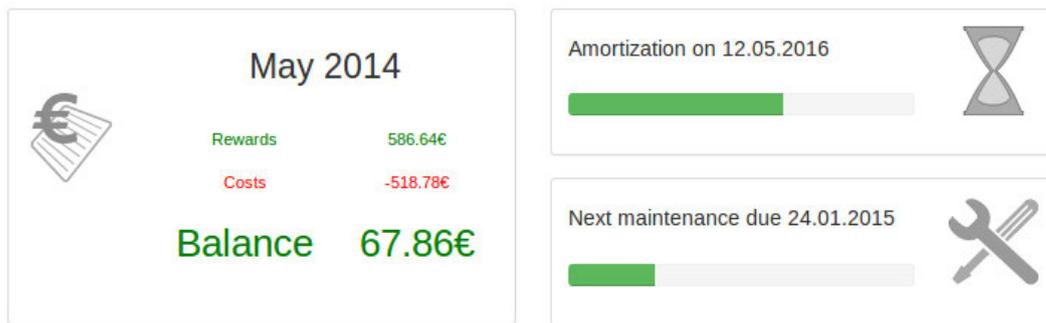


Abbildung 2.6: Übersicht über aktuelle Bilanz, Wartungsdatum und Amortisationsdatum

kann zum Beispiel als erste geöffnete Seite im Browser eingestellt werden. So sieht die Verwaltung jeden Tag die neusten Bilanzen.

Bilanzen

Die detaillierte monatliche Bilanz ist besonders wichtig für Abrechnungen. Der Nutzer erhält nicht nur eine Übersicht über die Einnahmen und Ausgaben eines bestimmten Monats. Mit ecoControl kann er auch die Bilanzen verschiedener Monate oder Jahre miteinander vergleichen (siehe Abbildung 2.7). So können Verbesserungspotentiale gefunden oder Gründe für Bilanzunterschiede in verschiedenen Monaten gesucht werden.

Um verschiedene Bilanzen zu vergleichen, ist eine grafische Darstellung sehr hilfreich. Die Balkengröße kann schneller erfasst und verglichen werden als Einträge in einer Tabelle. Je nach Wunsch können die Bilanzen mehrerer Jahre im Diagramm angezeigt werden. Es ist ebenfalls möglich, nur die Einnahmen oder die Ausgaben grafisch zu untersuchen.

Wurde ein Monat gefunden, der besonderes Interesse weckt, kann dieser durch einen Klick auf den jeweiligen Balken näher analysiert werden (siehe Abbildung 2.8). Zur detaillierten Analyse eignen sich Tabellen besser als Diagramme. Die Verwaltung ist mit Abrechnungen in Tabellenform vertraut und kann diese daher besser überschauen und analysieren.

Unsere Software erlaubt darüber hinaus den Datenexport in das gängige CSV-Format. So können Daten zu Dokumentationszwecken gespeichert und in andere Programme importiert werden. Der Export der monatlichen Bilanzen kann auch die Steuererklärung erleichtern.

Statistiken

Die Einnahmen und Ausgaben verschiedener Jahre sind wahrscheinlich die Daten, die am häufigsten verglichen und analysiert werden sollen. Da wir in unserem System aber eine Vielzahl von Daten sammeln und aggregieren, möchten wir dem Nutzer diese Daten zur Verfügung stellen.

Welche Werte von dem Nutzer verglichen werden möchten, ist sehr verschieden. Daher stellt ecoControl wie in Abbildung 2.9 gezeigt, ein Diagramm, das flexibel



May		2013	
Thermal Revenues	0.075 €	1666.95 kWh	125.02 €
Warmwater Revenues	0.065 €	166.7 kWh	10.84 €
Electrical Revenues	0.268 €	2964.88 kWh	794.59 €
Electrical Infeed Revenues	0.0917 €	100.31 kWh	9.2 €
Gas Costs	-0.0655 €	6275.29 kWh	-411.03 €
Electrical Purchase	-0.283 €	2172.89 kWh	-614.93 €
			939.64 €
		-1025.96	-1025.96 €
Total			-86.32 €

Abbildung 2.7: Vergleichsmöglichkeit der Einnahmen verschiedener Monate und Jahre im Diagramm

Abbildung 2.8: Detaillierte Übersicht der Einnahmen eines Monats in einer Tabelle

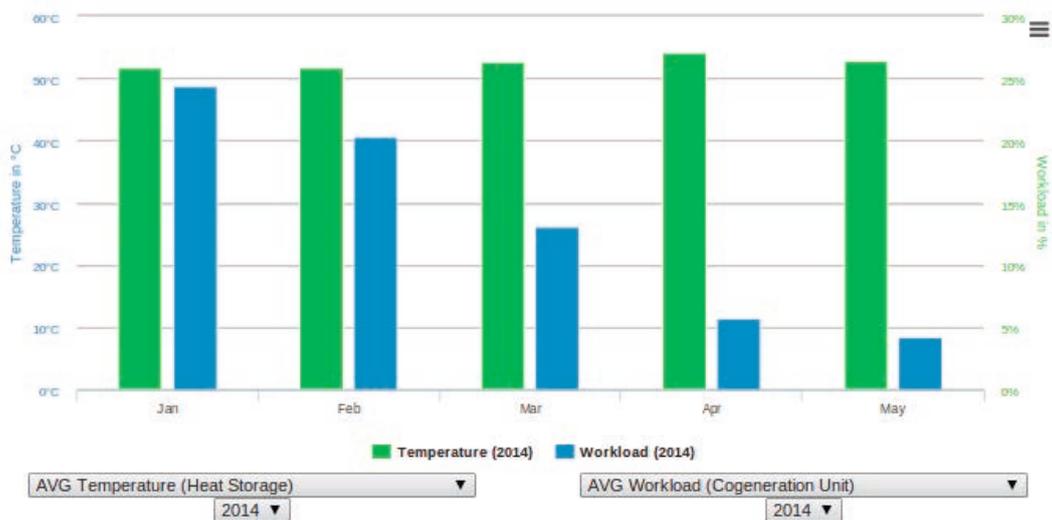


Abbildung 2.9: Vergleichsmöglichkeit beliebiger Werte in Diagrammform

an die eigenen Bedürfnisse angepasst werden kann, zur Verfügung. So können die Werte zwei beliebiger Sensoren verglichen werden.

Ganz spezifische Fragen, wie „Wie viel mehr Strom wurde im Januar 2014 im Vergleich zum Januar 2013 eingespeist?“, können so beantwortet werden, ohne jeden Nutzer mit allen möglichen Diagrammkombinationen zu überlasten. Besonders interessant sind solche Vergleiche, wenn in einem Mehrfamilienhaus ein neues Gerät, zum Beispiel eine Solaranlage, installiert wurde und geprüft werden soll, inwiefern sich die Investition gelohnt hat.

Erfüllung von V2: Fehlerfrüherkennung

Benachrichtigungen

Die Verwaltung hat genau wie der Techniker die Möglichkeit, sich die Fehlermeldungen des Systems anzuschauen. Wenn der Techniker nicht sofort auf Benachrichtigungen reagiert, kann er schnellstmöglich Maßnahmen ergreifen, wenn schwerwiegende Fehler vorliegen. Dadurch, dass beispielsweise ein beschädigtes Gerät möglichst schnell repariert wird, können hohe Kosten vermieden werden.

Bereich des Technikers

Der Techniker kann im Gegensatz zur Verwaltung Einstellungen am System vornehmen. Abbildung 2.10 zeigt eine Übersicht über den Bereich des Technikers. Die einzelnen Komponenten unter der Navigationsleiste werden im Folgenden auf die Anforderungen übertragen und näher erläutert.

Unsere Software kann sowohl im Demo-Modus als auch im Normal-Modus betrieben werden. Im Demo-Modus wird ecoControl mit einer Simulation gestartet. Die Zeit kann hier vorgespielt werden.

Im Normal-Modus werden die Daten des echten Systems in Echtzeit in unsere Software eingespeist und angezeigt. Die in Abbildung 2.11 gezeigte Hauptkomponente des Bereichs des Technikers ist das Live-Diagramm, welches das Verhalten aller wichtigen Komponenten und andere Messwerte in einem Diagramm darstellt.

Auf der linken Seite des Live-Diagramms befinden sich Vergangenheitswerte und auf der rechten Vorhersagewerte. Der Techniker hat nun die Möglichkeit, einzelne Einstellungen vorzunehmen und deren Auswirkungen im Diagramm anhand der Vorhersage-Seite zu prüfen. Außerdem werden dem Techniker zu Analyse Zwecken Statistiken und die Möglichkeit, Benachrichtigungen festzulegen, bereitgestellt.

Erfüllung von T1: Kontrolle der Tabellen

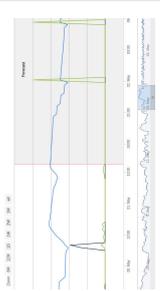
Konfigurationsseite

Um die Wahl der Dimensionierung des BHKWs zu prüfen, müssen zuerst alle Parameter der verschiedenen Geräte sowie das gewählte BHKW angegeben werden. In Abbildung 2.14 wird dargestellt, über welche Benutzeroberfläche die Konfiguration in unserer Software durchgeführt wird. Hier müssen alle Felder, gruppiert nach Geräten, ausgefüllt werden.

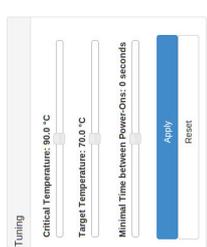
Ob das gewählte BHKW zu den anderen Geräten und dem spezifischen Szenario passt, kann nun vor dem Kauf des BHKWs mit Hilfe des Demo-Modus getestet werden. In die Datenbank von ecoControl können beispielsweise reale Daten für den Strom- und Wärmeverbrauch geladen werden. Dann kann eine Simulation mit

Bereich des Technikers

Übersicht



Live-Diagramm

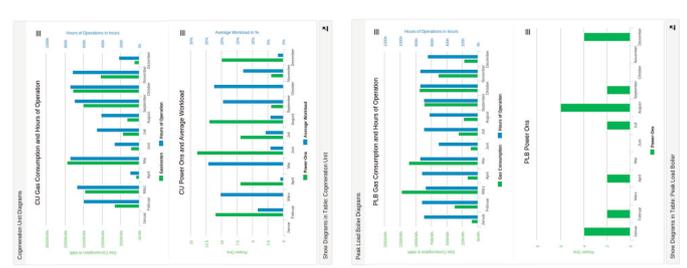


Tuning zur Justierung



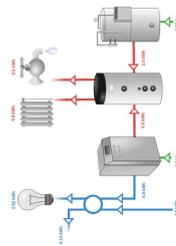
Code-Editor

Statistiken



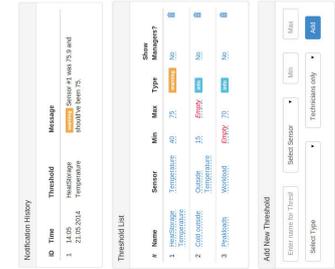
Statistik für CU und PLB

Live-Schema



Live Schema

Benachrichtigungen



Benachrichtigungen einsehen und erstellen

Konfiguration



Konfiguration/ Nachjustierung



Abbildung 2.10: Übersicht über den Bereich des Technikers

2 Nutzerstudien für die Entwicklung von ecoControl

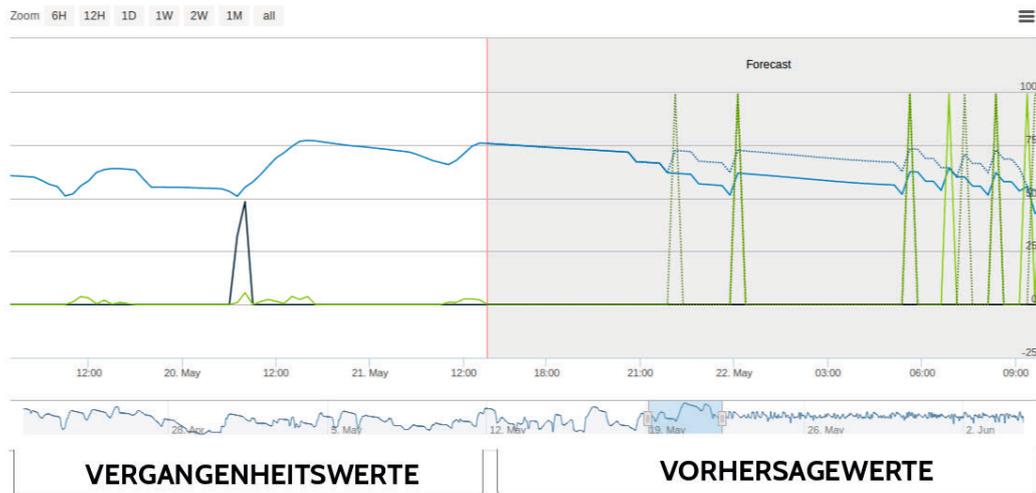


Abbildung 2.11: Live-Diagramm, Vergangenheitswerte auf der linken, Vorhersagewerte auf der rechten Seite

General

Avg. Rooms per Apartment:

Type of Windows:

Apartments:

Avg. Windows per Room:

Reward per kWh: €

Avg. Thermal Consumption: kWh

Peak Load Boiler

Maximal Gas Input: kWh

Thermal Efficiency: %

Abbildung 2.12: Eingabemaske für die Settings zu Programmstart

General

Apartments: 4

Avg. Rooms per Apartment: 4

Avg. Windows per Room: 3

Electrical Revenues: 0.268 €

Gas Costs per kWh: 0.0655 €

Residents: 8

Total Heated Floor: 500 m²

Type of Residents: 0

Warmwater Revenues: 0.065 €

Peak Load Boiler:

Maximal Gas Input: 45.0 kWh

Abbildung 2.13: Übersicht der Einstellungen und Möglichkeit der Justierung

realem Nutzungsverhalten von Strom und Wärme und dem virtuellen BHKW gestartet werden. Das Live-Diagramm zeigt das Verhalten der Komponenten. So kann beispielsweise geprüft werden, wie sich die Geräte bei Spitzenlasten verhalten.

Mit ecoControl kann schon vor Kauf geprüft werden, wie das BHKW für ein spezifisches Szenario dimensioniert werden muss.

Abbildung 2.14: Wizard (links) und Eingabemaske (rechts) für die Konfiguration

Usability-Testing: Wizard oder Eingabemaske? Lange bestand die Frage, wie unsere Software am besten auf ein bestimmtes Szenario eingestellt werden soll. Eine bekannte Möglichkeit, Programme zu konfigurieren, sind Wizards. Die andere Möglichkeit ist, eine große strukturierte Eingabemaske zu entwerfen. Beide Möglichkeiten sind in Abbildung 2.14 dargestellt. Wizards sind vor allem für eine große Anzahl an Daten sinnvoll, die von einem unerfahrenen Nutzer einmalig eingetragen werden müssen. Besonders wichtig werden sie, wenn weitere Fragen von der Antwort vorheriger Fragen abhängen. Dies ist bei uns nicht der Fall.

Große Eingabemasken zeigen alle Informationen, die benötigt werden auf einen Blick und könnten den Nutzer überfordern oder demotivieren. Auf der anderen Seite können sie schnell durch Drücken der Tabulator-Taste ausgefüllt werden [22]. In unserem Fall sprechen zwei Punkte für die Verwendung der Eingabemaske: Zum einen wird die Software von Technikern verwendet, die sehr genau wissen, welche Einstellungen sie vornehmen wollen. Das Klicken des Weiter-Buttons behindert die Techniker eher in ihrem Arbeitsfluss, als es ihnen hilft [39].

Zum anderen sollen die Einstellungen später auch noch zugänglich und änderbar sein. Daher ist es für den Techniker einfacher, wenn er den Aufbau und die Reihenfolge der Eingabemaske schon kennt. Die in Abbildung 2.12 dargestellte Konfigurationsseite, die bei Systemeinrichtung angezeigt wird, ist der in Abbildung 2.13 gezeigten Konfigurationsseite, auf die immer zugegriffen werden kann, sehr ähnlich. Im bekannten Umfeld kann der Techniker so später schneller Änderungen vornehmen.

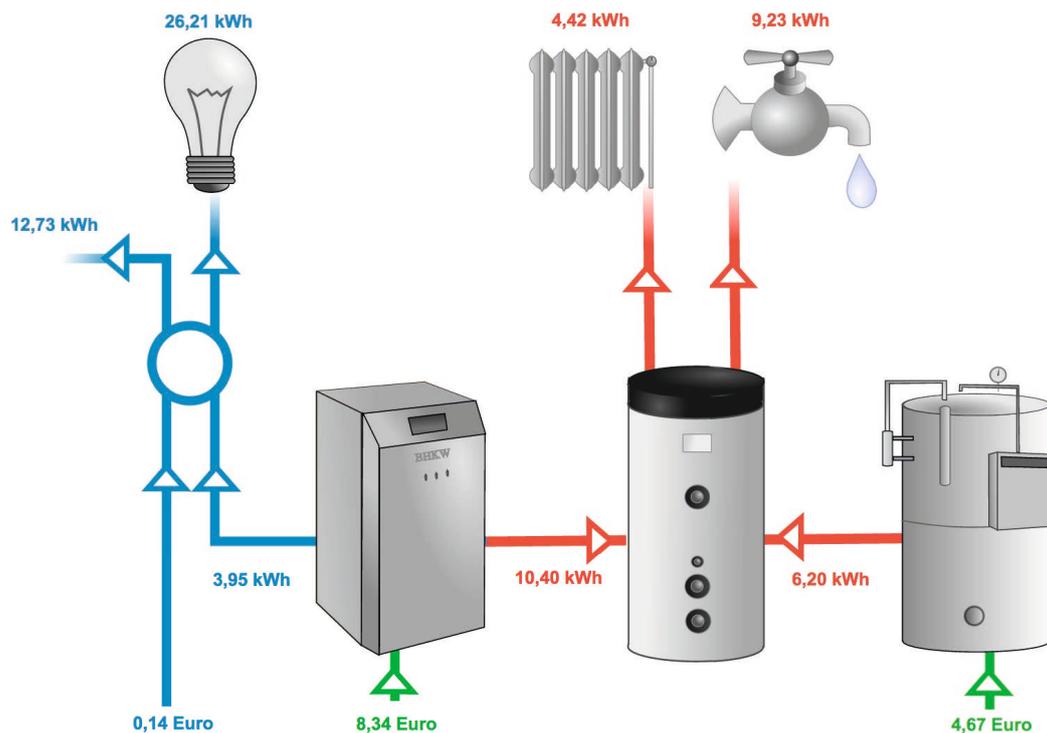


Abbildung 2.15: Live-Schema gibt eine Übersicht über die Geräte

Erfüllung von T2: Beobachtung und frühe Fehlererkennung

Live-Diagramm

Das Systemverhalten kann durch das Live-Diagramm jederzeit im Browser grafisch betrachtet werden. Bei Bedarf können vergangene Fehler detailliert analysiert werden. Durch die Vorhersage können darüber hinaus in Zukunft entstehende Fehler oder nicht optimale Verhaltensweisen im System schon vor Eintritt erkannt und verhindert werden.

Gerätespezifische Fehler, wie der Ausfall eines Gerätes, können hier nicht vorhergesagt werden. Nur Fehlverhalten, welches durch das Zusammenspiel von verschiedenen Faktoren zustande kommt, kann vorhergesagt werden. Ein einfaches Beispiel wäre eine zu hohe Wärmeproduktion bei warmen Wetter.

Live-Schema

Das Live-Schema bietet einen schnellen Überblick über alle Geräte im Haus (siehe Abbildung 2.15). Dies ist vor allem für Nutzer, die sich sehr für den aktuellen Zustand ihrer Geräte und der Verbräuche interessieren, wichtig zu sein. In der herstellereinspezifischen Software von vielen BHKWs ist eine solche Grafik integriert. Bei Vaillant fehlt beispielsweise eine solche Übersicht.

Benachrichtigungen

Jedes BHKW liefert gerätespezifische Fehlermeldungen mit. Es ist bisher nicht möglich, selbst definierte Schwellenwerte und Benachrichtigungen anzulegen. Damit es

Notification History

ID	Time	Threshold	Message
1	14:05 21.05.2014	HeatStorage Temperature	warning Sensor #1 was 75.9 and should've been 75.
2	14:05 21.05.2014	PLB on	info Sensor #6 was 2.2 and should've been 0.
3	14:05 21.05.2014	number 2	warning Sensor #11 was 24.6 and should've been 17.

Threshold List

#	Name	Sensor	Min	Max	Type	Show Managers?	
1	HeatStorage Temperature	Temperature	40	75	warning	No	
2	Cold outside	Outside Temperature	15	Empty	info	No	
3	Peakloads	Workload	Empty	70	info	No	
4	Consumption peek	Thermal Consumption	Empty	11	warning	No	

Add New Threshold

Abbildung 2.16: Liste der überschrittenen Schwellenwerte, aktive Schwellenwerte und Erstellung neuer Schwellenwerte

beispielsweise auch Benachrichtigungen bei ungewöhnlich hohem Stromverbrauch geben kann, zeigt Abbildung 2.16, wie Benachrichtigungen bei beliebigen Schwellenwerten angelegt werden können.

Der Techniker kann einen gewünschten Sensor auswählen, Schwellenwerte angeben, einen Namen vergeben und auswählen, ob die Benachrichtigung auch für die Verwaltung zu sehen sein soll. Danach wird dieser Schwellenwert in der Liste der aktiven Schwellenwerte angezeigt (siehe Abbildung 2.16) und kann ab diesem Zeitpunkt auch ausgelöst werden.

Die Übersicht über alle bisher ausgelösten Schwellenwerte gibt ihm die Möglichkeit, sich eine Übersicht über die aufgetretenen Fehlerfälle zu verschaffen. Im Rahmen seiner Optimierungen kann der Techniker nun auch Benachrichtigungen anlegen, welche ihn informieren, wenn das System gerade nicht optimal läuft.

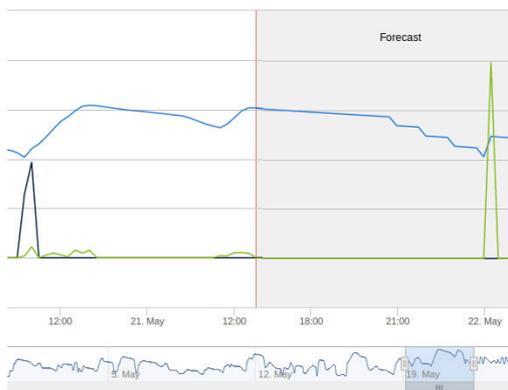


Abbildung 2.17: Live-Diagramm als Hauptkomponente mit der Vorhersage auf der rechten Seite

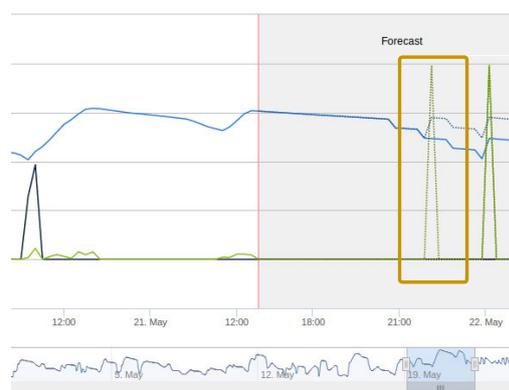


Abbildung 2.18: Live-Diagramm nach dem Tuning, gestrichelte Linien stellen unangewendete Änderungen dar

Erfüllung von T₃: Optimierung der Geräte

Live-Diagramm und Programmierschnittstelle

Die Hauptseite des Bereichs des Technikers zeigt ein Live-Diagramm (siehe Abbildung 2.17). Dieses Diagramm enthält den Verlauf aller wichtigen Messwerte über die Zeit. Unter dem Diagramm befindet sich die Komponente *Tuning*, in der einige Parameter geändert werden können. Über Schieberegler können hier verschiedene gerätespezifische Werte, wie zum Beispiel die Wärmespeichertemperatur, angepasst werden. Nach der Anpassung wird dem Nutzer direkt im Live-Diagramm auf der Seite mit Vorhersage gezeigt, wie sich Änderungen in Zukunft wahrscheinlich auswirken würden. Die gestrichelten Linien zeigen das veränderte zukünftige Verhalten dargestellt (siehe Abbildung 2.18). So können Änderungen vor Anwendung auf die echten Geräte geprüft und so lange geändert werden, bis sie optimal sind. Erst danach können die neuen Einstellungen durch den Klick auf den Button zum Bestätigen angewendet werden. Die gestrichelten Linien werden durch normale Linien ersetzt

und die Geräte arbeiten ab diesem Zeitpunkt mit den neuen Einstellungen. Für unerfahrene Techniker sinkt durch die Simulation des Weiteren die Hemmschwelle, komplexe Einstellungen vorzunehmen. Er kann seine Ideen vor Anwendung auf die echten Geräte validieren.



Abbildung 2.19: Programmierschnittstelle für Optimierungsalgorithmen

Eine Änderung der Parameter reicht allerdings nicht aus, um komplexere Optimierungsalgorithmen und Abhängigkeiten zu definieren. Daher haben wir eine Programmierschnittstelle eingefügt. Der Techniker kann hier durch Python-Code beispielsweise Regeln erstellen, die die Wettervorhersagen mit einbeziehen. Die Auswirkungen seines Programmcodes kann er durch einen Klick auf den Button zum Simulieren testen. Hier hilft das Live-Diagramm wieder, die Änderungen direkt zu validieren. Die Programmierstelle bietet den Vorteil, dass eine Optimierung des Systems durch einmalige Konfiguration durchgeführt werden kann.

Bei dem Beispiel der Wettervorhersagen müssen nur einmalig Regeln dafür angelegt werden, dass die Wettervorhersagen einbezogen werden, statt diese Werte einmal wöchentlich manuell einzutragen. Der in Abschnitt 2.4.2 beschriebene Zeitaufwand, der durch Optimierungsversuche entsteht, wird so minimiert.

Wünschenswert ist hier natürlich eine Oberfläche, die es ermöglicht, auch ohne Programmierkenntnisse Optimierungsalgorithmen zu erstellen.

Usability-Testing: Unmittelbares Feedback durch Klick oder Schieberegler? Da die Simulation dazu gedacht ist, schnell verschiedene Parameter auf ihre Auswirkungen zu prüfen, darf es für den Nutzer nicht aufwendig sein, mit verschiedenen Werten zu experimentieren. Wie Bret Victor in seinem Artikel „Magic Ink“ [49] beschreibt, soll jede Interaktion zu einer wahrnehmbaren Änderung in einer kontextsensitiven Informationsgrafik, bei uns im Live-Diagramm, führen. Unmittelbares Feedback reduziert die Menge an Manipulationen, welche der Nutzer durchführen muss, bevor er ein be-

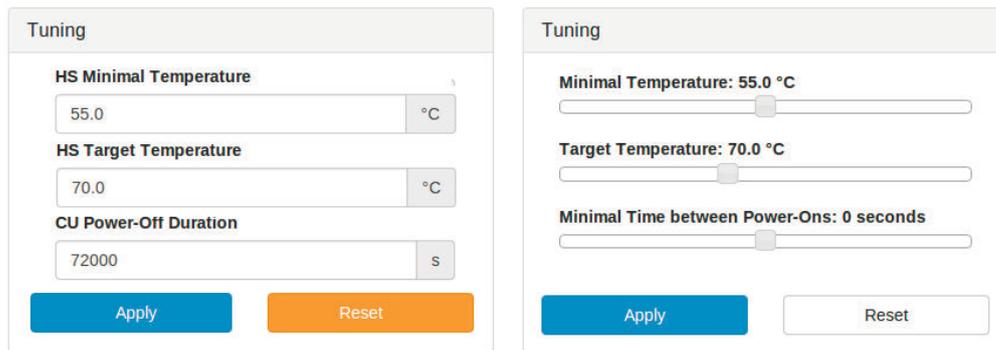


Abbildung 2.20: Tuning mit Eingabefeldern (links), Tuning mit Schiebereglern (rechts)

stimmtes Ergebnis erhält oder eine falsche Änderung feststellt und diese rückgängig macht. Jede Website mit einem Button zum Bestätigen erfüllt diese Anforderungen nicht. (Übersetzt nach [49]).

Daher wollten wir einen Button, der die Simulation mit den neuen Werten startet, vermeiden. Um unmittelbares Feedback bereit zu stellen, haben wir zwei verschiedene Designs ausprobiert. Die erste Version bestand weiterhin aus Eingabefeldern (siehe Abbildung 2.20 (rechts)). Wenn man nach Änderung eines Wertes in den Feldern an eine beliebige Stelle klickt, wird die Simulation dementsprechend angepasst.

Während des Usability-Testings hat sich herausgestellt, dass der zusätzliche Klick nicht gemacht wurde. Stattdessen wurde nach einem Button gesucht oder direkt der Button zum Anwenden geklickt. Mit JavaScript kann allerdings nicht eindeutig bestimmt werden, wann eine Eingabe in einem Eingabefeld beendet ist.

Daher haben wir uns im zweiten Schritt, wie in Abbildung 2.20 dargestellt, auf Schieberegler zurückgegriffen. Für jeden Wert gibt es nun statt einem Eingabefeld einen Schieberegler, der bei Änderung direkt eine neue Simulation startet. So kann schnell experimentiert werden.

2.5 Qualität der Ergebnisse

Nutzerstudien sollten der erste Schritt sein, wenn es darum geht, ein neues Produkt zu entwickeln. Aber auch während der Entwicklung ist es besonders wichtig, seine Nutzer nicht aus den Augen zu verlieren und kontinuierlich nachzuforschen, ob die Anforderungen der Nutzer erfüllt werden.

In diesem Abschnitt soll erläutert werden, welche Schwierigkeiten bei den Nutzerstudien auftraten und welche Kompromisse wir eingegangen sind.

Unser Projekt begann mit einer unklaren und sehr schwer zu erfassenden Zielsetzung, die für uns Nutzerstudien nicht als ersten Schritt beinhaltete. Bevor wir mit den Nutzerstudien beginnen konnten, mussten wir die Problemstellung verstehen und die Domäne kennen lernen. Für die Internet-Recherche haben wir relativ lange

gebraucht und die ersten Wochen damit verbracht, eigene Ideen zu sammeln. Erst danach wurde uns bewusst, dass wir ohne die Befragung verschiedener Personen unserer Nutzergruppen nicht weiterkommen.

Die ersten Versuche, Interview-Partner zu finden, waren nicht sehr erfolgreich. Die in Abschnitt 2.3.1 beschriebenen Herausforderungen machten es uns schwer, Teilnehmer zu finden, die sich die Zeit für eine Befragung oder ein Treffen nehmen wollten. Daraus resultiert, dass auch nicht alle Methoden aus Abschnitt 2.2, die wir gerne angewendet hätten, immer anwendbar waren. So konnten wir beispielsweise das Contextual-Inquiry, welches uns am meisten Erkenntnisse versprach, nur einmal durchführen. Alle anderen Teilnehmer hatten nur Zeit für kurze Interviews oder haben uns per E-Mail Fragen beantwortet.

Mehr Teilnehmer und mehr Zeit für einzelne Gespräche wären besonders im Hinblick darauf wichtig gewesen, die Aussagen anderer Teilnehmer zu validieren oder zu widerlegen. Es ist davon auszugehen, dass die Aussage einer Person einer bestimmten Nutzergruppe nicht unbedingt die Meinung aller anderen widerspiegelt. Unsere gefundenen Anforderungen beruhen daher manchmal auf Aussagen weniger Personen. Deshalb kann nicht für die Repräsentativität oder Verallgemeinerbarkeit garantiert werden.

Trotzdem sollte nicht außer Acht gelassen werden, dass wir an einer Problemstellung angesetzt haben, mit dem sich noch nicht viele Personen gut auskennen. Das Feedback eines BHKW-Beraters ganz am Ende des Projektes war sehr positiv. Im Rahmen der Open-Source Lösung ist es für andere Interessierte möglich, unsere Software weiter zu entwickeln und zu verbessern. Mit unserem Prototypen ging es uns vor allem darum, eine Inspiration zu geben, wie die Energie von BHKWs und anderen Energiesystemen im eigenen Haus effizienter genutzt werden kann.

2.6 Zusammenfassung

Nutzerstudien bildeten die Grundlage für die Entwicklung der Softwarelösung *ecoControl*. Am Anfang des Projektes konnten wir durch Nutzerstudien die Idee unseres Projektpartners verstehen und erste Lösungsansätze entwickeln. Mit Interviews und anderen Methoden war es möglich, die Anforderungen unserer Nutzer kennenzulernen und unsere Zielgruppe zu definieren. Durch Prototypen, die wir von Teilnehmern unserer Zielgruppe testen ließen, konnten wir unsere Ideen schon früh validieren. Usability-Testing konnte sicherstellen, dass *ecoControl* die Anforderungen von Technikern als auch verwaltenden Personen erfüllt und intuitiv bedienbar ist.

ecoControl ist eine Softwarelösung zur Optimierung von dezentralen Energiesystemen in Mehrfamilienhäusern. Derzeit gibt es kaum Personen, die versuchen, effizienzsteigernde Maßnahmen an BHKWs vorzunehmen, um Energiekosten zu sparen. Auch Softwarelösungen, die an dieser Stelle ansetzen, sind selten. Durch Nutzerstudien konnten wir schon bestehende Optimierungsansätze kennenlernen.

Es hat sich gezeigt, dass BHKW-Eigentümer ein großes Interesse an einer solchen Optimierung haben. Manchmal ist hierzu aber zusätzliches Wissen notwendig, welches die Optimierung zeitintensiv und risikoreich werden lässt.

ecoControl erleichtert es BHKW-Installateuren oder Heizungsmonteuren, Einstellungen vorzunehmen und Regeln zu definieren. Hier kann Zeit eingespart werden. Die sofortige Kontrolle der Auswirkungen auf das echte System macht die Optimierung für Techniker einfacher und sicherer.

Wenn Strom und Wärme in einem Mehrfamilienhaus selbst umgewandelt werden, soll ecoControl auch bei der Verwaltung helfen. Derzeit müssen wichtige Messwerte und Zahlen manuell zusammengeführt werden. BHKW-Eigentümer können mit unserer Software die finanzielle Seite des Szenarios betrachten und analysieren. Alle wichtigen Informationen für Abrechnungen und Steuererklärungen werden zusammengefasst und übersichtlich dargestellt.

Als Prototyp stellt ecoControl eine ganzheitliche Lösung für die Optimierung und Verwaltung von BHKWs und anderen Energiesystemen zur Eigenproduktion von Strom und Wärme dar.

3 Entwicklung einer Anwendung zu softwaregestützten Aussteuerung des Energiesystems eines Mehrfamilienhauses

Dieses Kapitel beschreibt die Software ecoControl und ihre Entwicklung. Momentan erfolgt die Aussteuerung heterogener Energiesysteme in einem Mehrfamilienhaus nicht so effektiv wie möglich. Um dieses Problem zu lösen, haben wir einen Prototyp entwickelt, der alle Energiesysteme eines Hauses verbindet und steuert. Konzentriert haben wir uns dabei auf Energiesysteme, die ein Blockheizkraftwerk enthalten. Die entstandene Software sammelt Daten und wertet sie mit Hilfe von Vorhersagen aus. Mit den Berechnungen vereinfacht sie Nutzern eine effiziente Steuerung der Geräte. Der entwickelte Prototyp wurde in einer Simulation eines Energiesystems getestet und zeigte darin Einsparungspotenziale, die durch die Nutzung der Software entstanden. Mit dem Ergebnis haben wir gezeigt, dass eine solche Software für echte Geräte lohnenswert wäre. Auf Basis des Prototyps kann eine Software weiterentwickelt werden, welche die Verbesserungspotenziale der Energiesysteme in Mehrfamilienhäusern nutzt.

3.1 Anwendungsgebiet von ecoControl

Das Anwendungsgebiet von ecoControl ist ein Mehrfamilienhaus, dessen Energiesystem aus mehreren Komponenten besteht. Diese Komponenten sind ein Blockheizkraftwerk (BHKW), ein Spitzenlastkessel und ein Wärmespeicher. Genauere Informationen über die einzelnen Geräte können in Kapitel 1 nachgelesen werden.

Die Geräte sind mit Sensoren ausgestattet, die den Gerätezustand messen können. Wir gehen davon aus, dass eine externe Geräteschnittstelle existiert. Sie wandelt Steuerbefehle an die Geräte so um, dass die Geräte das Signal verarbeiten können. Genauso wandelt sie die Messdaten der Geräte in eine Form um, die ecoControl verarbeiten kann. Von ihr erhält ecoControl die Messdaten der Geräte. Die Geräte wiederum können über die Komponente Steuerbefehle von ecoControl entgegennehmen.

ecoControl informiert die Nutzer über den Zustand des Energiesystems und nimmt Steuerbefehle der Nutzer entgegen. Zusätzlich nutzt ecoControl einen Internetzugang, um Daten aus dem Internet zu beschaffen. Einen Überblick über die Einbettung von ecoControl in die Umgebung gibt Abbildung 3.1.

