DIPLOMA THESIS

Energy Feedback and Electrical Load Identification

Submitted at the Faculty of Electrical Engineering and Information Technology,
Vienna University of Technology
in partial fulfillment of the requirements for the degree of
Diplom-Ingenieur (equals Master of Sciences)

under supervision of

O. Univ. Prof. Dipl.-Ing. Dr.techn. Dietmar Dietrich
Dipl.-Ing. Dr.techn. Friederich Kupzog

Institute of Computer Technology
Vienna University of Technology

by

Michael Rathmair, BSc
Matr.Nr. 0426124
Obere Augartenstraße 18/7/14, 1020 Wien

October 2011

Abstract

Today an individual power measurement and management of appliances get more and more popular. Especially for home automation and energy information feedback applications. On the other side the measurement of the aggregated power consumption of a household (smart metering) is a commonly used measurement method for customer billing aspects. The objective of this diploma thesis is to create a connection between smart metering and nonintrusive monitoring and detection of individual electrical loads. A significant task of this work is recording load profiles of different home appliances and aggregated load profiles of a household. A main challenge is to identify the individual measured devices out of an overlapped aggregated load signal. With the results of load profile analyzations an algorithm for an automatic detection of loads is designed. Different identification methods and sub-algorithms are collected in a object oriented load identification framework. So each load type can be identified at a maximum probability with the best qualified detection method. All information of the smart meter and load detection results are presented at an appropriate graphical user interface. The illustration of the current electrical consumption and activated devices in a household should be a motivation for customers using the available electrical energy in a more efficient way. This reduces the energy costs, flattens demand load profiles and finally increases the effectiveness in distribution and storage of energy.

Kurzfassung

Eine individuelle Messung und das Management von Einzellasten in einem Haushalt wird immer populärer. Besonders, wenn diese Messung mit Systemen für Heimautomatisierung und Energiefeedback in Verbindung gebracht wird. In jedem Haushalt erfolgt die Messung des Gesamtverbrauchs über einen Stromzähler, welcher von den Energieversorgern für die Abrechnung der Energiekosen verwendet wird. In modernen neuen Anlagen kommen vollelektronische Stromzähler, sogenannte Smart Meter, zum Einsatz. Die Hauptaufgabe in dieser Diplomarbeit besteht darin, die Einzel- mit der Summenverbrauchsmessung zu vereinen und die einzelnen Geräte aus dem Summenlastprofil einflussfrei zu identifizieren. Eine wesentliche Aufgabe des Projektes, ist die Aufzeichnung von Lastprofilen einzelner Geräte sowie des Summenlastprofils eines Haushaltes. Anschließend wird versucht aus dem gemessenen, sich überlappenden Summenverlauf die einzelnen Geräte zu identifizieren. Die Ergebnisse dieser Analyse sollen für den Entwurf eines automatisierten Lastidentifikationsalgorithmus herangezogen werden. Das Design besteht darin, dass verschiedenste Sub-Algorithmen in einer objektorientierten Struktur zusammengefasst werden. So kann jedes Gerät mit einer maximalen Detektionswahrscheinlichkeit und der am besten geeigneten Methode identifiziert werden. Alle Verbrauchsdaten des Smart Meter sowie die Ergebnisse der Lastidentifikation werden auf einer grafischen Benutzeroberfläche angezeigt. Die Präsentation der Verbrauchsdaten soll Kunden motivieren die verfügbare elektrische Energie effizienter zu nutzen. Dies reduziert die Stromkosten, führt zum Abflachen von Spitzenlasten und erhöht die Effektivität in der Verteilung und Speicherung von elektrischer Energie.

Acknowledgements

After having finished my diploma thesis, it is time to say thank you to all people that encouraged me during the course of this project. It was a very intense and interesting time and I profited very much by the experience I got from this scientific work. My special thanks go to my supervisors Prof. Dietrich and Dr. Kupzog who gave helpful suggestions for problem solutions, discussions and guided me during the writing of this thesis. Finally I would like to thank my parents, my family and all friends for their constant support and motivation.

Table of Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Scope . . . 3

2 Load Monitoring **5**
2.1 Line Power Measurement . . . 5
2.2 Implementation Methods for Power Meters . . . 7
2.2.1 Single Chip and Distributed Approaches . . . 7
2.2.2 State of the art Power Measurement Solutions . . . 9

3 Smart Metering **11**
3.1 Technical Structure and Motivation for Smart Metering . . . 11
3.1.1 Costs . . . 12
3.1.2 Data Management . . . 13
3.1.3 Smart Metering as a Nucleus for new Technologies . . . 13
3.1.4 New Smart Metering Tariff Schemes . . . 14
3.1.5 Smart Metering Structure . . . 15
3.1.6 Visions of Smart Metering . . . 16
3.2 Automatic Metering Information System . . . 16
3.3 ABB Smart Metering Solution . . . 22
3.4 Easy Meter and NZR - Metering Solution . . . 23

4 Energy Feedback Information **26**
4.1 State of the art Energy Feedback Systems for Home Applications . . . 27
4.1.1 Direct Energy Feedback Methods . . . 28
4.1.2 Feedback Methods used in this Work . . . 29
4.1.3 Visualization System from Bticino . . . 29
4.2 Hybrid System Architecture . . . 31
4.3 Impact of Energy Feedback on Energy Efficiency . . . 32

5 Electrical Load Identification **35**
5.1 Hardware based approach . . . 35
5.2 Software Algorithms . . . 36
5.2.1 Edge-Based Approach . . . 36
5.2.2 Model-Based Approach . . . 37

5.2.3 Pattern Recognition and Dynamic Time Warping 39
5.3 Analysis of Typical Aggregated Load Profiles 40
5.4 Load Profile Analysis of Typical Home Appliances 46
5.5 Conclusions for the Implementation 47

6 System Architecture and Implementation **50**
6.1 Requirements on the System 50
6.2 Overview and Architecture 51
6.3 Aggregated Load Profile Measurement 52
6.3.1 Functional Requirements 53
6.3.2 Hardware Overview and Setup 53
6.3.3 Load Profile Measurement Software 54
6.4 Load Database Structure and Access 57
6.4.1 File Structure 58
6.4.2 Template Modeling and Storage 58
6.4.3 File Access 59
6.5 Object oriented Load Identification Functionality 60
6.5.1 Requirements on the Identification System 60
6.5.2 Common Algorithm Implementation 61
6.5.3 Edge Detection and Parameter Matching Algorithm 64
6.5.4 Detection of Real and Reactive Power Characteristics 69
6.6 User Interface 72
6.6.1 Interface Requirements 72
6.6.2 Tools and Implementation 73

7 Results and Discussion **77**
7.1 Energy Feedback Results 77
7.2 Load Identification Results 77

8 Conclusion and Future Work **86**

9 Appendix **88**
9.1 Load Profiles of Home Appliances 88
9.2 Program Listings and Graphs 93

Literature **101**

Internet References **104**

Abbreviations

A/D	Analog to Digital
AC	Alternating Current
ADC	Analog to Digital Converter
AMIS	Automatic Metering Information System or german: Automatisches Metering- und Informations-System
AMM	Automatic Meter Management
AMR	Automated Meter Reading
CPP	Critical Peak Pricing
CTS	Clear to Send
DC	Direct Current
DLC	Distribution Line Communication
DSL	Digital Subscriber Line
DSP	Digital Signal Processor
DTW	Dynamic Time Warping
eHZ	ger. Elekronischer Hauhaltszähler - ABB Smart Meter
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GTK	GNU Image Manipulation Program Toolkit
GUI	Graphical User Interface
HMM	Hidden Markov Models
IC	Integrated Circuit
ID	Identifier
IEC	International Electrotechnical Commission
IP	Internet Protocol (RFC 791 and RFC 2460)
LED	Light Emitting Diode
MUC	Multi Utility Communication
NZR	Nordwestdeutsche Zählerrevision (Company Abbreviation)
OBIS	Object Identification System
PC	Personal Computer
PLC	Power Line Carrier
POSIX	Portable Operating System Interface for Unix
RFC	Request for Comments
RMS	Root Mean Square
RTP	Real-time Pricing
RTS	Ready to Send
TV	Television
ToU	Time-of-use Pricing
UTF-8	8-bit Universal Character Set Transformation Format
XML	Extensible Markup Language

Symbols

This table will hold all symbols used in mathematical expressions.

Symbol	Explanation	Unit
t	time	s
ω	angular frequency	s^{-1}
φ_v	voltage phase angle	rad
φ_i	current phase angle	rad
φ	phase difference between voltage and current	rad
j	imaginary unit	$\sqrt{-1}$
T	periodic time	s
v	voltage instantaneous value	V
i	current instantaneous value	A
V_{RMS}	voltage RMS (root mean square) value	V
I_{RMS}	current RMS value	A
P	real power	W
Q	reactive power	VAR
S	apparent power	VA
λ	power factor	1
$p_{i,j}$	probability for transmission from state i to j in a HMM (hidden Markov Model)	1
$T_{i,j}$	number of transmissions from state i to j at the HMM learning phase	1
$T_{i,x}$	number of transmissions from state i to any state in the HMM learning phase	1
τ_{env}	environment temperature	K
$r_{detected}$	detection rate	%
$n_{detected}$	number of successful load detections	1
$n_{activated}$	number of load activations	1
D	aggregated smart meter information set	none
E	smart meter energy information	kWh
$L1, L2, L3$	abbreviation for line 1, 2, 3	none
I_{L1}, I_{L2}, I_{L3}, I_N	currents in line 1,2,3 and neutral wire	A
V_{L1}, V_{L2}, V_{L3}	voltages of line 1,2 and 3	V
φ_1, φ_2, φ_3	angles between voltage and current in line 1,2 and 3	rad

Symbol	Explanation	Unit
τ	time-stamp	s
f	flag information	none
N_d	number of detectable loads	1
N_i	number of individually measured loads	1
ΔP	change of real power at one step in time	W
ΔQ	change of reactive power at one step in time	VAR
N	number of watcher objects	1
M	number of observer objects	1
$\{D_t\}$	time series of D sets	none
$\{P_t\}$	time series of measured real power values	W
$\{E_t\}$	time series of measured energy values	kWh
$\{I_{L1,t}\}$, $\{I_{L2,t}\}$, $\{I_{L3,t}\}$, $\{I_{N,t}\}$	time series of measured currents in line 1, 2, 3 and in the neutral conductor	A
$\{V_{L1,t}\}$, $\{V_{L2,t}\}$, $\{V_{L3,t}\}$	time series of measured voltages at line 1, 2, 3	V
$\{\varphi_{L1,t}\}$, $\{\varphi_{L2,t}\}$, $\{\varphi_{L3,t}\}$	time series of measured angles at line 1, 2, 3	rad
$\{\tau_t\}$	time series of time stamps	s
$\{f_t\}$	time series of flag register information	none
$P_{t,j}$	real power value of a device indexed by j at time t	W
$E_{t,j}$	energy value of a device indexed by j at time t	kWh
$I_{L1,t,j}$, $I_{L2,t,j}$, $I_{L3,t,j}$, $I_{N,t,j}$	current values of line 1, 2, 3 and the neutral conductor of a device j at time t	A
$V_{L1,t,j}$, $V_{L2,t,j}$, $V_{L3,t,j}$	voltages of line 1, 2, 3 of a device j at time t	V
$\varphi_{L1,t,j}$, $\varphi_{L2,t,j}$, $\varphi_{L3,t,j}$	angles of line 1, 2, 3 of a device j at time t	rad
$P_{t.i}$	real power value of an individual measured device	W
P_{rest}	real power value of not detectable devices	W
$I_{L1,t,i}$, $I_{L2,t,i}$, $I_{L3,t,i}$, $I_{N,t,i}$	current values of line 1, 2, 3 and the neutral conductor of a individual measured device indexed by i at time t	A
I_{rest}	current value of not detectable devices	A
δ	threshold value for the creation of a new system event	W, A, rad
R_k	rising edge event indexed by k	none
F_k	falling edge event indexed by k	none

Symbol	Explanation	Unit
id	event identifier	none
d	delta value of a sampled event	W, A, rad
s	source information of an event	none
ψ	set of events	none
$d_{template}$	delta value of the template event	W, A, rad
p_{match}	matching probability of a sample	1 or %
ϵ	reference value of the Gauss probability function	1
β	width parameter of the Gauss probability function	1
ΔP_{sample}	sampled change of real power	W
ΔQ_{sample}	sampled change of reactive power	VAR
$\Delta P_{template}$	template change of real power	W
$\Delta Q_{template}$	template change of reactive power	VAR
$p_{\Delta P}$	probability value for ΔP matching	1 or %
$p_{\Delta Q}$	probability value for ΔQ matching	1 or %
p_{result}	result probability for the ΔP - ΔQ matching algorithm	1 or %
$\hat{P}_{noise}$	maximum real power noise signal amplitude	W
$\hat{I}_{noise}$	maximum current noise signal amplitude	A
$\hat{\varphi}_{noise}$	maximum angle noise signal amplitude	rad
f_{noise}	noise factor	1
$random()$	random function - returns values between 0 and 1	1
P_{sample}	real power sample value with overlapped noise signal	W
I_{sample}	current sample value with overlapped noise signal	A
φ_{sample}	angle sample value with overlapped noise signal	rad

1 Introduction

Advances in computational technologies combined with state of the art energy systems (combination of energy and information technology (IT)) will become a key technology for future systems of electrical energy distribution. Today the development of micro-electronics and communication equipment, for informing customers of their current energy consumption and promoting actions to use the available energy in a more effective way is no further problem [WN06]. Embedded hardware components combined with algorithms integrated in high functional software modules enables the realization of a smart system for energy feedback, to users and power authorities. The identification and forecast of activated electric loads, effective integration and balance of demand and supplying components can be achieved.

1.1 Motivation

The supply of electrical energy at competitive prices, in sufficient quality and quantity and under the constraints of uninterrupted service is an activity that is difficult to manage, for electricity suppliers [SR08]. Rising consumption and increasing peak load periods requires effective energy feedback methods to counteract against these trends. Therefore an informational flow of power grid information and customized communication data between all active and passive nodes in the grid must be applied. This parallel architecture (power supply and informational flow) shown in Figure 1.1 is called a Smart Grid. The combination of power supply and communication between grid nodes is a inseparable unit for modern power applications. The challenge is to reach a well balanced distribution of loads, grid stability and high supply quality anywhere and anytime in a extremely decentralized network of suppliers, consumers and storages. This can be achieved with the help of modern computer technology.

At the customers side this continuous flow of information can also bring a gain in energy efficiency and change the users behavior in energy consumption. New fully electronic metering devices installed in the households (smart meters) offers on time information about the actual electric power consumption. This informational feedback illustrated in Figure 1.1 is also of special interest for energy customers. Due to the knowledge about the actual consumption, customers should be motivated to use electrical energy in a more effective and efficient way. Users should power off devices if they are not used, reduce the standby power level and even know exactly how much energy appliances consume. The feedback can be seen as a customers learning tool allowing energy users to teach themselves through experimentation [Dar06]. Exactly this positive customer effects will help power authorities for an easier reaching of load balancing requirements and decrease the energy use. In other words the feedback of the consumption of appliances in a households is a key to solve the challenge of reducing the customers energy level and a more precise load forecast for energy suppliers.

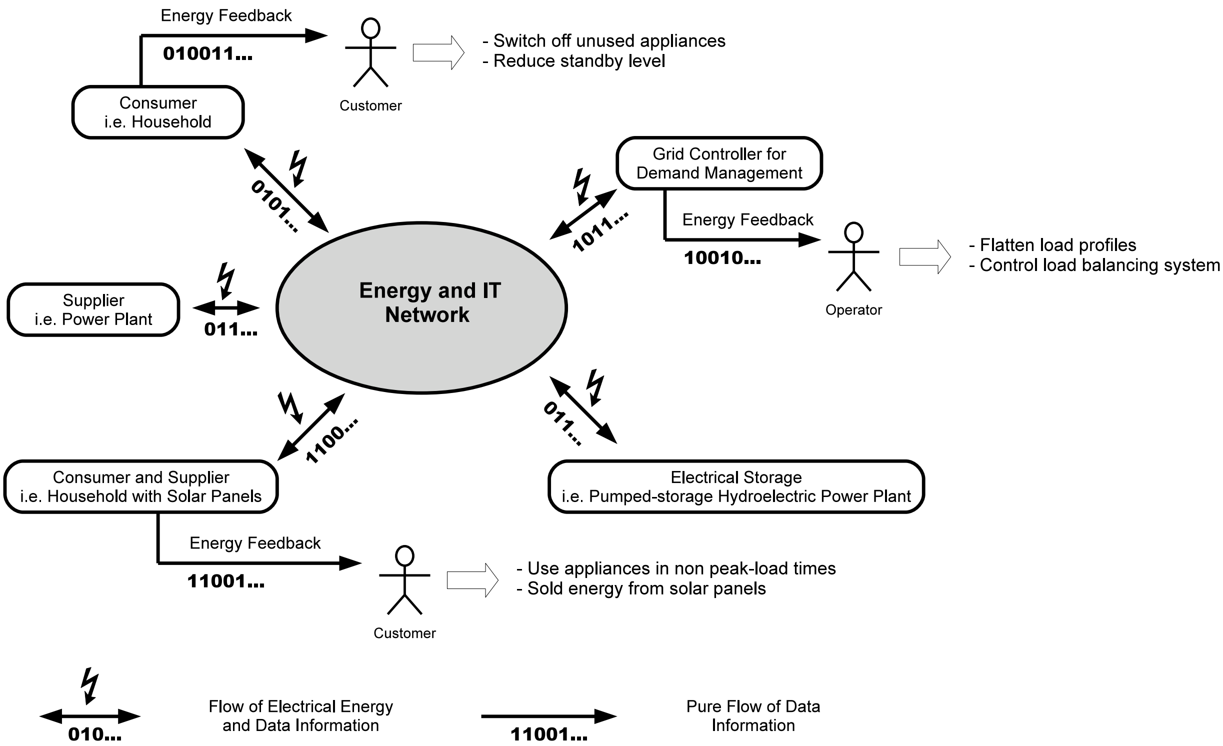

Figure 1.1: Schematically structure of a smart grid architecture with direct customer and energy supplier feedback. Source: Author

But the aspect of energy information feedback does not end in a simple presentation of the actual energy consumption. There can be offered much more different informational and control services around the core aspect of feedback. At the side of energy suppliers new flexible tariff schemes can be offered. With the help of smart metering these customer specific schemes can be selected, updated and easily managed direct at the metering device. This so called time-of-use tariffs enable more flexibility in pricing, because customers buy the available energy at different costs distributed over a day [DD09] [Zia10, p. 2]. This flattens demand load profiles, increases the effectiveness in distribution and storage of energy and enables suppliers running their power plants at optimal operating points.