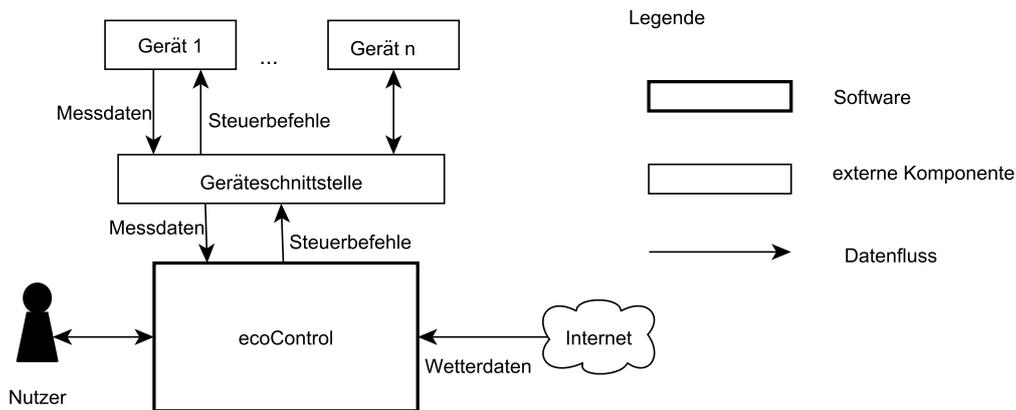


Abbildung 3.1: Systemgrenzen

3.2 Architektur von ecoControl

In diesem Kapitel werden wir den technischen Aufbau von ecoControl zeigen. Wir beginnen mit der Vorstellung des Servers inklusive der Datenbank und dem Client (Abschnitt 3.2.2, Abschnitt 3.2.3 und Abschnitt 3.2.4). Anschließend werden wir in Abschnitt 3.3 den Ablauf der Kommunikation zwischen dem Server und einem Client an einem Beispiel erläutern.

3.2.1 Überblick über die Komponenten von ecoControl

ecoControl besteht aus drei Hauptkomponenten, die zusammenarbeiten:

- einem Server,
- einer Datenbank,
- einem oder mehreren Clients.

Die Geräte gehören nicht zu ecoControl. Die Interaktion der Geräte mit ecoControl erfolgt über eine interne Geräteschnittstelle im Server und eine externe. Wir setzen voraus, dass ein anderes Team die externe Komponente entwickelt. So kann ecoControl von konkreten Geräten abstrahieren.

Der Server beinhaltet die meiste Logik ecoControl's. Er enthält Module für:

- die Gerätekommunikation,
- die Vorhersage,

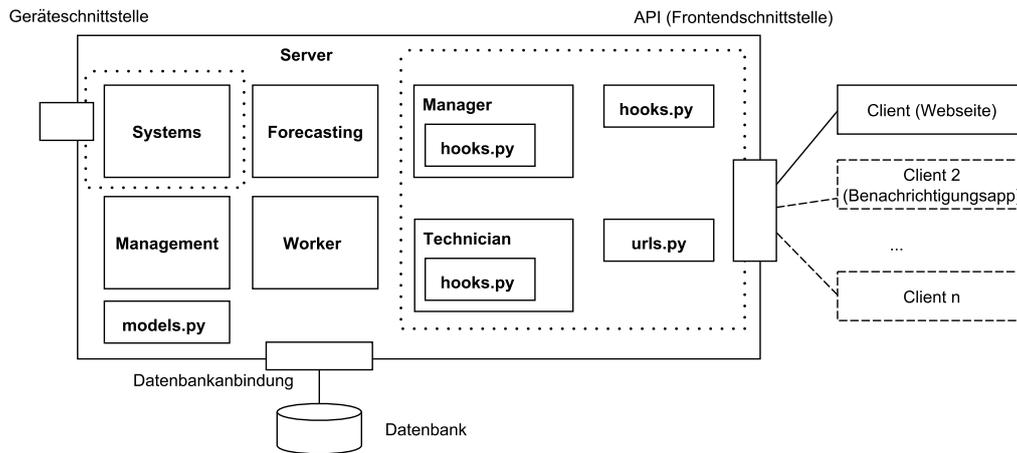


Abbildung 3.2: Grundaufbau von ecoControl

- die regelmäßige zentrale Steuerung der Geräte,
- die Kommunikation mit Clients,
- die Kommunikation mit der Datenbank.

Abbildung 3.2 gibt einen Überblick über die Module und Schnittstellen. Genauere Informationen zu den einzelnen Modulen geben wir in Abschnitt 3.2.2.2.

Die Datenbank speichert hauptsächlich gesammelte Sensordaten und Wetterdaten, außerdem enthält sie Einträge für den aktuellen Geräteaufbau und die Nutzer von ecoControl. Sie enthält auch Daten für angelegte Schwellenwerte, die das Geräteverhalten überwachen sollen.

Ein Client nutzt die Schnittstelle des Servers, oft mit API (englisch *application programming interface*) abgekürzt, um Informationen über das System abzufragen und den Nutzern zugänglich zu machen. Über die API kann ein Client den Nutzern auch die Steuerung des Energiesystems ermöglichen.

Innerhalb von ecoControl läuft die Kommunikation wie in Abbildung 3.2 gezeigt: Der Server speichert Daten in der Datenbank und ruft sie aus der Datenbank wieder ab. Der Server empfängt Informationsanfragen von Clients und antwortet mit entsprechenden Daten. Genauso empfängt er Steuerbefehle von Clients.

3.2.2 Server

Der Server bietet dem Client eine Programmierschnittstelle an. Diese API stellt Informationen aus der Datenbank und damit über das Energiesystem zur Verfügung. Sie gibt Zugriff auf Berechnungen, die der Server auf Basis der Gerätedaten und externer Daten angestellt hat. Die wichtigste dieser Berechnungen ist, eine möglichst genaue Vorhersage des Geräteverhaltens zu erstellen.

Der Server gibt über die API Auskunft, ob das Geräteverhalten Schwellenwerte verletzt hat, und erlaubt dem Client, Schwellenwerte zu manipulieren. Die API nimmt außerdem Steuerbefehle des Clients entgegen. Diese können aus einfachen Einstellungen an den Geräten bestehen. Daneben sind auch komplexere, bedingte Anweisungen möglich, die regelmäßig ausgeführt werden sollen. Solche komplexen, regelmäßig ausgeführten Konfigurationen werden von uns **Regeln** genannt. Regeln nimmt der Server in Form von Python-Code entgegen. Der Server bietet dem Client dafür eine Programmierschnittstelle an.

Die API bietet zusätzlich eine Authentifizierung des Clients an, weil der Zugriff auf ihre Anfragen durch Zugriffsrechte geschützt ist. Der Server nimmt über seine Geräteschnittstelle Sensordaten entgegen, welche die Geräte schicken, und speichert diese in der Datenbank. Genauso schickt er Steuerbefehle über die Geräteschnittstelle an die Geräte.

Der Server enthält zusätzlich eine Demosimulation. Diese stellt eine virtuelle Umgebung mit Testgeräten zur Verfügung, die schnell auf Einstellungen reagiert und jedem Entwickler jederzeit zur Verfügung steht. Die Demosimulation liefert Sensordaten, wie es echte Geräte tun würden, und sichert diese in der Datenbank. Die simulierten Geräte lassen sich vom Server genauso steuern, wie wir es von den echten Geräten erwarten.

Wir haben den Server mit der Sprache Python¹ (Version 2.7) entwickelt. Der Code des Servers liegt daher in einem Python-Paket *server*. Der Server wurde mit Hilfe von Django² entwickelt. Daher ist *server* ein Django-Projekt. Wir nutzen für den Server von ecoControl Django in der Version 1.6.

3.2.2.1 Django

Django ist ein Python-basiertes Webframework. Es bietet verschiedene Funktionen an, die den Betrieb eines datenbankgestützten Servers erleichtern.

Eine dieser Funktionen sind objektrelationale Abbildungen (ORM) von Python-Klassen auf Tabellen. So ist es möglich mit Hilfe der Python API Datenbankoperationen auszuführen. Django bietet eine bereits implementierte Nutzerverwaltung, die von ecoControl genutzt wird. Diese erlaubt Nutzer mit verschiedenen Rechten anzulegen und stellt Funktionalitäten für ein Login oder Logout zur Verfügung. Django bietet Funktionen, die das Einrichten eines Servers vereinfachen. Es erleichtert zum Beispiel die Erstellung von URLs und ihre Verknüpfung mit Funktionen.

3.2.2.2 Komponenten des Servers

Wie erwähnt, ist das Python-Paket *server* ein Django-Projekt. *server* enthält direkt Django-zugehörige Dateien, welche die Server-Funktionen bereitstellen. Diese definieren die URLs, über die der Server angesprochen werden kann. Sie definieren ebenfalls die Funktionen, die die Anfragen verarbeiten und beantworten. Zusätzlich

¹Python Software Foundation, <https://www.python.org/> (abgerufen am 19.11.2014).

²Django Software Foundation, <https://www.djangoproject.com/> (abgerufen am 19.11.2014).

enthält es Definitionen der Datenbanktabellen. Die zugehörige Datei heißt, wie bei Django üblich, *models.py*.

Nutzer Der Zugriff auf die API des Servers ist mit Hilfe von Django beschränkt. Zurzeit werden die Nutzer **Techniker** und **Verwalter** angelegt. Diese beiden Nutzer repräsentieren die Rolle Techniker bzw. Verwalter. Die verschiedenen Rechte, die jeder der Nutzer hat, beinhalten, dass ein Techniker alle Funktionen der API nutzen kann. Der Verwalter erhält nur eine beschränkte Auswahl von Funktionen. Diese Aufteilung und Beschränkung ist so eingerichtet, dass sie den Nutzerrollen entspricht, die wir für ecoControl erarbeitet haben.

Die Rolle Verwalters interessiert sich vor allem für die finanziellen Aspekte des Energiesystems. Er kennt sich nicht mit der Technik aus. Daher kann er keine Einstellungen an den Geräten vornehmen oder eine Vorhersage des Systemverhaltens anfordern, sondern nur den aktuellen Zustand des Systems abrufen.

Die Rolle des Technikers sieht vor, dass er sich mit den technischen Aspekten des Systems beschäftigt. Alle Steuerbefehle und Optimierungen sollen vom Techniker vorgenommen werden. Auch Schwellenwerte, die die Geräte überwachen, kann nur er einrichten. Genauere Informationen zu den Rollen finden sich in Kapitel 2.

In Zukunft wird ecoControl mehrere Nutzer mit der Rolle Techniker bzw. Verwalter kennen. Wir haben jedoch nur einen Nutzer pro Rolle erstellt, da dies für Demozwecke ausreichend war.

Systems-Paket Das Paket *systems* ist die Schnittstelle des Servers zu den Geräten des Energiesystems. Es enthält diejenigen Klassen, die den realen Geräten im Haus entsprechen.

Models.py Wie erwähnt, enthält *models.py* die Definition der Datenbanktabellen. Jede Klasse dieses Moduls erbt von der Django-Klasse für ein Model *django.db.models.Model*. Durch die Vererbung kann die Objektrelationale Abbildung, kurz ORM (englisch *object-relational mapping*), von Django genutzt werden: Die Tabellen und Datensätze haben eine Repräsentation als Klasse bzw. Objekt. Datenbankabfragen können so durch Methodenaufrufe ersetzt werden. Genauere Informationen zu den einzelnen Tabellen geben wir in Abschnitt 3.2.3.1.

Management-Paket Das Paket *management* definiert die Datenbankeinträge, die beim ersten Starten von ecoControl und dem Server enthalten sein sollen: Es erstellt für jede Rolle einen Benutzer und lädt ihn in die Datenbank. ecoControl wird mit einem Standardgeräteaufbau initialisiert gestartet. Zu diesem Aufbau definiert *management*

- Geräteinträge,
- Sensoreinträge,
- allgemeine Einstellungen und
- Geräteeinstellungen.

Diese Einträge müssen in der Datenbank enthalten sein.

Frontend-Schnittstelle Eine der wichtigsten Funktionen des Servers ist das Bereitstellen der Schnittstelle zum Frontend. Diese bietet er über eine HTTP-Schnittstelle an, die GET- und POST-Anfragen entgegennimmt. Einen Überblick über die Funktionen, die der Server zur Verfügung stellt, gibt die folgende List:

- Authentifikation
- Steuerung der Geräte*
 - Programmierschnittstelle*
- Vorhersage*
- Anzeige
 - Geräteinformationen
 - Messwerte
 - Aggregationen von Messwerten
- Schwellenwerte*
- Benachrichtigungen
- Bilanz

(Mit * markierte Punkte sind nur dem Techniker zugänglich)

Um die API anzubieten, werden mehrere Module des Servers genutzt (vergleiche auch den Bereich „Frontend-Schnittstelle“ in Abbildung 3.2). In der *urls.py* werden alle unterstützten URLs aufgezählt und einer Python-Funktion zugeordnet. Mit Hilfe dieser Informationen löst Django bei einer Anfrage an eine bestimmte URL die passende Funktion aus. Die von den URLs ausgelösten Funktionen sind auf mehrere Pakete verteilt. Die Datei, die diese Funktionen definiert, heißt in jedem Paket *hooks.py*. Sie sind nach ihren Anwendungsbereichen sortiert. Allgemeine Hooks, die nicht benutzerspezifisch sind, liegen direkt im Paket *server*. Ein Beispiel für eine solche Funktion ist das Login. Zusätzlich zu den allgemeinen Hooks gibt es für jede Rolle jeweils ein Paket, das die rollenspezifischen Hooks implementiert: Das Paket *manager* für die Rolle Verwalter und das Paket *technician* für die Rolle Techniker.

Ein Beispiel für die Funktionsweise der API ist die Anfrage nach allen Einstellungen von ecoControl. Der Server erhält eine GET-Anfrage an `/api/settings/`. In der in der *urls.py* definierten Auflistung wird nach der passenden Funktion gesucht. Gefunden wird die Funktion *hooks.list_settings*. Diese ist in der *hooks.py* mit den nicht nutzerspezifischen Funktionen definiert. Daher ist sie für beide Nutzer zugänglich. Zunächst überprüft die Funktion, ob die Anfrage von einem authentifizierten Client stammt.

Anschließend ruft sie die allgemeinen Einstellungen des Energiesystems und die Geräteeinstellungen aus der Datenbank ab (Genauere Informationen zu allgemeinen Einstellungen und Geräteeinstellungen gibt Abschnitt 3.2.3.1). Aus den Objekten

erstellt die Funktion ein JSON-Objekt. Einen Ausschnitt aus einem solchen JSON-Objekt, das erstellt wird, ist in Quelltext 3.1 zu sehen. Die Objektattribute sind zur Vereinfachung verallgemeinert. Das Objekt enthält ein Array. Das erste Element beinhaltet die allgemeinen Einstellungen, jedes weitere Element enthält die Einstellungen genau eines Geräts.

Quelltext 3.1: Vereinfachtes JSON-Objekt als Antwort auf eine Anfrage nach den Einstellungen

```
{
  "0": {
    "gas_costs": {},
    ...
  },
  "1": {
    "capacity": { "device": "Heat Storage", ...}
    ...
  },
  "2": {
    "max_workload": { "device": "Cogeneration Unit", ...}
    ...
  }
}
```

Forecasting-Paket Das Paket *forecasting* enthält die Vorhersage und die Demo-Simulation. Eine graphische Übersicht des Pakets befindet sich in Abbildung 3.3. Zu der Vorhersage und der Demo-Simulation kommt die Basis-Simulation, die die Grundlage der Demo-Simulation ist und auch von der Vorhersage genutzt wird. Das Paket enthält ein eigenes Paket *systems*, das die Geräte, die ecoControl unterstützt, als Simulationen enthält. Jede der Geräteklassen erbt von seiner allgemeineren Form aus dem normalen systems-Paket. Die Simulation besteht aus diesen Geräten. Die Demo-Simulation ersetzt im Demo-Modus einen realen Geräteaufbau für Demonstrationen der Software. Die Vorhersage nutzt die Simulation bei ihren Berechnungen als Grundlage für das Geräteverhalten. Weitere Informationen zu Details der Vorhersage geben wir in Kapitel 4.

Worker Das Paket *worker* enthält die Klasse Worker. Diese ist ein Thread, der regelmäßig auszuführende Aufgaben erledigt. So überwacht ecoControl Schwellenwerte und führt vom Techniker erstellte Regeln aus.

Tests Einige Module von ecoControl enthalten Unittests. Dadurch kann nicht nur sichergestellt werden, dass das Programm das gewünschte Verhalten aufweist, sondern die Tests geben auch Sicherheit beim Refactoring des Programmcodes. Wenn eine Änderung eine Funktionalität verändert oder zerstört, wird dies durch einen

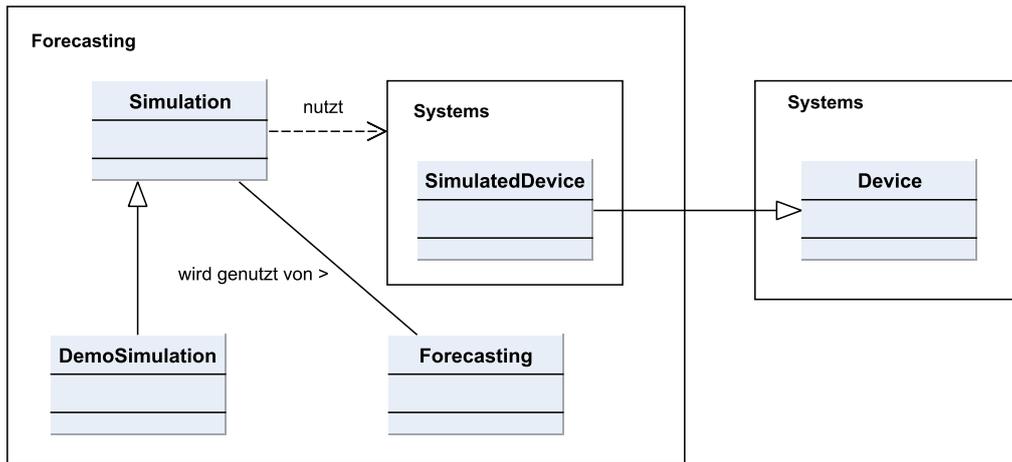


Abbildung 3.3: Das Paket Forecasting

oder mehrere fehlschlagende Tests angezeigt. Wir haben die Tests mit Hilfe des von Python bereitgestellten Testing-Frameworks *unittest* geschrieben.

Bei den Tests, die eine Datenbank benötigen, greifen wir auf Djangos Erweiterung der Unittests zurück (*django.test.TestCase*). Auch beim Testen der API stellt Django den Entwicklern eine Klasse *Django.test.Client* zur Verfügung, mit der man Abfragen durch einen Client simulieren kann. Für einige Tests setzen wir Mocks ein. Zur Mockerstellung haben wir die Python Bibliothek *mock*³ genutzt. Pythons *unittest* erlaubt uns, die Tests automatisiert auszuführen. Das war eine Voraussetzung, um Continuous Integration zu praktizieren. Jede Revision des Codes wurde vor dem Einchecken in das Repository mit unseren Unittests geprüft. Dazu nutzen wir die Continuous-Integration-Plattform „Travis CI“⁴.

server enthält das Paket *tests*, das nicht in Abbildung 3.2 abgebildet ist. Es beinhaltet einige API-Tests sowie Tests für die Funktionen des Pakets *technician*. Außerdem enthält das Paket *forecasting* ein eigenes Paket *tests*. Darin befinden sich sowohl Unittests für die simulierten Geräte als auch für die Vorhersage.

Unser System hat mit diesen beiden Paketen eine Testabdeckung von 54 % erreicht. Diese Zahl haben wir mit dem Tool *coverage*⁵ ermittelt, dass die Testabdeckung auf Basis der Anweisungen bestimmt.

³Mock, <http://www.voidspace.org.uk/python/mock/> (abgerufen am 19.11.2014).

⁴Travis CI, <https://travis-ci.com/> (abgerufen am 19.11.2014).

⁵Coverage.py, <http://nedbatchelder.com/code/coverage/> (abgerufen am 19.11.2014).

3.2.3 Datenbank

ecoControl verwendet als Datenbankmanagementsystem *PostgreSQL*⁶. Es kann große Datenmengen verarbeiten und wird von Django unterstützt. Das Initialisieren der benötigten Datenbank ist eine der Aufgaben, die Django in ecoControl übernimmt. Django stellt, wie erwähnt, ein ORM zur Verfügung, das erlaubt, die Tabellen als Python-Klassen zu definieren.

3.2.3.1 Tabellen

In diesem Abschnitt werden wir die einzelnen Tabellen der Datenbank erläutern. Definiert sind diese in *models.py*, die in Abbildung 3.4 in einem Klassendiagramm zu sehen sind.

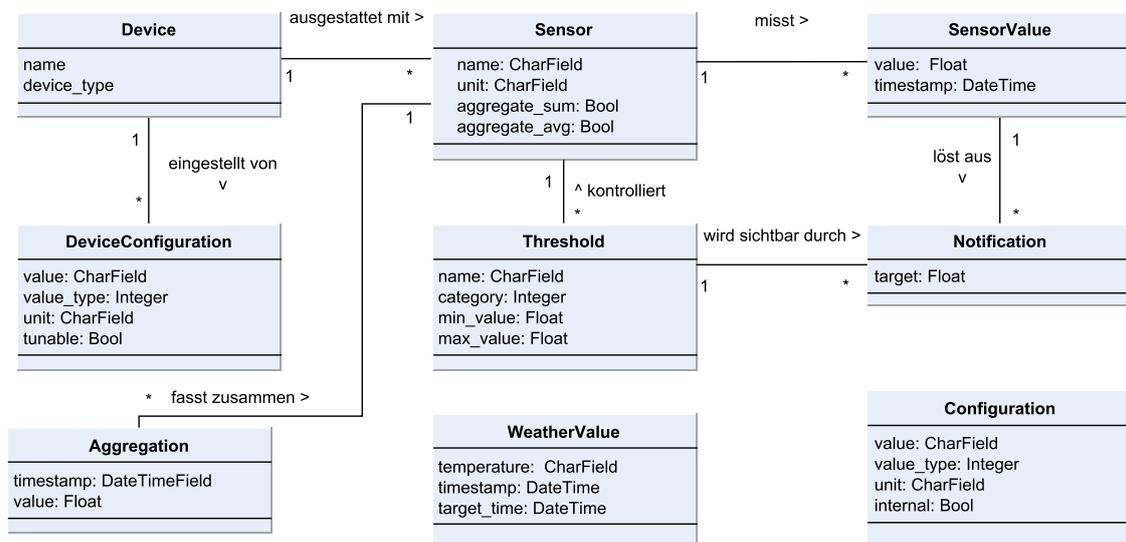


Abbildung 3.4: Datenbanktabellen wie in *models.py* definiert (Klassendiagramm)

Device enthält Einträge für alle Geräte, die Teil des Energiesystems des Hauses sind.

DeviceConfiguration beinhaltet alle Einstellungen, die ein Gerät betreffen. Neben Sensoren können Geräte auch noch Einstellungen haben, die von Gerätemodell zu Gerätemodell unterschiedlich sind. Ein BHKW kann beispielsweise einen anderen Wirkungsgrad erreichen als ein anderes.

⁶PostgreSQL, <http://www.postgresql.org/> (abgerufen am 19.11.2014).

Sensor enthält Einträge für alle Sensoren, die ein Gerät überwachen. An einem Gerät können mehrere Attribute überwacht werden, wie zum Beispiel am Stromzähler die eingespeiste Energie und die bezogene Energie.

SensorValue beinhaltet die Messwerte eines Sensors.

Threshold enthält alle Schwellenwerte für einen Sensor. Die Nutzer können diese Schwellenwerte anlegen, um bei deren Verletzung benachrichtigt zu werden.

Notification speichert diese Benachrichtigungen.

Configuration enthält allgemeine Einstellungen, wie den aktuellen Gaspreis, der alle Geräte betrifft.

WeatherValue beinhaltet Temperaturen, verbunden mit einem Datum. Diese Tabelle enthält Voraussagen und Messdaten.

Für einige Datenbankabfragen haben wir Sichten angelegt. Die Messdaten eines Sensors werden regelmäßig abgefragt. Dabei werden die Messdaten immer in der gleichen Form gespeichert. Alle Werte eines Sensors eines immer gleichen Zeitraums abgefragt, die Werte werden aufsummiert oder gemittelt. Solche *Aggregationen* haben wir in unserem System als folgende Sichten angelegt:

SensorValueHourly enthält alle Sensordaten des letzten Monats, pro Stunde und Sensor gemittelt.

SensorValueDaily beinhaltet alle Sensordaten, pro Tag und Sensor gemittelt.

SensorValueMonthlySum enthält pro Monat die Summe aller Sensordaten, deren Summe für Nutzer nützlich ist, zum Beispiel von Verbrauchsdaten.

SensorValueMonthlyAvg beinhaltet pro Monat den Mittelwert aller Sensordaten, deren Mittelwert für Nutzer nützlich ist, zum Beispiel die Außentemperatur.

Um häufige Abfragen zu beschleunigen, haben wir dabei auch materialisierte Sichten genutzt. Diese vereinfachen nicht nur die Abfragen wie normale Sichten, sondern berechnen ihr Ergebnis vor und halten es für den schnelleren Zugriff im Cache. Wenn diese Sichten häufiger abgefragt werden, als sie sich ändern, lohnt sich diese Art des Caching.

3.2.4 Client

Der Server enthält alle Funktionen die ecoControl benötigt. Das System ist daher theoretisch nur mit dem Server komplett. Es ist durch die API – mit Hilfe eines Tools, dass HTTP-Anfragen schicken kann – theoretisch bedienbar.

Für unsere Nutzergruppe wäre ecoControl so nur schwer nutzbar. Die Nutzer würden eine lange Einarbeitungszeit benötigen und auch danach wäre die Bedienung aufwändig, da die Informationen, nur in Textform präsentiert, schwer überschaubar wären. Daher benötigt ecoControl ein Frontend, das als Client die API

nutzt und sie für Menschen nutzbar macht. Dieser Client sollte Informationen über das System abfragen und aufbereiten. Eine gute Aufbereitung sind zum Beispiel Diagramme. Der Client sollte die Steuerung des Energiesystems durch Eingabehilfsmittel vereinfachen. Mögliche Hilfsmittel können Formulare, simple Schaltflächen oder Ähnliches sein. Zusätzlich muss der Client eine Authentifizierung beim Server unterstützen, da die API sonst nicht genutzt werden kann.

Wir haben als Frontend eine Webseite entwickelt. Eine genaue Beschreibung befindet sich in Kapitel 2. Die Webseite wird von einem Browser angezeigt. Theoretisch kann die Webseite von einem anderen Server an den Browser ausgeliefert werden. Diese Komponente stellen wir hier jedoch nicht dar.

3.2.4.1 Webseite

Die Webseite bildet eine interaktive graphische Oberfläche für die Techniker und Verwalter, die mit ecoControl arbeiten sollen. Sie zeigt Informationen über das Energiesystem in Diagrammen und Tabellen an. Sie hat ein Login implementiert, mit der sich der Nutzer beim Server authentifizieren kann. Die verschiedenen Rechte der Rollen hat sie durch zwei Webseitenbereiche abgebildet, die nur den jeweiligen Rollen zugänglich sind. Die Steuerung erleichtert sie durch Eingabefelder und Slider. Damit Techniker die Programmierschnittstelle ansprechen und auf diese Art Regeln formulieren können, enthält die Webseite einen Code-Editor. Dieser ist ein Editor, der mit Hilfe der Bibliothek *ace*⁷ in die Webseite integriert ist. Sie bietet zusätzlich noch einen Export der Daten in csv-Format auf die Clientmaschine an.

Die Steuerung Wie erwähnt, muss der Client von ecoControl Nutzereingaben entgegennehmen und diese an den Server senden. Hat der Nutzer eine Regel in Form von Code in den Code-Editor eingegeben und bestätigt die Eingabe, sendet die Webseite den Code an die Programmierschnittstelle des Servers. Dies geschieht per Ajax in Form einer Anfrage an die passende URL des Servers: `POST /api/code/`. Inhalt der Anfrage ist ein JSON, das den Code in folgender Form enthält:

```
{
  "code": <Python code>
}
```

Der Server schickt dem Frontend eine Meldung, ob die Regel erfolgreich geladen wurde. Diese Meldung erfolgt wieder im JSON-Format:

```
{
  "code": <Python code>,
  "status": 1
}
```

Die Eigenschaft `status` enthält dabei 1, falls die Anfrage erfolgreich war, sonst 0.

Visualisierung des Systems Genauso wichtig ist die Visualisierung der Gerätedaten. Eine detaillierte Beschreibung der Abläufe bei einer Visualisierung geben wir in Abschnitt 3.3.

⁷Ace (Ajax.org Cloud9 Editor), <http://ace.c9.io> (abgerufen am 19.11.2014).

3.2.4.2 Eingesetzte Bibliotheken

Die Benutzeroberfläche von ecoControl wurde mit Hilfe des Frameworks *Bootstrap*⁸ entwickelt. Bootstrap stellt viele vorgefertigte Komponenten für Benutzeroberflächen zur Verfügung. Diese sind optisch aufeinander abgestimmt, daher hat die Webseite ein einheitliches Design. Es bietet außerdem ein responsives Design. Daher werden Webseiten auch auf mobilen Geräten mit kleineren Bildschirmen richtig angezeigt. Bootstrap basiert auf CSS, HTML und JavaScript.

Das Frontend ist hauptsächlich in JavaScript geschrieben und nutzt dazu die Bibliothek *jQuery*⁹. jQuery erleichtert die Traversierung und Manipulation von HTML-Dokumenten, die Ereignisbehandlung und die Ajax-Nutzung. Die Webseite nutzt Ajax für die Kommunikation mit dem Server.

Die Diagramme haben wir mit Highstock und Highcharts umgesetzt. Dadurch unterstützt ecoControl auch Interaktionen mit den Diagrammen, beispielsweise Zoomen. Ein weiterer Grund für die Nutzung dieser Bibliotheken war, dass uns die Diagramme optisch gut gefielen.

Die Webseite bietet zusätzlich zu dem Userinterface zur Server-API auch eine Exportfunktion an. Diese ist direkt in der Webseite implementiert. Alle Diagramme können als Bilddatei und alle Tabellen im CSV-Format exportiert werden. Die Highcharts-Bibliothek stellt diese Funktionalität für ihre Elemente bereits zur Verfügung. Die restlichen Tabellen werden mit Hilfe von *jquery.table2csv*, einer jQuery Funktion, exportiert.

Die Webseite ist nur ein möglicher Client des ecoControl-Servers. Es könnte weitere Umsetzungen geben, die, wie die Webseite, den Systemzustand anzeigen und die Steuerung vereinfachen. Ein Beispiel wären Smartphone-Apps. Es wären aber auch Clients möglich, die nur einen Teil der API nutzen. Eine Idee wäre ein reines Benachrichtigungssystem, das nur die Benachrichtigungen über das Verletzen von Schwellenwerten überwacht und bei Änderungen einen Alarm gibt.

3.3 Die Kommunikation zwischen Server und Client

In diesem Abschnitt werden wir die Kommunikation von Server und Client am Beispiel eines Sensormesswerts zeigen.

Der Wärmespeicher sendet am 1.12.2013 um 0 Uhr an die Geräteschnittstelle, dass sein Temperatursensor 40° C gemessen hat. ecoControl speichert folgenden Datensatz in der Datenbank:

```
SensorValue(sensor = Waermespeicher, value = 40,  
            timestamp = "2013-12-01 00:00:00" )
```

Im Januar 2014 öffnet ein Nutzer im Webseitenbereich des Verwalters die Statistikseite. Wie in Kapitel 2 beschrieben, können auf dieser Seite Nutzer die Sensoren wählen, die sie vergleichen möchten. Er wählt dabei als zu vergleichende Werte die Auslas-

⁸Twitter Bootstrap <http://getbootstrap.com/> (abgerufen am 19.11.2014).

⁹jQuery, <http://jquery.com/> (abgerufen am 19.11.2014).

tung des BHKWs und die Temperatur des Wärmespeichers. Als Zeitraum legt er das Jahr 2013 fest. Die Webseite sendet nun mehrere Ajax-Anfragen an den Server, unter anderem nach der Temperatur des Wärmespeichers: `GET /api/avgs/sensor/1/year/2013/`. Die Anfrage enthält die Angabe, dass der Durchschnitt bestimmter Sensordaten gefragt ist (`avgs`). Sie beinhaltet den gefragten Sensor in Form einer Sensor-ID (1). Die API unterstützt als Zeiträume nur ein komplettes Kalenderjahr. Dieses ist hier das Jahr 2013. Beim Aufruf der Statistikseite hat die Webseite bereits eine Sensorliste beim Server erhalten, die alle Sensoren und Sensor-IDs aufzählt. Durch die Anfrage werden beim Server die entsprechenden Hooks aktiviert. Im Fall der Anfrage nach dem Wärmespeicher ist das die Funktion `manager.hooks.get_avgs`, aufgerufen mit den Parametern 1 und 2013.

Diese Funktion erstellt mit Hilfe des ORMs auf Basis der Sicht `SensorValueMonthlyAvg` die gewünschte Aggregation. Da `SensorValueMonthlyAvg` eine materialisierte Sicht ist, hatte die Datenbank bereits für jeden Monat aus jedem einzelnen Sensorwert (Tabellename: `SensorValue`) den Mittelwert berechnet und in einer Tabelle gespeichert. In diese Aggregation ist auch der einzelne Eintrag vom 1.12. eingegangen. Zuletzt wandelt die Funktion sie in ein JSON um und sendet es in einer HTTP-Response zurück an die Webseite (Quelltext 3.2).

Quelltext 3.2: Vereinfachtes JSON-Objekt als Antwort auf eine Anfrage nach gemittelten Sensorwerten

```
[
  {
    "timestamp": "2013-01-01",
    "total": 57.575760694008
  },
  {
    "timestamp": "2013-02-01",
    "total": 56.4845847954642
  },
  ..
  {
    "timestamp": "2013-12-01",
    "total": 56.7050982769989
  }
]
```

Das Highchart-Säulendiagramm ist bereits beim ersten Aufruf der Statistikseite erstellt worden. Wenn die Antwort mit dem JSON angekommen ist, ordnet der JavaScript-Code der Webseite den Diagrammen die neuen Beschriftungen aus der Anfrage und die neuen Werte aus der Antwort zu. Danach weist er das Diagramm an, sich zu aktualisieren. Anschließend wird die Prozedur mit der Tabelle wiederholt. Ihr werden die neuen Werte zugeordnet und sie aktualisiert sich. Der Nutzer kann nun eine Seite wie in Abbildung 3.5 sehen.

3 Entwicklung von ecoControl

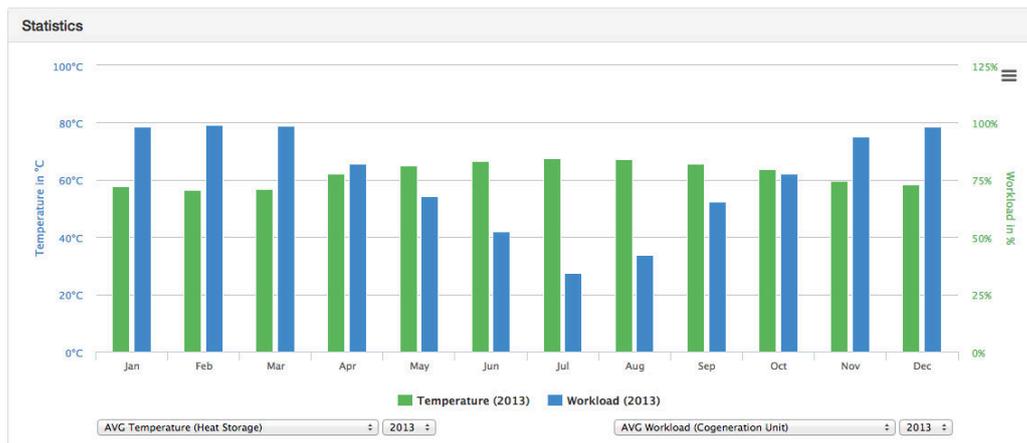


Abbildung 3.5: Die Statistikseite zeigt die Aggregation als Säulendiagramm an

3.4 ecoControl als Open Source Software

Eine Anforderung des Kunden, die von Anfang an bestand, war, dass ecoControl Open Source sein sollte. Es sprechen einige Gründe dafür, ecoControl auf diese Weise zugänglich zu machen. ecoControl ist erst ein Prototyp.