The embedded electronics of smart meters implements standardized interfaces for reading electrical on time data information like currents, voltages, real power, reactive power and angle information. These acquired datapoints can be presented to the user, and can be a basis for underlying control algorithms. Smart meters can be connected to full integrated automation systems and forward the received grid data information. With this direct informational link between energy supplier and the customers devices, appliances could be turned on or off depending on the grid utilization, energy price or even their significance. This requires the ability of a direct supply cut off. This functionality could be realized with modules direct in an appliance or as a socket module. A data link of the unit to all available grid information and control signals must be established. Such supply cut off units can also act as a distributed sensor nodes and feed back information about the device state and its environment. Power consumption, currents, voltage levels, etc. but also non electric data like humidity, temperature, light intensity, etc. depending on the implemented sensors are sent back to a centralized base station. This individual power measurement information can be user presented and fed into algorithms which guides customers to reach a maximum household economic level.

Today no standard appliance available at the market offers the opportunity of being controlled by an external instance and do not provide interfaces for customized data feedback. That is why external pre connected modules are the common way of use (external power measurement adapters or stationary installed measurement modules). The trend is to use such external socket modules only for non mobile and significant loads (i.e. fridge, washing machine, heating devices, etc). For power analyzation of all other and locally not fixed appliances nonintrusive load detection algorithms can be used. The strength of such algorithms is that the detection of activated appliances can be realized by analyzing summary power profiles delivered from a smart meter device [DK99]. With appropriate detection processes (model based, edge detection, pattern recognition, etc.) and a configured and trained load data base precise identification results can be accomplished. As a short summarization the motivation of this work is to combine smart metering, individually power measurement and nonintrusive load identification for developing a smart customer oriented energy information feedback and load detection system.

1.2 Scope

The main goal of this thesis is the design and implementation of a full integrated energy information feedback and load identification system. The architecture includes the measurement of aggregated power consumption with the help of a smart meter and sensor nodes for individually measured appliances. All information are brought to a common database. Smart meter information which includes real power, currents, voltages, angle information and meter reading are optically transferred to a controller (infrared reading head). Therefor the standardized communication protocol IEC-62056 (International Electrotechnical Commission) for smart meters is implemented. At the appliances side for individual load monitoring a controller for power measurement is implemented. Additional to the pure power data acquisition a digital interface temperature sensor collects absolute temperature values from the surrounding environment. For load cut off functionality a solid state relay enables a current zero-cross electric disconnection of the device. All implemented controller nodes are linked together with a wireless communication interface. A base station acts as a master node and collects all measurement data. The wireless sensor network offers a functional interface with a user specified communication protocol for requesting and responding data packets, load cut off commands and network administrative activities. At the user front-end a graphical interface presents all measured and computed data information. The functions, architecture and cooperation of all modules described above is illustrated in Figure 1.2. To maximize all positive aspects of energy feedback the system should be easy to use and installable in every standard smart meter equipped household.

An other challenge of this project is the design and implementation of a load identification software module. The scope is to analyze, compare and eventually mix up different approaches known in the literature. Data basis for the designed algorithm is like mentioned above a full smart metering data set containing power-, current-, voltage- and angle information. Due to a hybrid measurement architecture (aggregated and individually load data acquisition in one system according to Figure 1.2), individually measured load profiles can be subtracted from the aggregated data set. The resulting data points excludes the automatically identified appliances

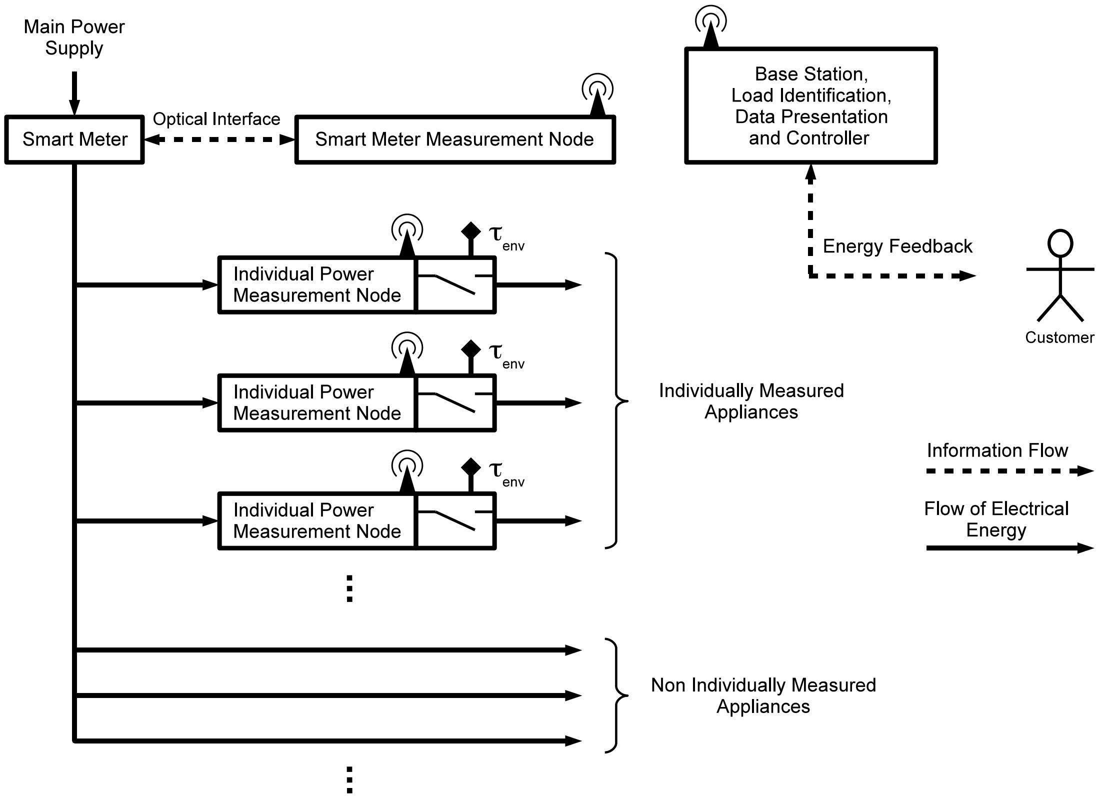

Figure 1.2: Architecture overview of the implemented energy feedback system realized in this work. The individual power measurement nodes are equipped with load cut off units, temperature sensors and radio interfaces. τ_{env} stands for the environment temperature in this illustration. Source: Author

and forms the input stream for the identification software module. Because of large variety in each type of loads, different manufactures, brands and technological innovations it will not be possible to realize a perfect all detecting algorithm [Zia10, p. 4]. However the implemented model provides a high flexibility object framework with a user specified load data base. An individual implementation of different detection algorithms and new additional rule sets into the model can be considerably achieved. All algorithms and rules will continuously be executed in a parallel way. The output of the load detection model are events including an unique load identifier and a probability value for its occurrence. As a future aspect with the help of this computed and real measured information intelligent virtual energy efficiency guides or agents can be implemented in the feedback system. The potentials of saving energy and reducing costs due to nonintrusive load monitoring are present in every household. Finally the driving key factor for the development of a nonintrusive load detection algorithm in this work, is to retrieve the activation of appliances out of an aggregated smart metering load profile.

2 Load Monitoring

This chapter will give an overview over the theory and implementation of load monitoring systems. All residential homes are normally equipped with only one main electrical meter. But a lot of energy saving strategies need real time energy consumption data of each of the main appliances [BXZ09]. One of the most expensive, but maximum reliable way to get individual energy consumption data is to attach a measurement node in each power supply unit (i.e. each socket of a household has an own measurement unit). On the other side with the help of nonintrusive load monitoring algorithms the individual active loads could found out of a summary measured load profile [BXZ09].

2.1 Line Power Measurement

An alternating current (AC) line power meter is used to measure the power consumption of an equipment which is drawn from 115/230 V, 50/60 Hz power line. This is the standard case for home applications. For energy control and optimization processes a precise current, power, and power factor measurement is the first step [25]. The following equations are only valid for sinusoidal signals which time function can be stated as Equation 2.1 [Sch98, p. 183].

$$x(t) = \hat{x} \cdot sin(\omega \cdot t + \varphi) \tag{2.1}$$

$x(t)$ can stand for current or voltage values in time domain. Due to linearity of the system, both will be sinusoidal at the the same frequency. Generally a home appliance load has an inductive or capacitive part (complex load). This will result in a phase difference between voltage and current signal. For power measurement root mean square (RMS) values of the electrical parts are required. They can be calculated with Equation 2.2 [25] and 2.3 [25] of the actual signal levels.

$$V_{RMS} = \frac{\hat{v}}{\sqrt{2}} = \sqrt{\frac{1}{T}\int v^2 dt} \tag{2.2}$$

$$I_{RMS} = \frac{\hat{i}}{\sqrt{2}} = \sqrt{\frac{1}{T}\int i^2 dt} \tag{2.3}$$

In an AC appliance with a complex impedance (capacitive or inductive part) apparent power can be calculated with the help of RMS values. Like stated in Equation 2.4 [Sch98, p. 183] the apparent power of an device is simply given by the product of RMS current and voltage [Sch98, p. 184].

$$S = V_{RMS} \cdot I_{RMS} \tag{2.4}$$

Because of the phase difference between voltage and current (inductive: $0° < \varphi \leq 90°$, capacitive: $-90° \leq \varphi < 0°$ due to the realtion $\varphi = \varphi_v - \varphi_i$) apparent power is a vector sum of real and reactive power (see Figure 2.1). The real part of this vector is called real power (see equation 2.5 [Sch98, p. 184]). This fraction of the vector sum can be converted to another useful form of energy (i.e. heat, light, rotation) [Sch98, p. 184].

$$P = V_{RMS} \cdot I_{RMS} \cdot cos(\varphi) \tag{2.5}$$

The imaginary part of the complex power vector is called reactive power (see Equation 2.6 [Sch98, p. 185]). In AC circuits energy is stored temporarily in inductive and capacitive elements, which can result in the periodic reversal direction of energy flow. This causes an reactive power part [Sch98, p. 185].

$$Q = V_{RMS} \cdot I_{RMS} \cdot sin(\varphi) \tag{2.6}$$

In Figure 2.1, P is the real power, Q is the reactive power (in this case positive), S is the complex power and the length of S is the apparent power. P, Q and S always form a rectangular power triangle. A result of basic equations based on rectangular triangles Equation 2.7 [27] states that the complex power vector $\underline{S}$ is a trigonometric vector sum of P and Q. Apparent power could be calculated with the pythagorean law.

$$\underline{S} = P + jQ \quad \Rightarrow \quad |\underline{S}|^2 = P^2 + Q^2 \quad \text{and} \quad \varphi = arctan\left(\frac{Q}{P}\right) \tag{2.7}$$

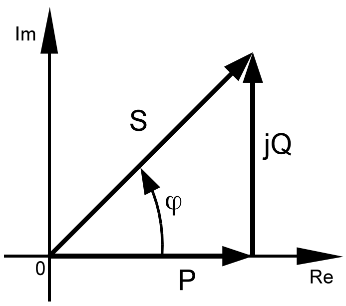

Figure 2.1: Complex power vector with real and reactive components. Source: [27]

The power factor λ of an AC electric power system is defined as the ratio of the real power consumed by the load to apparent power (see equation 2.8 [25]). λ is a dimensionless number between 0 and 1 (frequently expressed as a percentage). Circuits containing purely ohmic elements have a power factor of 1.0. Circuits containing inductive or capacitive elements have a power factor < 1.

$$\lambda = \frac{P}{S} = \frac{V_{RMS} \cdot I_{RMS} \cdot cos(\varphi)}{V_{RMS} \cdot I_{RMS}} \tag{2.8}$$

2.2 Implementation Methods for Power Meters

For the implementation of power measurement units several different hardware architectures are in use. Today single chip solutions are the most important and common ones. The different types of sensors and architectures differ in accuracy, cost, manufacturability, reliability and single or poly phase measurement mode.
In terms of accuracy modern power measurement chips have a measurement error of $< 1\,\%$ over their full dynamic range [Mou01, p. 3]. Generally the accuracy should be as low as possible. The accuracy includes two portions, which are reading and ranging, where latter is a constant error factor. Digital converters definitely have much better accuracies than analog converters [25]. Because there are only analog to digital (A/D) conversion errors which are smaller than analog converter's measurement errors [25].
Costs are divided in three different parts: Internal components costs (Accuracy of A/D converters, Sampling speed, internal analog-digital chip design), manufacturing costs (chip size, external connections, hardware test) and costs for maintenance and replacement [Mou01, p. 3]. For calibration [25] says that the main factor for calibration cycle times are: component rating, aging and drift. An analog converter power meter has an calibration cycle of about three months while a digital has one about one year [25].
The point of manufacturability goes in harmony with the costs for manufacturing [Mou01, p. 3]. Key points are: Ease of manufacturing (possibility of manufacturing a chip with the actual setup of the process line), simplicity of factory calibration and yield [Mou01, p. 3].
The requirements in reliability are especially for applications with billing aspects extremely important. For the measured electrical energy consistency of functionalities over time and a stable performance are the key points of an highly reliable measurement setup [Mou01, p. 3].
State of the art power measurement integrated circuits (ICs) are dedicated electricity measurement front-end that collects and calculates poly- or single phase voltage, current, power, energy, and many other metering and power-quality parameters of a load. The computed results can be retrieved by an external master through an on-chip implemented digital interface [Mou01, p. 3].

2.2.1 Single Chip and Distributed Approaches

As an architectural design of power measurement units single chip and distributed architectures are possible. The choice of the best design depends on the application and flexibility required on the measurement setup.

Analog converters with digital read out: With the help of analog square root, multiply and integration blocks the actual power consumption can be calculated according to Equations 2.2 to 2.5. The RMS converter blocks first creates a direct current (DC) signal which correspond to the root mean square value of the input sinusoidal waveform. At a second step the analog multiplier creates a signal $V_{RMS} \cdot I_{RMS}$ which is equal to the apparent power. A low pass filter eliminates high parasitic frequencies. This output signal will be A/D converted. The output signal is a digital representation of the apparent power and is ready for transmission, presentation on a display or storage in a database. All described components, interfaces and its connections can be seen in Figure 2.2 [25].

Digital converters with digital read out: For a digital calculation of power data it is necessary to sample and digitalize the signals at the front end of the measurement unit. An analog circuit part is only needed for leveling the sensor signals to the full input range of the A/D converters.

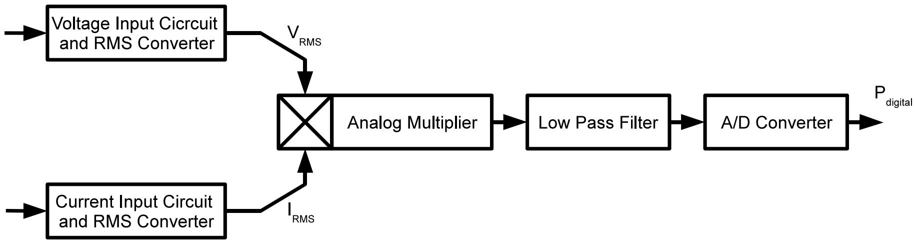

Figure 2.2: Architecture of an analog power meter with digital read out. Source: [25]

In the case of two separated A/D conversion units (like stated in the Block Diagram 2.3) the sampling process must be synchronized. Otherwise a time shifted sampling of current and voltage information, which results in an error in power calculation would be possible. An advantage of this architecture is that the samples of current and voltage are available simultaneously and an update of the power information can occur at each system clock cycle [25]. The digitalized electrical values are computed in a digital signal processor (DSP) or in a microcontroller unit. The first stage of the computation of power data will happen in an optimized digital processing unit (fast execution of mathematical operations). Then the calculated data will be transfered to a Data Management software Module. This module is responsible for the handling of the digital communication interface and for the timing and control of the system. The main advantage of such a distributed multi-module solution is the flexibility of the system. An individually design of each block to get a maximum optimization for the measurement system is possible. Disadvantages of the approach are a higher component count and hand in hand to this fact higher costs for the setup. The design of the analog front-end and the software for the DSP could become very complex [Mou01, p. 10]. Due to this disadvantages and a high flexibility is not required for standard applications single chip solutions become more and more popular.

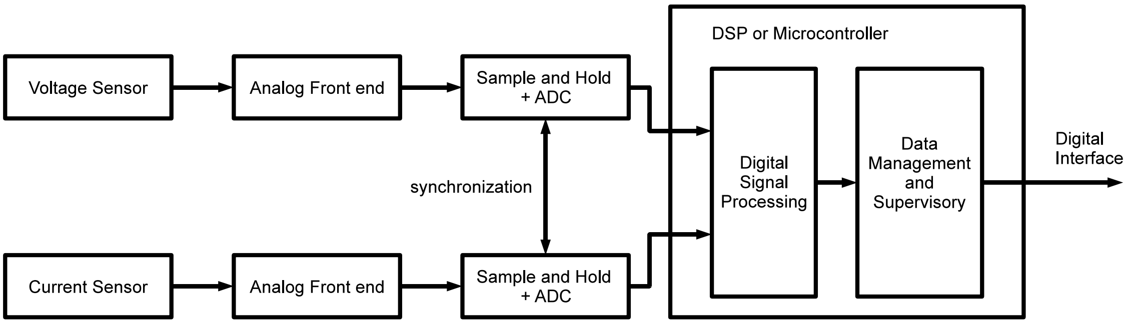

Figure 2.3: Architecture of a digital power meter with digital read out. Source: [Mou01, p. 9]

Single chip power measurement solutions: Single chip solutions are fitted for one specific standard measurement application. The reference designs are always stand-alone solutions without a large number of external components. Input current and voltage sensors (single or three phase) can be directly connected to the chip. The functionality of the blocks is principally the same as described in the last section but integrated as a System on Chip (SoC). The main parts can be seen in figure 2.4. There are a data acquisition unit, digital conversion, signal processing and calculation of power information and digital interface handling. The software of the DSP and microcontroler is burned in an internal read only memory (ROM). Some chips offers an additional programable read only memory (PROM) interface for the setup of calibration data, design of own algorithm implementations and error correction functionality [NF94]. The most important advantages according to [Mou01, p. 13] for a single chip design are:

- Non external analog front end needed. Sensors can be directly connected to the IC.
- Reduced component count for the chip external circuit.
- A ensured linearity better than 1 %.
- Lowest costs for high volume.

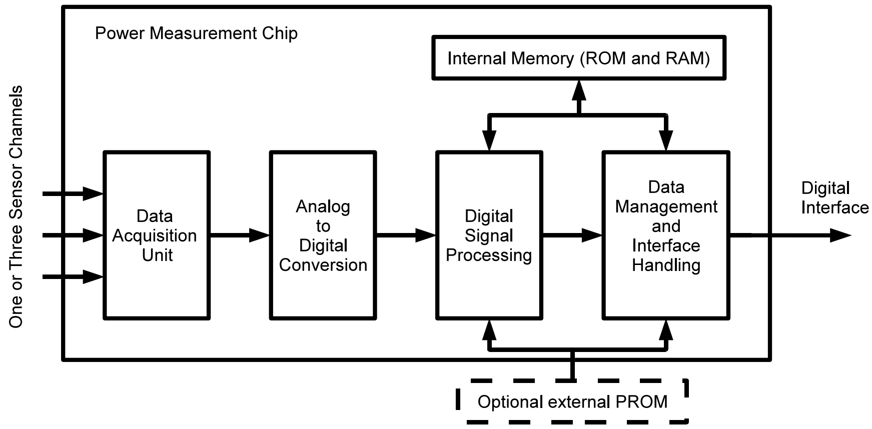

Figure 2.4: Single chip power meter architecture. Source: [Mou01, p. 17]

2.2.2 State of the art Power Measurement Solutions

In this section three state of the art power measurement chips will be compared. The advantages and disadvantages of the usage for individual power measurement of home appliances will be pointed out. Two chips described in this section are from the company Microchip (MCP3901 and MCP3909) and the other one is from Analog Devices (ADE7569).

MCP3901: Is a two channel analog front end IC. It contains two synchronous analog to digital Delta-Sigma converters for the digitalization of the two input signals. So the chip is able to interface a wide variety of standard sensors directly (shunts, current transformers, rogowski coils, etc). The 24-bit wide digital data words can be read out by an 20 MHz serial peripheral interface [9]. Additional functions of the chip are features for internal gaining, oversampling ration configuration, resolution setup, energy saving shutdown, etc [9].

MCP3909: The MCP3909 is a energy metering chip which supports energy measurement according to the international metering standard specification IEC 62053 (International Electrotechnical Commission) [4]. A frequency proportional and a digital serial interface output are available. The input signals of a shunt current sensor and direct voltage measurement are digitalized with 2 synchronized 16-bit Delta-Sigma converters [4]. An internal multiplier unit calculates a 20-bit real power value. Additional functions of the IC are a pulse output for the real power level, internal current and voltage filters, stepper motor output drivers for active power, etc [4].

ADE7569: This IC is a full integrated solution for metering and power measurement. The ADE7596 combines measurement hardware, a digital signal processor, a 8052 architecture microcontroller unit and a liquid crystal display driver in one IC [6]. The two analog inputs are ready to connect any standard current and voltage sensors and have a programmable gain amplifier at its first stage. The analog to digital converted sensor data are computed in a digital signal processor. The output results of the DSP are values for voltage, current, real-, reactive-, apparent power, etc. These data are internally transferred to a 8052 architecture microcontroller core. Power applications can be directly executed on the controller core of the ADE7596. Therefor several peripheral equipment like communication interfaces, flash and extension memories, watchdog timer and a 108-segment liquid crystal display driver are supported. Additional functions of the chip are an internal temperature sensor, power supply management with the ability of battery supply, an internal charge pump for the display connection, etc [6].

These are only three examples out of a pool of power measurement sensor types of different suppliers. The described sensor are single phase sensors. They are also available in three phase types. The chosen sensors above are rated in their supported functions and their complexity. The MCP3901 is only a two channel analog to digital converter with a special environmental setup for power measurement. Only current and voltage signals are digitally offered at a high resolution. Power levels must be calculated in external units. The advantages of the chip are a high resolution and fast delivery of current and voltage data. The MCP3909 has a similar internal structure but with a lower resolution of the digital values. The advantage of this chip is an internal multiplier. So power values are directly available from the sensor chip. But this is only a instantaneous power value. If real, reactive and apparent power levels are required, they must be calculated according to Equations 2.2, 2.3, 2.4, 2.5, 2.6. The ADE7569 handles these calculations in an integrated DSP. An advantage of this chip is that a full integrated microcontroller core is available for implementing user application software. Disadvantage of this sensor are that the internal structure is complex, the microcontroller must be programmed with the help of external programming tools, and if the integration of the user application into the power measurement chip is not required or not possible the microcontroller unit is disruptive and results in higher development costs.

Common properties of all state of the art sensors of all suppliers are a digital output interface for the measurement results and the capability of direct interfacing different current sensor types. As a final statement it could be said that for all different types of standard applications an appropriate single chip power measurement sensor could be found at the market.

3 Smart Metering

The introduction of smart metering is today one of the biggest challenges for energy supplying companies. There is a high suspense between investment costs and efficiency. The company internal structure of energy suppliers must change from pure electrical power supplying business to a provider of different customer depended tariff, energy and metering services. If the deployment of smart meter hardware will be finished in the active region of a power supplier, the pure metering functions could be updated to customer specific smart applications [KS09, p. 130].

Smart metering at the technical focus of electrical energy measurement is the used case scenario of summary power measurement of a household. In the view of the information and communication technology, smart metering is a transfer of consumption data from point A (a meter or sensor device) to a point B (a management or billing system). The metering system in a household gets really "smart" if all important measurement devices transfer their acquired values to a common database station. In addition to electricity and gas also water-, heating data and decentralized energy generation must be considered and integrated in an information network. For the future aspect of a full "Smart Home", automation and smart metering must be joint together and all data have to be presented to the user in a clear and structured way. To handle this hype of information powerful embedded processing systems and all internet protocol (IP) networks will become standard equipment in every modern household [KS09, p. 88].

3.1 Technical Structure and Motivation for Smart Metering

Smart metering is a full automatic and electronic measurement of consumptions in a household. The introduction of smart metering brings a lot of new requirements in technical structures of customers and energy suppliers. Today the real requirements for endusers are not known because smart metering is for them only a new technology without any special usage [KS09, p. 112]. The main tasks for energy supplies are to handle technical and ifrasrtuctional problems within the introduction, and being a customized service partner for the users. Smart metering can motivate customers using the available energy in a more efficient way and finally safe money. Future visions like smart grid applications, electromobility, distributed energy sources, etc. will require smart metering solutions anyway.

3.1.1 Costs

The costs for hard- and software are essential factors for an energy supplier. The most obvious cost factor at the first step is the installation of full electronic meters in each household. All meters must be connected to a higher instance data management network (data concentrators). This will result in additional costs for telecommunication topology or power-line communication techniques. First experiments in non-urban regions showed that point to point connections will be required, due to high distances to the next concentrator station. This will also result in higher costs because conventional General Packet Radio Service (GPRS) connections are used. A large part of summary costs is like mentioned above the installation of smart meters. First pilot projects showed, that an early integration of the installation and maintenance personal into the technology process, results in high efficiency in costs and much more important in a huge increase of technical competences [KS09, p. 19]. Figure 3.1 will give an coarse overview of the average final resulting costs for the introduction of smart metering in one household. The main part of costs is the hardware of the device (measurement electronic parts, display, power electronic components, communication interfaces). A second big part of the costs is the installation (mechanical installation of the device, meter an communication configuration). Small portions of concentrator costs and integration of the meter into the communication system will also fall to the customers. The summarization of all costs will result in 90 to 170 euros per installed smart meter device.

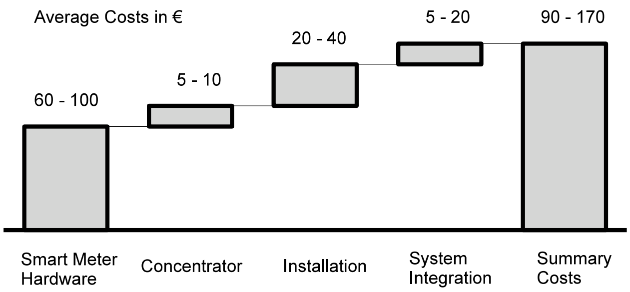

Figure 3.1: Average investment costs per smart meter in euros. Source: [KS09, p. 19]

A modern software system for the operation of a smart metering environment requires a high performance implementation of measurement- and database algorithms. The costs for the smart metering software system must not be underestimated and are a huge factor in the planing of a metering project. At an average calculation of 0.6 to 0.9 cent per month and customer the summarization of software costs can reach at an system size of 100000 meters one million euros per year and more [KS09, p. 23]. The main properties of the software package for smart metering applications are:

- High performance and scalability: At a full integrated smart metering network a large number of devices must be managed. Adding new meters or customer specific reconfigurations must be easy to perform and available any time. For effective process control and up to date billing information real time measurement data will be required.
- Integration of billing functionality: The metering software package is a standalone software module. But it is strongly connected to a companies enterprise resource planning (ERP) system which offers functions for automatic billing calculation and debt collection.

- High degree of automation: The system includes routines for full automatic post capture of metering data, estimation of consummations and a effective detection of communication errors [KS09, p. 24].

In addition to the costs of soft-, hardware and installation operating costs must be taken into consideration. A ferraris three phase AC meter has approximately a three times higher energy consumption than a modern smart meter. The smart meter has extra operation costs for the communication (GSM, GPRS or power-line) with data concentrators and base stations. The costs for training of the employes and field service personal are higher and changed from the classical work of manual meter reading into offering customer specific energy services [KS09, p. 25]. Data management, billing and reading of meter data will happen full automatic. Billing cycles gets shorter and shorter and are going to reach a one month period like mobile telecommunication services. This will result in high costs which must be suppressed with a higher effectiveness and usage of the collected data [KS09, p. 27].

3.1.2 Data Management

For the management of a high count of metering data powerful servers and databases are required. In addition to metering data the software system must hold personal data of the customers and parameters of the devices distributed in the metering network. All acquired values must be brought to a common data format (i.e. Object Identification System (OBIS)). The conversation of the data format should happen automatically direct after the communication process at the storage into the database. For effective revision of the converted data each point is identified with state- and time-stamp information and can hold user comments [KS09, p. 102]. Like mentioned above personal user data are required for different business processes of the energy supplier. All customer data has to be editable and must follow data security standards. Personal data are strongly joint with smart metering information and configurations. Data reconfigurations will become more and more flexible (new dynamic customer specific tariff schemes, address changes, real time consumption data information, new metering or communication firmware, etc.) in our modern way of life [KS09, p. 103].

3.1.3 Smart Metering as a Nucleus for new Technologies

Today and in near future the classical workspaces of generation, distribution and transportation of energy will be extended with enormous data and information processes. Smart metering is the central part between customer, energy generation company and distributor, and is therefor a base technology for innovative services beyond measurement applications. The following explained applications are based on smart metering and use this key technology as a source for real time customer specific energy consumption data.

Decentralized energy generation and virtual power plants are concepts where the generation of energy occurs at the customers home. In this part regenerative energy sources like wind-, sun- or biomass energy plays the main role. Such a microgrid which is a combination of electrical sinks and sources is connected to the energy supply system via well defined interfaces. With an energy management system microgrids can be bundled, and act as a virtual power plant at the suppliers view. So a virtual power plant is a summarization of several small energy generators, but they can act like a conventional power plant in the view of controllability [KS09, p. 184].

Electrical load management is the ability of shifting large loads to time windows with lower grid utilization. Therefore it can be distinguished between *time-tolerant* and *interruption tolerant* loads. At time tolerant loads the begin of the service of the appliance can be shifted in time (i.e. a dishwasher can be filled in the morning and the service must be finished at six o clock in the evening). Interruption tolerant devices can be interrupted and switched off during their operation. These types of appliances are heating, ventilation, air conditioning and cooling (HVAC) devices. For this application they must be connected with a temperature sensor and a control system. Effective load management with interrupts in the operation requires the utilization of naturally given temperature buffers like well insulated rooms or, hot water buffer reservoirs [KS09, p. 186]. A related topic to shifting the operational time of home appliances is electromobility. With the help of the accumulators of the electro vehicles decentralized sources and buffers for electrical energy will become available. Another challenge of smart metering is its usage in a charging infrastructure for electro vehicles. The customers requires a full automatic charging and billing system at any point of the grid, independent of the manufacturer of the vehicle. The advantages of the accumulators (buffers and decentralized sources) can only be used if the vehicle is connected to the grid if it is not in operation. Intelligent charging algorithms could also handle scenarios like charging the batteries at the cheapest energy tariff on weekends or selling energy back to the supplier from the vehicle at times of high grid load [KS09, p. 188]. The energy and information technology sector will be combined to a high functional unit.

The combination of smart metering with home automation can bring lots of new different advantages. Smart metering data can be implemented in a home automation system and different actuators could perform their actions under dependencies of the actual energy consumption or tariff. The presentation of real time energy data on in-home displays is a scenario which requires a cyclic and stable communication between home automation system and smart meter [KS09, p. 186].

3.1.4 New Smart Metering Tariff Schemes

Advanced metering structures can be implemented with different types of intelligence [DD09]. According to [DD09] typically three types can be implemented. Today only the third one - smart metering reaches a maximum of importance.

- Automated meter reading (AMR) is a automatic register reading procedure without a physical access to the meter.
- Automated meter management (AMM) updates the AMR system with the ability of managing the meters remotely (i.e. disconnection or dimming the energy supply for non-paying customers).
- Smart metering extends AMM with control abilities. This technology enables functionalities to shut down several customers simultaneously on short notice for load shifting or balance the grid in case of an incident. Smart meters are an indispensable enabler technology in the context of smart grids and communication to control the electrical grid [DD09].

Smart metering can implement several new tariff schemes for electrical energy. Tariff schemes can be time-invariant (flat rate, declining block rate, inverted block rate, etc.) or time variant (time-of-use pricing (ToU), critical peak pricing (CPP), real-time pricing (RTP), etc). At ToU the day is divided in a fixed number of timeslots with different prices for electrical energy. This scheme motivates customers to shift the activation times of large loads to cheap non peak time slots. At RTP prices various continuously (i.e. hourly), related to the prices on the whole sale market. In comparison with ToU prices in calm periods can be lower. Prices are typically fixed one day in advance. In CPP a period of time (typically also one day) is divided in several timeslots like ToU but timeslots with peak demand will become significantly more expansive. Such slots are not fixed and can vary over the day. Customers are informed late but they can safe a lot by avoiding these high price slots. In a view of an energy supplier the new tariff schemes wants to motivate the customers to use the energy more efficient and shift the activation of huge loads to non peak load times [DD09]. All there tariff schemes requires smart meter communication with a data concentrator. In case of TuO the new tariff information must be distributed to the customers meters every season (i.e. every three months). In RTP or CPP the effort of communication will become several times higher. At these two tariff schemes daily or sub-daily updates of the tariff information for timeslots are required [DD09].

3.1.5 Smart Metering Structure

Smart metering requires different flexible data communication structures for the transportation of user data to a central point. Basically three typical structures can be mentioned (see Figure 3.2): **Direct** - Central data collectors communicates direct with the modem in the meter. In Figure 3.2 smart meter one is connected direct to the system. **Data concentrator** - The central data collector communicates with a data concentrator and meters communicates with the concentrator. In Figure 3.2 smart meters two to four are connected to a data concentrator which is IP based connected to the central system. **Headend System/Transaction server** - The central data collector communicates with a headend- system. This headend system is a secondary transaction server which communicates direct with meters or data concentrators. In Figure 3.2 smart meters five to seven are connected via a data concentrator to a transaction server which transfers the collected datapoints to the central system [KS09, p. 99].

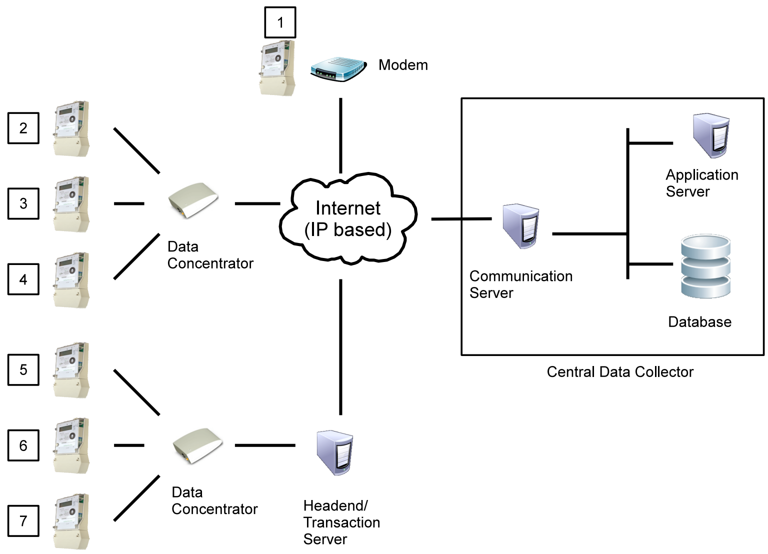

Figure 3.2: Different topologies for smart meter connections to a central data collector system. Smart meter one is connected direct, smart meters two to four via a data concentrator and smart meters five to seven with the help of a transaction server. Source: [KS09, p. 99]

At communication level actions can be divided in push- and pull transactions. Push transactions are commonly used in IP-based networks and means that the meter/concentrator sends its data automatically to a higher instance in a frequent way. In pull operation mode concentrators/-transaction servers can request data-points from an enddevice. So this mode of communication is used if a re-requirement of datapoints in the case of push operation errors is needed [KS09, p. 98]. For the communication-media and protocols between local- and global communication must be divided. The global communication is the transaction of data between central data collectors, transaction servers, data concentrators, etc. conventionally on this layer IP based communication is used - direct ethernet connections via fibre lines, Digital Subscriber Line (DSL) connections or wireless GSM/ GPRS connections. If the meters are not directly connected to a net (generally not) local communication between smart meters and data concentrators become necessary. In addition to wired interfaces like EIA-232 (Electronic Industries Alliance), EIA-485 and wireless short range technologies like wireless M-Bus, Bluetooth or ZigBee, power line carrier (PLC) transmission is mainly used [KS09, p. 100].