Da ecoControl Open Source ist, ist es einfach, relevantes Wissen über die Domäne zu bündeln und die Software mit diesem Wissen weiterzuentwickeln. ecoControl ist darauf angewiesen, alle Geräte eines Energiesystems zu unterstützen. Da es mehr Geräte gibt und geben wird, die in einem Haus installiert sein können, als ecoControl momentan unterstützt, muss ecoControl weiterentwickelt werden. Geräte, die benötigt werden, deren Typ aber noch nicht unterstützt wird, können von Programmierkundigen ergänzt werden. Ein Beispiel ist der Gerätetyp Photovoltaik. Gleiches gilt für verschiedene Modelle eines Geräts, falls ein Modell Besonderheiten hat, die in ecoControl beachtet werden müssen. Ein weiterer Vorteil ist, dass ecoControl kostenlos ist. Ein Nachteil, der mit der Open Source entsteht, ist die fehlende Garantie. Wird ecoControl eingesetzt, steuert es teure Geräte. Die fehlende Garantie könnte potenzielle Nutzer der Software abschrecken.

Auch Bibliotheken, die ecoControl benötigt, sind kostenlos oder sogar Open Source. Eine Übersicht über die genutzten Module befindet sich in Tabelle 3.1.

3.5 Nächste Schritte

Der nächste und wichtigste Schritt ist, den Quellcode von ecoControl zu veröffentlichen. Dazu gehört, noch eine Dokumentation zu schreiben, um den Einstieg in den Code von ecoControl zu vereinfachen.

ecoControl ist in seinem momentanen Zustand noch ein Prototyp. Es wurde bis jetzt nur an einer Simulation getestet, die ein idealisiertes Verhalten aufwies, und nicht an echten Geräten. Für die Weiterentwicklung ist es daher sehr wichtig, die Soft-

Tabelle 3.1: Überblick über genutzte Open Source Bibliotheken

Name	URL (jeweils abgerufen am 19.11.2014)	Version
Django	http://www.djangoproject.com/	1.6.2
Mock	http://pypi.python.org/pypi/mock/	1.0.1
NumPy	http://www.numpy.org/	1.8.1
Psycopg	http://initd.org/psycopg/	2.5.2
python-dateutil	http://labix.org/python-dateutil	1.5
pytz	http://pytz.sourceforge.net	2012d
SciPy	http://www.scipy.org/	0.14.0
south	http://south.aeracode.org/	0.8.4
PostgreSQL	http://www.postgresql.org/	9.3
Bootstrap	http://getbootstrap.com/	3.2.0
jQuery	http://www.jquery.com/	2.1.1
Highstock*	http://www.highcharts.com/	4.0.1
Highcharts*	http://www.highcharts.com/	2.0.1
Ace	http://ace.c9.io/	1.1.4
jQuery Address	http://www.asual.com/jquery/address/	1.5
jQuery Cookie	http://plugins.jquery.com/cookie/	1.4.1
jquery-dateFormat	http://github.com/phstc/jquery-dateFormat/	1.0.1
jQuery UI	http://jqueryui.com/	1.11.0
mustache	http://mustache.github.io/	0.8.2
X-editable	http://vitalets.github.io/x-editable/	1.5.1

*frei nutzbar in nicht-kommerziellen Projekten (*Creative Commons Attribution-NonCommercial 3.0 License*)

ware mit passenden Geräten zu testen. Damit das möglich ist, ist allerdings neben den Geräten auch noch die Geräteerweiterung Voraussetzung, die die Kommunikation der Geräte mit der Hardwareschnittstelle ermöglicht. Eine Oberfläche statt des Code-Editors oder neben ihm, die es ermöglicht Regeln ohne Code zu formulieren, wäre eine sinnvolle Weiterentwicklung der Webseite. Weil wir vermuten, dass nur wenige Nutzer programmieren können, wäre ecoControl so mehr Nutzern zugänglich.

3.6 Prozess

Neben der reinen Programmierleistung ist auch die Projektorganisation wichtig für ein Projekt. Bei der Entwicklung fielen auch Arbeiten wie die Aufgabenverteilung oder Dokumentation an. Eine der wichtigsten dieser Aufgaben war das Finden der konkreten Anforderungen an ecoControl. Zu Beginn der Entwicklung hatten wir nur wenige, nicht konkrete, Anforderungen an die Software. Um die Anforderungen zu konkretisieren, haben wir verschiedene Methoden genutzt. Eine davon war die User Research, die in Kapitel 2 ausgiebig vorgestellt wurde. Wir haben für die Entwicklung ein iteratives Vorgehen gewählt. In diesem Abschnitt untersuchen wir, wie sich eine iterative Vorgehensweise beim Entwickeln auf ecoControl ausgewirkt hat.

3.6.1 Unser Vorgehen bei der Entwicklung

Am Anfang des Projekts stand die Idee eines Kunden¹⁰. Im Laufe des Projekts kamen Anforderungen vom Kunden und aus den Erkenntnissen der Nutzerstudien zusammen. Aus den Anforderungen erstellten wir Aufgaben und priorisierten sie. Wir wählten die wichtigsten Aufgaben und verteilten sie unter uns. Die Aufgaben waren ungefähr auf die Länge einer Iteration zugeschnitten. Eine Iteration dauerte etwa drei Wochen und wurde mit der Präsentation des aktuellen Stands beendet. Das erhaltene Feedback arbeiteten wir in die Anforderungen ein, erstellten neue Aufgaben, priorisierten sie und begannen die nächste Iteration.

Als Hosting-Plattform für unser Repository nutzten wir GitHub¹¹. Daher konnten wir auch dessen Organisationshilfen nutzen. Wir legten Anforderungen und daraus resultierende Aufgaben als Issues an. Aktuelle Aufgaben waren Issues, die einem Entwickler zugewiesen waren. Zur Strukturierung der Issues nutzten wir Milestones und Tags. Die Tags setzen wir zur thematischen Strukturierung ein, ein Beispiel ist der Tag „frontend“. Milestones nutzten wir zur thematischen Strukturierung („THE Simulation“), aber auch für eine zeitliche Einordnung („Post-Presentation“). Zeitliche Grenzen, die Meilensteine setzen können, haben wir jedoch nicht verwendet.

¹⁰Agentur für Marketing und Informationsmanagement, <http://ami-net.de/> (abgerufen am 19.11.2014).

¹¹GitHub, <https://github.com/> (abgerufen am 19.11.2014).

Wir setzten ebenfalls das zum Repository gehörende Wiki ein. Es hatte bei uns zwei wichtige Funktionen: Archiv und Informationsplattform. Wir legten Interview-Mitschriften oder andere Notizen zeitlich sortiert im Wiki ab. Außerdem teilten wir über das Wiki Informationen wie Coding Standards oder die Nutzung einer Schnittstelle von ecoControl.

3.6.2 Verschiedene Entwicklungsstufen von ecoControl

So entstand ecoControl insgesamt über acht Iterationen hinweg. Nach der zweiten Iteration hatten wir bereits einen ersten Prototyp von ecoControl geschrieben. Er bestand aus einer BHKW-Simulation, einem Server, der die Simulationswerte über eine API zugänglich machte, einer Datenbank und einer einfachen Webseite. Als nächstes entwickelten wir ein Frontend, das es dem Nutzer erlaubte, Regeln zu erstellen. Dieses Frontend nutzten wir jedoch nicht weiter. Im nächsten Schritt erweiterten wir die Simulation und die zugehörige Webseite. Die Datenbank entfernten wir in diesem Entwicklungsstadium. In der weiteren Entwicklung bauten wir die Simulation zur Vorhersage aus. Parallel erweiterten wir die Webseite um die beiden Nutzerbereiche. In den letzten Iterationen wurden die Datenbank wieder eingebaut und die Funktionen des Servers weiter verfeinert.

Einen Überblick über die Entwicklungsstufen der Software gibt Abbildung 3.6.

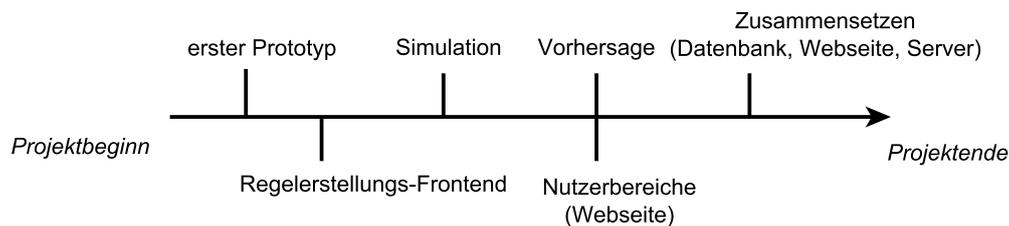


Abbildung 3.6: Die Entwicklungsstufen von ecoControl

3.6.3 Fallstudie: Regelwerk

Dieser Abschnitt beschreibt anhand eines Beispiels, wie wir eine Anforderung im Laufe der Entwicklung von ecoControl umgesetzt haben. Dabei wurde die erste Implementierung verworfen und später eine andere Version der Umsetzung gewählt.

Die Anforderung des Projektinitiators war: „Die Software soll durch Regeln konfiguriert werden können. Der Nutzer soll Bedingungen formulieren können, die vom Sensorzustand oder anderen Parametern bestimmt werden. Bei deren Eintreffen wird das System nach der Angabe des Nutzers gesteuert.“

3 Entwicklung von ecoControl

Diese Anforderung werden wir in den nächsten Abschnitten verkürzt Regeln oder Regelsystem nennen. Wie im vorherigen Abschnitt erwähnt, wurde eine Oberfläche zur Regelerstellung im Laufe des Projekts erstellt und wieder entfernt. Die Anforderung wurde letztendlich mit der Programmierschnittstelle erfüllt.

3.6.3.1 Ein Frontend zur Regelerstellung

Als wir begannen, eine Oberfläche zu Regelerstellung zu bauen, sahen wir das Regelsystem als eines der wichtigsten Features von ecoControl an. Der Grund war, dass es eine Art der Steuerung des Energiesystems ermöglichen würde, die neu für die Geräte war.

Die Regeln sollten in der Lage sein, mehrere Geräte zu beachten und ihr Verhalten in Relation zueinander zu setzen. Mit Hilfe der Regeln sollte ecoControl die Geräte flexibel steuern und auch auf unregelmäßige Ereignisse reagieren können. Solche Ereignisse könnten plötzliche Werteänderungen, zum Beispiel ein Wetterumschwung, sein.

Als Hauptnutzer der Regeln sahen wir bereits zu diesem Zeitpunkt die Rolle Techniker. Es erfolgte eine Umsetzung unserer Vorstellung einer Schnittstelle, um Regeln zu konfigurieren. Der Stand, den wir nach einigen Wochen präsentierten, bestand aus einer Frontendlösung. Das Backend war noch nicht funktionsfähig.

Das Frontend erlaubte es dem Nutzer, Sensorwerte zu vergleichen, und anhand des Ergebnisses Geräteeinstellungen oder andere Einstellungen zu setzen. Die Bedingung verglich Sensorwerte oder Konstanten miteinander. Die daraus folgende Konsequenz bestand nur aus einem Textbaustein, der für eine Konfigurationsänderung des Systems stand.

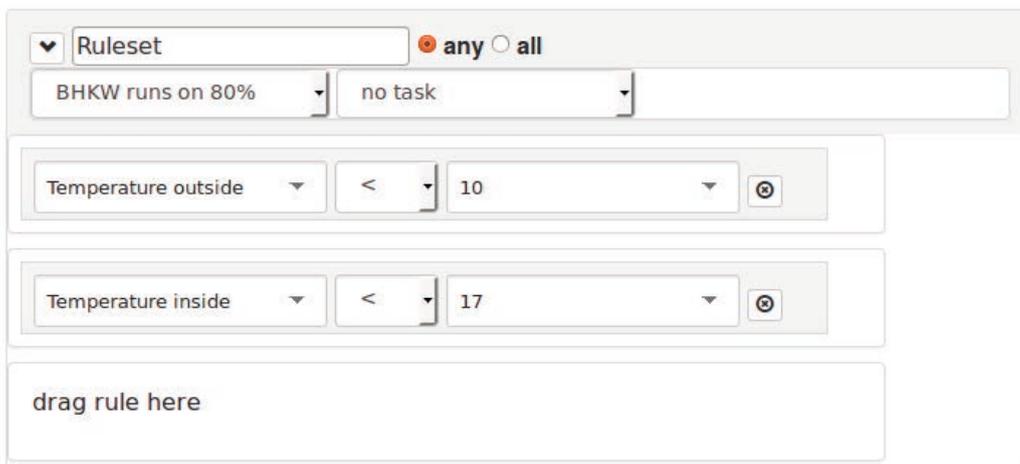


Abbildung 3.7: Beispielregel im Frontend zur Regelerstellung

In Abbildung 3.7 befindet sich eine Umsetzung der Regel: „Wenn es kalt ist, soll die Heizung warm sein.“ Wie in der Abbildung zu sehen, wurde dies im Frontend einge-

geben zu der Bedingung „Außentemperatur < 10 °C oder Innentemperatur < 17 °C“ und der Folge „*bhkw.workload* = 80 %“.

3.6.3.2 Analyse der Rückmeldungen

Wie üblich, präsentierten wir den Stand dem Kunden. Dabei nutzten wir das obige Beispiel, um das Frontend vorzuführen.

Während des Vortrags fiel bei diesem Beispiel der Gegensatz zwischen der geringen Komplexität der Regel und der hohen Komplexität der Regelerstellung auf. Die Regel hatte einen sehr einfachen Hintergrund. Die Erstellung erforderte dagegen, neben einer aufs Grad genauen Temperaturkonfiguration, ein Verständnis der booleschen Logik und Wissen über technische Parameter des Geräteaufbaus.

Die Zuhörer nannten keine konkreten Verbesserungsvorschläge, sondern äußerten Kritik am Konzept der Regeln. Einer der Einwände war, dass eine Heizung auch ohne unser Regelsystem und die formulierte Regel funktionieren würde. Ein zweiter war, dass die Einarbeitung in die Regelsyntax für Mieter zu aufwändig wäre. Sie nahmen an, dass Mieter die Zielgruppe der Regeln wären und zweifelten daher an deren Nutzen und Praktikabilität.

Weil wir eine so naive Regel präsentiert hatten, hatten wir in der Präsentation unsere eigentliche Zielgruppe, die Techniker, nicht ausreichend herausgestellt. Wir vermuteten, dass die Zuhörer keine Möglichkeit hatten, das Regelsystem so zu sehen, dass es überhaupt eine sinnvolle Erweiterung von *ecoControl* darstellte und deshalb kein konkretes Feedback geben konnten.

Während dieser Analyse stellten wir fest, dass wir zu diesem Zeitpunkt keine einzige Regel kannten, die, in einem System angewendet, Geld oder Energie gespart hätte. Wir konnten also auch uns selbst nicht beweisen, dass das Regelsystem nützlich war.

3.6.3.3 Konsequenzen und neue Anforderung

Wir erkannten, dass wir noch nicht genug Wissen über die Domäne hatten. Wir hörten auf, die Anforderung Regelsystem weiter zu bearbeiten, und entfernten das Frontend aus dem aktuellen Entwicklungsstand. Unsere aktuelle Aufgabe wurde die Entwicklungsdomäne Energiesysteme. Wir wollten uns besonders auf die BHKW konzentrieren.

Zu diesem Zweck recherchierten wir bei Anbietern von BHKW und anderen Informationsplattformen. Unser Kunde konnte uns an dieser Stelle einige Anlaufpunkte nennen. Die BHKW-Infothek [7] war einer dieser Anlaufpunkte und sehr hilfreich. In dieser Zeit meldeten wir uns auch in einem Forum an, das auf BHKW spezialisiert war. Dort knüpften wir mehrere Kontakte, die bei der User Research eine Hilfe waren (vergleiche Kapitel 2).

Zu den wichtigsten Informationen gehörte auch die Antwort auf die Frage, ob sich ein BHKW finanziell lohne. Dazu gab es verschiedene Beispielrechnungen, die wir nachvollzogen. Am Ende nutzten wir unser neues Wissen, um eine Simulation eines Energiesystems mit einem BHKW zu entwickeln (vergleiche Kapitel 1). Zwei Entwicklungszyklen später hatten wir die Simulation entwickelt, inklusive einer Webseite, um die Simulation ausgiebig zu testen. Als Entwickler nutzten wir innerhalb des Frontends eine Programmierschnittstelle, um die simulierten Geräte zu

steuern. Diese Schnittstelle erlaubte es uns, die Simulation so zu manipulieren, wie die Regeln des Regelsystems später die echten Geräte manipulieren können sollten.

Nach der Simulation entwickelte sich ecoControl dazu, Vorhersagen zu unterstützen. Die Programmierschnittstelle blieb im Frontend.

3.6.4 Bewertung des Prozesses

Da die Regeln Teil der Anforderungen waren, blieb die Programmierschnittstelle in ecoControl integriert. So erfüllten wir die gestellte Anforderung. In diesem Fall führte die Rückmeldung zu der Erkenntnis, dass wir die Domäne besser kennen lernen sollten.

Positiv an dem Ablauf war, dass wir dies erkannten. Außerdem haben wir die Anforderung so implementiert, dass sie weniger aufwändig als geplant geriet. Als Programmierschnittstelle war die Implementierung am Ende für uns Entwickler sehr praktisch, da wir Code besser lesen und editieren konnten, als wir die Drag'n'Drop-Elemente der Oberfläche bedienen konnten.

Dadurch, dass die Regeln weniger Arbeit kosteten als geplant, blieb Raum für andere Features, besonders die Vorhersage. Negativ war, dass die Regeln in ihrer jetzigen Form nur für Menschen nutzbar sind, die mit den grundlegenden Elementen von Python umgehen können. Dies ist Wissen, das wir bei einem Techniker nicht voraussetzen können. Da wir die Regelerstellung noch nicht mit Nutzern getestet haben, haben wir an dieser Stelle keine Rücksicht auf die vermutete fehlende Kenntnis der Nutzer genommen.

Wir halten den Verlauf für ein insgesamt gutes Ergebnis und sind daher der Meinung, dass der Prozess und das Ergebnis ecoControl von der iterativen Vorgehensweise profitiert haben. Eine Voraussetzung für das Gelingen des Prozesses war eine funktionierende Kommunikation, über die wir Feedback erhalten konnten. Die Präsentationen haben sich dafür als nützlich erwiesen.

3.6.5 Vorgehen im Vergleich zu Scrum

Unser Vorgehen war an das Softwareentwicklungs-Framework Scrum[40] angelehnt. Unsere Iterationen waren den Sprints ähnlich. Sie hatten in der Regel eine feste Dauer von drei Monaten und endeten mit einer Präsentation des Systems vor dem Kunden. Wir vollzogen allerdings keine strikte Trennung zwischen den aktuellen Aufgaben für diese Iteration und den übrigen Aufgaben. Auf Scrum übertragen haben wir Sprint Backlog und Product Backlog nicht sauber getrennt. Wenn uns ein wichtiger Input während Iteration erreichte, setzten wir daraus resultierende Aufgaben auch in derselben Iteration um. Sprint Review und Sprint Planning haben wir prinzipiell mit unserer Analyse und dem Planen neuer Aufgaben nach den Präsentationen umgesetzt. Unregelmäßiger und seltener als einmal pro Iteration setzten wir uns zu einer Retrospective zusammen. Wir haben kein Daily Scrum umgesetzt. Die Rollen, die Scrum definiert, waren bei uns nicht klar verteilt. Wir hatten keinen Product Owner, sondern haben dessen Aufgaben gemeinsam übernommen.

3.7 Zusammenfassung

Das Projekt, in dessen Rahmen ecoControl entwickelt wurde, ist fast beendet. Ziel unseres Projektes war es, eine Anwendung zur softwaregestützten Aussteuerung des Energiesystems in Mehrfamilienhäusern zu entwickeln. Einen ersten wichtigen Schritt in diese Richtung haben wir mit dem vorliegenden Produkt erreicht. Auch wenn das Projekt abgeschlossen ist, ist es unserer Meinung nach lohnenswert, die Software weiterzuentwickeln. Für den wichtigsten Schritt halten wir es, echte Geräte einzubinden.

Die Zukunft von ecoControl ist davon abhängig, ob es unserem Kunden und uns gelingt, noch mehr Menschen für die Idee von ecoControl zu gewinnen. Den Code von ecoControl auf GitHub zu veröffentlichen, ist der erste Schritt, mehr Menschen in das Projekt einzubeziehen. Idealerweise wird sich auf dieser Basis eine Open-Source-Community bilden. Menschen mit Zugang zu den technischen Geräten und mit Wissen über diese wären besonders wichtig, um ecoControl zu testen und weiter zu entwickeln. Mögliche Ansprechpartner sind BHKW-Besitzer. Unser Kunde ist auch im Gespräch mit anderen Universitäten, die als Kooperationspartner in Frage kämen. Möglicherweise sind auch Firmen, die Geräte vertreiben, und Planungsfirmen potentielle Ansprechpartner.

Wir sehen die aktuelle Version ecoControl als wichtigen ersten Schritt, um durch die effektive Steuerung moderner dezentraler Energiesysteme die begrenzten Energieressourcen zu schonen. Die mit dem Einsatz von ecoControl verbundenen Einspareffekten können unserer Meinung nach die Attraktivität der dezentralen Energiesysteme erhöhen. Wenn BHKW und Photovoltaikanlagen in Mehrfamilienhäusern verbreiteter werden, ist die Energiewende ein großes Stück weiter.

4 Prognose und simulationsgestützte Optimierung von Energiesystemen in Mehrfamilienhäusern

Im Zuge der Energiewende hat die dezentrale Energieversorgung in Deutschland in den letzten Jahren stark zugenommen. Eine steigende Anzahl von Mehrfamilienhäusern deckt ihren Energiebedarf durch Photovoltaikanlagen und Blockheizkraftwerke. Allerdings wird die Effizienz solcher Energiesysteme noch durch starke Verbrauchsschwankungen beeinträchtigt, da häufig zu viel oder zu wenig Strom und Wärme produziert werden. Die Geräte können sich nicht rechtzeitig auf Verbrauchsschwankungen einstellen, da keine Schätzungen des zukünftigen Verbrauchs vorhanden sind.

Dieses Kapitel stellt einen Ansatz zur Kurzzeitprognose von Nutzerverbräuchen in Mehrfamilienhäusern mittels einer Zeitreihenanalyse vor und nutzt diese Prognose zur Optimierung der Wirtschaftlichkeit des Systems. Die Verbräuche werden in Wochentage aufgeteilt und einzeln durch das Holt-Winters Verfahren vorhergesagt. Die Lastgangprognose wird zusammen mit der Wettervorhersage zur Simulation des zukünftigen Verhaltens der Energiesysteme verwendet. Die effiziente Aussteuerung des Systems wird durch Minimierung einer Kostenfunktion über der Simulation erreicht. Der in diesem Kapitel vorgestellte Ansatz liefert für Prognosen von bis zu einer Woche bessere Ergebnisse als das Holt-Winters Verfahren ohne Tagestrennung. Die Minimierung der Kostenfunktion konnte die Abdeckung der elektrischen Last durch Eigenproduktion um etwa 20 % steigern und somit die Wirtschaftlichkeit deutlich erhöhen.

4.1 Einleitung

Viele große Energieversorger verwenden Prognosen zur Vorhersage der Verbrauchslast, da die Kraftwerke verzögert auf Leistungsänderungen reagieren [21].

Auch in Mehrfamilienhäusern brauchen dezentrale Energiesysteme Zeit, um sich auf veränderte Lasten einzustellen. Bei Geräten, die Kraft-Wärme-Kopplung nutzen, ist es erstrebenswert, die Produktion von Strom und Wärme an den Nutzerbedarf anzupassen. Diese Fahrweise kann die Wirtschaftlichkeit erhöhen und die Energieverschwendung minimieren, wie in Kapitel 5 erläutert wird. Prognosen des Nutzerverbrauchs in Mehrfamilienhäusern helfen dabei, die Strom- und Wärmeproduktion an den tatsächlichen Verbrauch anzupassen.

4.1.1 Zielsetzung

Eine These zur Effizienzsteigerung lautet: Das Blockheizkraftwerk (BHKW) läuft am effizientesten, wenn zwei Bedingungen erfüllt sind. Es muss genau so viel Strom

selbst produziert werden, wie verbraucht wird. Gleichzeitig muss die Wärmeversorgung des Hauses gedeckt sein. Dies werden wir in Kapitel 5 näher erklären.

Mithilfe einer Prognose des Nutzerverbrauchs und der Gerätesimulation kann das Verhalten der Energiesysteme vorhergesagt werden. Anhand dieser Vorhersage kann das System dann so eingestellt werden, dass es die oben genannten Bedingungen erfüllt und somit effizient läuft. Zielsetzung dieses Kapitels ist es, ein Prognosesystem zu entwickeln, das sich zur Optimierung der Geräte eignet.

In Abschnitt 4.2 werden die verwendeten Prognosen kategorisiert und die Verfügbarkeit von Eingabedaten erörtert. Der Nutzerverbrauch kann mithilfe von statistischen Prognosen vorhergesagt werden. Dazu erläutern wir in Abschnitt 4.3 die exponentielle Glättung mit Fokus auf die Holt-Winters Methode. `ecoControl` wendet eine modifizierte Version von Holt-Winters an, um den elektrischen Bedarf für zwei Wochen vorherzusagen. Wir stellen in Abschnitt 4.4 vor, wie diese Prognose und die Wettervorhersagen dynamisch in `ecoControl` eingebunden und erneuert werden. In Abschnitt 4.4 stellen wir einen Algorithmus vor, der mithilfe des Prognosesystems die Geräte möglichst rentabel aussteuert. Die Evaluation und Diskussion der Ergebnisse erfolgt in Abschnitt 4.6 und Abschnitt 4.7.

4.2 Analyse

In diesem Abschnitt werden wir die Vorhersage in mehrere Teilprognosen kategorisieren und die dazu benötigten Vorhersagemodelle vorstellen. Da die Prognose auf Werten der Vergangenheit berechnet wird, werden wir die Verfügbarkeit der Daten in einem Mehrfamilienhaus erläutern.

4.2.1 Kategorisierung

Der erste Schritt bei der Erstellung eines Prognosemodells ist die Identifizierung der zu prognostizierenden Komponenten.

Das entscheidende Kriterium bei der Prognose des Heizungssystems ist die genaue Kenntnis des Strom- und *Wärmeverbrauchs* sowie der *Energieproduktion* im Haus. Die Gesamtprognose lässt sich also in mehrere Teilprognosen aufteilen. Alle Prognosen, die in `ecoControl` benutzt werden, lassen sich in eine der folgenden drei Kategorien einteilen.

4.2.1.1 Geräteprognosen

Die Geräte besitzen physikalische und technische Eigenschaften, wie beispielsweise die Leistung des BHKWs oder das Fassungsvermögen des Pufferspeichers, die ihr Verhalten bestimmen.

Die Modellierung der physikalisch-technischen Gesetzmäßigkeiten stellt eine eigene Kategorie an Prognosemodellen dar. Die beeinflussenden Faktoren auf ein abgeschlossenes System sind dabei ausreichend bekannt und können nachmodelliert werden. Die Prognose wird dann durch die *Simulation*, also der Realisierung

des Modells erstellt. In Kapitel 1 wurde gezeigt, wie die Geräte (Energiewandler, -speicher und -verbraucher) simuliert werden können.

4.2.1.2 Nutzerverhalten

Andererseits hängt die Fahrweise von BHKW und Heizkesseln natürlich vom Nutzerverhalten ab. Das Nutzerverhalten kann in elektrisches und thermisches Nutzerverhalten unterteilt werden.

Prinzipiell ist eine Simulation des elektrischen Nutzerverbrauchs zwar möglich [16], allerdings benötigt dies sehr genaue Informationen über die einzelnen Verbraucher im Haus. Somit ist es für den Einsatz in echten Wohneinheiten ungeeignet. Andererseits lassen sich bestimmte Verbrauchsmuster in den allgemeinen Verbrauchskurven abschätzen und mithilfe von statistischen Modellen vorhersagen.

Der thermale Verbrauch kann in Heizungsverbrauch und Warmwasserverbrauch unterteilt werden. Der Heizungsbedarf unterliegt deutlich weniger den Gewohnheitsmustern der Bewohner. Die Heizung hält die Räume auf einer vorgegebenen Temperatur, die Schwankungen in der bezogenen Heizleistung hängen vor allem von der Außentemperatur und den Baueigenschaften wie zum Beispiel der Dämmung ab. Die Heizleistung kann durch ein physikalisches Modell simuliert werden, wie in Kapitel 1 gezeigt wurde. Durch Einbeziehung der Wettervorhersage (siehe unten) kann nun der thermische Bedarf auch für die Zukunft prognostiziert werden.

Der Warmwasserbedarf hängt dagegen wieder stark vom Nutzerverhalten ab, beispielsweise gibt es Spitzenlasten am Morgen durch die Entnahme von Wasser zum Duschen. Durch diese Gewohnheitsmuster eignet sich somit auch der Warmwasserverbrauch für eine statistische Vorhersage. Die Voraussetzung hierfür ist, dass genügend Daten vorhanden sind, wie in Abschnitt 4.2.3 erläutert ist.

4.2.1.3 Externe Vorhersagen

In diese Kategorie fallen vor allem die Wettervorhersagen, die durch Online-Services bereitgestellt werden. Eine korrekte Wettervorhersage ist wichtiger Bestandteil anderer Prognosen, wie in Abbildung 4.1 erläutert ist.

4.2.2 Zusammenspiel der Prognosekomponenten

Die statistischen Prognosen bilden gemeinsam mit der Wettervorhersage ein Fundament für die Simulation der Geräte. Die Geräte greifen, wie in Abbildung 4.1 gezeigt, während der Simulation auf die Vorhersage des Strombedarfs zurück. Der thermale Bedarf setzt sich zusammen aus Heizlast, die in Abhängigkeit von der Wettervorhersage simuliert wird, und dem Warmwasserverbrauch, der statistisch vorhergesagt werden kann. Das Geräteverhalten und der thermale Bedarf werden wie in Kapitel 1 simuliert und für dieses Kapitel als gegeben betrachtet.

4.2.3 Vorhandene Daten und Datenbeschaffung

Die Software für die Erstellung der Prognose soll außerhalb der Forschungsumgebung auch in realen Mehrfamilienhäusern eingesetzt werden. Allerdings sind in Mehrfamilienhäusern meist nicht genug Daten verfügbar, die zu einer umfassenden

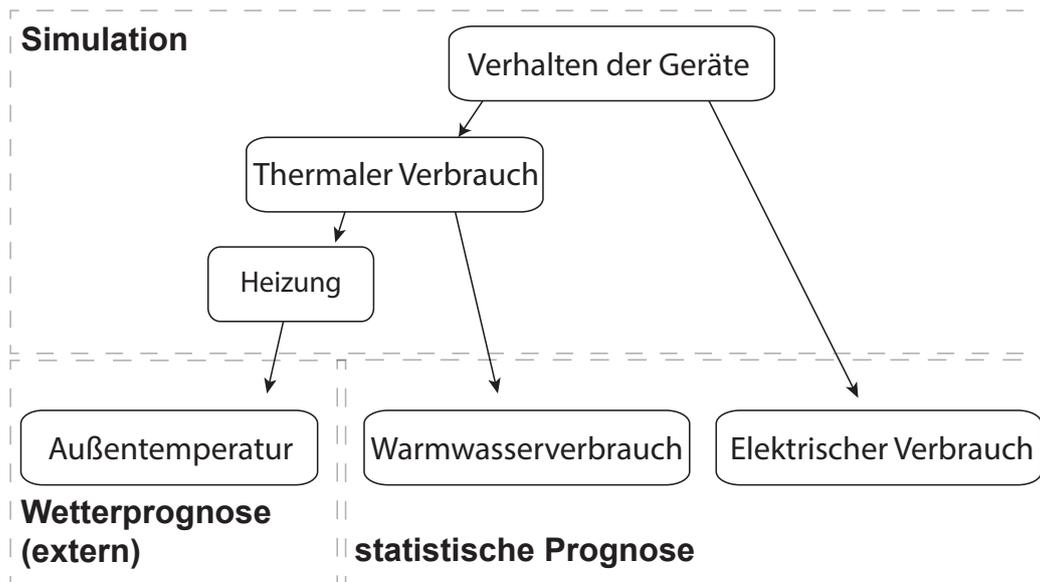


Abbildung 4.1: Abhängigkeit und Berechnungsmethoden der Prognosekomponenten

Prognoseerstellung gebraucht werden. Es stellt sich also die Frage, welche Daten bei Einbau dieses Systems verfügbar und nutzbar für Vorhersagen sind.

ecoControl soll vorrangig in Haushalten eingesetzt werden, die sich ein neues BHKW anschaffen oder bisher keine Optimierung an ihrem Energiesystem vorgenommen haben. Hier liegt das größte Potential zur Effizienzsteigerung.

Lastkurven sind elektrische oder thermische Verbrauchslasten eines Objektes (beispielsweise eines Mehrfamilienhauses), aufgetragen über die Zeit. Die Verfügbarkeit von Lastkurven des elektrischen und thermalen Bedarfs ist Voraussetzung für die statistische Prognose dieser Bedarfszahlen. In folgendem Abschnitt diskutieren wir daher die Verfügbarkeit der Daten.

4.2.3.1 Elektrischer Verbrauch

Herkömmliche, mechanische Zähler bieten als einzige Information die Anzahl der verbrauchten Kilowattstunden seit Beginn der Messung und einen aktuellen Stromverbrauch über die Drehgeschwindigkeit der Aluminiumscheibe des Zählers. Digitale Stromzähler, in Verbindung mit externer Datenverwaltung auch Smart-Meter genannt, speichern hingegen auch Lastkurven ab. Die aktuellen Verbräuche werden hierbei im 15 min Takt abgespeichert [32] und können je nach Installation für die gesamte Wohnanlage oder auch pro Wohnung gemessen werden. Wenn solche „intelligenten“ Stromzähler bereits in der Wohnung installiert sind, kann das System also auf eine ausführliche Datensammlung zurückgreifen. Die Verbreitung der Smart-Meter ist unter diesen Gesichtspunkten allerdings noch zu gering.

Die Europäische Union fordert bis 2020 den Einbau von Smart-Metern in 80 % aller Haushalte. In Deutschland ist der Einbau von intelligenten Energiezählern nach

§ 21c EnWG (Energiewirtschaftsgesetz) bei bestimmten Bedingungen inzwischen Pflicht. Das EnWG schreibt vor, bei Hausneubauten, Sanierungen und dem Kauf von KWK-Anlagen intelligente Stromzähler einzubauen.

Die tatsächliche Verbreitung in Deutschland ist bis dato aber eher gering. Die Chance auf vorhandene, ausführliche Lastkurven bei der BHKW Installation zuzugreifen, ist daher bis jetzt noch niedrig. In anderen europäischen Ländern, wie zum Beispiel Schweden oder Italien ist die Verfügbarkeit dagegen schon sehr hoch [20].

4.2.3.2 Thermischer Verbrauch

Detaillierte Lastkurven für thermische Verbräuche zu bekommen, gestaltet sich dagegen noch schwieriger. Allgemein wird nur der Gasverbrauch für den Heizkessel erfasst, im Normalfall auch nur als monatliche oder vierteljährliche Summe. Die Heizlast wird nur jährlich über eine Ablesung der Zähler erfasst. Eine Heizlastkurve wird dagegen üblicherweise nicht erfasst [29]. Die Heizlast muss aufgrund dessen und wegen der in Abschnitt 4.2.1.2 beschriebenen Abhängigkeit von der Außentemperatur simuliert werden und kann nicht statistisch ausgewertet werden.

Der Warmwasserbedarf muss statistisch ausgewertet werden, allerdings werden die benötigten stündlichen oder noch genaueren Warmwasserverbräuche üblicherweise nicht mitgeschrieben. Hierfür muss im Haus ein elektronischer Warmwasserzähler installiert werden, der periodisch über einen Datenlogger (zum Beispiel einen So-Recorder) ausgelesen wird [4].

4.2.3.3 Schlussfolgerung zur Datenverfügbarkeit

Bei der Inbetriebnahme eines BHKWs sind in der Regel zu wenige Daten vorhanden, um eine Prognose mit Vorjahresdaten auszuführen. Es kann allerdings nicht ein ganzes Jahr gewartet werden, bis diese verfügbar sind. Es werden also Verfahren gebraucht, die auch auf kurzfristigen Daten eine Prognose ausführen können.