The collection of metering data is a high-end activity in information technology. Modern server systems supports full parallelism in functionality, load balancing, intelligent routing, setting priorities and automatic error handling. In addition to software functionalities the hardware structure must be designed in a high scaleable way. Higher requirements must be simply satisfied by adding an additional server [KS09, p. 101]. Because of high availability requirements in the databases redundant systems (server clusters) and virtual severs (VMWare) became a standard [KS09, p. 101].

3.1.6 Visions of Smart Metering

The vision of a global smart grid with intelligent self optimizing resources can become more and more reality in the next years. A potentially key driver of this restructuring is the slight introduction of electrical powered vehicles [KS09, p. 95]. This trend started in the years 2009/ 2010 with global public advertisement for electric powered bicycles and will end in full electric powered cars. This requires a mobile smart metering structure with user bounded tariff models. For resident households the new tariff schemes will bring more flexibility. The users are motivated to get a demand-response way of thinking, use the available energy in a more efficient way and flatten the average load profile [DD09]. In this case automation joint with a smart metering structure could bring additional advantages to use naturally given energy buffers. A household could also act as a distributed source (i.e. photovoltaic systems) and sell the energy back to the local supplier. All these applications will require a strong junction of energy and information technology.

3.2 Automatic Metering Information System

The Automatic Metering Information System (AMIS) is a full smart metering solution from the company Siemens. According to Table 3.1 several important projects and pilot projects in Europe uses this smart metering technology.

The AMIS system is a summarization of devices which include the following components: End-devices (smart meters, gateways, load cut off devices, etc.), data concentrators, telecommunication equipment for the connection to a central server, AMIS central stations with SAP (company

Table 3.1: AMIS Smart metering projects of Europe sorted on the number of metering points. Source: [Ene11, p. 23]

Energy Supplier	Country	Number of metering points
Energie AG Oberösterreich	Austria	600.000
EnBW ODR	Germany	240.000
Ried Energie	Austria	35.000
EW Witrh	Switzerland	30.000
Stadtwerke Stein	Germany	20.000
GW Schübelbach	Switzerland	10.000
Arbon Energie	Switzerland	10.000

name no abbreviation) interfaces, network management systems, etc. An important AMIS system feature is the openness of the system architecture. The system design with digital signal processors enables an in field firmware upgrade without any mechanical manipulations or changing the device [Ene11, p. 4].

Because of the usage of AMIS technology for the implementation of this project the siemens smart metering device will be explained in a more detailed way in the following sections. The AMIS smart metering palette has three different devices for the satisfaction of customer requirements: The TD-3510 is a three phase smart meter which is designed for a maximum current of 63 A (right device in Figure 3.3). This device is used for houses and farms. The TD-3511 is also a three phase smart meter which can be used for a maximum current of 50 A (left smart meter in Figure 3.3). This meter is used for small households or flats. The TD-3512 is a single phase smart meter also for a maximum current of 50 A (the smaller smart meter in the middle in Figure 3.3). This meter is used for small single phase supplied flats or as sub smart meters in combination with a TD-3510 or TD-3511 [Hue09, p. 4].

All smart meters have the same electronics hardware and functionality, only power electronic components are different. A customer specific parameterization will occur at the installation of the device. The common main components of the meters are:

- Data Interfaces: An optical IEC 62056-21 interface can be used for setting configurations, communication with a Personal Digital Assistant (PDA), basic calibration and extended functions [Hue09, p. 5].
- State indicator: The left light emitting diode (LED) indicates the status of the meter. If the LED is green the device is authenticated in the metering system. If the LED is red the meter is in boot up state and connecting. A orange LED indicates that distributed line communication (DLC) commands will be received. If the LED is off the device is not supplied or an error occurred [Hue09, p. 11].
- Impulse LED indictor: The right LED is a pulse indication and can be configured for real or reactive power. If the LED is permanently red no energy will be imported or exported. Otherwise the LED will flash at a rate of 500 Impulses per KWh for active power and 500 Impulses per kVARh for reactive power. [Hue09, p. 11]
- Display: A monochrome display will present all metering datapoints (power, energy, currents, tariff information, etc). The data view can be changed by pressing the red button under or beside the display [Hue09, p. 5].

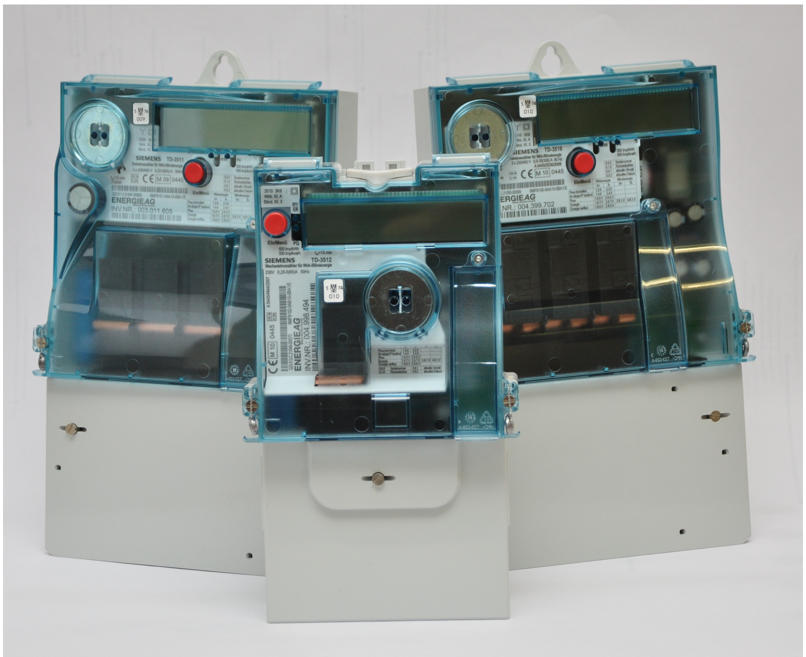

Figure 3.3: AMIS Smart Meter family. Source: [Hue09] With allowance of Energie AG Oberösterreich.

- Extension interface: For the extension of functionality or for the connection to secondary devices or automation systems a M-Bus extension interface is implemented [Hue09, p. 5].
- Load cut off Module: This can be used for an automatic or manual switch off of the connected load. The function can be used in the case of customers moving or for debt collection [Hue09, p. 5].
- PLC modem: The PLC modem is used for the DLC communication between smart meter and data concentrator. The DLC is a bi-directional communication via power line. DLC communication technology is not optimal because the power grid is optimized for an efficient transportation of energy and not for data packet transmission [Ene11, p. 4]. DLC can be divided into narrowband and broadband DLC systems. According to Table 3.2 narrowband DLC which is implemented in smart meters uses frequency ranges of lower than 500 kHz [Ari09]. This brings lots of advantages which are very important for communication in power grids: - Low attenuation which is optimal for long distance communications between smart meters and data concentrators. - Reduced number of repeaters. This brings lower costs and a higher efficiency. - An independent and parallel use with broadband DLC is possible [Ari09]. The data-rates at narrowband DLC are about 100 kbps up to 1 Mbps. The narrowband DLC communication is standardized in European Committee for Electrotechnical Standardization (CENELEC) standards. Table 3.2 lists different properties of narrow- and broadband DLC communications. But like mentioned above due to the requested data rate and high distance communication narrowband DLC is the key technology for smart meters [Ari09].

  AMIS technology uses narrowband DLC at frequencies of 9 kHz to 95 kHz [Ene09]. The DLC communication at this frequency spectrum is standardized by CENELEC A (Committee for Electrotechnical Standatdization) [Ari09]. Field experiments in Burgstein (Austria) results a communication distance of 3294 meters without any repeating component. The commu-

Table 3.2: Characterization of Power Line Communication (PLC) systems. Source: [Ari09]

Property	Low Data Rate Narrow Band	High Data Rate Narrow Band	Broad Band
Frequency Range	9 kHz - 148.5 kHz	9 kHz - 500 kHz A-Band 9-95 kHz B-Band 95-125 kHz BCD-Band 95-148.5 kHz	1.5 MHz - 50 MHz
Data Rate	<10 kbps	50 kbps< ... <1 Mbps	>10 Mbps
Coding Technology	frequency shift keying, binary phase shift keying, fast frequency hopping, dif. chirp shift keying	orthogonal frequency division multiplexing, multi carrier modulation	multi carrier modulation bit loading
Forward error correction	no or low	strong	medium
Applications	Automatic meter reading, European Installation Bus, Power line area network	Airfield lightning, Energy management, Smart Grids and Metering, AMR and AMM	Voice over IP, High definition television

nication happened via two grid subsections (980 V) and 3 transformers (400 V/980 V). All components are joint via underground cables [Ene11, p. 13].

- Manipulation contact: There is a small button between the casing and the connector coverage of the smart meter. This small button has 2 functions. At the installation and configuration the button must be released (connector coverage opened). After this process the connector coverage shield will be mounted, the button pressed and the meter connects to the data concentrator. If the customer is going to manipulate the installed smart meter the button contact gets opened and a DLC error message will be sent to the energy supplier [Hue09, p. 6].

AMIS smart meter properties, functions and services

The AMIS smart meter is a full electronic device which replaces convenient ferraris meters. One of the largest advantages for the energy supplier is an automatic reading of consumption data and direct import of this datapoints into the ERP (enterprise resource planning) system. But the smart metering also offers great functions for customers. In each smart meter device the load profiles of the last two month with time intervals of a quarter hour are buffered. After the analyzation of these load profiles the customer specific tariffs can be designed [Ene11, p. 5]. For this functionality and storage of the tariffs the smart meter has six tariff registers. The user can choose between the stored tariff models and the tariff will be activated at the appointed date [Ene11, p. 5]. But smart metering also holds costs for the energy supplier. Table 3.3 illustrates prices for AMIS measurement services of the Energie AG Oberösterreich. These are pure costs for metering services. The AMIS devices can handle different metering services independent of the used technology. The introduction of smart metering is free for the customers but the tariffs for metering services can vary in time. The mentioned prices are from the year 2011 and can be handled as guiding values [Ene11, p. 20]. With the help of relays each AMIS meter in the system can be locked. The locking command can be sent by the central station of the supplier

Table 3.3: Prices for smart metering services. Source: [Ene11, p. 20]

Class of metering service	Price in euros
Single phase AC - single tariff	0.73
Single phase AC - multi tariff	1.73
Three phase AC - single tariff	2.18
Three phase AC - multi tariff	4.00
Load switching	1.00
Power measurement - direct	9.00
Power measurement and reactive power measurement - direct	11.40
Load profile measurement	50.00

and receives the smart meter via DLC connection. The lock of the smart meter is also a standard procedure if a costumer is not able to pay for the consumed energy and services. Is the smart meter switched to the debt collection mode the user can buy credits for energy supply. If credits are paid the relays will automatically close. The locked debt collection meters can be easily administrated with the help of the SAP system. Today the prepaid mode smart meters are permanently switched on or off depending on the loaded credit. For future applications it could be requested that a minimum energy supply must be approved (i.e. 100 W). This functionality can also be realized with the AMIS smart meter hardware [Ene11, p. 15].

Several additional functionalities and services at the meter are implemented or could be updated by software. For example like mentioned above the detection of manipulations on the meter could start several processes at the energy suppliers server system (i.e. automatic lock). Also a magnetic field sensor is implemented to detect manipulations with the help of strong magnets. An other additional service is the monitoring of voltage quality for the supplier. The smart meter stores different voltage parameters (i.e. level, interruptions, etc.) in its internal registers. These registers could be read out and are a data basis for grid simulations, infrastructional optimizations and finally decision criteria for effective asset management [Ene11, p. 6].

The following itemization summarizes the main technical properties of the AMIS smart meter devices. (Source: [Ene11, p. 8])

- 2 x 6 registers for active energy measurement
- 1 x 1 register for reactive energy measurement
- Automatic or manual tariff change depending on time or power
- Load cut off relay for locking the smart meter
- Voltage quality measurement registers
- Optical IEC 62056-21 interface
- Extension interface via M-Bus
- Logbook functionality
- Firmware download possible
- Full electronic four quadrant (4-Q) metering unit for the measurement of real- and reactive energy import and export

- Load profile memory for 60 days at a cyclic storage time of 15 minutes
- Display for user feedback information

AMIS structure and business processes

In transformer stations smart metering data will be collected with the help of data concentrators. The connection is in AMIS always given by DLC [Ene11, p. 7]. The data concentrator in the transformer station is connected with a central transaction server via one of the following communication media (listed per priority): - Fiber communication. - Radio communication. - GPRS. - DSL (over private copper lines) [Ene11, p. 10]. Data concentrators can also act as a remote devices for the automation of tasks in the transformer station itself. Therefor input/output (I/O) modules and an optional accumulator can be connected to the concentrator. With the help of the accumulator communication could also be active in de-energized times. With the connection of short-circuit indicators to this automation system the interruption times in case of grid errors can be dramatically shortened [Ene11, p. 10].

During working processes in the grid the connection between smart meter and concentrator can be interrupted. In this case the smart meter is able to connect a "foreign" data concentrator. This makes no difference for the communication with the central server. Due to load management the smart meter tries to connect the old concentrator after eight hours (after this time generally working activities in the grid are finished). After a successful reconnection to the "own" concentrator station all tasks can be continuously done in a normal way [Ene11, p. 14].

A daily check of the availability of the end devices shows permanent DLC disconnections. If a AMIS device is permanently not available over five days the green LED at the device will be switched off and the energy supplier will automatically receive a maintenance request [Ene11, p. 14].

The installation and putting into service processes are handled with the help of full electronic workforce management. The following steps describes the business process of including a new customer and his smart meter to the system. (1) The customer places an installation request at the call center or direct at the local energy supplier department. (2) An installation request will be created in the SAP software system. This request will be transmitted to the mobility system Q4. (3) A service technician loads the installation request via GPRS or direct universal serial bus (USB) connection to his PDA (Personal Digital Assistant). (4) The service technician installs an appropriate AMIS device at the customers system. The parameterization of the device can also be done with the PDA. (5) The service technician retransmits the data of the installed AMIS device (i.e. serial number) back to the SAP system. (6) The device connects the transaction server via DLC. (7) The SAP system automatically registers the new AMIS device at the transaction server. (8) Finally the connection between registered SAP device and the real DLC connected device must be found. After these steps the new AMIS device is full integrated to the system [Ene11, p. 11].

The role of AMIS smart metering in this project

In this project AMIS smart meters are provided by the institute. According to the structure of the system one or more smart meters can be configured as summary power measurement devices. The smart meter is in this case a high precision reliable energy meter. An other advantage is that the smart meter offers several additional measurement points (currents, voltages, phase information, etc.) which could be displayed, and are important for system functionality. For the read of datapoints the optical interface will be used. A optical reading head and the implementation of the IEC 62056-21 protocol is required (see Section 6.3.2). This has the advantage of a simple serial application programming interface (API) and the reading head is easy to connect for the user. A large disadvantage of using the optical interface is that it is generally reserved for maintenance activities of the energy supplier. For a long time implementation it could be considered to change the communication to the AMIS M-Bus extension interface. As a summarization it could be said that the AMIS smart meters offers a large variety of functions and services which are optimal for the implementation of this project.

3.3 ABB Smart Metering Solution

The company ABB (no abbreviation) has two separated components for smart metering. The eHZ and MUC can be combined to form a smart metering solution or can also be used independent. eHZ stands for the german word "Elekronischer Hauhaltszähler" and is a modern smart meter device, MUC stands for multi utility communication and is a data gateway device. One of the main properties of the eHZ are small dimensions. The eHZ needs approximately half space of a classical ferraris meter. Additional hardware features are a fast non interruption exchange of the metering device and high operational safety [Chr10, p. 45]. For the non interruption exchange of the meter a special interface was developed. Figure 3.4 shows that interface. The new smart meter can be clipped onto this interface.

Figure 3.4: Interface for the ABB smart meter eHZ. With the help of this interface an non interruption exchange of the meter is possible. Source: ABB smart metering folder. With allowance of ABB.

The gateway device is like mentioned above a multi utility communication (MUC) device and supports communication between end user and energy supplier. The MUC guideline is a coordinated working basis in the view of communication, for users and especially for manufacturers of meters. The main facts of the guideline are: Standardized system for the acquisition of metering data for private customers. Compliance with regulations in the field of relevant to payroll measurement data. Support of consumption data information for business processes and future implemented intelligent services. Direct access to metering information by the customer via standardized interfaces [26]. The ABB MUC device is additionally to electrical energy data able to collect and forward datapoints of other smart meters (i.e. for gas or water). The MUC also offers an interface for the visualization of energy consumption data. Possible media devices for the presentation are an ABB-Panel (see Figure 3.5), personal computer or the display of a mobile phone. The respective connections between MUC and data viewer device are given by GSM or the internet [Chr10, p. 45].

For an extension of the metering functionalities the eHZ offers standardized interfaces for the connection of state of the art smart meters for gas, water and heating. They can also send their consumption datapoints to the MUC station. The MUC can handle sub smart meters for the measurement of sub-circuits in a household or an individual measurement of single appliances. The MUC data gateway will be able to support a full energy management including external decentralizes external sources like photovoltaic or wind turbines [Chr10, p. 45].