Die Prognose und die spätere Evaluierung werden mit Daten aus einem Mehrfamilienhaus im Raum Hamburg ausgeführt. Die Daten enthalten zwei Jahre elektrische Lastkurven, Außentemperaturen sowie Daten für die Leistung des BHKWs und Spitzenlastkessels (SLK). Abbildung 4.2 zeigt einen Ausschnitt aus dem Datenlogging-Programm des Objekts. Für dieses Kapitel sind vor allem der elektrische Verbrauch sowie die Speichertemperatur wichtig. Eine Aufschlüsselung der Energieentnahme aus dem Speicher in Warmwasser und Heizlast ist allerdings nicht möglich.

4.3 Prognose des elektrischen Lastverhaltens

Beim Aufstellen einer Prognose wird versucht, aus den bisherigen Werten einer Zeitreihe auf zukünftige Werte zu schließen. Es existieren eine Vielzahl an Prognoseverfahren, die jeweils eigene Einsatzgebiete haben und sich nur für bestimmte Daten eignen. Die einfachste Prognose ist die naive Prognose. Sie besagt, dass der zu erwartende Wert für einen Zeitpunkt f_{t+1} gleich dem Wert davor f_t wird und trifft damit die einfache Annahme „heute wird es so wie gestern“. Komplexere Verfahren müssen besser als die naive Prognose sein um ihren Einsatz zu rechtfertigen.



Abbildung 4.2: Beispieldaten für den 01.01.2014–07.01.2014

Statistische Verfahren analysieren Historiendaten und können daraus Trends und andere Muster ableiten. Ein Beispiel für ein solches Verfahren ist die exponentielle Glättung. Statistische Verfahren eignen sich vor allem dann, wenn eine Zeitreihe von Werten in die Zukunft weitergeführt werden soll, deren unterliegende Gesetzmäßigkeiten nicht genau bekannt sind oder nicht modelliert werden können. Das genaue Nutzerverhalten, zum Beispiel wann gekocht und wann geschlafen wird, ist unbekannt, erzeugt aber wiederkehrende Muster in den Lastkurven. Diese Muster gilt es mithilfe statistischer Methoden zu erkennen und vorherzusagen.

4.3.1 Grundlagen Statistischer Modelle

Eine Zeitreihe ist eine Folge y_t von Beobachtungen, die zu verschiedenen Zeitpunkten t gemessen wurden. Die Reihenfolge der Werte ergibt sich dabei durch die Zeit. Komponentenmodelle interpretieren eine Zeitreihe y_t als Überlagerung einfacher Funktionen. Vor der Erstellung des Modells wird eine geringe Anzahl isolierbarer Komponenten ausgewählt. Jede Funktion stellt eine Komponente dar, die neben der formalen Definition einen in der realen Welt auftretenden Faktor darstellen soll [28].

Die Komponenten dieses Komponentenmodells unterscheiden sich vor allem in der Periodizität.

Der *Trend* als Komponente ist eine glatte Kurve und ist Ausdruck von längerfristigen Einflussfaktoren, die nicht periodisch sind. Der Trend beim Bevölkerungswachstum der Welt ist beispielsweise steigend.

Das *Niveau* (englisch *Level*) ist die aktuelle durchschnittliche Höhe der Zeitreihe und kann sowohl zyklisch als auch azyklisch sein (beispielsweise die Konjunktur einer Volkswirtschaft).

Die *Saison* ist Ausdruck der saisonalen Schwankungen, wobei diese sich periodisch wiederholen. Beispiele für saisonale Einflussfaktoren sind die Jahreszeiten (Klima) oder institutioneller Einfluss (Ferien). Weiterhin wird die Zeitreihe beeinflusst durch

unvorhersehbare Faktoren (keine Komponenten im eigentlichen Sinn), zum Beispiel durch den Einzug neuer Mieter.

4.3.2 Einführung in die exponentielle Glättung

Die exponentielle Glättung als statistisches Mittel ist ein verbreitetes Verfahren zur Erstellung einer Kurvenglättung sowie zur Fortsetzung von Zeitreihen in die Zukunft. Mithilfe der exponentiellen Glättung können sowohl Ein-Schritt-Prognosen $y_{t+1|t}$ als auch k-Schritt-Prognosen $y_{t+k|t}$ erstellt werden.

4.3.2.1 Einfache exponentielle Glättung

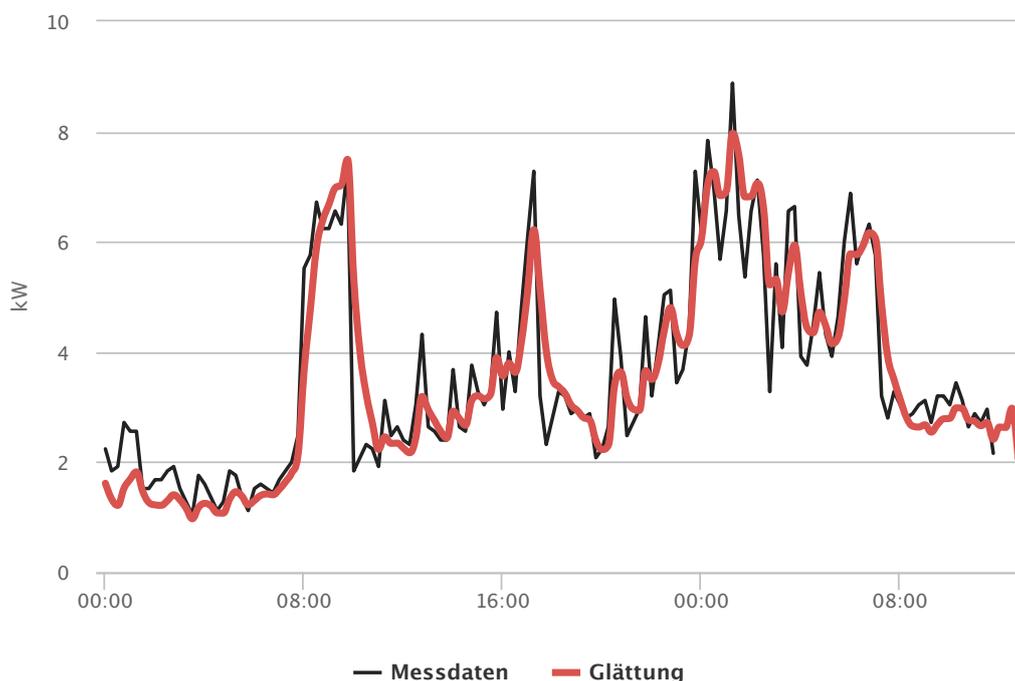


Abbildung 4.3: Glättung einer Kurve mittels Einschrittprognosen

Die einfachste Art der exponentiellen Glättung – die *einfache exponentielle Glättung* – ist anwendbar für Prognosen von Kurven, die keinen (ersichtlichen) Trend oder saisonale Komponente haben. Diese Prognose bietet eine Möglichkeit zur Gewichtung zwischen naiver Prognose und dem einfachen Mittelwert. y_t ist der tatsächliche Wert an Zeitpunkt t . $y_{t+1|t}^{\text{prog}}$ ist der Prognosewert für den Zeitpunkt $t+1$, der zum Zeitpunkt t berechnet wurde – die Ein-Schritt-Prognose. Der Glättungsparameter α bestimmt dabei die Gewichtung zwischen den beiden Einflussfaktoren (siehe Gleichung 4.1).

$$y_{t+1|t}^{\text{prog}} = \alpha y_t + (1 - \alpha) y_{t|t-1}^{\text{prog}} \quad \text{mit } (0 < \alpha < 1) \quad (4.1)$$

Abbildung 4.3 zeigt die Auswirkung der Glättung auf eine Kurve mit $\alpha = 0,5$. Die Kurvenglättung ergibt sich dadurch, dass der aktuelle Wert y_t der Messdaten nur zur Hälfte mit in die Schätzung des nächsten Punkts eingeht, die andere Hälfte besteht aus der letzten Schätzung (Zeitpunkt $t - 1$) für den aktuellen Punkt $y_{t|t-1}^{\text{prog}}$.

Durch simple Umformung lässt sich diese Gleichung auch in einer anderen Form, Fehlerkorrekturform genannt, schreiben.

$$y_{t+1|t}^{\text{prog}} = y_{t|t-1}^{\text{prog}} + \alpha(y_t - y_{t|t-1}^{\text{prog}}) = y_{t|t-1}^{\text{prog}} + \alpha F$$

Die Prognose für $t + 1$ wird dadurch gewonnen, dass der Prognosefehler F zur Korrektur von $y_{t|t-1}^{\text{prog}}$ (abhängig von α) benutzt wird. Bei einer Unterschätzung, also zu niedriger Prognose ($F > 0$ da $y_{t|t-1}^{\text{prog}} < y_t$) wird die Prognose nach oben korrigiert (dh. $y_{t+1|t}^{\text{prog}} > y_{t|t-1}^{\text{prog}}$). Bei einer Überschätzung wird die Prognose entsprechend nach unten korrigiert. Dass es sich tatsächlich um eine exponentielle Gleichung handelt, kann durch Einsetzung einer Gleichung in ihre Variable im nächsten Schrittes gesehen werden (eine Herleitung findet sich im Kaptielanhang). Durch Umformung und Summierung erhält man

$$y_{t+1|t}^{\text{prog}} = \sum_{j=0}^{t-1} \alpha(1-\alpha)^j y_{t-j} + (1-\alpha)^t \ell_0.$$

In dieser Form wird ersichtlich, dass die allererste Schätzung für das Niveau ℓ_0 bei hohem α einen wichtigen Einflussfaktor darstellt. Eine häufig genutzte Schätzung für ℓ_0 ist der Durchschnitt der ersten n Werte der Zeitreihe (n beliebig) [23].

4.3.2.2 Exponentielle Glättung mit Trend

Viele Zeitreihen besitzen allerdings einen Aufwärts- oder Abwärtstrend. Da dieser durch das einfache exponentielle Glätten nicht gut vorhergesagt werden kann, muss die Gleichung um eine Trendkomponente erweitert werden. Die neue Prognose, genannt Holt's lineare Methode (1957), setzt sich zusammen aus einem Niveau und einem Trend. Hier ist gleichzeitig auch die k -Schritt Vorhersage dargestellt. Diese ergibt sich durch k -malige Wiederholung der Trendkomponente b_t :

$$y_{t+k|t}^{\text{prog}} = \ell_t + kb_t.$$

Wobei ℓ_t , das Niveau, geschätzt wird durch

$$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$$

und b_t , der Trend, geschätzt wird durch

$$b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}.$$

Das Niveau ℓ_t ist ein gewichtetes Mittel zwischen den aktuellen Beobachtungen y_t und der Ein-Schritt Prognose für t durch $\ell_{t-1} + b_{t-1}$ (denn $y_{t|t-1}^{\text{prog}} = \ell_{t-1} + b_{t-1}$), genau wie bei der einfachen exponentielle Glättung.

Der Trend b_t ist wiederum ein gewichtetes Mittel zwischen dem geschätzten Trend zum Zeitpunkt t (durch $\ell_t - \ell_{t-1}$, der Steigung zwischen den Niveau-Schätzungen) und dem letzten geschätzten Trend b_{t-1} .

4.3.2.3 Holt-Winters Methode

Holt's lineare Methode funktioniert allerdings nicht gut, wenn die Zeitreihe saisonale Schwankungen aufweist. Um mit dem Modell auch Saisonalität abbilden zu können, wurde das Modell durch Winters (1960) erweitert. Die Methode enthält drei Glättungsgleichungen, jeweils eine für die Komponenten Niveau ℓ_t , Trend b_t , Saison s_t . Der Saisonlänge-Parameter m gibt die Anzahl der Messwerte innerhalb eines Zyklus an. So ist m bei mehrjährigen Daten die Anzahl der Saisons, zum Beispiel $m = 4$ für Quartalsdaten.

Die Holt-Winters Methode kann entweder in additiver oder multiplikativer Form berechnet werden. Bei der additiven Methode wird die saisonale Komponente als absoluter Wert in die observierten Werte eingerechnet, während die multiplikative Methode die saisonale Komponente relativ zu den Beobachtungen einrechnet.

Die multiplikative Methode lautet

$$y_{t+1|t}^{\text{prog}} = (\ell_t + kb_t)s_{t-m+1}$$

$$\text{mit } \ell_t = \alpha \frac{y_t}{s_{t-m}} + (1 - \alpha)(\ell_{t-1} + b_{t-1})$$

$$b_t = \beta(\ell_t - \ell_{t-1}) + (1 - \beta)b_{t-1}$$

$$s_t = \gamma \frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1 - \gamma)s_{t-m}.$$

Die Saisongleichung s_t ist ein gewichtetes Mittel aus der jetzigen Saisonkomponente $\frac{y_t}{(\ell_{t-1} + b_{t-1})}$ und der Vorjahres-Saisonkomponente s_{t-m} . Die multiplikative Methode wird dann angewendet, wenn die saisonalen Variationen mit steigendem Trend auch stärker schwanken.

Auf eine tiefergehende Erklärung und Diskussion dieser Methode wird an dieser Stelle verzichtet und auf [23] verwiesen.

Wir haben die multiplikative Methode eingesetzt, da sie bei Anwendung auf stündliche Lastkurven bessere Ergebnisse erzeugt hat als die additive Methode. Eine Erweiterung dieser Methode folgt später noch. Für die Parametrisierung der Holt-Winters Gleichung sind also vor allem die richtigen Gewichtungsfaktoren α , β und γ wichtig.

4.3.3 Prognose einer Lastkurve

Nun soll mithilfe der multiplikativen Holt-Winters Methode eine elektrische Lastkurve vorhergesagt werden. Für die Prognose müssen die Glättungsparameter für das Niveau α , den Trend β und die Saison γ richtig geschätzt werden. Wichtig ist hier wiederum die Beschaffenheit der Daten. Herkömmliche Schätzungen mittels exponentieller Glättung gehen von mehrjährigen Daten aus (weshalb auch von Saisons die Rede ist). Allerdings sind für den hier benannten Anwendungsfall kurzfristige Prognosen wichtiger. Sowohl die Eingabe als auch die Ausgabe ist auf weniger als ein Jahr beschränkt. Für das Beispielszenario wird eine Schätzung des zukünftigen

Stromverbrauchs anhand zweimonatiger Verbrauchsdaten erstellt. Daher muss der Saisonparameter m als Messwerte pro Tag interpretiert werden, also $m = 24$. Jede Stunde stellt somit eine Saison dar, ein Tag stellt einen Zyklus dar. Die Prognosedauer umfasst zwei Wochen.

Zunächst kann eine grobe Schätzung der zu erwartenden Parametern abgegeben werden. Der Niveau-Glättungsfaktor α sollte niedrig sein, da ein hoher α Wert nur die neuesten Messwerte berücksichtigt, die Prognose aber Werte aus dem gesamten Zeitraum berücksichtigen soll. Der Trend-Wert β sollte ebenfalls niedrig sein, da ein genereller Aufwärts- oder Abwärtstrend bei einer elektrischen Lastkurve nicht zu erwarten ist. Es ist zu beachten, dass je kleiner der Einflussparameter (α, β, γ) ist, desto mehr hängt die Ausgabe von den Initialisierungswerten (der ersten Schätzung für ℓ_t, b_t, s_t) ab.

Wir haben zuerst eine intuitive Schätzung mit $\alpha = 0,2, \beta = 0,3, \gamma = 1,0$ angesetzt, um die Auswirkung der Parameter zu verstehen und zu visualisieren. Holt-Winters wird anhand einer Woche an Eingabedaten getestet (entspricht sieben Zyklen). Die erste Hälfte der Woche wird als Trainingsdaten benutzt, aus denen die Vorhersage berechnet wird. Die zweite Hälfte wird genutzt, um die Vorhersage visuell zu validieren.

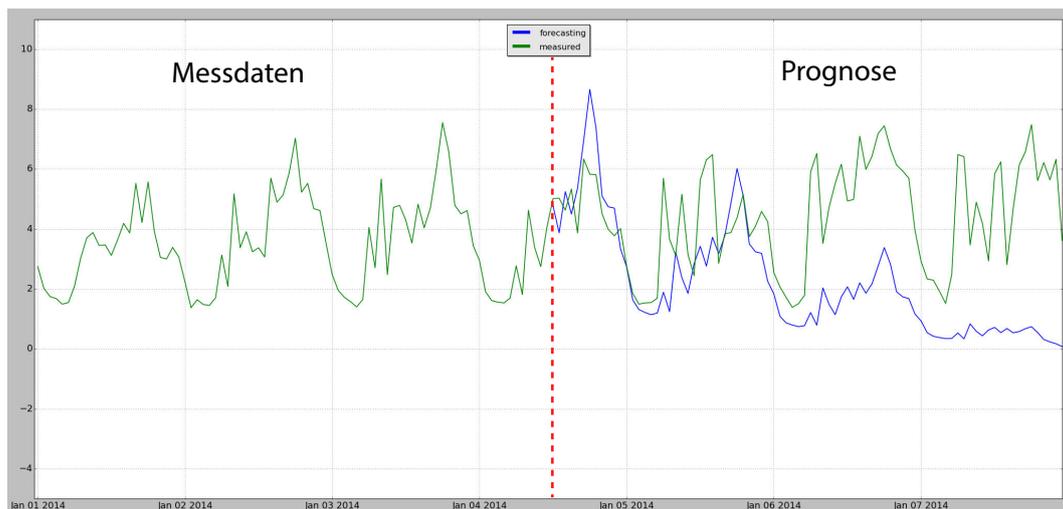


Abbildung 4.4: Trainingsdaten (links) und Vorhersage- und Testdaten (rechts)

In Abbildung 4.4 ist zu erkennen, dass die Vorhersage einen Trend nach unten berechnet hat, obwohl in den Testdaten optisch kein Trend zu erkennen ist. Der Vergleich zwischen Vorhersage und Testdaten zeigt somit, dass diese Parameter kein befriedigendes Resultat erzeugen. Um die manuelle Parameterfindung zu unterstützen, haben wir die Parameter mit Reglern justiert, um sofortiges Feedback über die Auswirkung zu bekommen.

Wie in Abbildung 4.5 zu sehen ist, hat die Vorhersage das Grundmuster (Wechsel zwischen Tag- und Nachtverbrauch/Verbrauchsspitzen am Nachmittag) richtig vor-

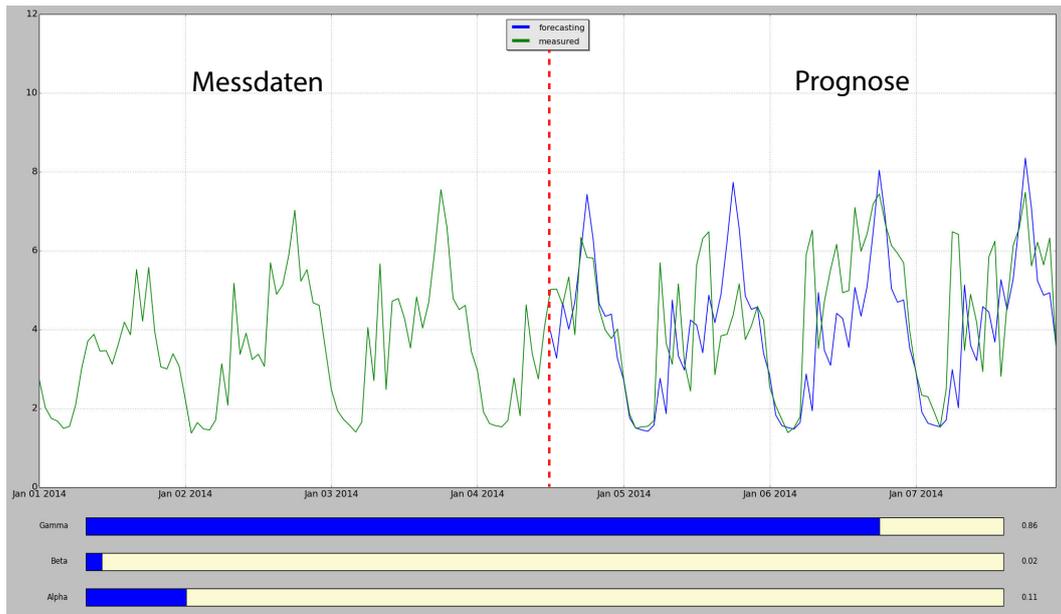


Abbildung 4.5: Screenshot der angepassten Kurve (erzeugt mit matplotlib)

hergesagt. Einzelne Spitzen konnten nicht richtig vorhergesagt werden, allerdings unterscheiden sich die Bedarfsspitzen in der ersten Wochenhälfte auch von den gemessenen Spitzen in der zweiten Hälfte. Bei den empirisch bestimmten $\alpha = 0,11$, $\beta = 0,0$, $\gamma = 0,86$ wird der fehlende Trend in Abbildung 4.5 durch die Wahl des Parameters β deutlich.

4.3.3.1 Fehlermessung mittels Quadratischer Abweichung

Neben der optischen Evaluation müssen für einen statistischen Algorithmus auch statistische Mittel zur Gütebestimmung eingesetzt werden. In Kapitel 1 wurden Fehler anhand der Mittleren Quadratische Abweichung berechnet. Für die Evaluierungen wird hier die Wurzel aus der Mittleren Quadratischen Abweichung, der RMSE (englisch *root mean squared error*) genommen, da dieser Fehlergrößen in der Dimension der Prognosewerte liefert und daher relativ intuitiv ist.

Für die erste Parameterschätzung ergibt sich ein RMSE von 1,76, während die zweite Parametermenge einen RMSE von 1,22 erzielt. Die zweite Prognose ist also auch quantitativ besser.

4.4 Automatisierte Prognosen

Das Holt-Winters Verfahren bietet die Möglichkeit, den Verlauf einer Lastkurve vorherzusagen, indem die Komponenten des Modells geschätzt und in die Zukunft weitergeführt werden. Bisher wurden Prognosen für den Verlauf einer einzelnen Lastkurve erstellt. Wie bereits erwähnt, müssen diese Einzelprognosen in die Simulation eingegeben werden, um das Verhalten der Geräte zu prognostizieren.

4.4.1 Automatische Prognose durch Optimierung

Da die Prognose für alle Typen und Ausprägungen von Lastkurven eingesetzt werden soll, müssen die Glättungsparameter α , β , γ für Holt-Winters automatisch gefunden werden können. Die Graphen können also nicht geplottet und manuell eingestellt werden, sondern es wird ein Algorithmus benötigt, der die besten Parameter findet.

Diese Problemstellung kann als ein Minimierungsproblem beschrieben werden. Eine Funktion f ordnet jeder Lösung $L = (\alpha, \beta, \gamma)$ durch $f : L \rightarrow \mathbb{R}$ einen Wert (hier den RMSE der Prognose) zu. Ziel ist es dann, $f(X)$ zu minimieren.

Um das Minimierungsproblem zu lösen, kann ein Solver für nichtlineare Gleichungssysteme verwendet werden. Da es zusätzliche Nebenbedingungen gibt (die Parameter müssen im Bereich 0 bis 1 bleiben), muss ein Algorithmus gewählt werden, der solche Beschränkungen beachtet. Wir haben uns an dieser Stelle für den L-BFGS-B (Limited Space Broyden–Fletcher–Goldfarb–Shanno (BFGS) with Boundaries) Algorithmus entschieden, da dieser gegenüber anderen Optimierungsalgorithmen mehrere Vorteile bietet [54]:

- Es muss keine Strukturinformation für die zu minimierende Funktion angegeben werden.
- Der Speicherverbrauch kann beschränkt werden.
- Die Kosten pro Iteration sind gering.

Der L-BFGS-B Algorithmus findet ein lokales Optimum anhand einer ersten Schätzung der Parameter. Die Iteration terminiert, wenn die Änderung pro Schritt in der Bewertungsfunktion klein genug ist.

$$\frac{(f_k - f_{k+1})}{\max(|f_{k+1}|, |f_k|, 1)} \leq \text{factr} \cdot \text{eps}mch$$

Wobei *eps*mch die Maschinenpräzision ist (automatisch generiert) und *factr* ein Faktor, der durch den Nutzer angegeben wird. Je kleiner der Faktor, desto genauer wird das Ergebnis und desto mehr Iterationen werden durchlaufen.

Die SciPy Bibliothek für Python bietet viele Algorithmen zur Optimierung von Funktionen. Sie bietet auch eine Implementierung der L-BFGS-B Methode an, die wir für die Minimierung des Fehlers benutzt haben.

Es gibt zwei Arten, den Fehler zwischen Prognose und Trainingsdaten zu Berechnen:

- Die Messdaten werden jeweils mit den Ein-Schritt Vorhersagen ($y_{t+1|t}^{\text{prog}}$) verglichen (In-Sample). Der Vorteil hierbei ist, dass auch die neusten Eingabedaten (bis auf den allerneusten Wert) in die Prognose einfließen. Die Prognose wird immer bereits nach einem Prognoseschritt mit den Realdaten verglichen, somit kann auch noch der vorletzte Wert mit einbezogen werden. Nachteil ist allerdings, dass die Ein-Schritt Vorhersage häufig keinen guten Aufschluss über die längerfristige Prognosequalität liefert.

- Die Eingabedaten werden in eine Trainings- und eine Testreihe unterteilt. Die Fehlermessung erfolgt zwischen der Vorhersage und den Testdaten (Out-of-Sample). Vorteil ist, dass hier eine längere Prognose auf die Testdaten angepasst werden kann.

Der zweite Ansatz erwies sich nach mehreren Tests als der Bessere und wurde ausgewählt. Die von uns implementierte `holt_winters()`-Funktion liefert also die k-Schritt Prognose (mit k in der Länge der Testdaten) zurück und der Fehler im Vergleich zum Testdatensatz wird berechnet.

Neben der Bewertungsfunktion erhält die L-BFGS-B Methode die Grenzen für α , β , γ sowie eine initiale Schätzung dieser Parameter. Der Algorithmus sucht ein lokales Optimum in der Umgebung dieser initialen Schätzung.

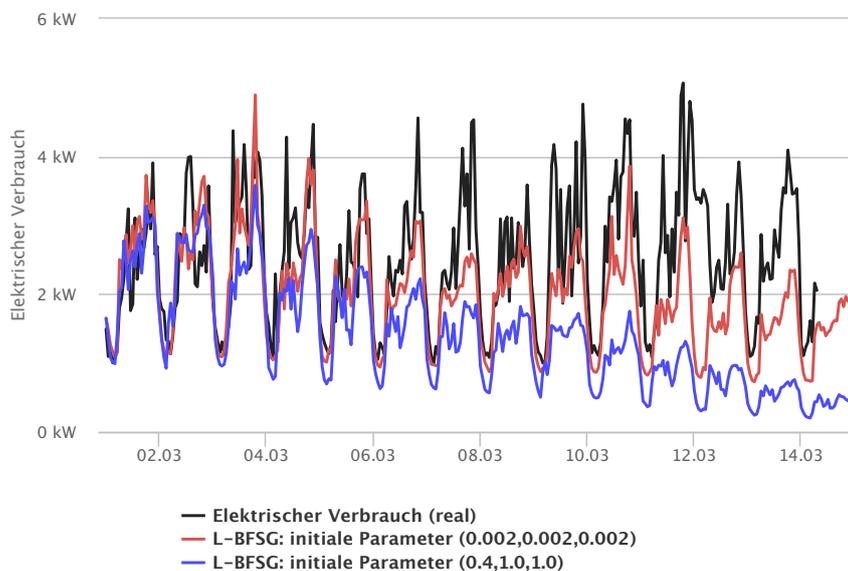


Abbildung 4.6: Vergleich zweier initialer Schätzungen

Abbildung 4.6 zeigt die Auswirkung zweier deutlich verschiedener Initialwerte des L-BFGS-B Algorithmus. Wir haben vier Wochen an Eingabedaten benutzt und mit zwei weiteren Wochen die Prognoseergebnisse validiert.

Tabelle 4.1: Statistische Auswertung von Abbildung 4.6

	Initiale Schätzung			Resultat Optimierung			Fehler
	α	β	γ	α	β	γ	RMSE
hohe Schätzung	0,400	1,000	1,000	0,001	1,000	0,053	1,527
niedrige Schätzung	0,002	0,002	0,002	0,036	0,002	0,226	0,927

In Tabelle 4.1 konnte gezeigt werden, dass der Algorithmus mehrere lokale Optima findet. Im Falle dieses Beispiels ist der RMSE der Testdaten durch eine niedrige Schätzung der Trendkomponente kleiner geworden. Dies lässt sich dadurch erklären, dass generelle Aufwärts- oder Abwärtstrends bei elektrischen Lastkurven nur schwach vorhanden sind, die Trendkomponente daher also die neuen Trendschätzungen nicht stark gewichten sollte.

4.4.2 Doppelte Saisonalität

Durch die algorithmische Optimierung können die Glättungsparameter nun automatisch angepasst werden. Allerdings ist, wie in Abbildung 4.6 ersichtlich, häufig ein zu starker Trend vorhanden und die Spitzenlasten werden an vielen Tagen falsch vorhergesagt.

Der Holt-Winters Algorithmus benötigt die Saisonlänge m , um periodische Daten berücksichtigen zu können. Lastkurven, die stündliche Werte erfassen, haben tägliche Zyklen, nachts wird deutlich weniger verbraucht als tagsüber. Also kann, wie in den vergangenen Beispielen, der Saisonalitätsparameter $m = 24$ gesetzt werden. Es gibt allerdings ein weiteres Muster in den Daten: Der Verbrauch an den Wochentagen ist anders als am Wochenende. Also kann zusätzlich ein wöchentlicher Zyklus mit $m = 24 * 7$ festgestellt werden. Diese doppelte Saisonalität ist im normalen Holt-Winters Verfahren nicht abgedeckt.

Es existiert eine Erweiterung des Holt-Winters Verfahrens, das auch doppelte Saisonalität berücksichtigt [46] und sich für die Vorhersage von elektrischen Lasten eignen soll. Hierbei wird eine weitere Saisonformel eingeführt, die die Saisonalität der Wochentage mit einberechnet. Allerdings ist diese Variante in vielen Fällen instabil und schwer kontrollierbar [27]. Es wurde eine andere Variante bevorzugt, die im Folgenden vorgestellt wird.

In Abbildung 4.7 wurde eine Aufteilung der Lastkurve je nach Wochentag vorgenommen und für jeden Wochentag eine Zeitreihe erstellt. Die Annahme ist, dass gleiche Wochentage ein ähnliches Profil haben. Jede dieser Zeitreihen wird nun mit Holt-Winters vorhergesagt und am Ende die Vorhersagen in der richtigen Reihenfolge zusammengesetzt. Die Berechnungen werden auf mehrere Prozesse verteilt und parallel ausgeführt, um den Zeitaufwand zu minimieren.

Die rote Kurve ist die Weiterführung der Prognose aus Abbildung 4.6 und hat über vier Wochen einen RMSE von 1.8. Mit der Tagesaufteilung konnte der Fehler auf $RMSE = 1.28$ minimiert werden. Die vollständige Evaluation der neuen Methode folgt in Abschnitt 4.6.

4.4.3 Dynamische Erneuerung der Prognose

Die Software `ecoControl` soll kontinuierlich laufen und auf neue Messdaten reagieren. Daher müssen auch die Prognosen bei Eingabe neuer Messdaten erneut berechnet werden. Dies bedeutet, dass dynamisch entschieden werden muss, wann eine Prognose erneuert werden soll.

Die Prognosequalität sinkt, je weiter in die Zukunft prognostiziert wird. Gründe dafür sind:

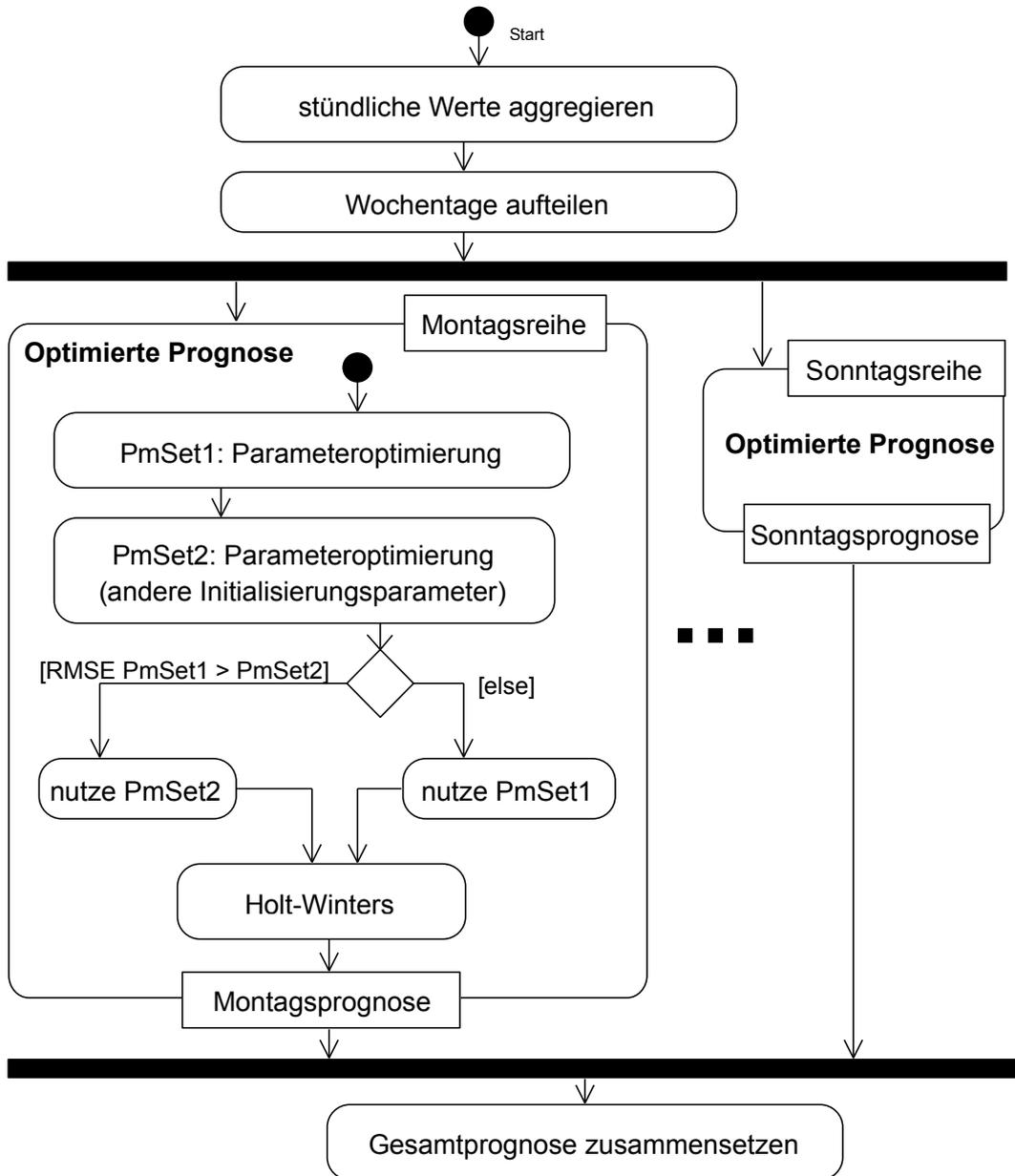


Abbildung 4.7: Datenverarbeitung für die Prognose

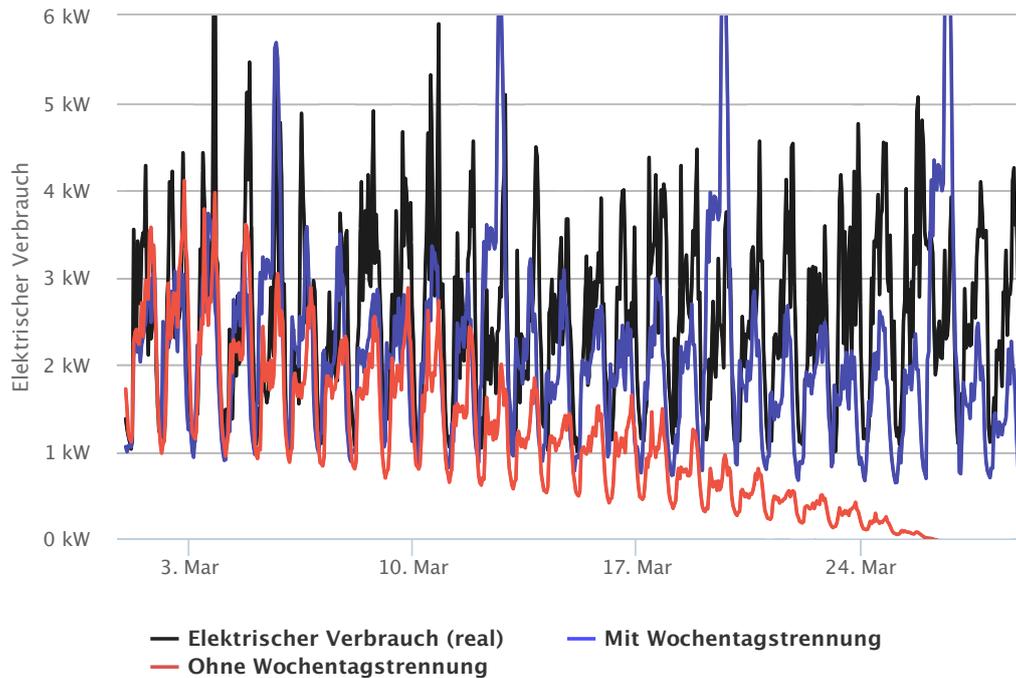


Abbildung 4.8: Verbesserung einer 4-Wöchentlichen Prognose durch Wochentagstrennung

- Kurzzeitige Anstiege werden als Trends erkannt und überbewertet.
- Änderung der Nutzergewohnheiten.
- Saisonale Unterschiede zwischen den Jahreszeiten, wie beispielsweise der Einfluss der Temperatur auf den Verbrauch werden nicht berücksichtigt.

Daher ist eine Begrenzung der Prognoselänge notwendig. Die Gesamtprognose (inklusive Simulation der Geräte) wird auf zwei Wochen begrenzt. Dies geschieht vor allem, weil Wettervorhersagen nur für maximal zwei Wochen Daten bereitstellen.

Um gleiche Bedingungen für jede Lastkurvenprognose zu schaffen, muss eine Prognose zu jedem Zeitpunkt die gleiche Menge an Daten als Eingabe bekommen. Hierzu wird eine Sliding Window Strategie benutzt.

Die neu gemessenen Daten werden zunächst in einen Puffer geschrieben und noch nicht in der Prognose berücksichtigt. Die bestehende Lastkurvenprognose gilt solange, wie weniger als ein Tag an Messwerten im Puffer liegt. Sind Daten von einem Tag oder mehr im Puffer, werden diese an die Zeitreihe angefügt. Am Anfang der Eingabedaten wird gleichzeitig ein Tag weggeschnitten. Das Zeitfenster von berücksichtigten Werten verschiebt sich also nach vorne (Abbildung 4.9). Somit bleibt stets die gleiche Menge an Eingabedaten vorhanden. Die angepasste Zeitreihe wird nun in die Holt-Winters Pipeline (Abbildung 4.7) eingespeist und die neue Vorhersage berechnet. Die Geräte greifen dann bei der Simulation auf diese Lastkurvenprognose zu, und entnehmen immer den Wert des aktuell simulierten Zeitpunkts.

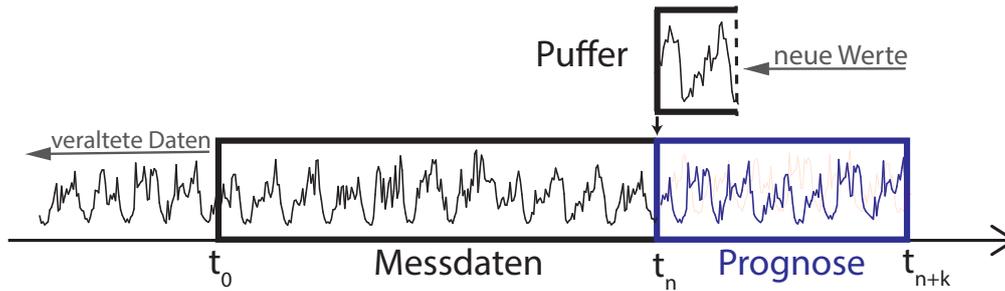


Abbildung 4.9: Nutzung einer Sliding Window Strategie zur Verarbeitung der einkommenden Daten

4.5 Kostenoptimierung mittels Prognosen

Mittels der erstellten Vorhersagen soll nun die Fahrweise der Geräte kosten-optimiert werden. Zwei Herangehensweise sind möglich:

- Das System wird mittels Regeln optimiert, dies erfordert allerdings die Kenntnis der langfristigen Auswirkungen von Änderungen auf das System und wird in Kapitel 5 behandelt.
- Das System wird algorithmisch verbessert, indem eine Kostenfunktion (Gesamtbilanz des Systems bei gegebenen Einstellungen) aufgestellt wird und diese minimiert wird.

Die zweite Variante wurde bereits beim Finden der Holt-Winters Parameter angewendet, nun soll ein Ansatz zur algorithmischen Optimierung für das Gesamtsystem entwickelt werden.

4.5.1 Formulierung der Kostenfunktion

Genau wie in Abschnitt 4.4 wird eine Bewertungsfunktion $f : L \rightarrow \mathbb{R}$, hier Kostenfunktion genannt, aufgestellt. Die Lösungsmenge L besteht aus Parametern, die das Systemverhalten beeinflussen. Obwohl es an dieser Stelle viele Parameter gibt, beschränken wir uns hier auf die Leistung des Blockheizkraftwerks, da diese in unserem Szenario den größten Einfluss auf die Kosten hat, also $X = \{\text{BHKW}_{\text{Auslastung}}\}$ für $f(X)$. Dieser Parameter wird in die Simulation gegeben und nach Durchlauf der Simulation werden verschiedene Verbrauchs- und Leistungsdaten für die Berechnung der Kosten verwendet. In den folgenden Formeln wird von der Verwendung der Simulation abstrahiert und nur deren Ausgabe benutzt.

Ausgabe der Kostenfunktion f ist die Bilanz des Systems zu einem gegebenen Zeitpunkt, in Euro. Die Bilanz wird aus der Summe S der Ausgaben abzüglich der

Summe der Einnahmen aufgestellt,

$$\begin{aligned} f(X) &= S_{\text{Ausgaben}} - S_{\text{Einnahmen}} \\ S_{\text{Ausgaben}} &= (\text{BHKW}_{\text{Gas}} + \text{SLK}_{\text{Gas}}) \cdot K_{\text{Gas}} + E_{\text{Bezug}} \cdot K_{\text{Strom}} \\ S_{\text{Einnahmen}} &= E_{\text{Einsp}} \cdot V_{\text{Einsp}} + E_{\text{Verb}} \cdot V_{\text{Verb}} + T_{\text{Verb}} \cdot V_{\text{Therm}}. \end{aligned}$$

Enthalten sind die Geräteverbräuche $\text{BHKW}_{\text{Gas}}, \text{SLK}_{\text{Gas}}$ (Blockheizkraftwerk und Spitzenlastkessel) und der thermische Verbrauch T_{Therm} jeweils in kWh. Der Strom wird unterteilt in Eigenverbrauch E_{Verb} , Einspeisung E_{Einsp} und Bezug E_{Bezug} . Die gemessenen Werte werden jeweils mit den entsprechenden Konstanten K (Kosten) bzw. V (Vergütung) pro kWh multipliziert. Nun können einer Lösungsmenge die Kosten in Euro zugeordnet werden.

Allerdings ist eine gute Lösung nicht nur durch möglichst geringe Kosten gekennzeichnet, sondern sie erfüllt noch andere, physikalische, Nebenbedingungen. Eine einfache Bedingung lautet, dass der Wärmespeicher (WS) nie über- oder unterhitzt sein darf. Die Minimierungsfunktion kann allerdings nur Kosten optimieren, andere Faktoren werden nicht beachtet. Deshalb müssen Nebenbedingungen als Kosten kodiert werden, die so hoch sind, dass eine minimierte Lösung niemals diese Nebenbedingungen verletzen wird.

Hierzu wird $f(X)$ erweitert zu

$$f(X) = S_{\text{Ausgaben}} - S_{\text{Einnahmen}} + S_{\text{Nebenbedingungen}}. \quad (4.2)$$

Für die Beschränkung der Speichertemperatur WS_T gilt

$$\begin{aligned} S_{\text{Nebenbedingungen}} &= C_{\text{unter}} + C_{\text{ueber}} \\ C_{\text{unter}} &= \begin{cases} 10^4 & \text{wenn } WS_T < WS_{\text{min}T} \\ 0 & \text{sonst} \end{cases} \\ C_{\text{ueber}} &= \begin{cases} 10^4 & \text{wenn } WS_T > WS_{\text{max}T} \\ 0 & \text{sonst} \end{cases}. \end{aligned}$$

Wobei der Wert 10^4 hier für eine beliebig große Konstante steht, die auf jeden Fall größer als die zu erwartende Bilanz ist. Fällt die Temperatur unter die Minimaltemperatur ($WS_{\text{min}T}$) des Wärmekessels, wird C_{unter} diesen hohen Wert annehmen und somit die getesteten Lösungsparameter als ungeeignet markieren.

Ein Beispiel: Das BHKW läuft mit maximaler Leistung also $X = \{100\}$. Die Gaskosten für einen Tag betragen ca. 30 €, 5 € Bezugskosten aus dem Stromnetz kommen hinzu. Auf der Einnahmeseite konnten 5 € durch Einspeisung, 25 € durch Stromverkauf an die Mieter und 15 € durch die Wärmebereitstellung verbucht werden. Die Bilanz beträgt -15 € (die Einnahmen überwiegen). Allerdings wurde die Maximaltemperatur des Pufferspeichers (90 °C) überschritten und somit ist $C_{\text{ueber}} = 10000$. Das Ergebnis von $f(X) = 9985$ kann nun nicht mehr als minimal bezeichnet werden und somit ist die einhundertprozentige Auslastung des BHKWs für diesen Zeitraum ungeeignet.

Mit dieser Kostenfunktion sind die wichtigsten Kosten und Bedingungen modelliert, andere Faktoren wie zum Beispiel Wartungskosten können durch Erweiterung der Funktion berücksichtigt werden.

4.5.2 Parametersuche

Algorithmus 1 : Optimierung der Simulation im laufenden System

input : Simulation Sim_t zum Zeitpunkt t_0 , Prognoselänge in Stunden h
output : Optimierter Simulationsdurchlauf Sim_{t+k}

```

1 while Simulation nicht beendet do
2   if Optimierung ausgelöst then
3     Wettervorhersagen cachen;
4     Stromverbrauch vorhersagen und cachen;
5      $BHKW_{Auslastung} \leftarrow 0$ ;
6     while  $Kosten_{Gesamt}$  nicht minimal do
7        $Vorschau_t \leftarrow Sim_t$  kopieren;
8        $BHKW_{Auslastung}$  ändern und in  $Vorschau_t$  setzen;
9        $Vorschau_{t+h} \leftarrow Vorschau_t$  um  $h$  Stunden vorspulen;
10       $Kosten_{Gesamt} \leftarrow$  Bilanz  $Vorschau_{t+h}$  berechnen;
11    end
12     $BHKW_{Auslastung}$  in  $Sim_t$  setzen;
13  end
14  Step-Funktion der Geräte in  $Sim_t$  ausführen;
15 end
```

Algorithmus 1 zeigt den Ablauf der Optimierung am Beispiel der BHKW Auslastung (es können allerdings auch andere Parameter des Systems optimiert werden). Durch Ausprobieren mehrerer Parameter in Zeile 8 kann eine Lösung gefunden werden, die die niedrigsten Kosten verursacht bzw. den größten Ertrag erwirtschaftet. Das „Ausprobieren“ kann realisiert werden durch:

- Anwenden eines Brute-Force Ansatzes, indem über alle möglichen Werte (mit einer bestimmten Schrittgröße) iteriert wird.
- Verwenden eines Minimierungsalgorithmus, wie beispielsweise L-BFGS-B, der bereits in Kap. 4.4 benutzt wurde.

Wir verwenden einen zweistufigen Ansatz, bei dem zuerst eine kleine Anzahl Parameter aus dem gesamten Parameterbereich ($BHKW_{Auslastung} \in [0,100]$) getestet wird. Die günstigste Variante wird genommen und in der Umgebung dieses Parameters wird mittels des L-BFGS-B Algorithmus ein Minimum gefunden.

4.5.3 Beispiel einer automatischen Systemoptimierung

Es folgt ein Beispiel für die Verwendung der Kostenminimierung, das konkrete Einsparungspotential wird in Kapitel 5 erläutert. Der Vergleich bezieht sich auf die Simulation einer Woche im April, einmal ohne und einmal mit automatischer Optimierung. Die optimale BHKW-Auslastung wurde immer für eine Stunde voraus berechnet (Prognoselänge $h = 1$) und nach einer Stunde vergangener Zeit eine neue Optimierung angestoßen.

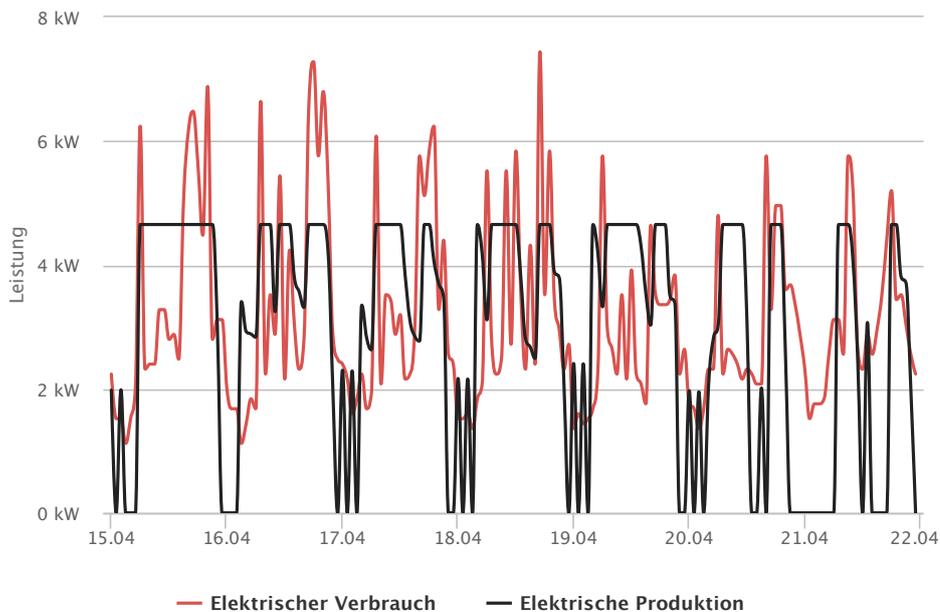


Abbildung 4.10: Stromverbrauch und -produktion ohne Optimierung

Der Vergleich zwischen Abbildung 4.10 und Abbildung 4.11 zeigt, dass sich die elektrische Produktion im optimierten Fall besser an den Verbrauch anpasst. Hierdurch wird die Überproduktion stark vermindert und die Abdeckung des elektrischen Grundbedarfs vergrößert. Die Verlängerung der Bedarfsabdeckung kann gut in den Nächten vom 18.–20.04 gesehen werden, zusätzlich schaltet sich das BHKW in der optimierten Fahrweise hier auch weniger ein und aus, was sich positiv auf die Gerätelebensdauer auswirkt. Insgesamt konnte die Abdeckung der elektrischen Last durch die optimierte Aussteuerung um etwa 20 % gesteigert werden.

4.5.4 Verbesserungsmöglichkeiten

Der vorgestellte Ansatz liefert bereits vielversprechende Ergebnisse, hat allerdings noch Verbesserungspotential. Problematisch ist der Kompromiss zwischen kurzen und längeren Optimierungsschritten:

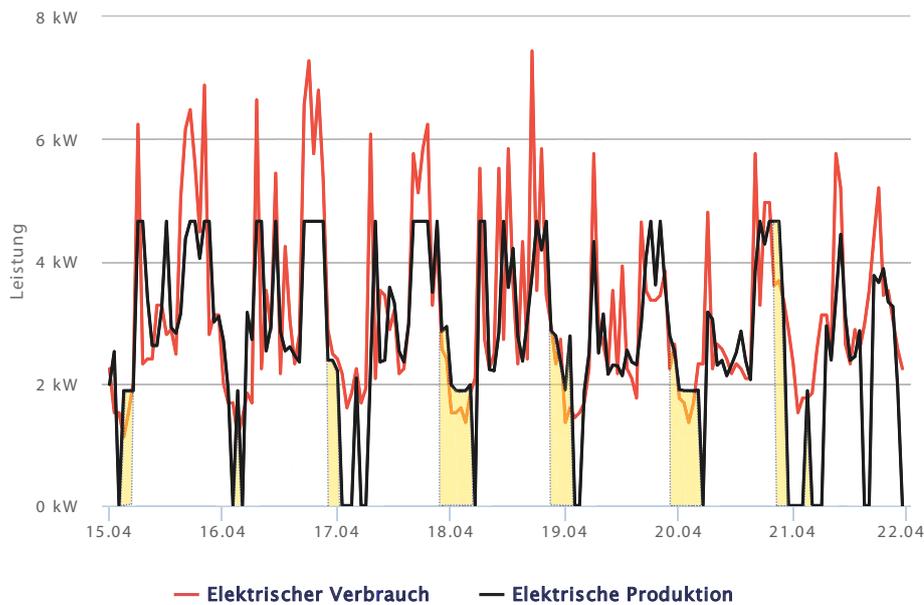


Abbildung 4.11: Stromproduktion mit Maximierung des Eigenverbrauchs (in gelb)

Wird ein langer Optimierungsschritt (Prognoselänge h) ausgeführt, werden längerfristige Prognosen des Wetters und des Strombedarfs mit einbezogen. Allerdings kann immer nur ein konstanter Wert für die gesamte Periode ermittelt werden, was die Anpassung an den realen Strombedarf und die Pufferspeicherkapazität schwierig macht. Die Stromspitzenlasten treten beispielsweise häufig nur für ungefähr eine Stunde auf.

Kurze Optimierungsperioden können die Produktion deutlich besser an den Bedarf anpassen, dafür aber längerfristige Prognosen (wie zum Beispiel das Wetter in vier Tagen) nicht berücksichtigen.

Bei einer Verbesserung könnte die Methode so erweitert werden, dass die Optimierungsfunktion statt einem einzelnen Wert eine Wertereihe für den Optimierungszeitraum errechnet. Zum Beispiel würde die Funktion eine Kurve der BHKW-Auslastung, statt einer festen Auslastung, zurückgeben. Somit kann langfristig geplant und trotzdem auf kurze Bedarfsschwankungen reagiert werden.

4.6 Evaluation

Im Folgenden evaluieren wir die Qualität der Vorhersagen. In Abschnitt 4.6.2 werden wir die Vorhersagen für ein ganzes Jahr jeden Tag einmal ausführen und das Ergebnis evaluieren. In Abschnitt 4.6.3 wird der Effekt der Wettervorhersage auf das Gesamtsystem evaluiert.

Für einen Vergleich der Gerätesimulation mit Echtdaten verweisen wir auf Kapitel 1.

4.6.1 Fehlermessung mittels Skalierter Fehler

Die Messung der Abweichung mittels RMSE ist für den Vergleich verschiedener Methoden an der gleichen Zeitreihe ausreichend. Will man die Prognosemethoden auf verschiedene Zeitreihen anwenden und miteinander vergleichen, sind Prognosefehler wie der RMSE allerdings nicht aussagekräftig, da sie von der Skalierung der Zeitreihe abhängen. Um die Evaluationsergebnisse vergleichbar mit anderen Methoden und anderen Zeitreihen – beispielsweise dem elektrischen Bedarf eines Einfamilienhauses – zu machen, muss ein *skalierungs-unabhängiger* Fehler verwendet werden. Der „Mean Absolute Scaled Error“ (MASE) [1] teilt die Vorhersagefehler durch die Fehler der naiven Prognose auf dem gleichen Datensatz.

$$q_t = \frac{y_t - y_{t|t-1}^{\text{prog}}}{\frac{1}{N-1} \sum_{n=2}^N |y_n - y_{n-1}|}$$
$$\text{MASE} = \text{avg}(|q_t|)$$

Da sowohl Zähler als auch Nenner in der gleichen Dimension sind, ist der resultierende Fehler unabhängig von der Skalierung der Daten.

4.6.2 Evaluation der Lastkurvenprognose

Das Prognosesystem wurde mit den Daten des Jahres 2013 evaluiert. Die konstante Eingabe neuer Daten, die zur Erneuerung der Prognosen gebraucht werden, wurden aus den Messdaten des in Abschnitt 4.2.3.3 erwähnten 12-Familienhauses entnommen. Für jeden Tag in diesem Zeitraum haben wir eine neue Holt-Winters Prognose ausgeführt, die Fehler gemessen und am Ende den Durchschnittsfehler gebildet.

4.6.2.1 Evaluation der Fehler nach Uhrzeit

Abbildung 4.12 zeigt, dass der Fehler morgens um 6, mittags zwischen 13 und 15 Uhr und abends zwischen 20 und 23 Uhr am höchsten ist. Hier ist auch der RMSE aufgeführt, um zu zeigen, dass die Vorhersage um 13 und 14 Uhr die größte Abweichung von der tatsächlichen Last aufweist. Die Spitzen bei den Fehlern können zwei Ursachen haben:

- Die Fehler treten durch höchst variable Spitzenlasten auf und sind somit nicht vorhersagbar.
- Das Prognosemodell unter- oder überschätzt die Lasten an diesem Punkt konsequent. An diesem Punkt ließe sich das Modell dann noch verbessern.

Um die Fehlerursachen weiter zu erforschen, wird in Abbildung 4.13 der durchschnittliche Strombedarf eines Haushalts betrachtet.

Abbildung 4.13 zeigt die Wahrscheinlichkeit des Auftretens eines bestimmten elektrischen Leistungsbedarfs eines durchschnittlichen Einfamilienhauses. Der maximale Strombedarf zeigt um 14 und 15 Uhr starke Spitzen, obwohl der durchschnittliche Strombedarf niedrig bleibt. Somit gibt es an dieser Stelle starke, sporadische Ausreißer, die nicht korrekt vorhergesagt werden können. Der erhöhte Fehler um 6 und

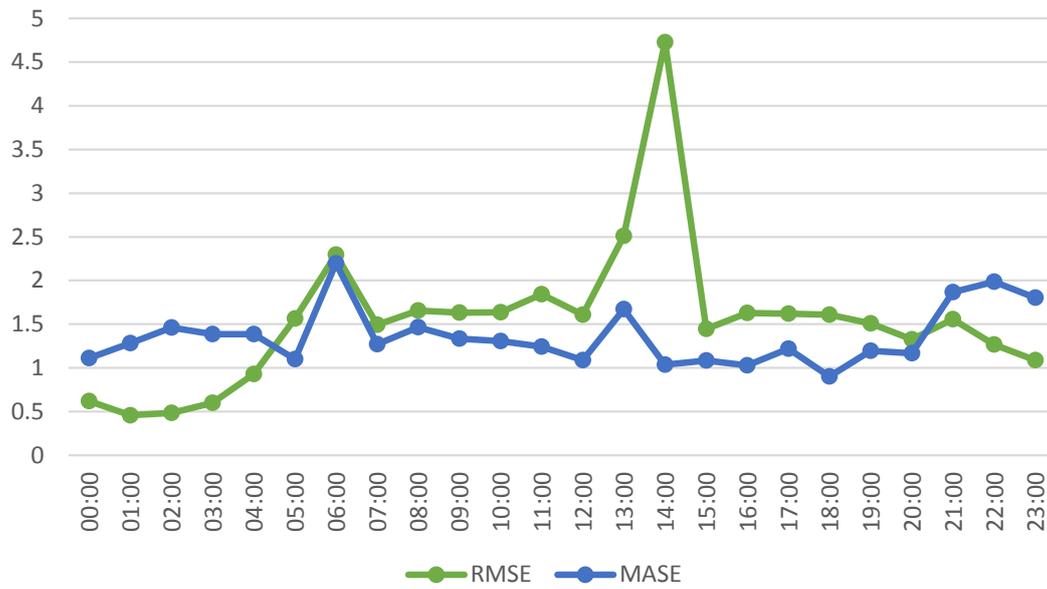


Abbildung 4.12: Fehlerdurchschnitt nach Stunde für das Jahr 2013

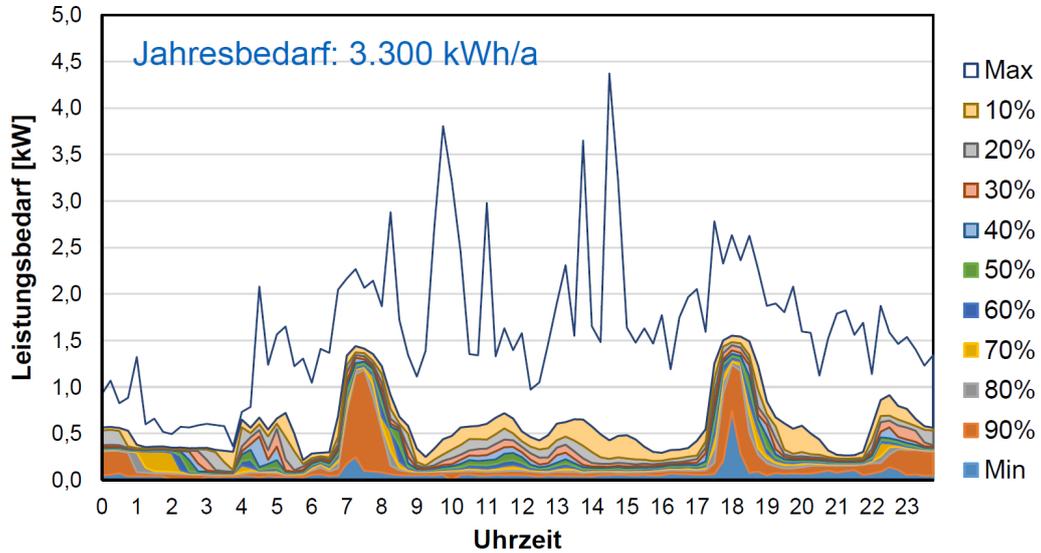


Abbildung 4.13: Wahrscheinlichkeitsintervalle für den Stromverbrauch eines Haushalts (Quelle: Tzeutschler, TU München [47])

7 Uhr in Abbildung 4.12 korreliert dagegen mit der Spitzenlasterhöhung in Abbildung 4.13 und zeigt damit eine Stelle, an welcher der Algorithmus noch verbessert werden muss.

4.6.2.2 Vergleich mit einem Einfamilienhaus

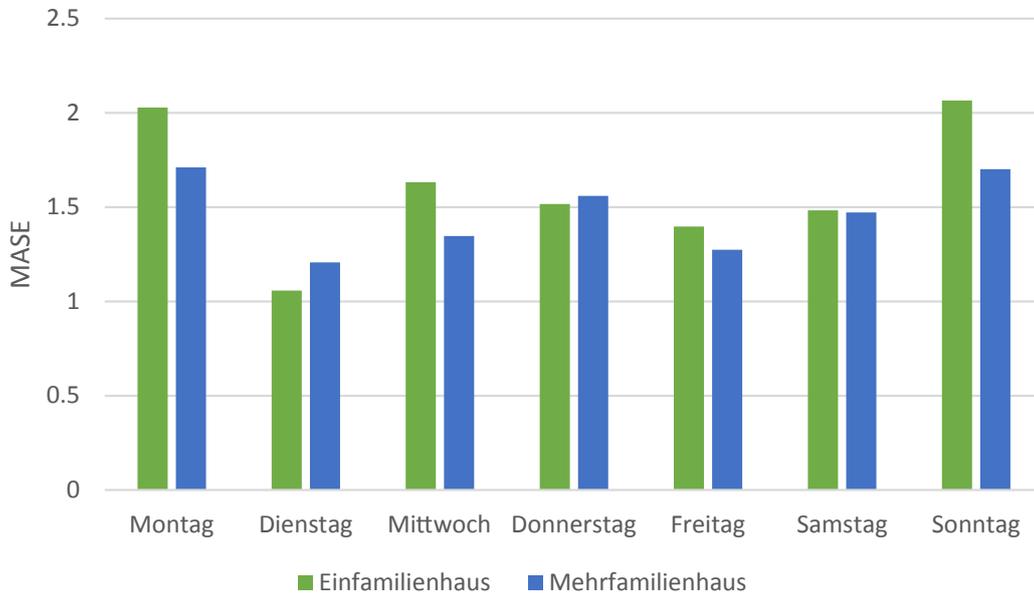


Abbildung 4.14: Vergleich der Originaldaten mit einem Einfamilienhaus im gleichen Zeitraum

Die Originaldaten wurden mit den Daten eines Einfamilienhauses bei Nürnberg verglichen. Der Vergleich in Abbildung 4.14 zeigt einerseits, dass der Fehler zwischen zwei unterschiedlichen Zeitreihen nicht stark divergiert. Andererseits ist auch zu sehen, dass verschiedene Wochentage unterschiedlich gut vorhergesagt werden. Die Fehlerwahrscheinlichkeiten für die Wochentage in beiden Zeitreihen scheinen sogar miteinander zu korrelieren.

Weiterhin ist zu sehen, dass die Abweichung vom Mittel aller Fehler beim Mehrfamilienhaus (Mittel: 1.46, Standardabweichung: 0.174) kleiner ist als beim Einfamilienhaus (Mittel: 1.6, Standardabweichung: 0.3). Wie statistisch zu erwarten, steigt also die Vorhersagbarkeit mit der Anzahl der Einwohner.

4.6.2.3 Vergleich der Prognosemethoden

Abbildung 4.15 vergleicht die Holt-Winters Methode mit Tagestrennung mit der normalen, multiplikativen Methode und der naiven Prognose.

Die naive Prognose wird in diesem Fall erzeugt, indem der gleiche Tag aus dem Jahr zuvor (2012) genommen wird. Der Fehler der Holt-Winters Methoden liegt

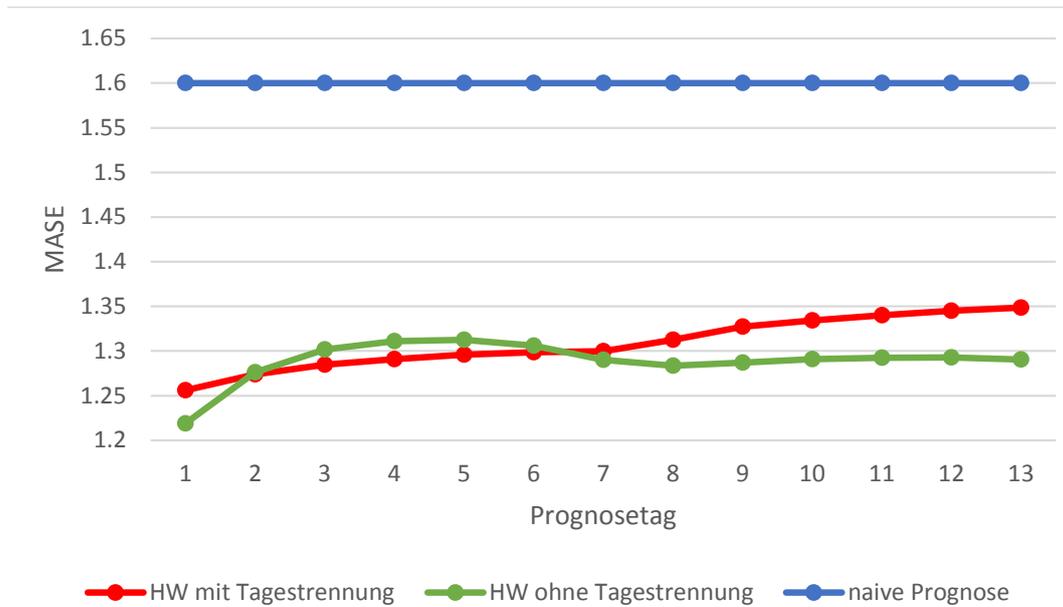


Abbildung 4.15: Fehler der Prognosen (Holt-Winters und naiv) am n-ten Tag der Prognose

immer unter dem Fehler der naiven Prognose und rechtfertigt somit den Einsatz von Holt-Winters.

Die Tagestrennung liefert für die erste Woche bessere Ergebnisse als die Methode ohne Tagestrennung, wird allerdings in der zweiten Woche von der normalen, multiplikativen Methode übertroffen. Der konstante Anstieg der Fehlerrate könnte darauf zurückzuführen sein, dass für die einzelnen Tage ein falscher Trend angenommen wird und somit die Fehlerrate mit der Zeit steigt. Die Methode ohne Tagestrennung kann dagegen beim Optimieren des RMSE die Prognoseergebnisse mit der siebenfachen Anzahl an Tagen am Stück vergleichen, womit die Trendschätzung (falls kein Trend vorhanden) auch deutlich konservativer ausfallen sollte.

Die Auftrennung nach Wochentagen liefert also für die erste Woche gute Ergebnisse und sollte hier der normalen multiplikativen Methode vorgezogen werden. Um Prognosen für zwei Wochen oder mehr zu erhalten, sollte auf Holt-Winters ohne Tagestrennung zurückgegriffen werden.

Ob eine zweiwöchige Prognose sinnvoll ist, hängt nicht nur von der Nutzerprognose, sondern auch von der Fehlerrate der Wettervorhersage ab.

4.6.3 Evaluation der Wettervorhersage

Die Außentemperatur spielt eine zentrale Rolle im Heizungssystem. Alle anderen Komponenten sind von ihr abhängig. Die Prognose des thermalen Bedarfs kann deshalb nur so genau wie die Wettervorhersage sein. Wettervorhersagen können maximal zehn Tage an stündlichen Daten liefern. Für Abbildung 4.16 wurden

720 konsekutive Vorhersagen aus einer Wetterstation in Berlin gesammelt (30 Tage, 24 Abfragen pro Tag) und der durchschnittliche RMSE gebildet.

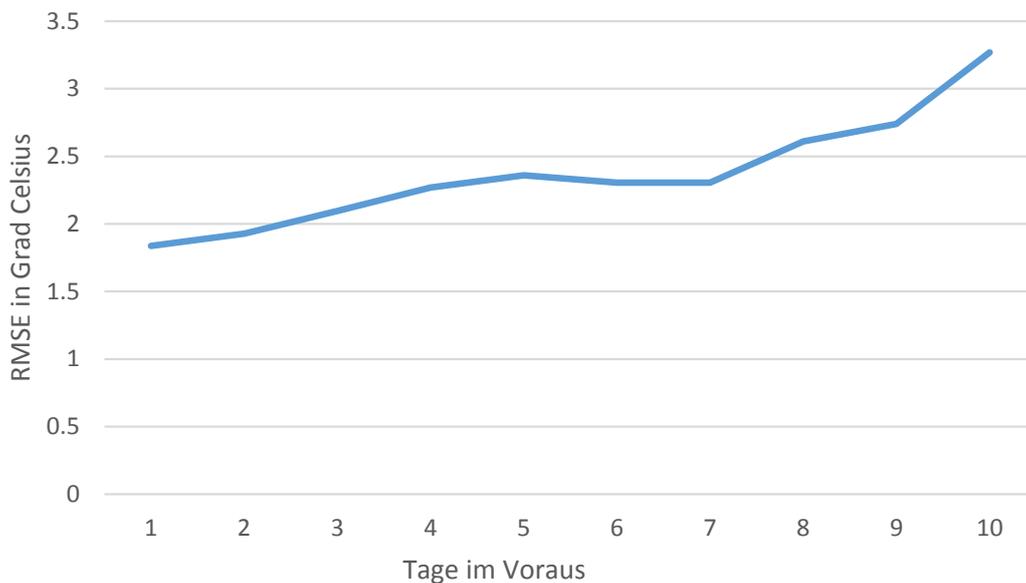


Abbildung 4.16: Fehlerdurchschnitt für Vorhersagen vom 15.05.–15.06.2014

Abbildung 4.16 zeigt, dass die Prognose im Durchschnitt weniger als 2,5 °C von der realen Messung abweicht. Laut dem Deutschen Wetterdienst kann das als gute Prognose eingeschätzt werden [53]. Allerdings steigt die Fehlerrate ab dem 7. Tag stark an.

Abbildung 4.17 zeigt beispielhaft eine Wettervorhersage und dessen Effekt auf den Pufferspeicher. Um zweiwöchentliche Prognosen verwenden zu können (die in der Zielsetzung erwähnte Prognoselänge), wurden die letzten vier Tage durch die naive Klimaprognose (den Werten des letzten Tages) aufgefüllt.

Wie in Abbildung 4.17 zu sehen ist, wirkt sich der Fehler in der zweiten Woche der Prognose bereits deutlich auf die Temperatur des Wärmespeichers aus. Bei den gemessenen Daten ist es auf Grund warmer Temperaturen nicht zu einem Anschalten der Heizung gekommen. Die Prognose schätzt die Außentemperatur dagegen niedriger ein, wodurch die Innentemperatur im Gebäude absinkt und die Heizkörper durch Bezug aus dem Wärmespeicher mehr Leistung abgeben. Aufgrund dieses Fehlers ist es nur bedingt sinnvoll, die Fahrweise des Blockheizkraftwerkes anhand der Wettervorhersage auf zwei Wochen genau einzustellen. Mehr als sieben Tage an Prognosen zu nutzen ist (nach der Aussage eines BHKW-Planers) allerdings aufgrund der Begrenzung der Wärmespeicher auch in den meisten Fällen nicht nötig.