The eHZ devices are also able to receive time invariant tariff information. With the help of the ABB visualization solution and wireless ABB switched socket-outlets the customers can use a home automation system for selecting the optimal tariff scheme and management of the energy state of the household [Chr10, p. 46]. The consumptions of sub-circuits and appliances can be controlled. Actions can be set manually or a device could be activated automatically if an appropriate tariff is available. For customers in treatment of the technical realization, the energy supplier can offer a module based metering solution: Pure smart metering solution with a simple visualization of consumption data. Smart metering with a higher resolution visualization and interface for connection and a manual control of wireless controlled switchable sockets. In addition to the last point a intelligent and home automated management of appliances in combination with a smart metering could be implemented. This individualy combinable modules will result in a customer specific smart metering solution [Chr10, p. 46].

Advantages and the most important properties of the ABB smart metering solution are:

- The eHZ meets all relevant international standards and the german TAB2007 (ger. "technische Anschlussbedingungen") standard.
- The eHZ is in combination with an official verification a measurement device which is standardizes for billing.
- The eHZ has small dimensions and is due to simple connectors easy adaptable.
- The installation of the smart meter can be done without supply interruption.
- An update and connection of a second eHZ is possible. Figure 3.5 shows a system with two eHZ smart metering devices.
- For the update of older electrical household systems a standardized interface adapter is available for the installation of an ABB smart meter [Chr10, p. 47].

The ABB smart metering system is a well specified module based metering technology. The harmonization with ABB home automation devices make that solution very attractive for households equipped with such components. The implementation of standardized communication interfaces makes the ABB solutions ready for smart grid applications.

3.4 Easy Meter and NZR - Metering Solution

The two german companies NZR (Nordwestdeutsche Zählerrevision) and Easy Meter developed a metering solution by combining two devices of them. NRZ developed a multi utility communicator (MUC) which offers storage and communication functionality for different energy consumption

Figure 3.5: Left: Two mounted ABB eHZ smart meters. Right: ABB visualization touch screen panel. Source: ABB smart metering folder. With allowance of ABB.

meters. The MUC is connectable to different smart meter devices and bring all data to a common format for presentation and dissemination to the energy supplier. The requirements on the meters is due to a high intelligence of the MUC device extremely low [KS09, p. 132]. No storage of data or the implementation of tariff registers are needed. Therefor an optimal partner for the MUC is the Q3D metering device of the company easy meter. The Q3D is a full electronic replacement device for the conventional used ferraris meters. The device is available for a maximum current of 60 A or 100 A three phase AC line. The smart meter is also licensed for single phase measurement mode and can therefore used for sub-metering and other single phase appliances [KS09, p. 132]. The combination of the two devices is called MUCMeter (see Figure 3.6). The MUC is fitted on the top of the smart meter case. The interface between the two devices is the optical D0 port which is automatically connected by mechanical mounting. The power supply of the MUC part is accomplished with the help of external jumper connections. All three phases of the smart meter are connected to the MUC. So there will be the availability of all power lines for a three phase DLC communication. For the power supply connectors two versions are available. The metering device can be supplied before or after the internal energy measurement part. So the difference is whether the customer or the energy supplier pays for the MUC service [Chr10, p. 118]. Due to a well isolated construction of the supply connectors the update of a Q3D with a MUC controller could definitely be achieved by non specialized staff [Chr10, p. 118].

The MUC can also be connected to other types of smart meters and offers additional interfaces like M-Bus, current line interface (4 mA - 20 mA) and optical serial interface. In the case of M-Bus the adaption for the connection to the MUC is extremely easy. The automatically defined media identifier of a M-Bus device, makes the system "Plug and Play" useable in the case of one connected meter per type (i.e. gas, water, etc.) [KS09, p. 134]. No additional manual configurations are required. In the case of wireless data transmission wireless M-Bus will be used. Wireless M-Bus is a standardized short range communication protocol for measurement applications. It

Figure 3.6: MUCMeter - Combination of NZR MUC device and a Easy Meter smart metering device. Source: NZR - `www.nzr.de`. With allowance of NZR

uses the industrial, scientific, medical (ISM) band at 868MHz. At the initialization the two wireless devices share their bus addresses and starts an authentication process. For authentication a 128-bit code is used which must be configured and deployed per MUCTool software at each wireless device [KS09, p. 135].

For the communication to the energy supplier GPRS, PLC or DSL are implemented standards. An early conversion of the data packets to an IP-based communication is one of the most important key factors in the communication infrastructure. At the background an IP telemetry server called "IPT-Smart Gate" handles data requests and connections to end devices. The server software allows grouping of customers and offers functions for customer management. The IPT-Smart Gate offers at its software interface the standardized protocol DIN 43683-4 (ger. Deutsches Institut für Normung) and is so a vendor independent solution [KS09, p. 137].

For a customer specific visualization of consumption data a user interface for direct access to all connected meter device information is realized [KS09, p. 133]. The MUC device implements a standalone web-server and the smart message language protocol (SML). Consumption data information can be displayed on a browser without the installation of any operating system dependent software [KS09, p. 138].

As an update of the NZR hardware product portfolio the company plans a socket measurement device. This will communicate via PLC with the MUC device and enables a individual load profile measurement of appliances [KS09, p. 140]. As an additional service NZR will more and more turn to custom specific energy-related advisory services. The NZR smart metering solution is a state of the art metering technology and will help to reach an increased energy efficiency on user and supplier side [KS09, p. 139].

4 Energy Feedback Information

Many examples in our daily life shows that systems naturally contains feedback. But for engineers it is not trivial to implement such high efficiency and clear feedback mechanisms in technical systems [PI10, p. 16]. Feedback mechanisms need to be planed, and they must serve a particular purpose or objective. Data must be computed and displayed, and it must be clear how to impact the system in a way to reach a desired objective. This means, that the user translates feedback information into actions on the system [PI10, p. 16].

For the design of a feedback system the following common aspects and questions according to [PI10, p. 16] should be considered:

- What is the purpose of the informational feedback and which objective should be realized?
- Can the objective be verified. Which types of sensors must be used and are there sensor data nobody can verify?
- How can sensor data converted into actions? Even if there are sensor information available how can the user decide which action is the best to steer the system in a higher performance or desired state.
- How should the user act on the system? Which actuators are necessary? - "Knowing what to do is good, but could it be done?"

An other aspect in planning a feedback system containing a human actor, is to find an adequate abstraction of a system (changing from a presentation illustrated in Figure 4.1 a to Figure 4.1 b). An aspect of modeling is that the detailed internal signals passing the system are not visible for the user. Only the main structure (block diagram) and really important internal information should be displayed (see Figure 4.1 b). Like illustrated in Figure 4.1 a human is a part of the system which receives feedback information and can based on this feedback data set actions to change the systems internal states.

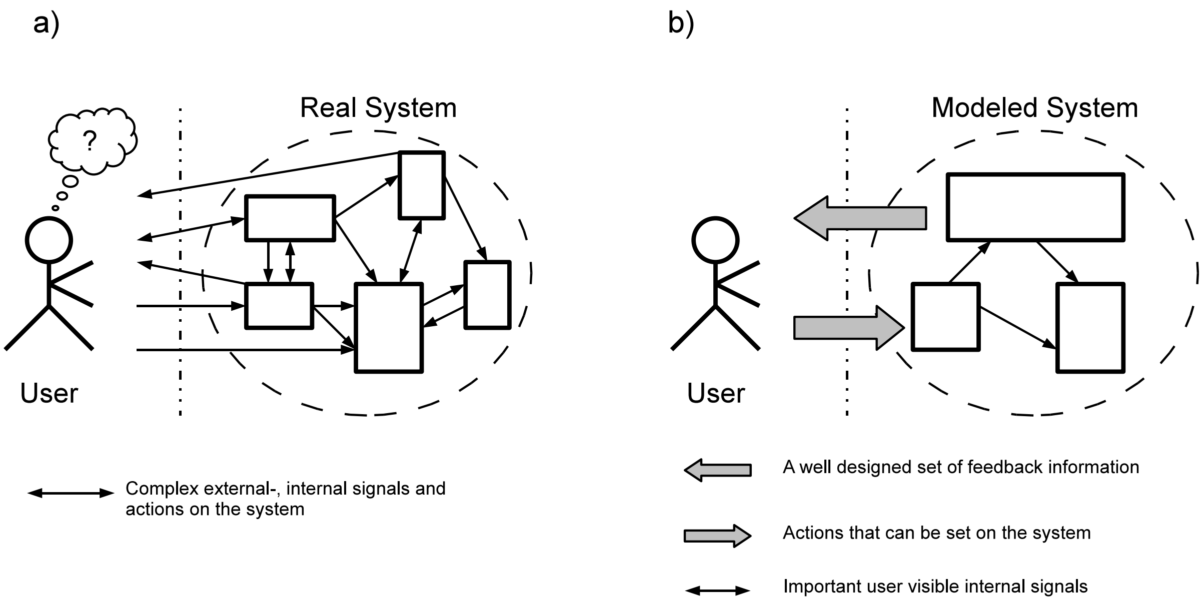

Figure 4.1: a) The real system contains several sub-systems which are connected and passing internal messages. It is hard for the user to handle such a complex full system presentation and to find the best actions to reach the desired system state. b) An abstracted presentation and a well defined number of feedback information helps the user to transform feedback data into actions for reaching an other system state. Source: [PI10, p. 17]

4.1 State of the art Energy Feedback Systems for Home Applications

Because of long term effects of energy feedback mechanisms, the feedback topic is strongly connected with new smart metering tariff schemes, (see Section 3.1.4) and the economic aspects for introducing smart metering technology in standard households (see Section 3.1) [Dar06]. New tariffs and different user specific parameterization of the supply system can only be presented by using an appropriate visual feedback system. Energy feedback at household systems could be divided into two main sections:

- **Direct Feedback:** The role of the meter or other common measurement systems is that data are available at any time. One or more display units informs the customer about the actual energy state of the household. The presentation of information can occur in different ways (i.e. ambient display, mobile, television (TV), etc). This type of feedback allows a quick engagement into the households energy state and a short term learning effect of the user.
- **Indirect Feedback:** Describes methods that has been processed in some way before reaching the customer, normally via billing or other informal notifications. This form of energy feedback is more suitable than direct feedback, for demonstrating effects in changing some structural aspects of a household (i.e. investments in thermal insulation, replacement of high consuming appliances, etc).

For this work only direct feedback methods are of interest. For more detailed information on indirect energy feedback methods see [Dar06]. The common idea of all direct feedback methods is building up a body of know-how about the energy useage of appliances and its optimal time of activation. Customers learn to act, and can interpret the changes at the direct feedback within a short period of time. These three basic elements (receive information about the actual state, act and change something, receive feedback and interpret the actions) are the driving key factors of all described state of the art feedback methods.

4.1.1 Direct Energy Feedback Methods

The following methods describe how a state of the art energy feedback systems works and how some of them are implemented within this project.

Direct displays on monitors separated from the meter: Direct displays show exactly the same information which can be read from the meter device itself. Information of different metering devices (i.e. also for gas or water) can be connected, and the meter readings can be displayed. Additional features of such direct display systems are the storage and presentation of historical values, setup of alarm levels, etc. The core property of direct displays is that they are a pure presentation of the meters data without any mathematical computation [Dar06].

Disaggregated feedback: Disaggregated feedback methods offers in addition to direct displaying the consumption data an analyzation of the measured values [HKR95]. A norwegian study showed that the breakdown of the total energy consumption into six main domestic device types printed in a pie chart on the bill will be the first step to change the customers behavior in the view of energy efficiency. 81 % of the customers responded that this form of additional information is useful, and 38 % of them said that they have learned something new from it [Dar06]. And this is exactly what disaggregated feedback systems do. The fragmentation of the total energy consume will be broken down into main load types and this is shown on time at a display.

Use of TVs, PCs and mobile displays: Graphical presentation of data requires higher performance displays (higher resolutions , color display, touch input, etc). This will result in increased costs for customers. That is why some companies use displays which are already installed in a household. State of the art high definition (HD) television systems for displaying the data is a cost reducing system solution, but has the disadvantage that the input of parameters with the help of a remote controller is poor in performance aspects. Other solutions are to use web pages which could be loaded at any internet connected private computer (PC) or display solutions for smart phones.

Ambient displays: Ambient displays are a smart and minimalistic solution for energy feedback information. They do not show any text or numbers. These type of display alerts the customer if there are some relevant changes in the electricity supply or consumption of the household [Dar06]. Due to the simple presentation of relevant information the hardware costs for the system are reduced to a minimum. In data acquisition this system configuration is not simplified in comparison to disaggregated or direct displays. Reading smart meter data, storage and computation at a central unit is also required at this feedback system type.

4.1.2 Feedback Methods used in this Work

In this thesis three of the methods described above are implemented. This are a direct illustration of smart meter data, disaggregated load view and an ambient display style traffic light view. The feedback information at the graphical user interface is divided into different modules representing the different available views. The smart metering life view is a direct display feedback method and presents the results of the data management software module. The most important data of the smart meter (A full smart meter data set is shown) are presented and history values can be requested. In addition to the measured values of currents, voltages and angles reactive- and real power for each line can be calculated and displayed according to Equations 2.5 and 2.6. Smart meter data are read continuously and are the input information for the load identification algorithm. The module analyzes the aggregated load profile data and tries to filter out activated devices. The result is an overview over the appliances which are included in the total load profile. The division will happen at every time-step when new smart meter data are available for the identification algorithm. If the detection fails or the load could not be identified due to bad matching with any stored device in the database the display will show historical values which are signed with a time-stamp. At these aspects the load identification live view sub-window is a disaggregated feedback mechanism. The disaggregation in this work is accomplished by a nonintrusive load identification software module.

In addition of presenting smart meter data values a so called "quick view area" at the main window of the GUI is realized. This shows the actual power consumption measured from the smart meter. As an additional indicator traffic lights for the power level are shown. The red light indicates that a maximum energy consumption limit is exceeded. The yellow one indicates that the consumption is in a mid area, and the green light indicates that the actual energy consumption is at a low level. The energy levels for the yellow and red indication are customer configurable at the preferences view. This traffic light functionality is a non textual representation of the energy state which can be interpreted by the customer at any quick view at the display. Such an ambient display could be also realized at any location of the underlying communication network. The interpretation of the smart meter data message and presentation of the levels with LEDs is also realizable in extern units independent of a graphical display. For the presentation of the graphical user interface (GUI) a standard PC display (i.e. resolution of 1280 x 960) is used. An implementation for other display types like TVs, smart-phones or presentation of the GUI at a web page is realizable but not part of this work.

4.1.3 Visualization System from Bticino

The Italian company Legrand/Bticino offers a state of the art energy visualization add on for their home automation concept. This system is based on their installation bus system. The additional components for energy measurement transfers consumption data via this bus to visualization panels. At a panel the consumption of electrical energy, gas, water, etc. can be displayed in graphical or tabular style (see Figure 4.2-b).

Implemented functions: The system is able to handle data from different consumption measurement sources. There are sensor element available for the acquisition of electrical energy, connecting water or gas smart meters, manage solar collectors and photovoltaic systems, etc. Distributed data acquisition units, for current and electrical power build a network for individual measurement of appliances. All collected values are presented at a display unit. Different views for monthly, weekly or daily data are available. Actuator modules enables automatic switching actions for connected loads. Energy levels for switching off devices can be configured by the user. These bus connected actuator modules can be categorized into 63 independent importance classes. So a customer specific load management is established. All sensor and display units are powered via the communication bus.

Measurement, interface and visualization modules: Depending on the application different modules for energy and current measurement can be used.
Current Measurement: A module for independent measurement of three current signals is available. The externally connected sensors are torroide ferrite core sensors surrounding the conducting wire. The module is designed for a nominal measurement current of 16 A and a maximum current of 90 A [bti11] [16].
Real Power Measurement: Two modules is for real power measurement on a single line are available. Power values are internally stored for a history value presentation and average values for 12 hours, 2 weeks and 12 months are offered. The first module is a pure measurement module for consumptions up to 18 kW (see Figure 4.2-a) [bti11] [16]. Measurement errors are depending on the power value and can reach a maximum of $\pm 20\,\%$ [bti11] [16]. At the second module in addition to the measurement part the installed appliance can be disconnected from power supply. A bistable semiconductor relay accomplishes the switching action which happens at a current zero cross. Maximum switching currents are: 16 A pure ohmic, 10 A with electrical light bulbs and 4 A for inductive loads with $cos(\varphi) = 0.5$ [bti11] [16].
Impulse counter: For the connection of smart meters the energy feedback system includes a module for interfacing state of the art metering devices with a pulse output connector. At the module different configurations for electrical power, gas volume, water volume, calories, etc. can be selected by a hardware switch. The counted pulses from the smart meter are divided by a factor according to the device configuration and the calculated value is transferred to a display unit [bti11] [16].
Displays: Each display available of the series can be updated for the energy management system. Data are presented in table or graph style (see Figure 4.2-b). The range of products reaches from small socket displays up to ten inch multimedia touch screen panels with universal serial bus (USB) interface and memory card slots [bti11] [16].

Advantages and Disadvantages: The energy visualization of bticino is a solution for solving energy feedback at different levels. The visualization of aggregated measurement results of smart meters, individual power measurement and individual current measurement is possible. For the connection of a smart meter an electrical pulse output interface is required. This interface only enables to read the actual aggregated power consumption. Additionally, at the display of the meter presented information (i.e. meter reading, currents, voltage levels, angles, etc.) are not available. Individual power measurement modules are supplied via the bus connection and qualified for a wide range of standard applications. At the time of writing no three phase power measurement module is available. The device management system enables automatic appliances switching actions. Configurations (timing and power levels) for the load management are direct stored in the actuator modules and not in the display controller unit. So the system has a distributed control approach and no single point of failure exists. User friendly installation, a subsequent addition of modules and a small number of configurations are a large advantage of the system. The ability of using different types of displays offers a customer specific solution for the presentation of information. A disadvantage of the system is the wired bus structure for communication. Wireless solutions with a meshed network of controller nodes would offer more

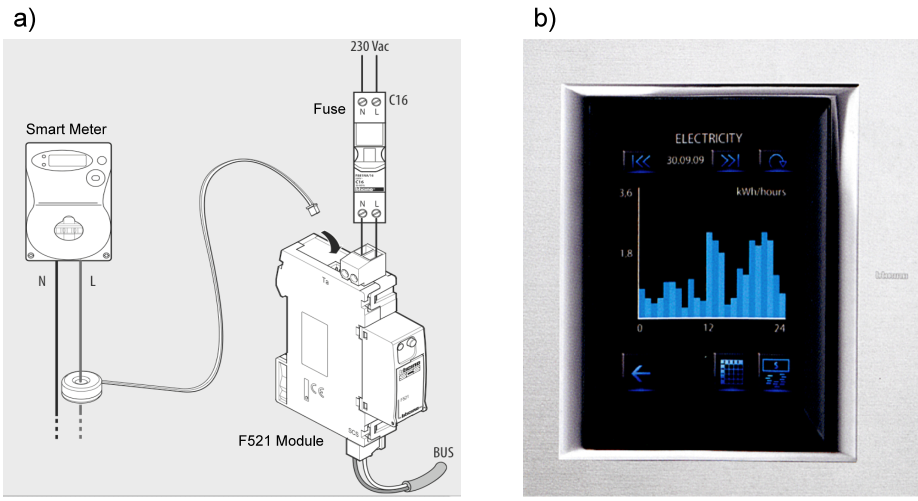

Figure 4.2: a) Installation scheme of the F521 power measurement module. At this setup the module is used for aggregated power measurement. b) Graph visualization of consumption at a touch screen panel. Source: [bti11] [16] With allowance of Legrand Austria

flexibility in the installation and in realization of mobile remote systems. However the energy visualization system from bticino is a state of the art solution for the integration of energy feedback in new built homes, or houses still equipped with a bticino installation bus system.