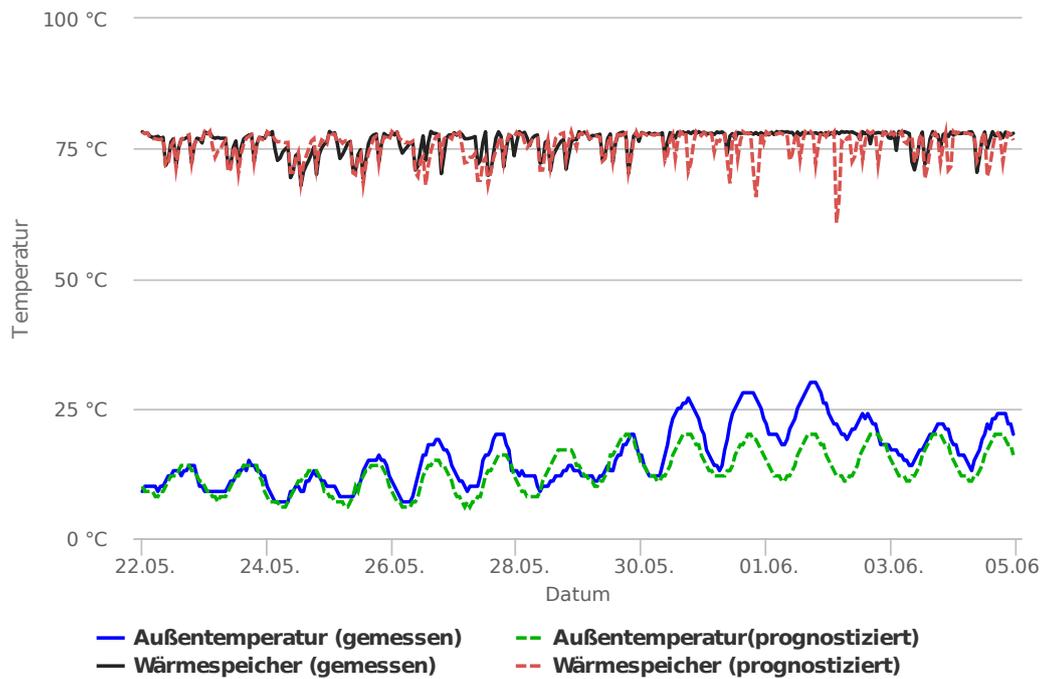


Abbildung 4.17: Effekt des Temperaturfehlers auf den Pufferspeicher

4.7 Zusammenfassung

In diesem Kapitel haben wir eine Anwendung der Holt-Winters Methode zur dynamischen Vorhersage von Stromverbräuchen vorgestellt. Die Vorhersage wird mittels einer Optimierungsroutine an die Eingabedaten angepasst und in die Gerätesimulation eingebunden.

Die vorgestellte Methode schneidet im Vergleich deutlich besser ab als eine naive Prognose, die Werte aus dem Vorjahr nutzt. Die Vorhersage kann genutzt werden, um die Grund- und Spitzenlasten im Haushalt vergleichsweise genau vorherzusagen. Die Prognose fällt dabei desto besser aus, je mehr Bewohner im Haus leben, wie wir in Abschnitt 4.6.2.2 gezeigt haben.

Wir haben in Abschnitt 4.4.2 mit der Aufteilung der Holt-Winters Prognosen auf die verschiedenen Wochentage eine Anpassung der Methode an doppelte Saisonalitäten vorgestellt. Diese Aufteilung erweist sich für eine Kurzzeitprognose bis zu einer Woche als sinnvoll, schneidet allerdings ab der zweiten Woche weniger gut als die originale Holt-Winters Methode ab. In einer weitergehenden Arbeit muss die Tagestrennung mit der Double-Seasonal Holt-Winters [46] Methode verglichen und evaluiert werden, um die beste Variante für eine kurzfristige, dynamische Vorhersage von elektrischen Lasten zu ermitteln.

Wir haben den Effekt der Wettervorhersage auf die Simulation des Wärmespeichers evaluiert und festgestellt, dass die Temperatur im Speicher zwischen Realität und Prognose für die erste Woche gut miteinander übereinstimmen. In der zweiten

Woche ist die Abweichung von der tatsächlichen Temperatur allerdings schon so groß, dass eine längerfristige Prognose nicht sinnvoll ist.

Des Weiteren haben wir einen Algorithmus vorgestellt, der mithilfe der Nutzerprognosen und Gerätesimulation das System optimiert, um so den maximalen Ertrag zu erwirtschaften. Der Algorithmus ist einfach durch weitere Kostenfaktoren zu erweitern und liefert vielversprechende erste Resultate. Eine Evaluation erfolgt in Kapitel 5. Diese Aussteuerung könnte potentiell direkt nach Installation auf den Geräten angewendet werden und erfordert keine individuelle Feinjustierung im Gegensatz zu einem regelbasierten Vorgehen.

4.7.1 Ausblick

Die bisher vorgestellten Konzepte können in zukünftigen Arbeiten an verschiedenen Stellen noch erweitert werden:

- Erweiterung des Kostenoptimierungsalgorithmus, wie in Abschnitt 4.5.4 beschrieben,
- Reduktion des Fehlers bei Spitzenlasten, wie Abschnitt 4.6.2.1 nahelegt.

Die bisher vorgestellten Prognosen sind nur auf einen kurzen Zeitraum begrenzt: Wettervorhersagen können, wie in Abschnitt 4.6.3 beschrieben, nur für zehn Tage abgerufen werden. Auch die Fehlerquote bei Holt-Winters steigt mit der Zeit an (Abbildung 4.15). Für den Nutzer mit Verwaltungsaufgaben, der in Kapitel 2 beschrieben wurde, wären allerdings auch langfristige Schätzungen sehr wichtig. Vor allem die geschätzte Jahresbilanz oder das Amortisationsdatum des BHKWs benötigen eine solche langfristige Schätzung. Die in diesem Abschnitt vorgestellten Kurzzeitprognosetechniken eignen sich allerdings kaum für solche Langzeitprognosen. Bei einer langfristigen Bilanz müssen grobe Abschätzungen über den Strom und Wärmebedarf aufgestellt werden. Mittels einer Heuristik muss dann die benötigte Anzahl an Betriebsstunden des BHKWs abgeschätzt werden. Dies entspricht dem Vorgehen von Planern, die vor dem Kauf eines BHKWs solche Schätzungen anstellen, wie wir in Kapitel 2 herausgearbeitet haben. Der entscheidende Unterschied ist allerdings, dass beim Einsatz von `ecoControl` zusätzlich noch der Effekt diverser Optimierungen in die Schätzung eingehen muss. Die Entwicklung von solchen Heuristiken kann Bestandteil zukünftiger Arbeiten sein.

Anhang

Herleitung der exponentiellen Form der einfachen Exponentiellen Glättung

$$y_{t+1|t}^{\text{prog}} = \alpha y_t + (1 - \alpha) y_{t|t-1}^{\text{prog}} \text{ mit } (0 < \alpha < 1)$$

Die erste Vorhersage wird auf ℓ_0 gesetzt, und die folgenden Vorhersagen ausgerechnet.

$$\begin{aligned} y_{2|1}^{\text{prog}} &= \alpha y_1 + (1 - \alpha) \ell_0 \\ y_{3|2}^{\text{prog}} &= \alpha y_2 + (1 - \alpha) y_{2|1}^{\text{prog}} \\ y_{4|3}^{\text{prog}} &= \alpha y_3 + (1 - \alpha) y_{3|2}^{\text{prog}} \\ &\vdots \\ y_{T+1|T}^{\text{prog}} &= \alpha y_T + (1 - \alpha) y_{T|T-1}^{\text{prog}} \end{aligned}$$

Durch Ersetzen jeder Gleichung in die nächste Gleichung folgt

$$\begin{aligned} y_{3|2}^{\text{prog}} &= \alpha y_2 + (1 - \alpha) [\alpha y_1 + (1 - \alpha) \ell_0] \\ &= \alpha y_2 + \alpha(1 - \alpha) y_1 + (1 - \alpha)^2 \ell_0 \\ y_{4|3}^{\text{prog}} &= \alpha y_3 + (1 - \alpha) [\alpha y_2 + \alpha(1 - \alpha) y_1 + (1 - \alpha)^2 \ell_0] \\ &= \alpha y_3 + \alpha(1 - \alpha) y_2 + \alpha(1 - \alpha)^2 y_1 + (1 - \alpha)^3 \ell_0 \\ &\vdots \\ y_{T+1|T}^{\text{prog}} &= \sum_{j=0}^{T-1} \alpha(1 - \alpha)^j y_{T-j} + (1 - \alpha)^T \ell_0. \end{aligned}$$

Herleitung aus [23].

5 Evaluation und Diskussion von Optimierungsalgorithmen für Energiesysteme in Mehrfamilienhäusern

Eine dezentrale Energieversorgung ist ein wichtiger Schritt in Richtung Energiewende. Um die Bewohner in einem Mehrfamilienhaus rund um die Uhr mit Strom und Wärme aus Eigenproduktion versorgen zu können, müssen heterogene Energiesysteme zum Einsatz kommen. Bisher ist es schwierig, diese effizient miteinander zu betreiben.

In diesem Kapitel werden Optimierungsmöglichkeiten von Energiesystemen anhand eines Fallbeispiels vorgestellt. Dabei haben wir herausgefunden, dass das Primärziel solcher Optimierungen die maximale Deckung des Energiebedarfs durch Eigenproduktion ist. Wir haben erklärt, wie man beispielsweise das Zusammenspiel von Blockheizkraftwerk und Spitzenlastkessel durch bessere Nutzung von Wärmespeichern optimieren kann. Dadurch wird Energie gespart und die Wirtschaftlichkeit verbessert.

Mit unserer Softwarelösung ecoControl können dazu Optimierungsalgorithmen entwickelt, getestet und angewendet werden. Wir haben solche Optimierungen simuliert und anschließend evaluiert. Dabei stellt sich heraus, dass mit einfachen Algorithmen bereits viel Energie eingespart werden kann. Unter der Verwendung von Prognosen lassen sich sogar noch bessere Ergebnisse erzielen. In der Simulation des Fallbeispiels konnten wir Energie und Kosten mit ecoControl einsparen. Sobald geeignete Hardwarekomponenten vorhanden sind, kann man die Software an echten Energiesystemen testen und Optimierungen weiter evaluieren.

5.1 Motivation

Die von uns vorgestellte Software ecoControl ist darauf ausgelegt, Energiekosten zu senken, indem der Betrieb von heterogenen Energiesystemen wie Blockheizkraftwerken, Wärmespeichern oder Photovoltaikanlagen effizienter koordiniert wird. Vor Projektbeginn haben wir Einsparungspotenziale von 25 % bis 40 % vermutet.

Innerhalb dieses Projekts wurde ein System entwickelt, mit welchem Messdaten von unterschiedlichen Energiesystemen gesammelt, ausgewertet und weiter verarbeitet werden können. Unter der Verwendung der von ecoControl zur Verfügung gestellten Werkzeuge, wie beispielsweise der Programmierschnittstelle oder der Prognosen, ist es möglich, das Zusammenspiel verschiedener Geräte zu verbessern. Dies kann erreicht werden, indem die Arbeitsweisen der Geräte voneinander abhängig gemacht werden. So kann es beispielsweise sinnvoll sein, ein bestimmtes Energiesystem immer nur dann zu betreiben, wenn ein anderes ausgeschaltet ist. Bisher war dies nur bedingt möglich. Damit Energiekosten mit ecoControl unter echten

Bedingungen eingespart werden können, bedarf es allerdings geeigneter Hardwarekomponenten. Diese Komponenten müssen die Messwerte von Energiesystemen in regelmäßigen Intervallen an die Software senden und Steuerungssignale empfangen und ausführen können.

5.1.1 Zielsetzung

In Abschnitt 5.2 werden wir zunächst an einem Fallbeispiel verdeutlichen, wie eine Optimierung der Energiesysteme aussehen kann und wie groß der Gewinn ist, der sich daraus ergibt. Dazu soll ein Szenario dienen, in welchem ein Blockheizkraftwerk (BHKW), ein Spitzenlastkessel (SLK), mehrere Wärmespeicher (WS) und eine Photovoltaikanlage zusammen betrieben werden. Da wir vermuten, dass in der Winterzeit, in der ein BHKW normalerweise überwiegend unter Volllast betrieben wird, die Einsparungsmöglichkeiten nicht sonderlich groß ausfallen, werden wir Echtdaten eines Tages im Mai 2014 betrachten. Es gibt verschiedene Optimierungsziele und -potenziale, die wir anhand des Beispiels erklären werden. Dazu zählen unter anderem die Maximierung der Laufzeit des Blockheizkraftwerks sowie die Verlängerung der Lebenszeit aller Energiesysteme. Außerdem werden wir argumentieren, warum durch einen besseren Einsatz eines BHKWs Energie gespart werden kann.

Anschließend diskutieren wir in Abschnitt 5.3 das Ergebnis des Fallbeispiels weiter. Danach werden die mit ecoControl möglichen Einsparungen in dem gegebenen Szenario für ein Jahr abgeschätzt.

Schließlich werden wir in Abschnitt 5.4 darauf eingehen, wie man die vorgestellte Software einsetzen kann, um Optimierungen vorzunehmen. Dazu werden wir erst zeigen, wie man die Aussteuerungen von Blockheizkraftwerk und Spitzenlastkessel mit geringem Aufwand verbessern kann. Darüber hinaus werden wir die Fahrweisen der Energiesysteme beispielhaft durch die von ecoControl bereitgestellten Prognosen weiter optimieren. Außerdem evaluieren wir den ersten Ansatz zur automatischen Optimierung weiter, der bereits in Kapitel 4 vorgestellt wurde. Abschließend werden wir die Gesamtergebnisse diskutieren.

5.2 Fallbeispiel

Für das Fallbeispiel sollen echte Messdaten eines Mehrfamilienhauses dienen. Diese wurden uns von Herrn Schmidt bereitgestellt. Er ist Besitzer eines Mehrfamilienhauses, in dem mehrere Energiesysteme eingesetzt werden. Anhand dieser Daten werden wir in einem ersten Schritt Optimierungsmöglichkeiten aufzeigen.

5.2.1 Beschreibung des Beispielszenarios

Die Ausgangssituation bildet eine Kombination bestehend aus einem Blockheizkraftwerk der Ausführung ecoPOWER 4.7 von Vaillant und einem auf 50 kW gedrosselten Spitzenlastkessel G324L des Herstellers Buderus. Außerdem gibt es mehrere gekoppelte Puffer- beziehungsweise Warmwasserspeicher mit insgesamt 2500 l Fassungsvermögen sowie drei Photovoltaikanlagen mit insgesamt 9,66 kW_p Maximal-

leistung. Von allen anderen Komponenten werden wir abstrahieren, um das Beispiel einfach zu halten. Das BHKW kann modulierend angesteuert werden, während der Heizkessel nur ein- oder ausgeschaltet werden kann. Beide Systeme werden mit Erdgas betrieben. Die Wärmespeicher, also Puffer- und Warmwasserspeicher, sind vergleichsweise groß ausgelegt, was vor allen bei Optimierungen später von Vorteil sein wird. Echte Messdaten vom Blockheizkraftwerk genauso wie die Schaltzyklen des Heizkessels konnten unter Verwendung der zugehörigen Technikersoftware ecoServ exportiert werden und liegen in 15 Minuten Intervallen vor. Zum Sammeln der Messdaten aller anderen Geräte hat der Betreiber sogenannte So-Recorder eingesetzt, die unter anderem Temperaturen sowie den Verbrauch von Gas und Strom messen können. Im Gegensatz zu den mit ecoServ ermittelten Messdaten, wurden diese in 20 Sekunden Intervallen aufgenommen.

Tabelle 5.1: Gegebene Daten für das Fallbeispiel

Blockheizkraftwerk [48]		
Maximale elektrische Leistung	p_{el}^{bhkw}	4,7 kW
Maximale thermale Leistung	p_{th}^{bhkw}	12,5 kW
Maximale Feuerungswärmeleistung	p_{max}^{bhkw}	19,0 kW
Spitzenlastkessel		
Maximale Feuerungswärmeleistung	p_{max}^{slk}	50,0 kW
Wirkungsgrad	η^{slk}	91,6 %
Preise		
Gaskosten	K_{Gas}	0,0655 €/kWh
Stromkosten	K_{Strom}	0,2660 €/kWh
Wartungskosten BHKW	$K_{Wartung}$	0,0220 €/kWh _{el}
Einspeisevergütung inkl. KWK-Zuschlag ¹	$G_{Einspeisung}$	0,0917 €/kWh
Mieterpreis für Wärmeenergie	G_{th}	0,0980 €/kWh
Mieterpreis für Strom ohne Energieumlage	G_{el}	0,2500 €/kWh

Die Daten in Tabelle 5.1 dienen neben den eigentlichen Messwerten als Grundlage für die Berechnung des Fallbeispiels. Die technischen Daten des Blockheizkraftwerks basieren auf Herstellerangaben. Alle anderen Daten wurden vom Betreiber des Beispielsystems berechnet und zur Verfügung gestellt.

5.2.2 Grundlagen für die Berechnung

Um das Fallbeispiel berechnen zu können, bedarf es einiger Grundlagen, die wir nun vorstellen werden. Ein Großteil der Berechnungen beruhen auf der Definition von Leistung. Es gilt $P = \frac{\Delta E}{\Delta t}$. Da die Feuerungswärmeleistungen, also der maximal mögliche Gasverbrauch, vom Blockheizkraftwerk und Spitzenlastkessel bekannt sind, kann man damit die jeweiligen Energiemengen über einen Zeitraum ermitteln. Bei den Leistungsangaben zum BHKW handelt es sich um Maximalwerte. Um die tatsächliche Leistung vereinfacht zu ermitteln, muss die entsprechende Maximaleistung mit der durchschnittlichen Auslastung im Zeitraum Δt multipliziert werden.

Da der Heizkessel nur an oder aus sein und deshalb nur mit 100 % oder 0 % multipliziert werden kann, entfällt dieses Vorgehen an dieser Stelle. Dafür ist die thermische Leistung über die maximale Feuerungswärmeleistung und den thermischen Wirkungsgrad zu bestimmen. Hier gilt $P_{th}^{slk} = P_{max}^{slk} \cdot \eta^{slk}$. Preisangaben zu den unterschiedlichen Energien lassen sich berechnen, indem die Energie mit dem jeweiligen Preis pro Kilowattstunde multipliziert wird.

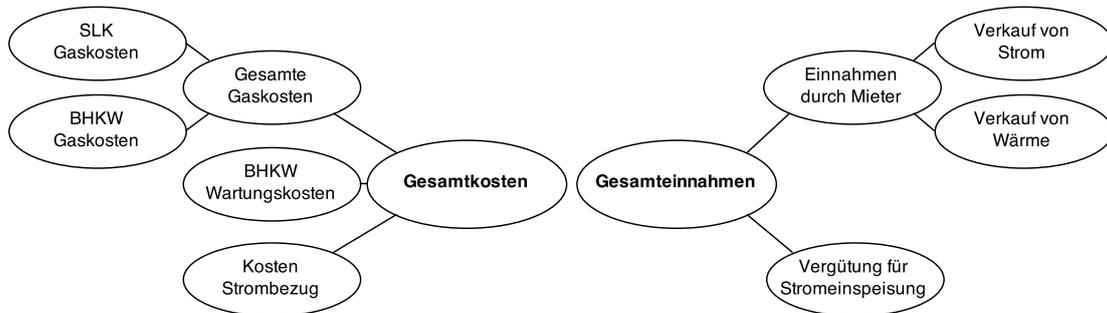


Abbildung 5.1: Zusammensetzung von Kosten und Einnahmen

Aus Abbildung 5.1 gehen die Abhängigkeiten der Gesamtkosten und -einnahmen hervor. Der Gasverbrauch und somit die Kosten für BHKW und SLK hängen darüber hinaus vom Betrieb der Systeme ab. Eine Besonderheit ist, dass die Wartungskosten des Blockheizkraftwerks von der umgewandelten elektrischen Energie abhängen. Es ist allerdings üblich, Wartungskosten so auszudrücken. Entsprechend setzen sich die Gesamtbetriebskosten aus den Gaskosten, den BHKW-Wartungskosten und den Kosten für den Einkauf von Strom zusammen. Die Einnahmen des Gesamtsystems hängen zum Großteil davon ab, wie viel Strom und Wärme an die Mieter verkauft werden konnten, aber auch von der Einspeisevergütung. Schließlich lässt sich der Gewinn über die Differenz von Einnahmen und Kosten berechnen. Die Formeln zur Berechnung von Einnahmen und Kosten durch BHKW und Heizkessel sind in Tabelle 5.2 aufgelistet.

Wir wollen noch einmal deutlich machen, dass der Verkauf von Strom und Wärme ausschließlich und direkt vom Bedarf der Mieter abhängt. Genauso richtet sich

Tabelle 5.2: Formeln für die Berechnung von Einnahmen und Kosten

Δt :	Zeitraum in Stunden
p^{bhw} und p^{slk} :	Durchschnittliche Leistung in Prozent im Zeitraum Δt
e :	Anteil des eingespeisten Stroms in Prozent im Zeitraum Δt
Gewinn _{el} ^{bhw} :	$(1 - e) \cdot G_{\text{el}} + e \cdot G_{\text{Einspeisung}}$
Einnahmen ^{bhw} :	$p^{\text{bhw}} \cdot \Delta t \cdot (p_{\text{th}}^{\text{bhw}} \cdot G_{\text{th}} + p_{\text{el}}^{\text{bhw}} \cdot \text{Gewinn}_{\text{el}}^{\text{bhw}})$
Kosten ^{bhw} :	$p^{\text{bhw}} \cdot \Delta t \cdot (p_{\text{max}}^{\text{bhw}} \cdot K_{\text{Gas}} + p_{\text{el}}^{\text{bhw}} \cdot K_{\text{Wartung}})$
Einnahmen ^{slk} :	$p^{\text{slk}} \cdot \Delta t \cdot (p_{\text{max}}^{\text{slk}} \cdot \eta^{\text{slk}} \cdot G_{\text{th}})$
Kosten ^{slk} :	$p^{\text{slk}} \cdot \Delta t \cdot (p_{\text{max}}^{\text{slk}} \cdot K_{\text{Gas}})$

die Aussteuerung des BHKWs und des SLKs danach. Allerdings ergeben sich hier weitere Abhängigkeiten, wie zum Beispiel zu dem Füllstand der Wärmespeicher. Üblicherweise sind Blockheizkraftwerke im Vergleich zu Spitzenlastkesseln wesentlich kleiner dimensioniert.

Wenn ein Heizkessel die Wärmespeicher vollständig gefüllt hat, kann ein Blockheizkraftwerk, da es neben Strom auch immer Wärme aus Gas umwandelt, unter Umständen den Bedarf an Strom nicht ganz decken. In diesem Fall wird Strom von außen bezogen, wodurch Kosten anfallen. Andersherum kann es aber auch durch erhöhten Wärmebedarf zur Überproduktion von Strom durch das BHKW kommen. Dieser kann dann eingespeist werden, wofür man eine Vergütung erhält.

5.2.3 Limitierungen und Abstraktionen bei der Berechnung des Fallbeispiels

Die Berechnung des Fallbeispiels unterliegt einigen Limitationen und aus Gründen der Vereinfachung müssen wir an einigen Stellen abstrahieren. Da die Messdaten in unterschiedlichen Granularitäten vorliegen, haben wir alle Daten auf 15 Minuten Intervalle gemittelt. Dies hat offensichtlich zur Folge, dass die Rechnung ungenauer wird. Außerdem kann es zu Messungenauigkeiten bei der Erfassung der Daten gekommen sein. Allerdings wollen wir nicht möglichst genau rechnen. Vielmehr soll das Beispiel demonstrieren, wie Energiesysteme prinzipiell besser angesteuert werden können, um die Energieeffizienz des Gesamtsystems zu steigern.

Genauso wollen wir davon absehen, dass der tatsächliche Wirkungsgrad eines Blockheizkraftwerks von der aktuellen Drehzahl des Motors abhängt. Diese Schwankungen im Wirkungsgrad können sich bei einem Erdgas-BHKW auf etwa $\pm 5\%$ belaufen [2]. Die ecoPower Modelle von Vaillant sind dafür bekannt, dass sie durch

einen Wechselrichter über noch geringere Wirkungsgradschwankungen verfügen². Davon abgesehen haben wir die Leistungsdaten des BHKWs den Datenblättern entnommen. Es ist anzunehmen, dass diese Angaben nur unter Laborbedingungen erreicht werden können.

Der Betreiber setzt zwei Pufferspeicher mit je 1000 l und einen Warmwasserspeicher mit 500 l Fassungsvermögen ein. Um nicht mit den Potenzialen aller drei Speicher rechnen zu müssen, stellen wir uns deshalb einen großen Wärmespeicher mit insgesamt 2500 l vor. Ebenfalls abstrahieren wir davon, dass Wärme nicht verlustfrei gespeichert werden kann. Aber auch der Verlust durch den eigentlichen Speicher, durch Rohre oder durch die Außentemperatur kann im Fallbeispiel vernachlässigt werden, da diese nahezu unabhängig von Optimierungen auftreten.

Es bleibt weiter festzuhalten, dass die gegebenen Preisangaben Werte sind, die von Region zu Region unterschiedlich sein und sich darüber hinaus jederzeit aus den verschiedensten Gründen ändern können. Die staatliche Subvention durch den KWK-Zuschuss hat der Betreiber Schmidt bei der Berechnung der gegebenen Werte berücksichtigt. Diese können sich aber, ähnlich wie die Gaspreise, jederzeit ändern. Zur Zahlung der EEG-Umlage machte er keine Angaben. Einige Werte, wie beispielsweise der Preis für Wärmeenergie für die Mieter, konnten vom Betreiber nur schätzungsweise angegeben werden. Wiederum andere, wie zum Beispiel die Wartungskosten des Spitzenlastkessels, wurden gar nicht erst in der Rechnung berücksichtigt. Spitzenlastkessel verfügen üblicherweise über einen einfachen Brenner und sind deshalb wartungsärmer als Blockheizkraftwerke, die einen Motor einsetzen, der unter anderem mit Öl versorgt werden muss. Wir nehmen deshalb an, dass die Wartungskosten des Heizkessels ebenfalls vernachlässigt werden können.

Nichtsdestotrotz wird die Berechnung eine verbesserte Fahrweise und die daraus resultierenden Einsparungen aufzeigen können. Schließlich kann es sich hierbei auch immer nur um Schätzungen handeln.

5.2.4 Berechnung des Beispiels

Um eine verbesserte Aussteuerung von Blockheizkraftwerk und Spitzenlastkessel zu erklären, wollen wir in diesem Beispiel die Auslastungen beider Geräte zunächst analysieren, berechnen und schließlich optimieren. Abbildung 5.2 zeigt die unterschiedlichen Auslastungen eines Tages im Mai 2014.

Es fällt auf, dass das Blockheizkraftwerk an diesem Tag nur 8 h 45 min betrieben wird, obwohl der Spitzenlastkessel 2 h 45 min verwendet werden muss. Darüber hinaus liegt die Leistung des BHKWs, während es eingeschaltet ist, zwischen 40 % und 65 %. Offensichtlich kam es am 24.05.2014 zu Problemen bei der Aussteuerung. Wir vermuten, dass der Spitzenlastkessel das BHKW an diesem Tag blockiert hat. Es kann aber auch sein, dass es zu einem Fehler in der Aussteuerungssoftware der Energiesysteme gekommen ist. Technische Fehler, zum Beispiel an der Hydraulik, sind ebenfalls nicht ausgeschlossen.

²Auskunft eines BHKW-Beraters.

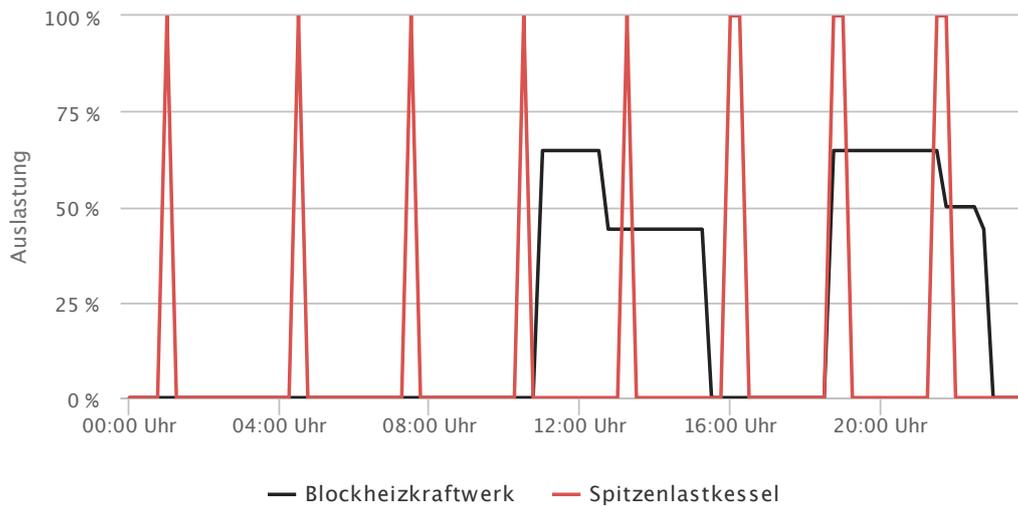


Abbildung 5.2: Ursprüngliche Auslastung von Blockheizkraftwerk und Spitzenlastkessel am 24. Mai 2014

Unsere Idee ist nun, das Verhalten beider Geräte so zu verändern, dass das Blockheizkraftwerk möglichst lange läuft und der Spitzenlastkessel dabei so wenig wie möglich verwendet werden muss. Es ist möglich, dies umzusetzen, da, wie sich herausstellen wird, die Wärmeenergie anstatt vom Spitzenlastkessel vollständig vom Blockheizkraftwerk bereitgestellt werden kann. Zunächst berechnen wir die Wärmeenergie, die der Spitzenlastkessel in 2 h 45 min umwandeln kann.

$$E_{\text{SLK}} = 2 \text{ h } 45 \text{ min} \cdot 50 \text{ kW} \cdot 91,6\% = 125,95 \text{ kWh}$$

Das Blockheizkraftwerk hat unter Volllast eine thermale Leistung von 12,5 kW. Um 125,95 kWh Wärmeenergie in diesem Zustand umzuwandeln, braucht es also

$$t_{\text{BHKW}} = \frac{125,95 \text{ kWh}}{12,5 \text{ kW}} \approx 10 \text{ h } 5 \text{ min.}$$

Das bedeutet, dass das Blockheizkraftwerk anstatt den Betrieb einzustellen, besser 10 h 5 min auf Volllast betrieben werden sollte. Insgesamt beläuft sich die Zeit, die das Blockheizkraftwerk ausgeschaltet ist auf 15 h 15 min. Das wiederum heißt, dass innerhalb dieser Zeit die nötige Menge Wärmeenergie theoretisch allein vom BHKW bereitgestellt werden kann. Dann würde das Blockheizkraftwerk allerdings immer noch rund 5 h stillstehen.

Sinnvoller ist es, das Blockheizkraftwerk modulierend anstatt unter Volllast zu betreiben, um den Maximalbetrieb zu verringern und zusätzlich die Laufzeit zu verlängern. Dadurch können wir außerdem eine zu extreme Nutzung des Wärmespeichers verhindern, die im schlimmsten Fall die Aussteuerung des BHKWs unnötig begrenzt. Denn nur wenn der Wärmespeicher noch über Kapazitäten verfügt, kann das Blockheizkraftwerk zeitweilig stromgeführt betrieben werden.

Wir haben herausgefunden, dass die optimale Nutzung des Wärmespeichers an dieser Stelle besonders wichtig ist, weil dieser den limitierenden Faktor darstellt. Zu einem ähnlichen Ergebnis kommen auch Peter Tzscheuschler und Urs Wehmhörner vom Lehrstuhl für Energiewirtschaft und Anwendungstechnik der Technische Universität München [47, 52]. Beide haben die bessere Nutzung von Wärmespeichern intensiv untersucht. In Abschnitt 5.4.2 werden wir genauer darauf eingehen.

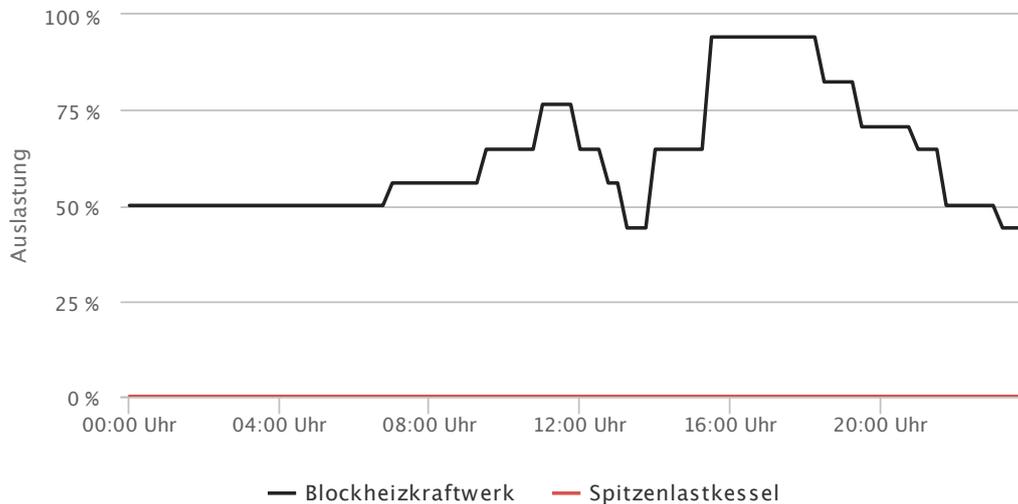


Abbildung 5.3: Verbesserte Auslastung Blockheizkraftwerk und Spitzenlastkessel

In Abbildung 5.3 haben wir eine mögliche Aussteuerung des Blockheizkraftwerks skizziert, die zur Folge hat, dass der Spitzenlastkessel nicht eingesetzt werden muss. Die einzelnen Leistungen beruhen auf Schätzungen. Wir haben aber darauf geachtet, dass ungefähr dieselbe Menge an Wärmeenergie durch das BHKW umgewandelt wird, die vorher durch den Heizkessel bereitgestellt wurde.

Als Nächstes werden wir uns die Auswirkungen der Verhaltensänderungen in Bezug auf die Kosten beziehungsweise die Einnahmen genauer ansehen. Vorher wurde der Spitzenlastkessel für insgesamt eine Stunde eingesetzt. Das Rechenergebnis ist in Tabelle 5.3 zu finden.

Tabelle 5.3: Rechnung bei Verwendung des SLKs wie in Abbildung 5.2

	Energie	Preis
Kosten durch Gasverbrauch	137,5 kWh	-9,01 €
Einnahmen durch Wärmeverkauf	125,95 kWh	12,34 €
Gewinn		3,33 €

Nun nehmen wir zunächst an, dass die zusätzliche Wärmeenergie vollständig vom Blockheizkraftwerk zur Verfügung gestellt und dass der dabei umgewandelte Strom restlos an die Mieter verkauft werden kann. Die zusätzliche elektrische Energie wirkt sich positiv auf die Bilanz aus, wie in Tabelle 5.4 erkennbar ist. Wenn die elektrische Energie allerdings komplett eingespeist werden muss, fällt die Bilanz entsprechend schlechter aus.

Wir konnten demnach zeigen, dass die Energiekosten schließlich nicht nur davon abhängen, wie viel Wärmeenergie vom Blockheizkraftwerk anstatt vom Spitzenlastkessel bereitgestellt wird, sondern auch davon, wie viel des Stroms im eigenen Haus direkt umgesetzt werden kann. Wenn zu viel Strom eingespeist werden muss, ist es sogar wirtschaftlicher, doch den Spitzenlastkessel einzusetzen. Das wird deutlich, wenn man Tabelle 5.3 und Tabelle 5.4 miteinander vergleicht. In diesem Fall haben wir durch das BHKW eine Bilanz von 3,10 € anstatt den ursprünglichen 3,33 € berechnet.

Tatsächlich wurden in der Zeit, in der das Blockheizkraftwerk ausgeschaltet war, ungefähr 26,75 kWh Strom eingekauft, die nun vom BHKW bereitgestellt werden können. Diese Annahme können wir treffen, denn es gab an diesem Tag nur ein Intervall, in dem der elektrische Bedarf über 4,7 kWh, also der maximalen elektrischen Leistung des Blockheizkraftwerks, lag. Wenn wir schätzungsweise davon ausgehen, dass wirklich 26,75 kWh im Haus verkauft werden können, fällt die Bilanz mit 7,34 € gut aus. Verglichen mit der ursprünglichen Bilanz sind das insgesamt rund 4 € beziehungsweise etwa 120 % mehr.

5.2.5 Ergebnis des Fallbeispiels

Wie wir in Abschnitt 5.2.4 zeigen konnten, können in den meisten Fällen mehr Einnahmen erzielt werden, indem man so viel Wärmeenergie wie möglich durch ein Blockheizkraftwerk bereitstellt und weniger beziehungsweise gar nicht auf den Spitzenlastkessel zurückgreift. Durch die zusätzliche elektrische Energie kommt man auf eine bessere Bilanz. Aus dieser Erkenntnis haben wir die Regel abgeleitet, dass das Blockheizkraftwerk theoretisch immer dann und mit möglichst hoher Auslastung betrieben werden muss, wenn der Spitzenlastkessel eingeschaltet ist.

Im Gegensatz zur ursprünglichen Nutzung des Heizkessel, können wir durch das beschriebene Vorgehen praktisch keinerlei Gas sparen. Genau genommen wird sogar mehr Gas verbraucht. Allerdings wird dabei auch elektrische Energie umgewandelt, die deutlich mehr Wert ist als Wärmeenergie. An dieser Stelle muss sorgfältig argumentiert werden, um zu erklären, warum insgesamt dennoch Energie eingespart werden konnte. In Abschnitt 5.2.6 gehen wir auf diese Problematik weiter ein.

Die veränderte Fahrweise von Spitzenlastkessel und Blockheizkraftwerk hat einen weiteren Vorteil. Die Schaltzyklen beider Energiesysteme werden erheblich verringert. Dadurch werden nicht nur Heizkessel und BHKW geschont, sondern auch die Wartungskosten auf lange Sicht reduziert. Das An- und Ausschalten von Motoren und von elektrischen Geräten verursacht generell Verschleiß, der so verringert werden kann.

Tabelle 5.4: Rechnung bei Verwendung des Blockheizkraftwerks

Fixkosten / Fixeinnahmen	
Gaskosten	−12,54 €
Einnahmen (thermisch)	12,34 €
Wartungskosten	−1,04 €
Vollständige Eigennutzung	
Maximale Einnahmen (elektrisch)	11,85 €
Maximaler Gewinn	10,61 €
Vollständige Einspeisung	
Minimale Einnahmen (elektrisch)	4,34 €
Minimaler Gewinn	3,10 €
Tatsächliches Ergebnis	
Einnahmen durch Mieter	6,69 €
Einspeisevergütung	1,89 €
Gewinn	7,34 €

Des Weiteren werden wir noch auf die eingesetzte Photovoltaikanlage eingehen. Da die entsprechende Leistung einzig und allein von der Sonneneinstrahlung abhängt, kann die Anlage nicht gesteuert werden. Die Kombination aus Photovoltaikanlage und Blockheizkraftwerk ist sinnvoll, obwohl der zusätzliche Strom aus Sonnenenergie das BHKW prinzipiell beeinflussen kann. Genauso wie die thermale Begrenzung durch den Wärmespeicher, kann der zusätzliche Strom das BHKW in seiner Leistung ebenfalls beschränken, wenn es beispielsweise im rein stromgeführten Modus arbeitet. Gerade hier kann durch eine bessere Aussteuerung die Effizienz des Energiesystems eines Hauses weiter optimiert werden, indem Leistungslimitierungen durch anderen Energiesysteme so gut wie möglich verhindert werden. Neuere Blockheizkraftwerke können zwar teilweise schon mit Spitzenlastkessel effizienter zusammenarbeiten, allerdings fehlt ihnen jegliche Möglichkeit, ihren Betrieb direkt von weiteren Energiesystemen wie einer Photovoltaikanlage abhängig zu machen.

5.2.6 Energiesparen trotz höherem Gasverbrauch

Nun wollen wir erklären, warum durch den Einsatz von Blockheizkraftwerken Energie gespart werden kann.

Der Wirkungsgrad von Kohlekraftwerken weltweit beträgt im Mittel 31 %, in Deutschland liegt er bei 38 % [19].

Siemens konnte 2011 mit dem Kraftwerk Irsching, einem Gaskraftwerk, welches einen Gesamtwirkungsgrad von über 60 % vorweisen kann, sogar einen Weltrekord für Gaskraftwerke ohne Kraft-Wärme-Kopplung aufstellen. Mit diesen Werten wollen wir verdeutlichen, dass die großen Stromversorger vergleichsweise große Verluste bei der Umwandlung einfahren, besonders dann, wenn die Abwärme nicht weiter verwendet wird. Deutlich effizienter wird es, wenn man die Energie direkt vor Ort mit einem Blockheizkraftwerk zur Verfügung stellt. Diese Kleinkraftwerke können Gesamtwirkungsgrade von bis zu 95 % erreichen [13].

Ein weiterer Faktor sind sogenannte Netzverluste. Es handelt sich hierbei um die Verluste, die bei der Übertragung von elektrischer Energie vom Kraftwerk bis hin zum Endverbraucher im Haus entstehen. Eine dezentrale Energieversorgung durch Blockheizkraftwerke oder Photovoltaikanlagen hat demnach nicht nur den Vorteil, dass keine zusätzlichen Durchleitungsgebühren anfallen, sondern auch, dass weniger Strom und Wärme beim Transport verloren gehen. Das wiederum heißt also, dass sich Energieeinsparungspotenziale aus längeren Laufzeiten eines Blockheizkraftwerks ergeben.

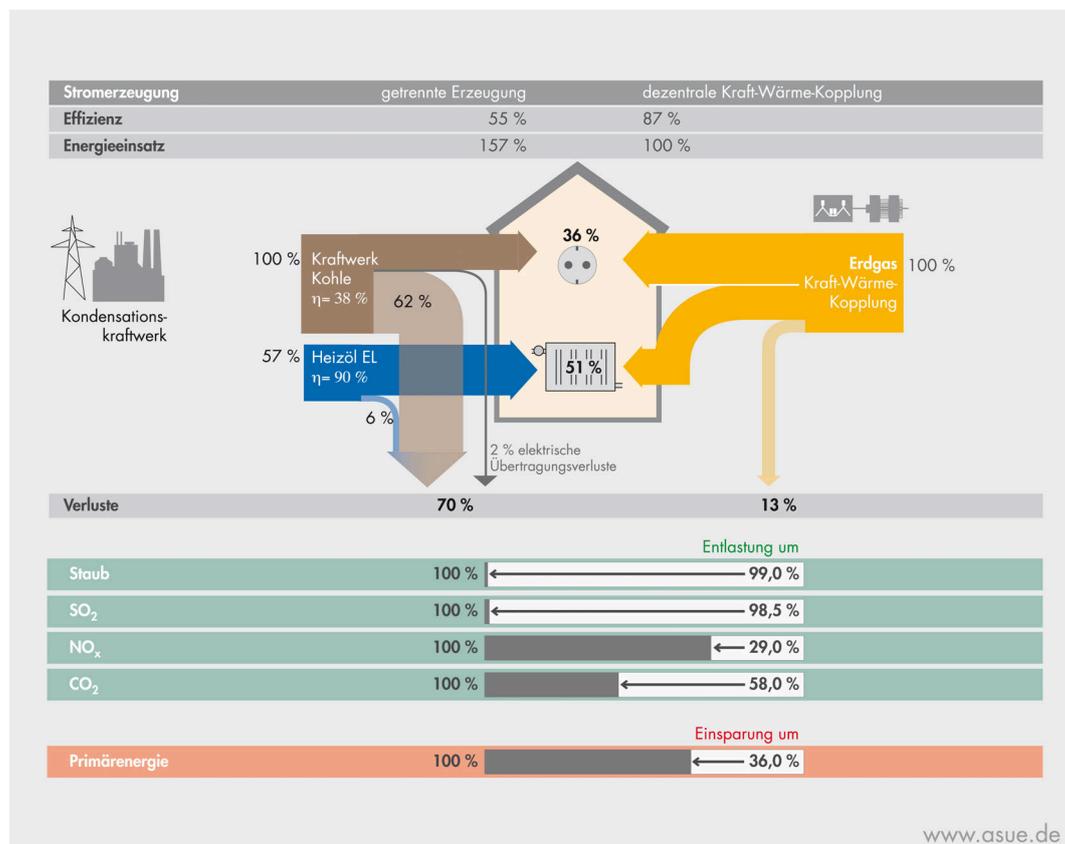


Abbildung 5.4: Vergleich der Effizienz und Umwelteigenschaften von Kohle- mit Blockheizkraftwerken (Quelle: asue.de [3])

Man kann zusammenfassend sagen, dass obwohl mehr Gas im Haus benötigt wird, mehr Energie und somit auch Energiekosten eingespart werden können, weil insgesamt weniger Verluste entstehen. Die Auswirkungen der Energieeinsparungen werden in Abbildung 5.4 weiter verdeutlicht.

Wie viel Energie tatsächlich gespart werden kann, lässt sich nur schwer verallgemeinern. Am besten einschätzen können dies nur die Energiekonzerne. Dennoch wollen wir versuchen, die Energieeinsparungen am Fallbeispiel abzuschätzen. Dazu vergleichen wir die ursprünglich benötigte Energie mit der Energie, die nach der Optimierung theoretisch benötigt wird.

Tabelle 5.5: Energieeinsparungen im Fallbeispiel

Bedarf an Strom	152,84 kWh
Bedarf an Wärmeenergie	137,50 kWh
Gesamtenergiebedarf vor Optimierung	290,34 kWh
Gesamtenergiebedarf nach Optimierung	191,52 kWh
Energieeinsparung	98,82 kWh
Relative Energieeinsparung	34,04 %

Wie aus Tabelle 5.5 hervorgeht, hätten wir am Tag des Fallbeispiels mit unserer Optimierung rund 98,82 kWh bei der Umwandlung sparen können. Da Steinkohle über einen Heizwert zwischen $7,5 \text{ kWh/kg}$ und 9 kWh/kg [51] verfügt, entspricht dies ungefähr 12 kg Kohle, die an einem Tag nicht im Kohlekraftwerk verbrannt werden müssen. Bei der Berechnung haben wir angenommen, dass der Strom von einem Kohlekraftwerk mit 38 % Wirkungsgrad bereitgestellt wird und sich die Netzverluste auf etwa 7 % belaufen. In dieser Größenordnung liegen laut Statistischem Bundesamt die durchschnittlichen Netzverluste in Deutschland [44]. Kohle- und sogenannte Gas- und Dampf-Kombikraftwerke können ebenfalls durch Kraft-Wärme-Kopplung höhere Gesamtwirkungsgrade von bis zu 90 % erreichen [13]. Die Wärmeenergie wird in diesen Fällen überwiegend in das örtliche Fernwärmenetz eingespeist. Aber auch dort entstehen weitere Transmissionsverluste.

Prinzipiell ist davon auszugehen, dass Blockheizkraftwerke aktuell die besten Gesamtwirkungsgrade vorweisen können, eben weil sie Energie sehr effizient umwandeln können und das direkt vor Ort, wo Wärme und Strom benötigt werden.

5.3 Verallgemeinerung des Ergebnisses

Aus dem Fallbeispiel ist hervorgegangen, wie man das Zusammenspiel von Blockheizkraftwerk und Spitzenlastkessel optimieren kann. Nun werden wir das Ergebnis auf ein Jahr hochrechnen, um den Gewinn besser abschätzen zu können.

5.3.1 Vorgespräch

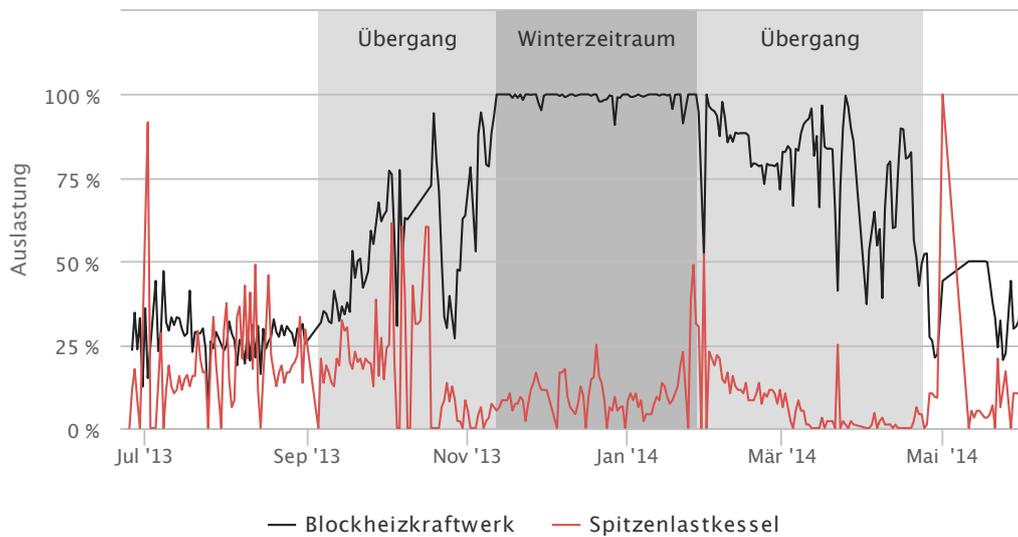


Abbildung 5.5: Auslastungen von Blockheizkraftwerk und Spitzenlastkessel in einem Jahr

Zunächst einmal sehen wir uns die Nutzung beider Geräte während eines Jahres genauer an. Abbildung 5.5 zeigt aggregierte Echtdata des vorher untersuchten Szenarios.

Als Erstes fällt auf, dass im Winterzeitraum das Blockheizkraftwerk nahezu die ganze Zeit unter Vollast betrieben wurde. Der Heizkessel sprang gelegentlich an, um Spitzenlasten auszugleichen. Im Winter werden wir das System daher nur sehr geringfügig optimieren können.

Im Frühling modulierte das Blockheizkraftwerk überwiegend. Hier lässt sich das Zusammenspiel mit dem Heizkessel weiter verbessern, auch wenn dieser laut dem Datensatz nur selten eingesetzt werden muss. Dies kann allerdings auch daran liegen, dass viele Messdaten in diesem Zeitraum aufgrund von technischen Störungen fehlen.

Anders sieht es allerdings im Sommer aus. Wir vermuten, dass die beiden größten Leistungsspitzen des Spitzenlastkessels wahrscheinlich durch Wartungsarbeiten am BHKW oder durch Ausfälle und technische Probleme verursacht wurden. Dennoch wurde gerade im Sommer der Heizkessel am meisten verwendet, obwohl das Blockheizkraftwerk im Durchschnitt nicht einmal annähernd auf halber Leistung betrieben wurde. Das heißt, dass das BHKW noch ungenutztes Potenzial hatte und den Spitzenlastkessel in der meisten Zeit hätte ersetzen können. Hier steckt somit ein großes Einsparungspotenzial.

Auch im Herbst war das Zusammenspiel zwischen Kraftwerk und Heizkessel nicht besonders gut. Deshalb vermuten wir weiter, dass man im Herbst mit Optimierungsalgorithmen ebenfalls Erfolg hat, auch wenn die Einsparungen nicht so groß ausfallen dürften wie im Sommer. Die allgemeine These ist, dass besonders im Sommer und den Übergangszeiten am meisten Energie in Mehrfamilienhäusern eingespart werden kann.

5.3.2 Schätzung der Einsparungen innerhalb eines Jahres

Um nun die Einsparungen schätzen zu können, werden wir die Daten aus dem Fallbeispiel weiter analysieren und mit den Jahresdaten vergleichen. Am 24.05.2014 lief das BHKW insgesamt 8 h 45 min und der Spitzenlastkessel lief 2 h 45 min. Wir nahmen an, dass das Blockheizkraftwerk den Heizkessel an diesem Tag komplett ersetzen kann. Dies wird aber nur an sehr wenigen Tagen im Jahr tatsächlich möglich sein, weil davon auszugehen ist, dass SLK und BHKW nicht immer so schlecht zusammenarbeiten wie im Fallbeispiel. Die größten Energieeinsparungspotenziale weist das Szenario jedoch im Sommer und im Herbst auf.

Tabelle 5.6: Statistische Auswertung von Echtdateien eines Jahres

	Frühling	Sommer	Herbst	Winter	Total
Blockheizkraftwerk					
Betriebsstunden	1023 h	950 h	1606 h	1894 h	5473 h
Vollbenutzungsstunden	198 h	20 h	953 h	1430 h	2599 h
Ausschaltzeit	202 h	850 h	314 h	27 h	1392 h
Gaskosten	-495 €	-340 €	-1449 €	-2142 €	-4301 €
Durchschn. Auslastung	38,89 %	28,77 %	72,51 %	90,92 %	63,15 %
Spitzenlastkessel					
Betriebsstunden	52 h	385 h	272 h	217 h	925 h
Ausschaltzeit	1173 h	1415 h	1649 h	1703 h	5939 h
Gaskosten	-169 €	-1262 €	-889 €	-710 €	-3029 €
Durchschn. Auslastung	2,15 %	21,40 %	14,14 %	11,29 %	13,48 %

In Tabelle 5.6 sind die saisonalen Unterschiede innerhalb eines Jahres bei den Laufzeiten von BHKW und SLK gut zu erkennen. Die angegebenen Jahreszeiten sind hier nach dem Kalender festgelegt. Wie schon in Abschnitt 5.3.1 erwähnt, läuft das Blockheizkraftwerk fast doppelt so lange im Winter wie im Sommer. Dabei arbeitet es überwiegend unter Volllast. Auch der erhöhte Einsatz des Spitzenlastkessels ist deutlich zu erkennen. Wir stellen nun die Vermutung auf, dass im Sommer die Laufzeit des Heizkessels um schätzungsweise 75 % und im Herbst um 50 % durch

Optimierungen verringert werden kann. Dies würde bedeuten, dass zusätzliche

$$385 \text{ h} \cdot 75 \% \cdot 50 \text{ kW} \cdot 91,6 \% = 13\,224,75 \text{ kWh}$$

im Sommer und

$$272 \text{ h} \cdot 50 \% \cdot 50 \text{ kW} \cdot 91,6 \% = 6\,228,80 \text{ kWh}$$

im Herbst an thermaler Energie durch das Blockheizkraftwerk bereitgestellt werden müssen. Unter Volllast braucht das BHKW dafür circa 1057,98 h beziehungsweise 498,30 h. Aus Tabelle 5.6 geht hervor, dass das Kraftwerk im Sommer ganze 850 h ausgeschaltet war und dass es nur 20 h von insgesamt 950 h unter Volllast betrieben wurde.

Im Herbst betrug die Ausschaltzeit 314 h und von 1606 h lief es 953 h unter Volllast. Wir können demnach davon ausgehen, dass das Blockheizkraftwerk durchaus die Laufzeit des Spitzenlastkessels im Sommer um 75 % und im Herbst um 50 % verringern kann. Dabei wird wieder wertvollere elektrische Energie umgewandelt.

Wenn wir zusätzlich noch annehmen, dass davon 60 % an die Mieter im Haus verkauft werden können und 40 % eingespeist werden, können wir berechnen, wie groß der Gewinn durch die bessere Aussteuerung von BHKW und Spitzenlastkessel ausfällt. Das 60:40 Verhältnis wird häufig beim Schätzen von Eigennutzungsanteilen genutzt und findet deshalb beispielsweise auch in der „Berechnung der Energiekosteneinsparung beim Einsatz eines Mikro-KWK“ [24] von Professor Oschatz Anwendung.

Tabelle 5.7: Berechnung der geschätzten Energiekosteneinsparungen

	Frühling	Sommer	Herbst	Winter	Summe
Vorher					
SLK Gewinn	62,49 €	467,46 €	329,44 €	263,00 €	1122,40 €
Nachher					
SLK Gewinn	62,49 €	116,87 €	164,72 €	263,00 €	607,08 €
BHKW Gewinn	0 €	908,23 €	426,71 €	0 €	1334,93 €
Gesamtgewinn	62,49 €	1025,09 €	591,43 €	263,00 €	1942,01 €
Einsparungen	0 €	557,63 €	261,99 €	0 €	819,62 €

Insgesamt schätzen wir die möglichen Energieeinsparungen bei dem gegebenen Szenario auf mehr als 800 €. Die Tabelle 5.7 zeigt eine genauere Aufschlüsselung der Rechnung. Zuerst berechnen wir, wie groß der Gewinn durch den Spitzenkessel ursprünglich war. Das Blockheizkraftwerk wurde auch schon vor der Optimierung

betrieben. Den Gewinn daraus haben wir zur Vereinfachung nicht betrachtet, da dieser nicht durch die Optimierung verändert wird. Der neue Gewinn besteht aus dem Gewinn des optimierten SLKs und aus dem zusätzlichen Gewinn des BHKWs. Diesen neuen Gewinn können wir dann mit dem ursprünglichen Gewinn vergleichen. Die Energiekosteneinsparungen beziehen sich demnach ausschließlich auf die Kosten und Einnahmen, die durch eine geringere Spitzenlastkessel- und eine höhere Blockheizkraftwerk-Laufzeit entstanden sind.

Der Gesamtgewinn der Anlage wird vom Betreiber auf etwa 2400 € pro Jahr geschätzt. Bei der Schätzung hat er bis auf die Investitionskosten sämtliche Nebenkosten berücksichtigt. Die Anschaffungskosten eines BHKWs der Ausführung ecoPower 4.7 von Vaillant liegen bei ungefähr 20 000 €. Dementsprechend belaufen sich die Abschreibungskosten pro Jahr für einen Zeitraum von zehn Jahren auf etwa 2000 €. Der tatsächliche Gewinn liegt also bei nur etwa 400 €. Eine Verbesserung um rund 800 € entspricht in diesem Fall einer Steigerung von 200 %.

Darüber hinaus haben wir errechnet, dass etwa 8330 kWh mehr Gas im Haus verbraucht werden. Warum dadurch Energie gespart werden kann, haben wir bereits in Abschnitt 5.2.6 ausführlich erläutert.

5.4 ecoControl zur Optimierung von Energiesystemen

Nachdem wir erklärt haben, wie sich ein Blockheizkraftwerk noch effizienter einsetzen lässt, werden wir nun zeigen, wie man ecoControl einsetzen kann, um vergleichbare Optimierungen vorzunehmen.

5.4.1 Entwicklung von Optimierungsalgorithmen

Die Software ecoControl bietet die Möglichkeit, optimierende Algorithmen für die Aussteuerung von unterschiedlichen Energiesystemen zu entwickeln, zu testen und anzuwenden. In Kapitel 4 wurde bereits die Vorhersagefunktion vorgestellt. Diese soll beim Entwickeln von Algorithmen helfen, indem man die Änderungen am System für die nächsten 14 Tage im Vorfeld sofort nachvollziehen kann. Die Algorithmen werden in der Programmiersprache Python geschrieben und können über die Technikersicht direkt im Browser programmiert, simuliert und angewendet werden. Es hat sich herausgestellt, dass sich Optimierungsalgorithmen für Energiesysteme nur schwer auf einfache Konstrukte abbilden lassen, weil sie beliebig komplex werden können. Aus diesem Grund haben wir uns für eine Programmierschnittstelle entschieden. Um den Nutzern so wenig wie möglich einzuschränken, hat er deshalb die Möglichkeit, Programmcode auszuführen. Hierzu stellt ecoControl eine einheitliche Schnittstelle für alle Geräte bereit.

Da es sich bei Python allgemein um eine sehr offene Programmiersprache handelt, ist die Programmierschnittstelle ein potentiell Sicherheitsrisiko. Dessen muss sich der Betreiber im Klaren sein. Wir gehen aber davon aus, dass die Nutzer, die Algorithmen entwickeln und Geräte damit steuern wollen, ohnehin vollen Zugriff auf das System benötigen. Dass man den Server direkt im Browser programmieren

kann, dient also in erster Linie dazu, die Entwicklung von Algorithmen so einfach wie möglich zu gestalten.

An dieser Stelle wollen wir noch einmal betonen, dass wir uns auf ein Szenario beziehen werden, das dem Fallbeispiel ähnelt. Eine Kombination aus schlecht aufeinander abgestimmten Energiesystemen, wie zum Beispiel falsch bemessene Wärmespeicher oder überdimensionierte Blockheizkraftwerke, wird auch durch Optimierungen niemals wirtschaftlich betrieben werden können. Wir nehmen deshalb an, dass die Planung sorgfältig vorgenommen wurde und die Geräte aufeinander abgestimmt sind.

5.4.2 Ziele von Optimierungsalgorithmen

Bevor wir einige Beispiele für Optimierungsalgorithmen vorstellen, werden wir zunächst noch einmal verdeutlichen, auf welchen Grundlagen solche Algorithmen entwickelt werden und welche unterschiedlichen Ziele man mit ihnen verfolgt.

ecoControl stellt dem Entwickler geeignete Prognosen der Geräteaussteuerungen zur Verfügung, die auf dem vorhergesagten Energiebedarf der Mieter, dem Verhalten der Geräte und auf dem Wetter basieren. Da Wettervorhersagen besonders wichtig für Optimierungen sein können, werden sie zusammen mit den Prognosen der Geräte bereitgestellt. Darüber hinaus kann der Entwickler auf alle Messdaten in der Datenbank zurückgreifen, die ecoControl gespeichert hat. Schließlich hat er über die Programmierschnittstelle die Möglichkeit, selbst-errechnete Werte zwischen zu speichern. Dazu kann am Ende des Programms ein beliebiges Objekt, wie zum Beispiel eine Liste, zurückgegeben werden, auf welches bei der nächsten Ausführung wieder zurückgegriffen werden kann.

Das wichtigste Optimierungsziel ist die Maximierung der Laufzeit eines Blockheizkraftwerks, weil dadurch der Gewinn vergrößert und Energie eingespart werden kann. Um dieses Ziel zu erreichen, müssen die Wärmespeicher besser genutzt werden.

Wärmespeicher sind die Kernkomponenten für einen flexiblen Betrieb [47]. Zu diesem Schluss kommt Peter Tzscheutschler vom Lehrstuhl für Energiewirtschaft und Anwendungstechnik der Technischen Universität München in einer seiner Präsentationen. Er folgert weiter, dass Strom- und Wärmebedarf durch eine verbesserte Nutzung der Wärmespeicher entkoppelt werden können. Dadurch wird es möglich, Blockheizkraftwerke zeitweilig stromgeführt zu betreiben. Wenn der Strombedarf in einem Haus beispielsweise steigt, sollte ein BHKW für möglichst viel Eigendeckung sorgen. Zu diesem Zeitpunkt ist es wichtig, dass die Speicher noch genug Wärmeenergie aufnehmen können und das sie das BHKW deshalb nicht in seiner Leistung einschränken.

Sind die Speicher hingegen ausreichend gefüllt, kann die Leistung des BHKWs gesenkt werden, sobald der Strombedarf nachlässt. Dadurch kann die Einspeisung minimiert werden. Wie Wärmespeicher in Zusammenspiel mit Blockheizkraftwerken besser eingesetzt werden können, um die Eigenstromnutzung zu maximieren und trotzdem den Wärmebedarf jederzeit decken zu können, wird ausführlich in der Dissertation von Urs Wehmhörner diskutiert [52].

Daneben gibt es noch andere Ziele, die man mit Optimierungsalgorithmen erreichen kann. Durch das Verringern der Schaltzyklen können Wartungskosten gesenkt und die Lebensdauer der meisten Energiesysteme verlängert werden, weil weniger Verschleiß anfällt. Dies lässt sich zum Beispiel durch größere Pausen zwischen zwei Einschaltvorgängen erreichen. Außerdem kann man sich auch Optimierungsalgorithmen vorstellen, die rein gewinnorientiert arbeiten. Einen solchen Algorithmus stellen wir in Abschnitt 5.4.7 vor.

5.4.3 Vorgehen bei der Evaluation der vorgestellten Optimierungsalgorithmen

Anhand der folgenden Beispiele werden wir zeigen, wie ecoControl zur Optimierung von Energiesystemen eingesetzt werden kann. Zunächst starten wir ecoControl im Demo-Modus mit der Konfiguration des Szenarios aus dem Fallbeispiel. Die generierten Messdaten werden, wie später beim Einsatz an echten Energiesystemen, in der Datenbank gespeichert.

Abbildung 5.6 zeigt die daraus resultierenden Verläufe von Wärmespeichertemperatur und den Auslastungen von BHKW und SLK innerhalb eines Jahres. Da die Daten tageweise aggregiert sind, liegt die durchschnittliche Auslastung des Heizkessels zwischen 0 % und 100 % und auch die des BHKWs liegt in einigen Zeiträumen unter der eigentlich möglichen Minimalauslastung.

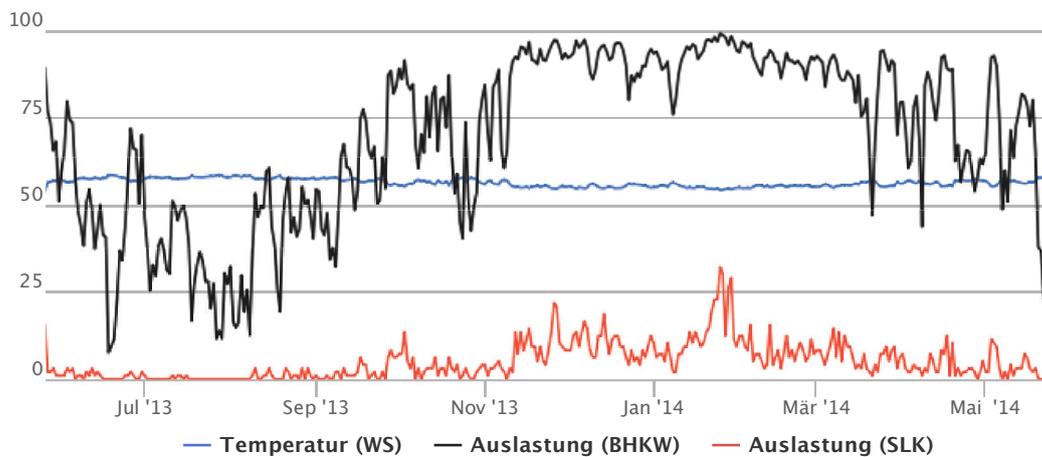


Abbildung 5.6: Simulierte Daten von Juni 2013 bis Mai 2014

Den Quelltext der folgenden Optimierungsbeispiele haben wir leicht vereinfacht in diesem Kapitel eingebunden. Da ecoControl immer noch weiterentwickelt wird, kann sich die genaue Spezifikation jederzeit ändern. Wir haben uns bemüht, das prinzipielle Vorgehen der Algorithmen möglichst gut darzustellen. Bei der Generierung der jeweiligen Ergebnisse kommt dieselbe ecoControl Instanz zum Einsatz.

Unter echten Bedingungen wird bei jeder Ausführung des Optimierungsalgorithmus eine 14-Tage-Prognose berechnet und dem Algorithmus zur Verfügung gestellt. Dies hätte jedoch die Simulation eines Jahres, so wie wir sie für die Evaluation beider Beispiele durchgeführt haben, deutlich verlangsamt. Schließlich wird der Algorithmus alle zwei Minuten in der Simulation ausgeführt, also insgesamt $365 \cdot 24 \cdot 60/2 = 262800$ Mal. Aus diesem Grund haben wir die Prognosen etwas eingeschränkt. Anstatt jedes Mal eine Prognose zu berechnen, wird einmal eine Prognose für die nächsten sieben Tage erzeugt und erst nach fünf Tagen neu berechnet. Das heißt, dass die Vorhersagen etwas ungenauer sind und dass in den Beispieloptimierungen nur die Prognosedaten für die nächsten zwei Tage verwendet werden können.

5.4.4 Eine einfache Beispieloptimierung

In Abschnitt 5.2.5 haben wir bereits eine Regel aus dem Fallbeispiel ableiten können. Diese besagt, dass das Blockheizkraftwerk möglichst unter Volllast laufen soll, wenn der Heizkessel eingesetzt wird. Zumindest sollte das BHKW niemals ausgeschaltet sein, was aber, wie die Echt Daten belegt haben, durchaus der Fall sein kann. In der verwendeten Simulation, die in Kapitel 1 genauer vorgestellt wurde, tritt dieser Fall niemals ein, weil die Energiesysteme fehlerfrei angesteuert werden. Fehlerhaftes Verhalten, welches wir in Abschnitt 5.2.4 identifizieren konnten, wird hingegen nicht simuliert. Stattdessen versuchen wir, die Laufzeit des Blockheizkraftwerks im Frühjahr, Sommer und Herbst auf anderem Wege zu erhöhen.

Quelltext 5.1: Einfacher Optimierungsalgorithmus

```
if 3 < heute.monat < 11 and WS.temperatur() < 80.0:
    BHKW.automatischer_Modus('AUS')
    # Schalte BHKW ein bzw. erhoehle Leistung um 2 %
    if BHKW.auslastung() == 0:
        BHKW.setze_Auslastung(BHKW.minimale_Auslastung)
    else:
        BHKW.setze_Auslastung(BHKW.auslastung * 1.02)
else:
    BHKW.automatischer_Modus('AN')
```

Quelltext 5.1 zeigt einen Optimierungsalgorithmus, der in den Monaten April bis Oktober das BHKW einschaltet beziehungsweise die Leistung des BHKWs um 2% pro Ausführung erhöht, wenn der Wärmespeicher dies zulässt. Damit ist sichergestellt, dass der Speicher eine Maximaltemperatur von 80 °C nicht überschreitet. Ansonsten soll das BHKW, wie gewöhnlich, automatisch, das heißt selbstständig modulierend, angesteuert werden. Es fällt auf, dass wir bei dieser sehr einfachen Optimierung auf keinerlei Vorhersagen zurückgreifen müssen.

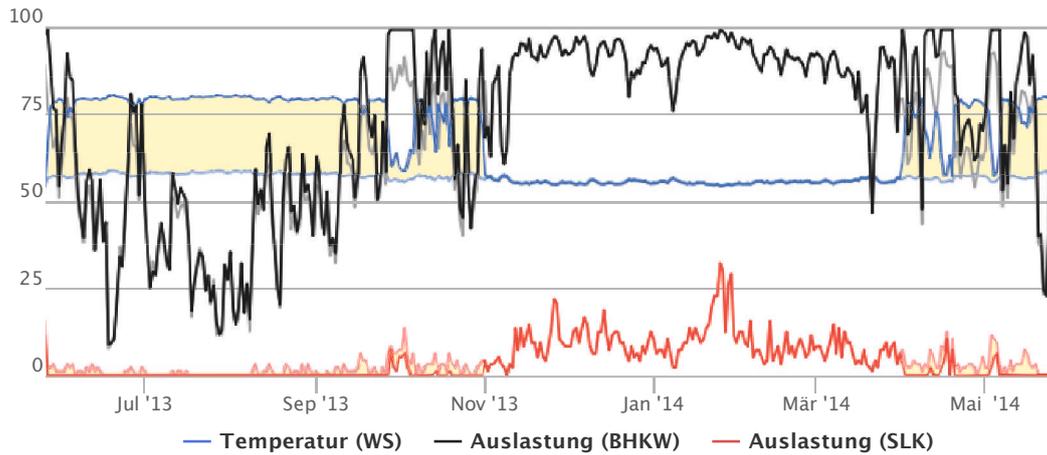


Abbildung 5.7: Einfache Optimierung von BHKW und SLK, wichtige Bereiche wurden eingefärbt

Das Ergebnis der Optimierung ist Abbildung 5.7 zu entnehmen. Verglichen mit den ursprünglichen Daten aus Abbildung 5.6 konnten wir die Laufzeit des BHKW erhöhen und die des Heizkessels in den entsprechenden Monaten verringern. Grund dafür ist eine stärkere Nutzung des Wärmespeichers, der sich in den optimierten Monaten größtenteils nahe der festgelegten Maximaltemperatur von 80 °C befindet.

Die genauen Auswirkungen dieser ersten Optimierung auf die Energiekosten werden wir in Abschnitt 5.4.6 zusammen mit dem Ergebnis einer weiteren Optimierung vergleichen.

5.4.5 Optimierung unter Verwendung von Vorhersagen

Die Vorhersagen über das Verhalten der Energiesysteme sind nicht nur beim Finetuning der Geräte oder während der Entwicklung von Optimierungsalgorithmen eine Hilfe. Die von ecoControl erzeugten Prognosen stehen in der Programmierschnittstelle zur Verfügung und können deshalb auch dazu genutzt werden, Algorithmen abhängig von zukünftigen Ereignissen zu programmieren.