4.2 Hybrid System Architecture

A hybrid system architecture combines smart metering and individual device power measurement. The smart meter is used for the acquisition of aggregated load profiles. The sum of consumptions of all appliances in a household is measured. Individual power sensors are distributed measurement elements for analyzing the consumption of devices. For this purpose, state of the art power measurement chip solutions (see Section 2.2.2) are used. They are integrated in external socket or installation units (i.e. system of Bticino, Section 4.1.3). Today no standard appliance is available with internally integrated solutions for power measurement and an appropriate interface for sharing this data with other automation instances. The measurement data from the individual sensor nodes at the devices and the smart meter must be transferred to an energy feedback system for presentation. This can happen via a wireless solution, power line communication, an installation bus system, etc. All information will be collected and brought to a common format at the visualization system (see Figure 4.3). In this project no hardware based hybrid system architecture with individual power sensors is used. The detection and measurement of individual appliances is realized with a non intrusive load detection algorithm.

At a distributed power measurement node the implementation of additional sensors is possible. Chips for the analyzation of temperature, humidity, light intensity, etc. give extra information about the surrounding of the appliance. These data also must also be transferred to the energy feedback system. A challenge in the design of a hybrid direct energy feedback system is the clear presentation of information. The more data available the harder is an appropriate presentation. If the usability of the system decreases due to an overload of information, the positive effects of direct energy feedback described in the next section will be not present.

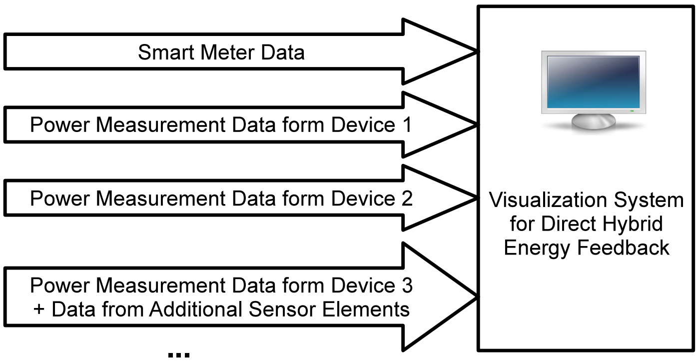

Figure 4.3: Hybrid data input containing smart meter, individual measured power consumption and information of additional sensor elements (i.e. temperature-, humidity values, etc). Source: author

4.3 Impact of Energy Feedback on Energy Efficiency

The importance of internal motivation opposed to external incentives, and the need of feedback over time to allow customers to monitor the positive effects in energy efficiency and any changes in their lifestyles, housing and appliances [You93]. It is worth saying that the saving from energy feedback not only depend on the technology under consideration. There are several more factors like institutional and cultural backgrounds, quality of feedback, regional aspects, etc. However all reviewed studies do show the usefulness of having energy feedback information that is specific to customers and allows them to control their energy use more effectively [Dar06]. As a summarization the most important factors for increase the effectiveness with energy feedback methods are:

- General context like social factors, energy infrastructure, etc.
- Scale and time of usage. The duration of a study for analyzation of energy efficiency is a driving factor, because for the customers a period of about three month is required to change their behavior in appliance usage [CKP+09].
- Synergies between energy feedback and other forms of information. The separation of different effects for a decrease of the energy consumption of a household is a complex activity and requires intensive and detailed studies [Dar06].
- The impact of long term effects (i.e. investment in insulation of a house, replacement of old appliances, new heating systems, etc.) must be taken under consideration. The usage of direct feedback with displays is a so called "moment-to-moment" strategy and change the short term behavior of a customer best [Dar06].

One of the most effective and base methods in reducing the total energy consumption of a household is the introduction of smart metering (see Section 3). Smart metering in combination with customized tariff schemes as an indirect feedback method will get more and more attention in several energy saving projects. The technology will help people to pay more attention on their electrical consumption, to cut energy bills. At a recent project in the UK the government announced that a smart meter device will be installed in every household until 2020 [ZXM10]. First analyzation of the consumption data shows that an average customer is possible to save 2 % to

3 % of energy per year and will cut £25 to £35 off their bills [ZXM10]. That means that this country wide smart meter roll-out can reduce CO_2 emissions by 2.6 megatons per year [ZXM10]. In the future in addition to the decrease of the energy consumption, the usage of renewable energy is the trend to reduce the gas and carbon emission from the generation.

In comparison to pure smart metering at direct energy feedback higher energy saving rates can be reached. This is caused by more flexible reactions and learning effects of the user at any action on the system see Section 4.1. A recent project in Korea uses so called "In Home Displays" for the direct energy feedback. The display presents actual smart meter data and historical consumption values. 48 of 53 households in Cheongju showed reductions in daily power consumption. The average decrease was 15.9 % from 7.92 kWh to 6.66 kWh [CKP+09]. In a second study 22 of 24 households in Seoul showed an energy reduction of 7.5 % from 9.77 kWh to 9.04 kWh [CKP+09]. The reasons why the study in Cheongju reached a higher rate of energy savings are social factors and different variations of the day/night temperature profile in the cities. For more details on the different results see [CKP+09]. According to the division of direct feedback methods into different methodologies (see Section 4.1.1) the following results could be found. For direct displays the savings are typically in the order of 10 % and can reach up to 15 % as shown in the Korean study above [Mou06]. An American study with ambient displays results in 16 % saving over a three week period [SDB78]. The functionality of the system is that a red light flashed in case of the temperature dropped below 20 degree celsius. This tells the customer to open the windows for cooling and the air conditioning will be turned off automatically. The best results could be found by using TVs or PCs displays for direct energy feedback. The rate of electricity savings reached up to 18 % for ten householders who took part in a ten month trial [UIST05]. As a summarization Table 4.1 will show typical average energy saving rates reached by a specific feedback structure.

Table 4.1: Typical average energy saving rates which could be reaches by the stated energy feedback methods

Method	Savings	Study / Source
Indirect feedback with smart meters and customized tariff schemes	2 % to 3 %	UK, [ZXM10]
Direct feedback with In-Home Displays	7.5 % to 15.9 %	Korea and UK, [ZXM10]
Ambient displays with simple non textual feedback	16 %	America, [SDB78]
Direct feedback with presentation on TVs or PCs displays	18 %	Japan, [UIST05]
Direct feedback with presentation on interactive web page	8.5 %	Netherlands, [UIST05]

Different tariff schemes in a smart meter equipped test environment and implemented energy feedback information systems can result in typical changes on the total load profile profile. The optimizations can be split in six individual areas illustrated in Figure 4.4 a) to f). The resulting total load profile will be a mix of all optimization types. Figure 4.4 a) shows load building. This will make the middle part more continuing and flattens the demand of the generators. A steady load profile is better for load prediction and a more effective scheduling of the available energy resources. Figure b) shows a general decrease of the energy consumption without any changes in the shape. Such a constant offset could be reduces by decreasing the consumption of appliances which are activated 24 hours a day. Illustration c) shows the load shifting optimization. Peak loads caused by simultaneous activation of a high number of devices could be shifted to low demand times. Customers must be motivated to shift the activations by getting cheaper energy from the supplier at non peak consumption times. Figure 4.4 d) shows a full shape variation of the load profile. This could be caused by replacing constant grid loads with new ones which have other individual characteristics. Picture e) illustrates valley filling which is very familiar with load shifting and could also be reached with customized tariff schemes. The last Figure 4.4 f) shows peak clipping. This method simply cut off the requests on energy at a maximum level and flattens the peaks. The disadvantage is that additional requests could not be served which results in a decreased customers performance satisfaction.

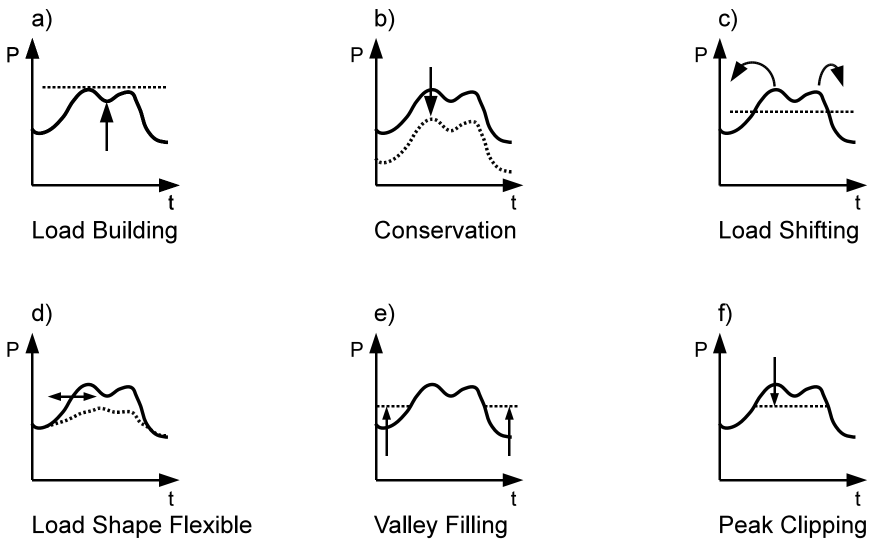

Figure 4.4: Optimizations on an aggregated load profile. Source: [KS09, p. 196]

5 Electrical Load Identification

The identification of active electrical loads in a household offers several advantages for customers. Most important is that users have the knowledge of their actual electric power consumption and which devices causes this consumption. With the help of an energy feedback system the users can identify which appliances are turned on or off at any time. In combination with an automation system, this feedback can help and motivate users to turn off unused devices. This reduces the consumption of electrical energy and finally decreases costs. The automation system enables automatic switching actions of the loads. According to configuration of controlling algorithms the automation system individually changes parameters of the household automatically (i.e. air conditioning parameters, setting of the heating system, lighting parameters, etc). The required identification of loads can be accomplished by controlling them in an individual way, or by the analyzation of aggregated load profile (nonintrusive load identification).

5.1 Hardware based approach

A system for controlling individual electrical loads via switches is stated in Figure 5.1. This is a totally centralized system. That means that the software algorithms are computed in a centralized processing unit [Fri00, p. 110]. Multi-core and inter module distributed solutions are possible but all components are locally centralized in one unit. The usage of multi-core processors enables the distribution of computational activities. The activation state of a load can easily be displayed by resolving the state of the internal switching relays. Due to the implementation of a switching elements and connectors for each device hardware costs and costs for the maintenance of the mechanical modules are high. A disadvantage of such system architecture is that each device must be connected individually to the central controller station. This results in long supply lines and a high number of individual connections at the module. The long wires increases costs for connecting a device and the limited number of interface ports results in a bad scalability of the system. The centralized controller is also a single point of failure. These disadvantages at the described architecture are the reasons why state of the art load identification has a totally distributed intelligence and the complexity changed from hard- to software.

State of the art load identification system offers more power in algorithmic design and software with less hardware components. One key technology is the distribution of small sensor nodes to change from a pure centralized architecture to a distributed measurement approach. These sensor nodes are small microcontroller devices with its own power supply and software. The units are linked together with a field bus system or wireless communication network. The data acquisition

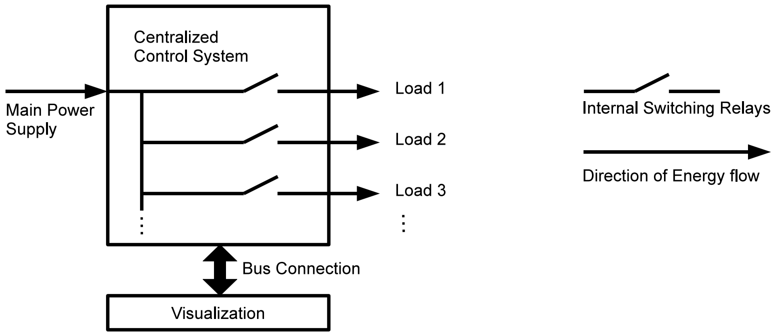

Figure 5.1: Centralized implementation of an energy feedback system. Source: author

offers in addition to load state information (on or off) a power and energy measurement of the connected appliances (sometimes the distributed nodes also implement other types of sensors. I.e. temperature, humidity, etc.) (see Section 4.2). All these collected data are transmitted to a central controller station. The central data acquisition unit in this case is a computer or embedded system which has an implementation of an appropriate load identification algorithm. An advantage of this system approach is the flexibility of the system architecture and functional extensibility. It does not matter how data is transmitted from the measurement sensors and linked to the identification process. In case of extensibility, modern computer or embedded hardware offers lots of functions like implemented web servers, database connectivity, GUI libraries, etc. As an summarization it can be said that this type of load monitoring technique deals with monitoring of energy consumption at the level of individual loads [Zia10, p. 10]. An alternative approach for a nonintrusive identification of loads is to monitor the aggregated load profile. An algorithm recognizes the signatures that indicates the energy consumption of the appliances as they are turned on and off [DK99].

5.2 Software Algorithms

All algorithms stated and compared in the next subsections are applicable nonintrusive load identification procedures. This means that the input data for the algorithm is a summary load profile containing of overlapped profiles from individual appliances.

5.2.1 Edge-Based Approach

The idea behind edge-based approaches is to analyze edges in the real and reactive power profile. For example if a refrigerator has a real power consumption of 200 W, a positive step increase of 200 W in the power load profile or life data measurement indicates that the refrigerator has been turned on. A 200 W falling edge indicates the deactivation of the device [DK99]. All types of appliances have their own but not unique signatures in the real- and reactive power profile (see Figure 5.2).

The detection of loads with this algorithm is typically divided into five main phases: edge detection, cluster analysis, cluster matching, anomaly resolution and appliance identification [DK99]. At the edge detection phase, rising or falling edges in the load profile are detected and recorded as events of the detection system. At the method described in [DK99] the event is a spot (point

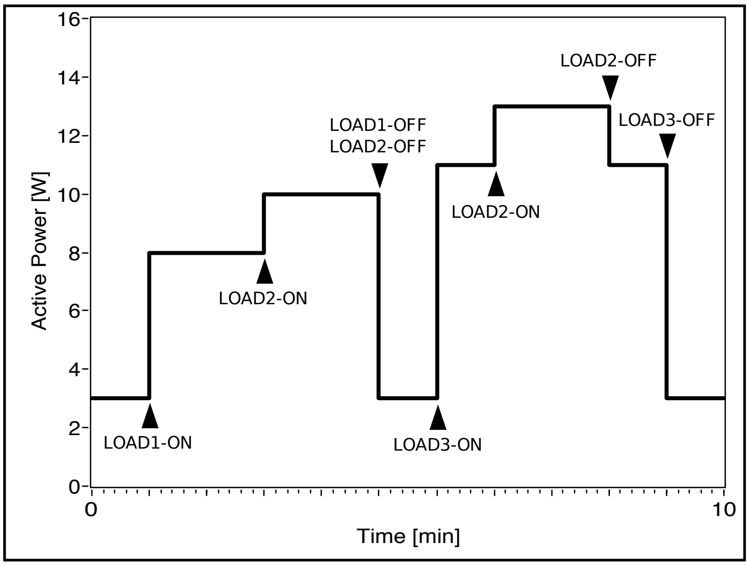

Figure 5.2: Appliance signatures on an arrgregated load profile. Three example appliances are switched on and off according to the time-markers in the profile. Source: [DK99]

with tolerances) in the two-dimensional real- and reactive power space ($\Delta P - \Delta Q$ plane). As a second step, the cluster analysis module will group the recorded events. After the grouping the cluster matching algorithm tries to pair up the created clusters. A created pair contains of a positive and a negative event cluster with the same edge magnitude [DK99] (According to the example in Figure 5.2 LOAD3 on and off events could be matched). The anomaly resolution step collects unpaired events and tries to match them again by taking the sum or difference of each (According to Figure 5.2 LOAD1 and LOAD2 could be matched with their common off event). The last step is the appliance identification which tries to match the paired clusters with an entry in a predefined load database [DK99].

The advantage of this algorithm type is that the described five parts can be distributed and executed individually in different hardware or software modules. For example one module can handle the edge detection and data communication, and an other including software algorithms based on pattern matching methods do the load identification [DK99]. An optimal balance of computational load, interruption safety, and redundancy aspects are other advantages which come with a distribution of the modules. This algorithm type delivers accurate results for so called two-step devices [DK99]. This means that the load profile of the appliance starts at a low power level (i.e. standby consumption), rises then in a hard edge to an activation power level and back to the base level after activation time. Extensions for multi-state load signatures and three phase loads were developed. The problems and disadvantages within this algorithm are the detection of ramps, fast fluctuations in the power level and other fading state changing events. For example a fan could generate different power states and transitions between the states depending on the rotation speed [DK99].

5.2.2 Model-Based Approach

The identification of loads by modeling them with the help of finite state machines is an effective way to avoid pure edge event observation. In this special application, hidden Markov models

(HMM) - a variant of finite state machines - will meet all the given requirements [Zia10, p. 69]. In HMM each state describes a steady power level through a probability density function. A steady state is defined as a level in the power signal where the level does not change over a certain threshold. If power changes above that tolerances the transmission into an other steady state will happen which is marked by an vector and a probability level in the HMM. Finally the curves above the states denote a probability of static staying in the actual state (sum of all transmission and feedback state must be 1) [Zia10, p. 70].

Figure 5.3 (a) shows an example power load profile which is a two-step load signature for a device. According to defined threshold levels the load profile implies a HMM model with three states. The transition between the three steady states will be reached by crossing the according threshold power levels. This results in a model transition path containing the following state sequence: S1, S3, S2, S1, S3, S2, S1 for this type of load. The transitions are also illustrated in Figure 5.3 (b). After determining the structure of the load model (see Figure 5.3 (c)) the next step involves learning the parameters of the model [Zia10, p. 75]. An available set of training data can be encoded into transmission probabilities between the states. The according transmission probability for a general state i to state j ($p_{i,j}$), can be calculated by taking the relative frequency of transmissions from i to j ($T_{i,j}$) with respect to total number of transmissions from state i ($T_{i,x}$) (see Equation 5.1 [Zia10, p. 76]). After a long term learning phase the states example load this probabilities could reach the values stated in Figure 5.3 (c).

$$p_{i,j} = \frac{T_{i,j}}{T_{i,x}} \tag{5.1}$$

The identification of modeled loads now happen with a HMM forward algorithm (see [Zia10, p. 60]). After a pre-processing the matching probability for all loads in the appliance database will be calculated. Finally the matching load model with the highest probability will be selected and retuned.

One of the most important advantages of the model-driven load identification process is that it is state-dependent. For the change from one power level to an other a steep edge is required. It is shape independent for the transmission between the steady states. Only threshold levels between the states must be defined for the change event. The steady power levels could be mixed with a relatively high noise magnitude (depending on threshold level definition) without creating any recognition errors (see Figure 5.3). A correct setup of states depending on the power profile of the load must happen. Finding an appropriate number of states could become a problematic limit of the algorithm. A high number of states makes the model complex and a low number of state definitions will harm the identification success. At edge detection based systems the magnitude of the edge is characteristic for the load and so this problem is not present. An other disadvantage is that a learning process for the calculation of the calculation of transmission probabilities is required. For some types of loads the modeling (especially the definition of the sates) and learning phase of the HMM could become a very complex activity. On the first side a big set of training data which reflects a high number of different possible signals and variations of model transitions can harm the flexibility of the model and decrease the load identification performance. On the other side an under-trained model can not catch enough characteristic signals and has also a bad recognition performance [Zia10, p. 77].

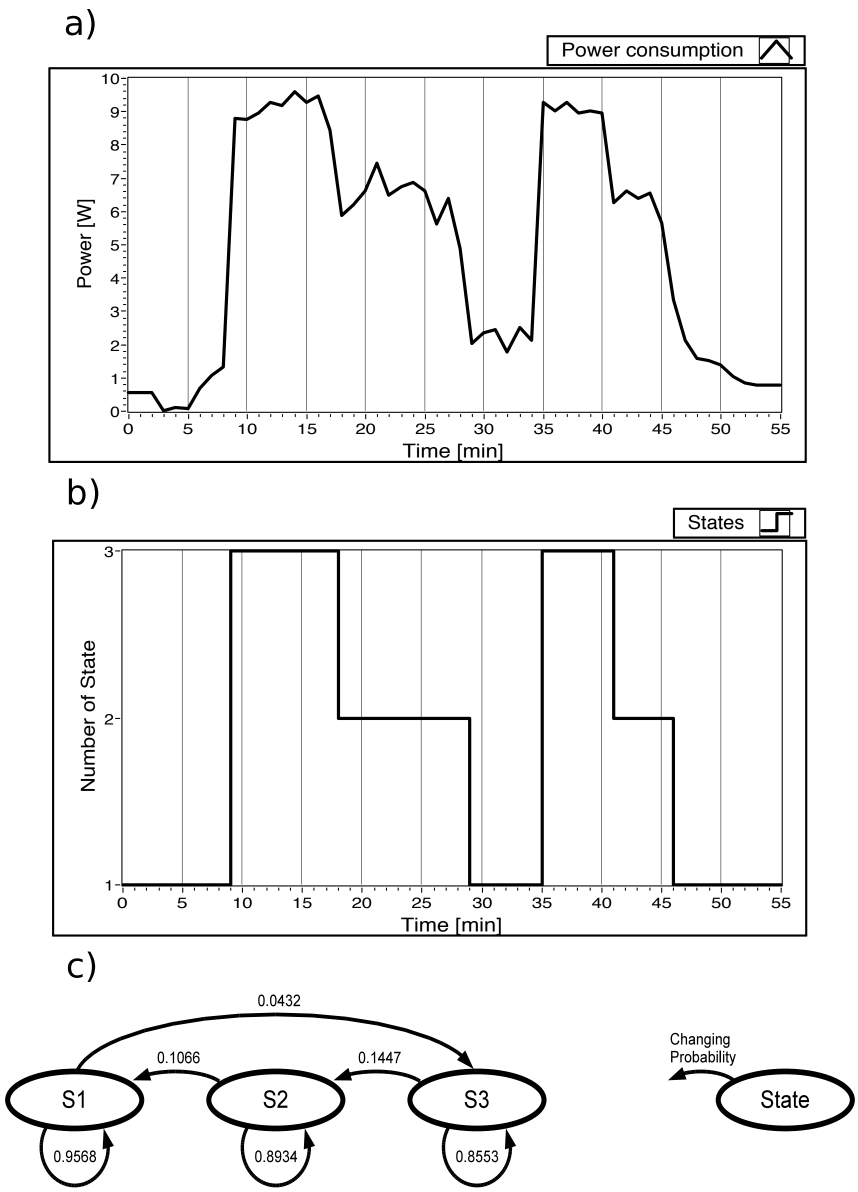

Figure 5.3: (a) Power load profile of an example load. (b) State transmission diagram for the power profile stated in (a). (c) Hidden Markov model structure with calculated transmission probabilities for a load with the power profile stated in (a). Source: [Zia10, p. 73 and p. 76]

5.2.3 Pattern Recognition and Dynamic Time Warping

Another effective approach for load detection is to use detection processes which are typical for application areas like handwriting, sign recognition or environmental sound recognition. They use sample matching and especially dynamic time warping (DTW) algorithms. The identification of the load is performed by calculating a ranking for the distance of the measured sample to a template [KP00]. The difference between DTW and sample matching is that latter compares only peak with peak values and valley with valley samples in the profile. DTW offers a more time flexible solution by matching corresponding samples in both areas (peaks and valleys) using dynamic programming algorithms. The load identification system consists of three main subsystems: pre-processing, distance computation and a load template database. The template profiles could be built at a training phase of the system. After the pre-processing module, which is used for an adjustment of the signal, the distance to each template sample in the database is computed. Therefor the calculation of distance vectors stated in [Zia10, p. 43] are required. The template sample reachable with the shortest distance vector is the matching point in the database.

As a result this type of load identification method returns an average distance vector of all samples between the signal under test and all templates in the data base. The minimum distance vector element has the highest probability for the identification. One of the design challenges of such a system is to find an appropriate length of the patterns. Loads can have extremely different profiles depending on their usage [Zia10, p.47]. For example the activation of a coffee machine over a day could vary from single event up to high frequently used times. An other disadvantage is that the calculation of the distance vectors for all database entries is a very computational expensive complex work [Zia10, p. 47]. This will require high performance central processing units (CPUs) and a fast database implementation. A big advantage especially of DTW is that the identification offers a high recognition potential for distorted loads. The identification can happen independent of timing because no timing parameters of the input load profile have an impact on the recognition result. In combination with an appropriate set of training data such pattern matching algorithms could be an very effective method for load identification applications.

5.3 Analysis of Typical Aggregated Load Profiles

This first paragraph will give some background information about the recorded household and the used measurement setup. All load profiles stated in this work were recorded at an average single-family household in Upper Austria. At the time of recording three adult persons lived in the household at 120 square meters of living space. The date of the recorded aggregated profile under discussion and stated in Figure 5.5 is the third April 2011. The average temperature was 15 degree Celsius and it was a dry high-pressure weather sunny day [24]. Because of well thermally isolated walls and windows there were no need of electrical heating in the evening. The third April was a Saturday and all three family members stayed at home the whole forenoon. The largest electrical loads installed are: electric cooker, two freezers, a fridge, a dishwasher, an electric water pump, a sauna and a whirlpool. The rest of the appliances are standard devices for lightening, kitchen equipment, additional heating elements, home entertainment, computers, etc.

Figure 5.4 illustrates the electrical parameters of the household and the structure of the measurement setup. The house is supplied via a overhead low-voltage distribution line. The three phase voltage levels of the supply are 400 V between the phases and 230 V between the phases and the neutral conductor at a frequency of 50 Hz (according to the Austrian national standard). The customers service connection is geographically located between other households approximately in the middle of a distribution grid branch. The distance to the next transformer station, which is interesting for the analyzation of the voltage levels is about one kilometer. The measurement setup itself (Figure 5.4) in the house includes two devices. A laptop is connected to an AMIS smart meter via an optical serial interface (see Section 6.3). The measurement software reads all displayed information from the smart meter and stores them in an appropriate Extensible Markup Language (XML) file (see Section 6.3.3). All appliances in the household are electrically connected behind one single smart meter. There are no additional branches into two or more

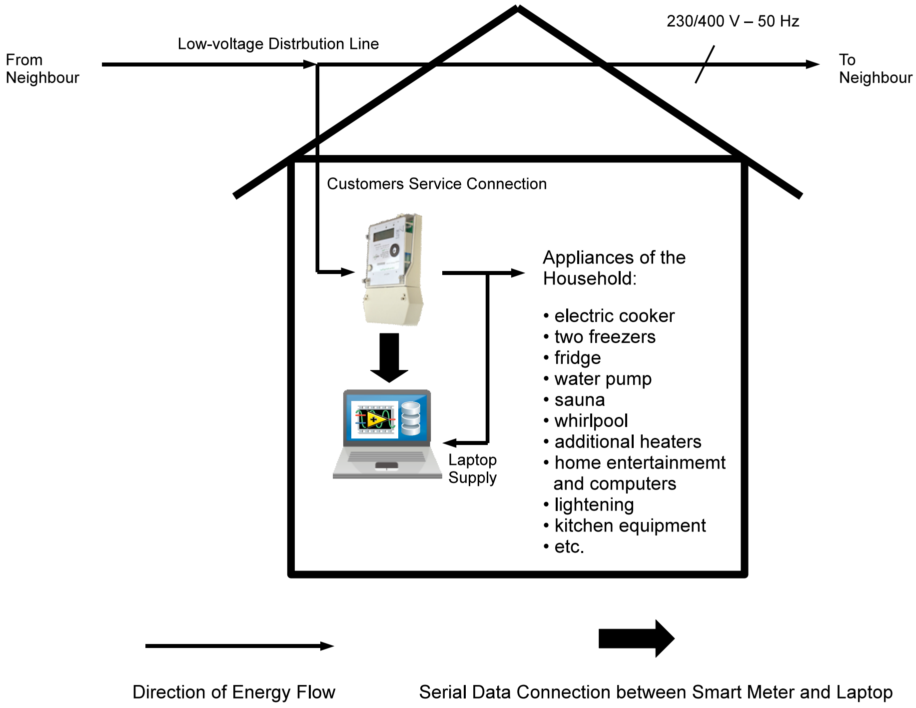

Figure 5.4: Measurement setup for aggregated load profile recording. Source: author

meters for separated billing in the house. A fact that should be kept in mind is that the measurement laptop is also supplied behind the metering device and its consumption is always present as a measurement error in the recorded profiles.

For the first analyzation of the profiles 24 hours will be point of interest. At a second examination details will be discussed and focused out of the aggregated load profile. Figure 5.5 shows the real power profile of the third April 2011 of the household under test.

Time interval one shown in Figure 5.5 has a relatively high average power level of 400 W from 1:30 to 8:00 in the morning. This high consumption offset level caused by a PC which was turned on during night. This results also in a constant current level of 1.7 A in line two in this time window (illustrated in Figure 5.5). All other appliances activated during night are overlapped with this constant offset power level. The significant peak at 2:45 will be described in more details below.

The structure of the power profile in time interval two shows a typical increase of the power level at the midday period between 10:00 and 13:30. In this time period kitchen appliances with a high power consumption were activated for preparing lunch. Especially between 12:00 and 13:00 (this is time period 3, not marked in Figure 5.5) there are high frequent fluctuations in the power profile which indicates thermostat controlled cooktops. The same structures at the same time intervals can also be found in the current profile of line one illustrated in Figure 5.6. That indicates that the corresponding heating plate is connected to line one and turned on and off periodically to hold a constant temperature.

With advancing time the consumption level sinks to a minimum at 15:00. This minimum value of 110 W is the significant value for the base load parameter of this profile. At time interval four from 15:00 to 23:00 nobody was at home. This period contains only overlapped load profiles from appliances which are permanently activated (fridge, freezer, etc.) or turned on automatically (heating elements, etc). The significant peak at 18:00 will be explained in detail below.

The typical increase of the consumption in the evening which will happen approximately at 18:00 on weekends is delayed to 22:30 in this example. There is a large rising consumption at 22:30. The current for this appliances came only from line two and three which corresponds to single phase socket connected devices (see Figures 5.7 and 5.8). The current profile for line one is totally flat in this time period (see Figure 5.6). At about 23:30 the water regeneration phase of the installed whirlpool started (see time interval 4 in Figure 5.5).

Table 5.1: Time intervals for structural analyzation of the aggregated power profile

Interval Nr.	Time Interval	Denotation
1	00:00 - 8:00	Night and early morning
2	10:00 - 13:30	Midday peak
3	12:00 - 13:00	High frequency fluctuations due to thermostat controlled cooktops (Not shown in Figure 5.5)
4	15:00 - 22:00	Afternoon and evening
5	22:30 - 00:00	Night

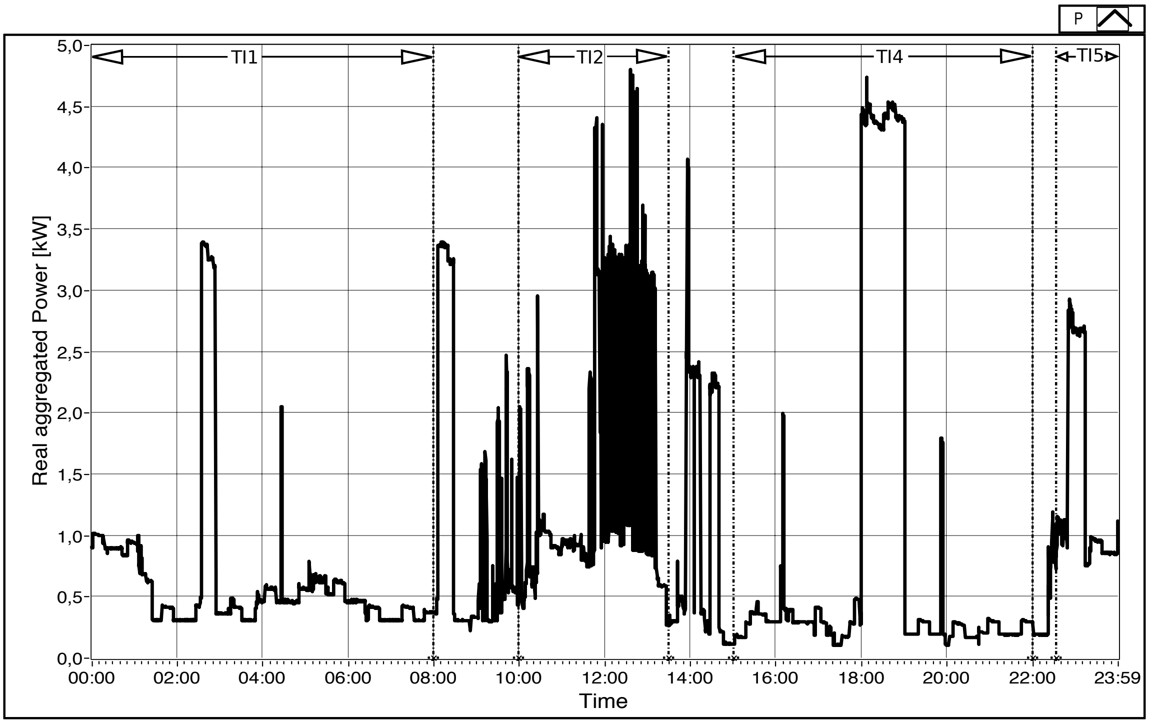

Figure 5.5: Aggregated real power load profile recorded for one day (TI in the illustration stands for Time Interval). Source: author