Quelltext 5.2 zeigt einen solchen Algorithmus, welcher auf Schaltzyklen vom Spitzenlastkessel und Außentemperaturveränderungen reagiert. Unsere Idee ist, das Blockheizkraftwerk höher auszusteuern, wenn der Spitzenlastkessel in naher Zukunft verwendet werden muss oder wenn es draußen wesentlich kälter wird. Besonders interessant ist die Möglichkeit, auf Wettervorhersagen zurückgreifen zu können. Da normalerweise bei jedem Ausführen des Algorithmus eine Prognose für die nächsten 14 Tage berechnet und innerhalb des Optimierungsalgorithmus bereitgestellt wird, kann man insgesamt auf $14 \cdot 24 \text{ h} \cdot 4/\text{h} = 1344$ Werte pro Sensor zurückgreifen. Durch die von uns in Abschnitt 5.4.3 getroffenen Einschränkungen stehen diesem Algorithmus allerdings nur $2 \cdot 24 \text{ h} \cdot 4/\text{h} = 192$ Sensorwerte zur Verfügung.

Quelltext 5.2: Vorhersagen-basierter Optimierungsalgorithmus

```

Temp_Tag_1 = Aussentemperaturen[0:96]
Temp_Tag_2 = Aussentemperaturen[96:2 * 96]

Max_Delta = max(Temp_Tag_1) - max(Temp_Tag_2)
Min_Delta = min(Temp_Tag_1) - min(Temp_Tag_2)
Delta = max(Max_Delta + Min_Delta, 0)

# Zaehle Intervalle, in denen SLK eingeschaltet ist
SLK_Intervalle = 0
for Wert in SLK_Auslastungen[0:2 * 96]:
    if Wert > 0:
        SLK_Intervalle += 1

if ((Delta > 4.0 or SLK_Intervalle > 0) and
    WS.temperatur() < 80.0):
    BHKW.automatischer_Modus('AUS')
    # Schalte BHKW ein bzw. erhoehe Leistung um 2%
    if BHKW.auslastung() == 0:
        BHKW.setze_Auslastung(BHKW.minimale_Auslastung)
    else:
        BHKW.setze_Auslastung(BHKW.auslastung * 1.02)
else:
    BHKW.automatischer_Modus('AN')

```

In dem konkreten Programmcode hinterlegen wir zunächst die vorhergesagten Außentemperaturen für den nächsten und den übernächsten Tag in den Variablen *Temperatur_Tag_1* und *Temperatur_Tag_2*. Dann bilden wir die Differenzen zwischen den beiden höchsten und den beiden niedrigsten Temperaturen der beiden Tage. *Delta* setzt sich dann aus den Summen beider Differenzen zusammen, kann aber mindestens nur 0 sein. Je größer dieser Wert also ist, desto größer ist auch die Temperaturänderung von warm nach kalt. Anschließend ermitteln wir, wie häufig der Spitzenlastkessel in den nächsten zwei Tagen anspringen wird. Schließlich wird ähnlich, wie im vorherigen Optimierungsbeispiel, das BHKW manuell eingeschaltet oder die Leistung um 2 % erhöht, wenn *Delta* größer 4 °C ist oder der Heizkessel bald verwendet werden muss. An dieser Stelle überprüfen wir ebenfalls wieder, ob der Wärmespeicher eine höhere Aussteuerung des BHKWs zulässt. Ansonsten wird das Blockheizkraftwerk wieder in den automatischen Modus versetzt.

Abbildung 5.8 zeigt das Resultat der zweiten Optimierung. Der Spitzenlastkessel läuft, ähnlich wie zuvor, wesentlich weniger in den Übergangszeiten. Es fällt auf, dass der Wärmespeicher im Vergleich zur vorherigen Optimierung in dieser Zeit nicht durchgängig auf etwa 80 °C gehalten wird. Dies deutet auf eine effizientere Wärmespeicherung hin und lässt vermuten, dass das BHKW aus diesem Grund auch weniger Strom überproduziert hat. In der Winterzeit wird das Blockheizkraftwerk außerdem fast durchgehend unter Volllast betrieben. Dadurch wird die Laufzeit des Heizkessels auch im Winter weiter reduziert. Insgesamt kann man also sagen,

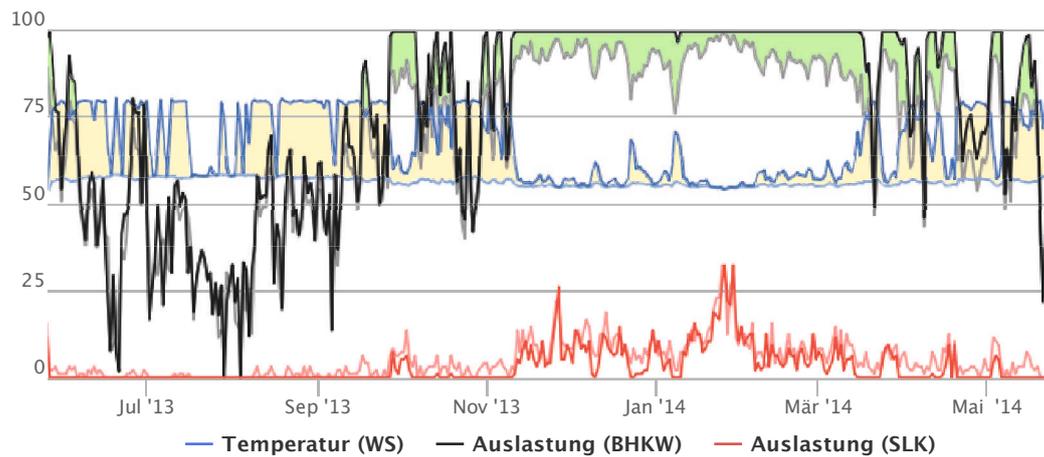


Abbildung 5.8: Vorhersagen-basierte Optimierung von BHKW und SLK; die wichtigen Bereiche wurden eingefärbt

dass das Ergebnis noch besser ausgefallen ist als das Ergebnis der Optimierung aus Abschnitt 5.4.4.

5.4.6 Auswertung der Optimierungsergebnisse

Tabelle 5.8: Statistische Auswertung der Optimierungen

	Unverändert	Optimierung aus 5.4.4	Optimierung aus 5.4.5
Blockheizkraftwerk			
Durchschn. Auslastung	69,1775 %	73,0240 %	76,1247 %
Gaskosten	-7541,61 €	-7960,95 €	-8298,98 €
Spitzenlastkessel			
Durchschn. Auslastung	4,9196 %	3,9616 %	2,9839 %
Gaskosten	-1411,38 €	-1136,54 €	-856,05 €
Gewinn	5095,13 €	5247,53 €	5348,54 €

Wie aus Tabelle 5.8 hervorgeht, haben wir die durchschnittliche Leistung pro Jahr des Blockheizkraftwerks mit dem in Abschnitt 5.4.4 vorgestellten Algorithmus um 3,8465 % erhöhen und die des Spitzenlastkessels um 0,9580 % verringern können. Im ersten Augenblick scheint die Verbesserung sehr gering ausgefallen zu sein. Al-

lerdings werden die Ersparnisse deutlicher, wenn man Kosten und Einnahmen im gesamten Jahr betrachtet. Die Kosten des Kraftwerks sind um 419,34 € gestiegen, dafür können die Gaskosten des Heizkessels um 274,84 € gesenkt werden. Da das BHKW neben Wärme auch noch wertvolleren Strom aus Gas umgewandelt hat, werden dennoch ganze 152,40 € mehr Gewinn erzielt.

Es ist anzumerken, dass sich die Gewinne in Tabelle 5.8 ausschließlich auf BHKW und Heizkessel beziehen und weder zusätzliche Kosten für den Strombezug noch Investitionskosten oder weitere Kosten eingerechnet sind. Deshalb ist es an dieser Stelle nicht möglich, eine relative Gesamteinsparung anzugeben.

Mithilfe der Optimierung aus Abschnitt 5.4.5 haben wir die Nutzung des Blockheizkraftwerks noch weiter maximieren können. Auf den Spitzenlastkessel wird fast nur noch halb so lange zurückgegriffen. Dadurch haben wir den Gewinn insgesamt um 253,41 € steigern können. Im Vergleich zu der einfachen Optimierung aus Abschnitt 5.4.4, fällt der zusätzliche Gewinn durch die Verbesserung über 66 % besser aus. Es hat sich also gezeigt, dass mit den Prognosen von ecoControl noch bessere Ergebnisse bei der Einsparung erzielt werden können.

5.4.7 Automatische Optimierung

An einem letzten Optimierungsalgorithmus werden wir schließlich zeigen, dass die reine Maximierung der Laufzeit eines BHKWs unter bestimmten Bedingungen nicht immer am wirtschaftlichsten ist. In Abschnitt 5.2.4 haben wir bereits feststellen können, dass die Gesamtbilanz nicht nur von den Laufzeiten der Energiesysteme abhängt, sondern auch davon, wie viel Energie innerhalb des Hauses direkt von den Mietern abgenommen wird. Da die Einspeisevergütung pro kWh in dem gegebenen Szenario geringer ist, als die Einnahmen durch den Verkauf von Wärme, sollte die Überproduktion von Strom so gering wie möglich ausfallen. Ein Algorithmus, der diese Abhängigkeiten berücksichtigt, kann den Gewinn deshalb weiter erhöhen.

An dieser Stelle soll noch einmal betont werden, dass Optimierungsstrategien nicht nur von den Eigenschaften und der Einsatzmöglichkeiten der Energiesysteme abhängen, sondern auch von externen Faktoren, wie Marktpreise oder Gesetzeslagen. Eine Steigerung des Gaspreises kann sich drastisch auf die Wirtschaftlichkeit auswirken. Genauso verhält es sich mit der gesetzlich vereinbarten Einspeisevergütung für Strom. Würde die Einspeisevergütung pro kWh über dem Mieterpreis für eine Kilowattstunde Wärmeenergie liegen, wäre die Überproduktion von Strom wieder attraktiv. Läge sie deutlich darüber, wäre es zumindest theoretisch sogar noch wirtschaftlicher, überschüssige Wärmeenergie überhaupt nicht zu nutzen und zum Beispiel an die Umwelt in irgendeiner Weise abzugeben.

Die Ausgangslage ist erneut ein Szenario, welches dem ursprünglichen Fallbeispiel ähnelt. Diesmal werden wir allerdings an nur einem Monat beispielhaft zeigen, wie man einen entsprechenden Optimierungsalgorithmus bewerten kann, und setzen deshalb dieselbe ecoControl Instanz ein, die schon in Kapitel 4 verwendet wurde. Die mithilfe der Simulation erzeugten Daten für den Monat März finden sich in Abbildung 5.9. Es fällt auf, dass in dieser Zeit das BHKW häufig an und ausgeschaltet wird und dass der Spitzenlastkessel nicht verwendet werden muss. Vergleichen wer-

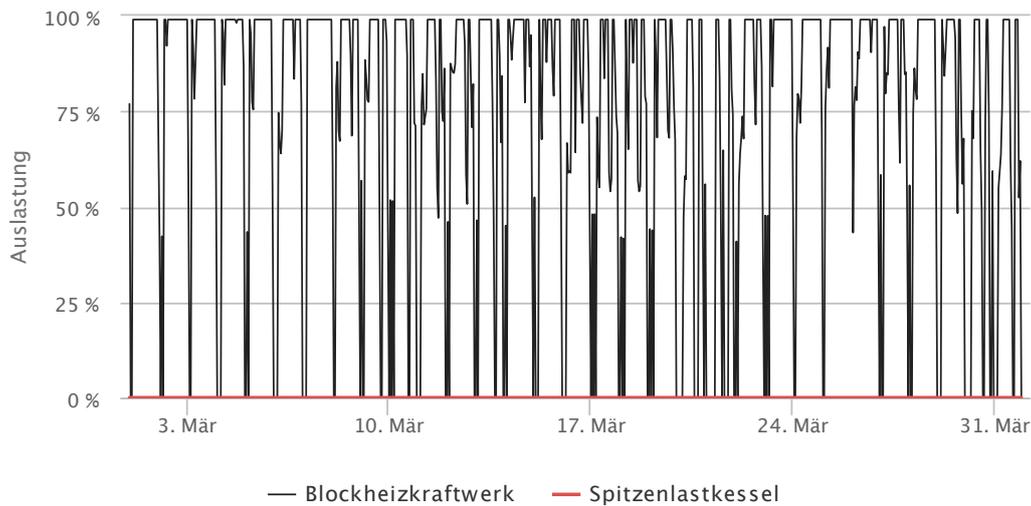


Abbildung 5.9: Nicht optimierte Fahrweisen von BHKW und SLK

den wir diese Daten nun mit der automatischen Optimierung, die bereits in Kapitel 4 genauer vorgestellt wurde.

Abbildung 5.10 zeigt eine deutliche Veränderung der Fahrweisen. Der Spitzenlastkessel kommt in einigen Zeiträumen zum Einsatz und das Blockheizkraftwerk moduliert an einigen Stellen mehr. Das besonders Auffällige ist, dass der Anteil der Eigenstromnutzung durch den Algorithmus gesteigert werden kann. Eingespeist werden nur noch etwa 11,08 % des selbstproduzierten Stroms, ursprünglich waren es ungefähr 22,94 %. Wir halten noch einmal fest, dass der Algorithmus ausschließlich versucht, den Strombedarf so gut wie möglich zu decken und dadurch den Gewinn zu maximieren. Dabei kann die Anzahl an Schaltzyklen und somit der Verschleiß der Energiesysteme zugenommen haben. Allerdings ist in den beiden Abbildungen zumindest keine große Änderung der Schaltzyklen ersichtlich.

Die Auswirkungen auf die Bilanzen haben wir in Tabelle 5.9 aufgeschlüsselt. Die durchschnittliche Auslastung des BHKWs wird um circa 8,2 % verringert, wodurch auch dessen Bilanz um 11,13 € sinkt. Durch den Einsatz des Spitzenlastkessels können dafür 30,07 € zusätzlich erwirtschaftet werden. Aufgrund der gestiegenen Eigenstromnutzung muss weniger Strom eingekauft werden. Insgesamt ergibt sich so eine Einsparung von knapp 32 €. Diese Einsparung ist nicht besonders groß in diesem Monat ausgefallen. Dennoch konnten wir zeigen, dass es in wenigen Fällen wirtschaftlicher ist, den Heizkessel trotzdem zu bevorzugen. Auch in anderen Monaten haben wir mit diesem Algorithmus den Gewinn vergrößern können.

5.4.8 Diskussion

Im Abschnitt 5.4 haben wir erklärt, wie `ecoControl` eingesetzt werden kann, um das Zusammenspiel von Blockheizkraftwerk und Spitzenlastkessel und deren Nutzung des Wärmespeichers zu optimieren. Dabei ist aufgefallen, dass man bereits mit einfa-

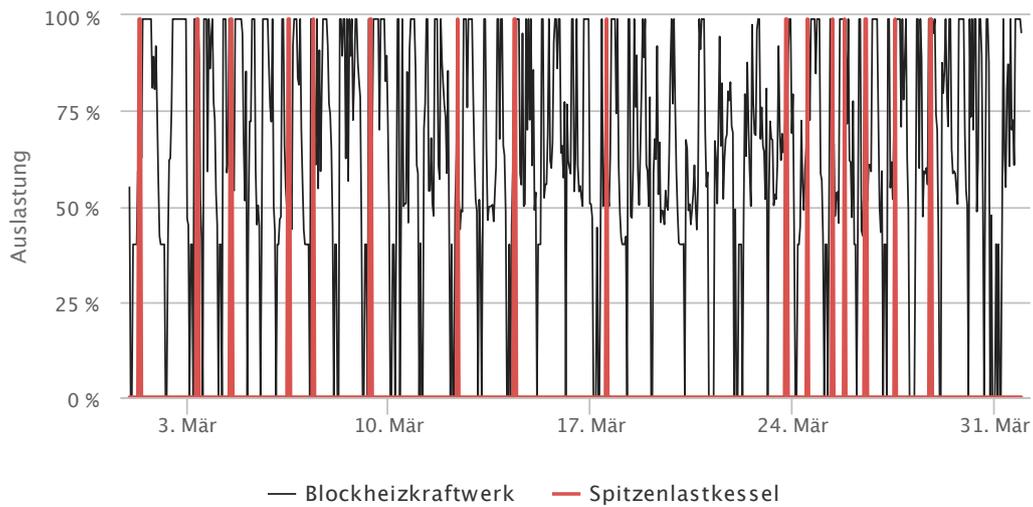


Abbildung 5.10: Gewinnerorientierte Optimierung von BHKW und SLK

Tabelle 5.9: Statistische Auswertung der automatischen Optimierung

	Unverändert	Optimiert
Blockheizkraftwerk		
Durchschn. Auslastung	73,6625 %	65,4182 %
Bilanz	483,08 €	471,95 €
Spitzenlastkessel		
Durchschn. Auslastung	0 %	3,3311 %
Bilanz	0 €	30,07 €
Bilanz BHKW & SLK	483,08 €	502,02 €
Eigenstromnutzung	77,06 %	88,92 %
Stromeinkaufskosten	186,98 €	173,95 €
Gesamtbilanz	296,10 €	328,07 €

chen Optimierungsalgorithmen, die nicht auf Prognosen zurückgreifen, vergleichsweise viel Einsparungen erzielen kann. Ein noch besseres Ergebnis konnten wir mit Optimierungen erzielen, die die von `ecoControl` bereitgestellten Vorhersagen mit einbeziehen. Bei beiden vorgestellten Optimierungsalgorithmen handelt es sich um einfach gehaltene Beispiele. Wir nehmen an, dass durch weitere Anpassungen der Parameter oder durch zusätzliche Abhängigkeiten der Aussteuerungsbefehle noch mehr eingespart werden kann.

Mit einem weiteren Beispiel haben wir dann zeigen können, dass durch das Minimieren der Laufzeit des Heizkessels der reine Gewinn nicht immer vergrößert werden kann. Welcher Optimierungsalgorithmus zum Einsatz kommt, muss der jeweilige Techniker beziehungsweise die Person festlegen, die `ecoControl` bedient. Wie aus Abschnitt 5.4.2 hervorgegangen ist, gibt es unterschiedliche Optimierungsziele.

Nach unserer Meinung kann es durchaus Sinn ergeben, einen Algorithmus so auszurichten, dass er mehrere dieser Ziele verfolgt. Solange sich die einzelnen Komponenten nicht gegenseitig ausschließen, kann dies durch einfache Komposition mehrerer Algorithmen erreicht werden. Das daraus entstandene Ergebnis ist zwar ein komplexerer Algorithmus, dafür kann er die Aussteuerung von Energiesystemen noch genauer festlegen. Vorstellbar ist auch die Komposition von zeitabhängigen Algorithmen, wie zum Beispiel Algorithmen für unterschiedliche Jahreszeiten. Prinzipiell sind dem Entwickler solcher Algorithmen keine Grenzen gesetzt. Da wir uns dafür entschieden haben, dass Optimierungsalgorithmen mit Python-Code definiert werden, ist es möglich, den Quelltext unter den Entwicklern auszutauschen, um gemeinsam Verbesserungen vorzunehmen.

Damit `ecoControl` unter echten Bedingungen eingesetzt werden kann, ist zusätzliche Hardware notwendig. Der Projektpartner hat vor, ein weiteres Projektteam mit der Entwicklung einer geeigneten Hardwarekomponente zu beauftragen. Sobald solche Komponenten vorhanden sind, kann man `ecoControl` an echten Energiesystemen ausprobieren und verschiedene Optimierungen weiter evaluieren.

5.5 Zusammenfassung

In diesem Kapitel haben wir zunächst an einem Fallbeispiel deutlich gemacht, wie durch eine verbesserte Aussteuerung eines Blockheizkraftwerks Energiekosten eingespart werden können. Wir haben gezeigt, dass durch längere Laufzeiten nicht nur mehr Gewinn erzielt, sondern insgesamt auch Energie eingespart werden kann. Ist ein Blockheizkraftwerk besser an den Strombedarf der Mieter angepasst, muss weniger Strom eingekauft werden. Auf dem Weg zwischen einem großen Kraftwerk und dem Haus fallen Netzverluste an, die so ebenfalls eingespart werden können. Außerdem weisen die meisten Kraftwerke der Energieversorger einen wesentlich geringeren Gesamtwirkungsgrad als Blockheizkraftwerke auf. Deshalb ist es sinnvoll, ein BHKW so einzusetzen, dass es den Eigenstrombedarf so gut wie möglich deckt.

Anschließend haben wir die möglichen Einsparungen mit `ecoControl` für das gegebene Szenario für ein Jahr berechnet. Es hat sich gezeigt, dass pro Jahr rund

800 € durch eine bessere Aussteuerung der Energiesysteme eingespart werden können. Unter Berücksichtigung sämtlicher Nebenkosten und den Investitionskosten entspricht dies ungefähr einer Verdreifachung des Gewinns.

Schließlich haben wir vorgestellt, wie ecoControl eingesetzt werden kann, um effizienzsteigernde Optimierungen vorzunehmen. Dabei haben wir festgestellt, dass bereits mit einfachen Algorithmen viel eingespart werden kann.

Noch bessere Ergebnisse konnten wir mit Optimierungsalgorithmen erreichen, die auf Prognosen von ecoControl zurückgreifen. Einen ersten Ansatz für einen automatischen Optimierungsalgorithmus mit dem Ziel, die Eigenstromnutzung zu verbessern, konnten wir schließlich ebenfalls vorstellen. Daran haben wir gezeigt, dass die reine Maximierung der Laufzeit eines Blockheizkraftwerks unter bestimmten Voraussetzungen nicht immer am wirtschaftlichsten ist.

Der nächste Schritt ist, ecoControl unter echten Bedingungen an realen Energiesystemen zu testen und zu evaluieren. Voraussetzung dafür ist eine geeignete Hardwarekomponente, die Messdaten an ecoControl senden und Steuerungsbe-
fehle von ecoControl empfangen und verarbeiten kann.

Schlussbetrachtung

Das Ziel des Projekts war es, eine Software zu entwickeln, mit der die Effizienz bei der Eigenproduktion von Strom und Wärme gesteigert und Kosten gesenkt werden können.

Zuerst haben wir uns anhand einer Simulation die Domäne erschlossen. Die Simulation wurde mit den Messdaten eines realen Mehrfamilienhauses verglichen. Hier konnten wir zeigen, dass die Nutzung des Wärmespeichers innerhalb eines Jahres gut simuliert werden kann. Die Simulation der anderen Geräte muss noch verfeinert und genauer auf echte Gegebenheiten abgestimmt werden.

Nutzerstudien haben uns dabei geholfen, die Nutzerrollen und Anforderungen genau zu definieren. Die Optimierung der Energiesysteme mithilfe von ecoControl soll durch einen Techniker vorgenommen werden.

Damit ecoControl erweitert werden kann, ist die Anwendung modular aufgebaut. Durch einen agilen Entwicklungsprozess konnte Feedback schnell berücksichtigt und auf neue Anforderungen reagiert werden.

Ein wichtiger Bestandteil von ecoControl ist die Berechnung von Prognosen. Der Nutzerverbrauch kann auf zwei Wochen gut vorhergesagt werden. Das Verhalten der Energiesysteme kann durch diese Vorhersage und durch die Simulation im Vorfeld berechnet werden. Die Prognosen konnten für einen ersten Algorithmus verwendet werden, der die Systeme automatisch so effizient wie möglich aussteuert.

Wir haben weiterhin gezeigt, wie ecoControl die eigene Entwicklung von Optimierungsalgorithmen unterstützt. Schließlich konnten wir an einem Fallbeispiel zeigen, wie Kosten und Energie in einem Mehrfamilienhaus eingespart werden können. Durch eine Hochrechnung haben wir ermittelt, dass der Gewinn durch den Einsatz von ecoControl in einem Jahr um rund 30 % gesteigert werden kann.

Bei ecoControl handelt es sich um einen Prototypen, der durch eine zentrale Steuerung verschiedener Energiesysteme hilft, Kosten zu sparen. Im nächsten Schritt muss der Prototyp mit echten Geräten getestet werden. Dazu ist eine Hardware-schnittstelle nötig, über die ecoControl mit den Energiesystemen kommunizieren kann. Diese Schnittstelle muss Messdaten erfassen sowie Steuerungsbefehle ausführen können.

ecoControl wurde auf <http://www.sec-i.org/ecocontrol/> unter der MIT-Lizenz veröffentlicht.

Literaturverzeichnis

- [1] P. Andres und M. Spiwox. *State of the Art der statistischen Ex-post-Beurteilung von Prognosen*. Sofia-Studien zur Institutionenanalyse, 2000. ISBN: 3933795222.
- [2] ASUE e.V. *BHKW-Kenndaten 2011*. 2011. URL: <http://asue.de/cms/upload/broschueren/2011/bhkw-kenndaten/asue-bhkw-kenndaten-0311.pdf> (abgerufen am 25.08.2014).
- [3] ASUE e.V. *Klimaschutz durch KWK mit Erdgas*. 2010. URL: http://asue.de/themen/blockheizkraftwerke/grafiken/grafik_05a_2010_008.html (abgerufen am 30.06.2014).
- [4] M. Baranski. *Energie-Monitoring im privaten Haushalt*. Cuvillier Verlag, 2006. ISBN: 3867270201.
- [5] *BHKW Online Forum*. URL: <http://www.bhkw-forum.de/> (abgerufen am 20.06.2014).
- [6] *BHKW-Consult - Ingenieurbüro für BHKW-Planung und KWK*. URL: <http://www.bhkw-consult.de/> (abgerufen am 20.06.2014).
- [7] BHKW-Infothek. *Planung und Umsetzung, BHKW-Infothek*. URL: <http://www.bhkw-infothek.de/bhkw-informationen/planung-und-umsetzung/> (abgerufen am 20.06.2014).
- [8] Bundesamt für Wirtschaft und Aussenkontrolle. *Kennzahlen zur BAFA-Förderung für Mini-KWK-Anlagen*. 2013. URL: http://www.bafa.de/bafa/de/presse/pressemitteilungen/2013/23_mkwk.html (abgerufen am 14.06.2014).
- [9] Bundesministerium der Justiz und für Verbraucherschutz. *Verordnung über Heizkostenabrechnung in der Fassung der Bekanntmachung vom 5. Oktober 2009*. BGBl. I S.3250. 2009. URL: http://www.gesetze-im-internet.de/heizkostenv/_9_.html (abgerufen am 25.08.2014).
- [10] H.-J. Bungartz, S. Zimmer, M. Buchholz und D. Pflüger. *Modeling and Simulation*. Springer Undergraduate Texts in Mathematics and Technology. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. ISBN: 9783642395239.
- [11] A. Cooper, R. Reimann und D. Cronin. *About face: Interface- und Interaction-Design*. Hüthig Jehle Rehm, 2010, Seite 541. ISBN: 3826658884.
- [12] Department of Defense, USA. *A Practitioner's Perspective on Simulation Validation*. 2001.
- [13] K.-P. Dielman und I. Kuperjans. *Kennzahlen typischer KWK-Anlagen*. 2009.
- [14] DigitalGov. *10 Tips for Better Hallway Usability Testing, DigitalGov*. URL: <http://www.digitalgov.gov/2014/02/19/10-tips-for-better-hallway-usability-testing/> (abgerufen am 20.06.2014).

- [15] M. Gailfuss und E. Gailfuss. *BHKW-Planung*. URL: http://www.bhkw-planung.de/bhkw-planung_planung.html (abgerufen am 21.06.2014).
- [16] T. Gobmaier, W. Mauch, M. Beer, S. von Roon, T. Schmid, T. Mezger, J. Habermann und S. Hohlenburger. *Simulationsgestützte Prognose des elektrischen Lastverhaltens*. Bericht StMWFK-002 KW21 BY 3E. Forschungsstelle für Energiewirtschaft e.V., 2012.
- [17] E. Godman, M. Kuniavsky und A. Moed. *Observing the User Experience*. Second Edi. Morgan Kaufmann, 2013.
- [18] Hamburg Gas Consult. „BHKW-Check“. In: *Unternehmen für Ressourcenschutz* 2 (2011).
- [19] W. Hess. „Das leistungsfähigste Kraftwerk der Welt“. In: *bild der wissenschaft* (2011).
- [20] R. Hierzinger und A. J. Scott. „European Smart Metering Landscape Report 2012 – update May 2013“. In: *SmartRegions Deliverable* (2013).
- [21] *Hintergrundinformationen zum deutschen Strommarkt*. 2013. URL: https://www.lichtblick.de/pdf/strom/info/der_deutsche_strommarkt.pdf (abgerufen am 25.08.2014).
- [22] M. Hughes. *Wizards Versus Forms :: UXmatters*. 2011. URL: <http://www.uxmatters.com/mt/archives/2011/09/wizards-versus-forms.php> (abgerufen am 10.06.2014).
- [23] R. Hyndman und G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2012. ISBN: 0987507109.
- [24] Institut für Technische Gebäudeausrüstung Dresden. *Berechnung der Energiekosteneinsparung beim Einsatz eines Mikro-KWK*. 2011.
- [25] P. Jones und P. Carberry. „A technique to develop and validate simulation models“. In: *Agricultural Systems* 46.4 (Jan. 1994), Seiten 427–442. ISSN: 0308521X.
- [26] K. Kobayashi und M. U. Salam. „Comparing Simulated and Measured Values Using Mean Squared Deviation and its Components“. In: *Agron. J.* 92 (2000), Seiten 345–352.
- [27] R. Lawton. „On the Stability of the Double Seasonal Holt-Winters Method“. 2009.
- [28] P. von der Lippe. *Deskriptive Statistik*. Oldenbourg, 2006. ISBN: 9783486578638.
- [29] T. G. Loga und M. Knissel. „Auswirkung des Nutzerverhaltens auf die Heizkosten – Konsequenzen für die verbrauchsabhängige Abrechnung“. In: *Eine Untersuchung im Auftrag der Viterra Energy Services AG, Essen*. IWU. Darmstadt (2003).
- [30] Louis-F. Stahl. *BHKW-Infothek*. März 2014. URL: <http://www.bhkw-infothek.de/bhkw-informationen/einleitung/> (abgerufen am 25.08.2014).
- [31] H. Mühlbacher. „Verbrauchsverhalten von Wärmeerzeugern bei dynamisch variierten Lasten und Übertragungskomponenten“. Dissertation. Technische Universität München, 2007.

- [32] K. J. Müller. „Gewinnung von Verhaltensprofilen am intelligenten Stromzähler“. In: *Datenschutz und Datensicherheit - DuD* 34 (2010), Seiten 359–364. ISSN: 1614-0702.
- [33] *Planungshandbuch und Funktionsbeschreibung XRGI 15-TO*. Hinnerup: EC Power.
- [34] Y. Rogers, H. Sharp und J. Preece. *Interaction Design: Beyond Human - Computer Interaction*. John Wiley & Sons, 2002, Seite 585. ISBN: 0470665769.
- [35] C. Rohrer. *When to Use Which User Experience Research Methods*. NN/g Nielsen Norman Group.
- [36] D. Ryder-Cook. *Thermal Modelling of Buildings*. Technischer Bericht. Cavendish Laboratory, The Inference Group, 2009, Seiten 1–23.
- [37] R. G. Sargent. „Verification and Validation of Simulation Models“. In: *Winter Simulation Conference*. Herausgegeben von S. Jain, R. Creasey, J. Himmelpach, K. White und M. Fu. University Syracuse, NY, 2011, Seiten 183–198. ISBN: 9781457721090.
- [38] R. G. Sargent. „Verification and validation of simulation models“. In: *Journal of Simulation* 7 (2013), Seite 13.
- [39] T. Scanlon. *When to Develop a Wizard*. 1997.
- [40] K. Schwaber und M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002. ISBN: 0130676349.
- [41] A. Sears und J. A. Jacko. *Human-Computer Interaction: Development Process*. CRC Press, 2009, Seite 356. ISBN: 1420088890.
- [42] C. Snyder. *Snyder Consulting - Paper Prototyping*. URL: http://www.snyderconsulting.net/article_paperprototyping.htm (abgerufen am 10.06.2014).
- [43] E. Sprenger und H. Recknagel. *Heizflächenarten*. Wolfenbüttel, 2007.
- [44] Statistisches Bundesamt. *Wirtschaftsbereiche - Energie - Erzeugung*. 2014. URL: <https://www.destatis.de/DE/ZahlenFakten/Wirtschaftsbereiche/Energie/Erzeugung/Tabellen/BilanzElektrizitaetsversorgung.html> (abgerufen am 29.06.2014).
- [45] M. Sturm. *Simulation einer Gebäudeheizung*. Forschungsbericht FKI-227-98. Technische Universität München, 1998.
- [46] J. W. Taylor. „Short-Term Electricity Demand Forecasting Using Double Seasonal Exponential Smoothing“. In: *The Journal of the Operational Research Society* 44 (2003).
- [47] P. Tzscheutschler. „Optimierung des Betriebs von kleinen KWK-Systemen durch Wärmespeicher“. In: *8. Fachtagung Mini- und Mikro-Kraft-Wärme-Kopplung*. 2014.
- [48] Vaillant GmbH. *Kraftwerke im Keller: die Strom erzeugende Heizung*. 2014.
- [49] B. Victor. *Magic Ink: Information Software and the Graphical Interface*. 2006.

- [50] D. Wallach und B. Goffinet. „Mean squared error of prediction as a criterion for evaluating and comparing system models“. In: *Ecological Modelling* 44.3-4 (Jan. 1989), Seiten 299–306. ISSN: 03043800.
- [51] J. Weber und M. Herrmann. *Öfen und Kamine: Raumheizungen fachgerecht planen und bauen*. Beuth Verlag, 2011. ISBN: 9783410213079.
- [52] U. Wehmhörner. „Multikriterielle Regelung mit temperaturbasierter Speicherzustandsbestimmung für Mini-KWK-Anlagen“. Dissertation. Universität München, 2012.
- [53] R. Wengenmayr. *Wie gut sind Wettervorhersagen? Qualitätsprüfung beim DWD*. 2009. URL: http://www.dwd.de/bvbw/generator/DWDWWW/Content/0effentlichkeit/KU/KUPK/Wir__ueber__uns/Broschueren/pdf/Wie_gut_sind__Wettervorhersagen,templateId=raw,property=publicationFile.pdf/Wie_gut_sind_Wettervorhersagen.pdf (abgerufen am 25.08.2014).
- [54] C. Zhu, R. H. Byrd, P. Lu und J. Nocedal. „Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-scale Bound-constrained Optimization“. In: *ACM Trans. Math. Softw.* 23 (1997), Seiten 550–560. ISSN: 0098-3500. DOI: 10.1145/279232.279236.

Aktuelle Technische Berichte des Hasso-Plattner-Instituts

Band	ISBN	Titel	Autoren / Redaktion
92	978-3-86956-317-6	N/A	N/A
91	978-3-86956-303-9	Weak conformance between process models and synchronized object life cycles	Andreas Meyer, Mathias Weske
90	978-3-86956-296-4	Embedded Operating System Projects	Uwe Hentschel, Daniel Richter, Andreas Polze
89	978-3-86956-291-9	openHPI: 哈索·普拉特纳研究院的 MOOC (大规模公开在线课) 计划	Christoph Meinel, Christian Willems
88	978-3-86956-282-7	HPI Future SOC Lab : Proceedings 2013	Christoph Meinel, Andreas Polze, Gerhard Oswald, Rolf Strotmann, Ulrich Seibold, Bernhard Schulzki (Hrsg.)
87	978-3-86956-281-0	Cloud Security Mechanisms	Christian Neuhaus, Andreas Polze (Hrsg.)
86	978-3-86956-280-3	Batch Regions	Luise Pufahl, Andreas Meyer, Mathias Weske
85	978-3-86956-276-6	HPI Future SOC Lab: Proceedings 2012	Christoph Meinel, Andreas Polze, Gerhard Oswald, Rolf Strotmann, Ulrich Seibold, Bernhard Schulzki (Hrsg.)
84	978-3-86956-274-2	Anbieter von Cloud Speicherdiensten im Überblick	Christoph Meinel, Maxim Schnjakin, Tobias Metzke, Markus Freitag
83	978-3-86956-273-5	Proceedings of the 7th Ph.D. Retreat of the HPI Research School on Service-oriented Systems Engineering	Christoph Meinel, Hasso Plattner, Jürgen Döllner, Mathias Weske, Andreas Polze, Robert Hirschfeld, Felix Naumann, Holger Giese, Patrick Baudisch (Hrsg.)
82	978-3-86956-266-7	Extending a Java Virtual Machine to Dynamic Object-oriented Languages	Tobias Pape, Arian Treffer, Robert Hirschfeld
81	978-3-86956-265-0	Babelsberg: Specifying and Solving Constraints on Object Behavior	Tim Felgentreff, Alan Borning, Robert Hirschfeld
80	978-3-86956-264-3	openHPI: The MOOC Offer at Hasso Plattner Institute	Christoph Meinel, Christian Willems

ISBN 978-3-86956-318-3
ISSN 1613-5652