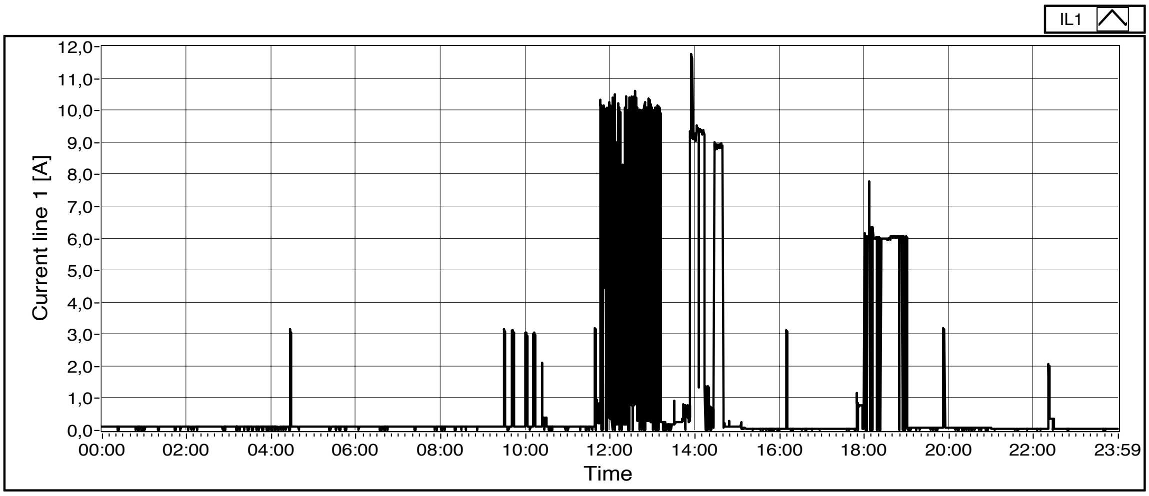

Figure 5.6: Line 1 current profile recorded for one day. Source: author

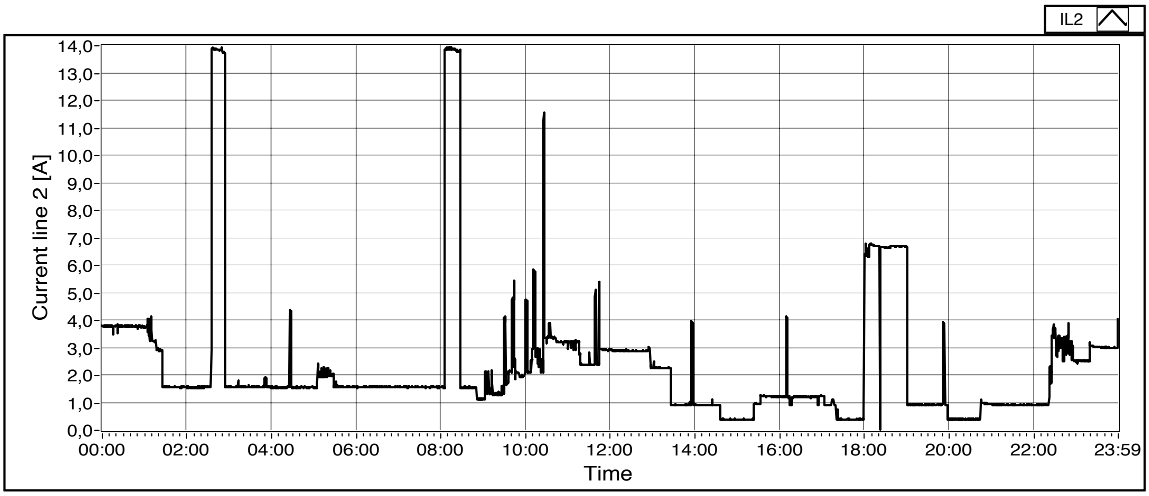

Figure 5.7: Line 2 current profile recorded for one day. Source: author

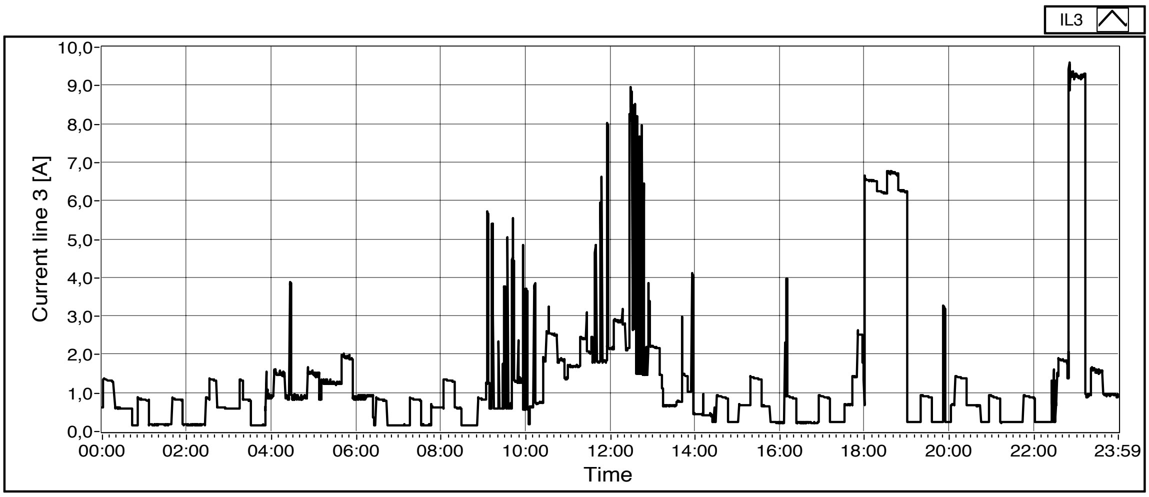

Figure 5.8: Line 3 current profile recorded for one day. Source: author

For a further analyzation of the recorded load profile some sections in combination with current and angle information will be focused. This analysis builds the knowledge base for the implementation and database configuration. Characteristic peaks and structures in the load profile can be assigned uniquely dedicated appliances.

First the power peak at 2:45 illustrated in Figure 5.5 will be discussed. According to Figure 5.9 the power peak has a length of 20 minutes and a hight of 3400 W. Its profile is nearly a perfect rectangular shape and the upper level is held constantly for the whole 20 minutes. In Figure 5.10 all three line currents are shown at the same time interval. The current of line two (bold non broken line in Figure 5.10) follows the shape of the power signal (peak hight is 13.9 A). The other currents are approximately zero. This indicates a single phase load which is connected to L2. Finally Figure 5.11 shows the angle information for the time interval. According to current in line two the angle between voltage and current has a negative edge to 4.4 degree between 2:35 and 2:55. The high consumption and the low phase angle indicates that this peak is caused by a heating element of the installed whirlpool. At the recorded day exactly the same heating procedure occurred again at 8:05 to 8:25.

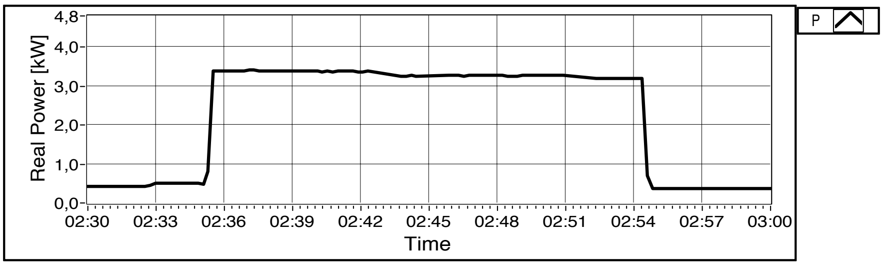

Figure 5.9: Power profile for the heating element of the whirlpool. Source: author

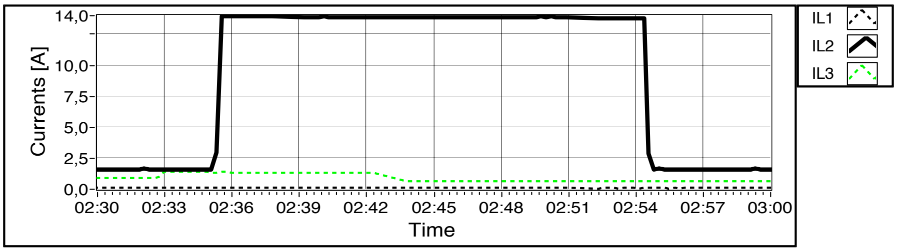

Figure 5.10: Current profiles for the heating element of the whirlpool. Source: author

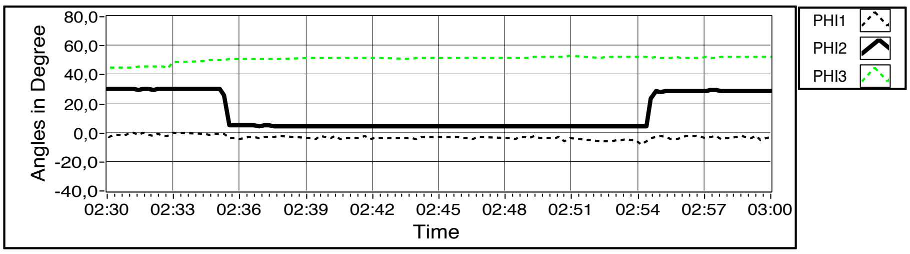

Figure 5.11: Angle profiles for the heating element of the whirlpool. Source: author

Next the significant power peak between 18:00 and 19:00 will be discussed. It has a hight of 4500 W and the duration of the activation is nearly exactly one hour (these parameters could be seen at Figure 5.12). Figure 5.13 shows the according currents of all three lines and the dotted one is the current of the neutral conductor. All three line currents have approximately the same level of 6.5 A and the neutral current keeps at a low level (about 1 A). The angles between voltages and currents shown in Figure 5.14 are near by zero degrees for the activated time period. So this load can be identified as a three phase resistive heating element which is exactly switched on at 18:00 for one hour. The real device behind this profile is a domestic hot water boiler which is triggered by a timer.

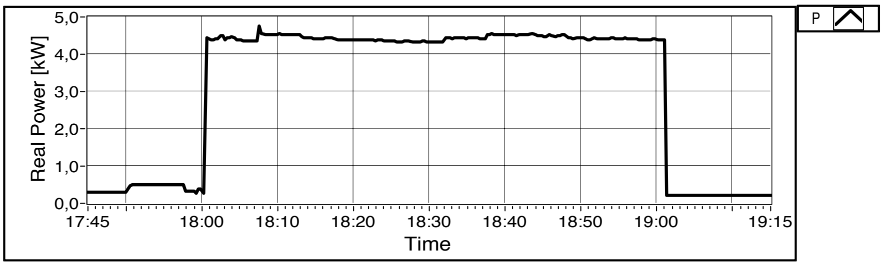

Figure 5.12: Power profile for the domestic hot water boiler. Source: author

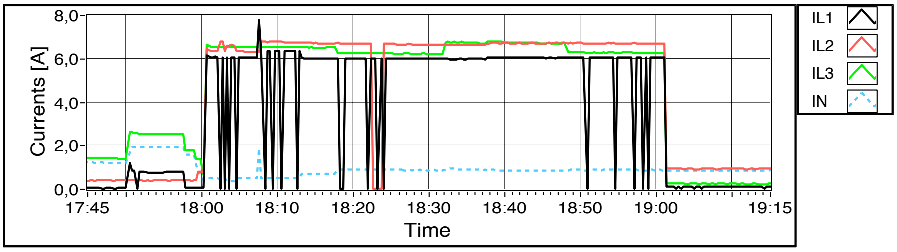

Figure 5.13: Current profiles for the domestic hot water boiler. Source: author

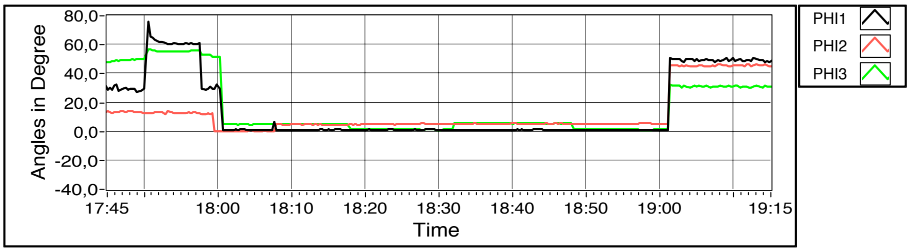

Figure 5.14: Angle profiles for the domestic hot water boiler. Source: author

At Figure 5.15 the electric water pump device of the household will be analyzed. It is a three phase load with a peak current of 3 A. The three line currents are illustrated in Figure 5.15 as broken lines with the right scale. The left scale and the non broken lines are the angle information of the lines. They have a value of 35 to 40 degrees in the interval of activation. This indicates the load as a inductive motor device. The time of activation is triggered by the water level in the reservoir. As illustrated in the chart the pump needs about 2 minutes to fill the reservoir from minimum to maximum level. The power consumption at activation can be read from Figure 5.5 and the peak level at 16:10) is 1600 W. The activation of the pump occurred periodically which can be read out of the current peaks of L1 illustrated in Figure 5.6. Activation times are 4:27, a series of activations happens in the morning at 9:30, 9:42, 10:01 and 10:12, at 11:38, 16:10 and 19:53. The activation time of this device is due to the exactly defined water levels constant, but the time of activation varies over the day depending on the water consumption.

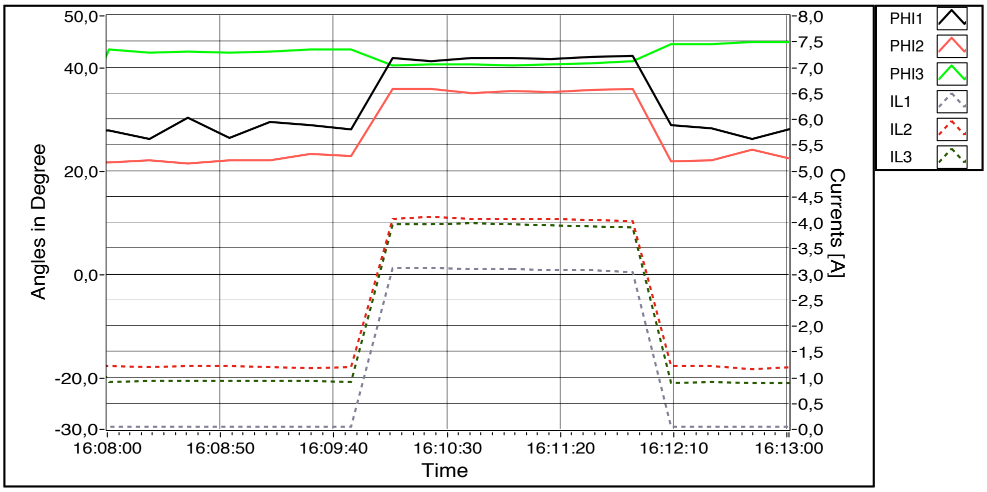

Figure 5.15: Current and angle profiles for the water pump. Source: author

5.4 Load Profile Analysis of Typical Home Appliances

This section describes some load profiles of typical household devices. The key elements for the identification of loads will be pointed out and typical characteristics of domestic home appliances will be shown. The measurement setup for the recorded load profiles in this section is similar to the structure described in Figure 5.4.

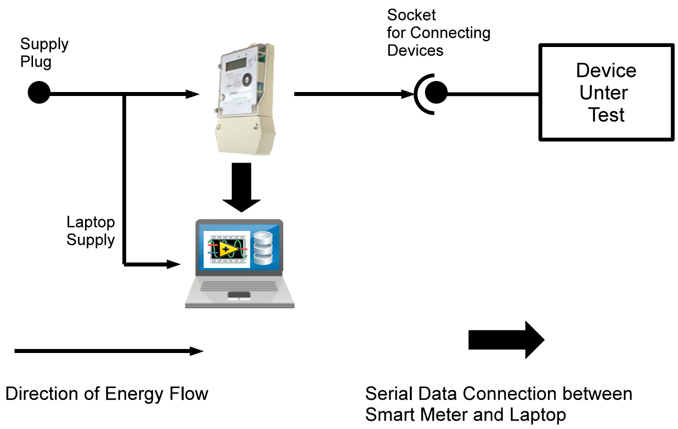

Figure 5.16: Measurement setup for recording load profiles of connected home appliances. Source: author

Figure 5.16 illustrates the configuration for the measurement. A second sub smart meter device is used for the recording. It is equipped with a standard plug and a socket at its ends. The supply of the laptop can now happen before the smart meter. So the error impact of the laptop into the measurement result is zero. The appliance under test could be connected to the output of the smart meter by plugging into the installed socket. The whole setup is built up as a single phase system for standard household devices. The recorded datapoints are transmitted via an optical serial link to the laptop and stored in a XML data file (see Section 6.3.3).

For a collection of load profiles of typical household devices to appendix Section 9.1 will be referred. In this section additional details of some load profiles of the collection will be pointed out.

The halogen lamp which load profile is shown in Figure 9.3 and 9.4 has a internal structure of two components. A switch-mode power supply which transforms the voltage from 230 V AC to 12 V DC, and the halogen bulb itself. An interesting effect for modeling, characterization and measurement of this load is that the appliance has a falling edge in the angle signal between voltage and current vector at an switch off event. Due to the capacitive elements in the switch-mode power supply a negative peak of reactive power (generation of reactive power) according to Figure 9.3 second 85 happen. The angle difference of the levels is -38 degrees from 17 degrees inductive to -20 degrees capacitive characteristic. This results in a level of -5 VAR reactive power. For the measurement results two possible signal structures for the falling edge of the real and reactive power can be found. One possibility is that the smart meter takes the measurement sample during the effect and delivers a negative reactive power level for one sample period like stated in Figure 9.3. The other possibility is that the effect could not be sampled and is not visible at the recorded load profile, then the falling edge looks exactly like the rising one. These two different shapes of the switch off event must be considered at the algorithm implementation for an automatic load detection.

Figure 9.5 and 9.6 illustrates a general valid typical load profile structure for cooling devices. A two-level controller thermostat will hold the inner temperature of the appliance at a constant level. The period times for switching on and off depends on the adjusted temperature and the temperature difference between inside and outside of the cooling device. At the turn on period the shape of real power profile is rectangular overlapped with a negative exponential function. The exponential decrease of the real power is caused by an increasing efficiency of the cooling liquid at a rising temperature difference between in- and outside [23]. The reactive power during the on period stay at a constant positive level due to the independent inductive characteristic of the compressor.

Finally for this analyzation two types of TVs included in the load profile collection will be compared. Figure 9.9 and 9.10 illustrates the load profile of a state of the art plasma TV and Figure 9.17 and 9.18 a ten years old cathode ray tube TV. The plasma TV has a capacitive characteristic, which results in a negative reactive power consumption shown in Figure 9.9 between 175 and 560 seconds. This electric capacity is caused of the internal structure of the plasma display of the TV. The cathode ray tube TV illustrated in Figure 9.17 has a inductive characteristic due to the internal coils of the tube. The two devices have approximately the same real power consumption of 115 W but different reactive power characteristics. This could be a significant parameter for the determination between the two appliances at an automatic load identification algorithm (for example at an edge based $\Delta P - \Delta Q$ plane approach explained in Section 5.2.1).

5.5 Conclusions for the Implementation

The analysis of aggregated load profiles and individually measured home appliances described in the last two subsections is the basis for the implementation of the load identification algorithms.

The main challenge of the identification algorithm is to match patterns and significant characteristic parameters of stored load device templates into a live streams of aggregated load data. Based on the examples described above some important facts for the identification of loads will be pointed out.

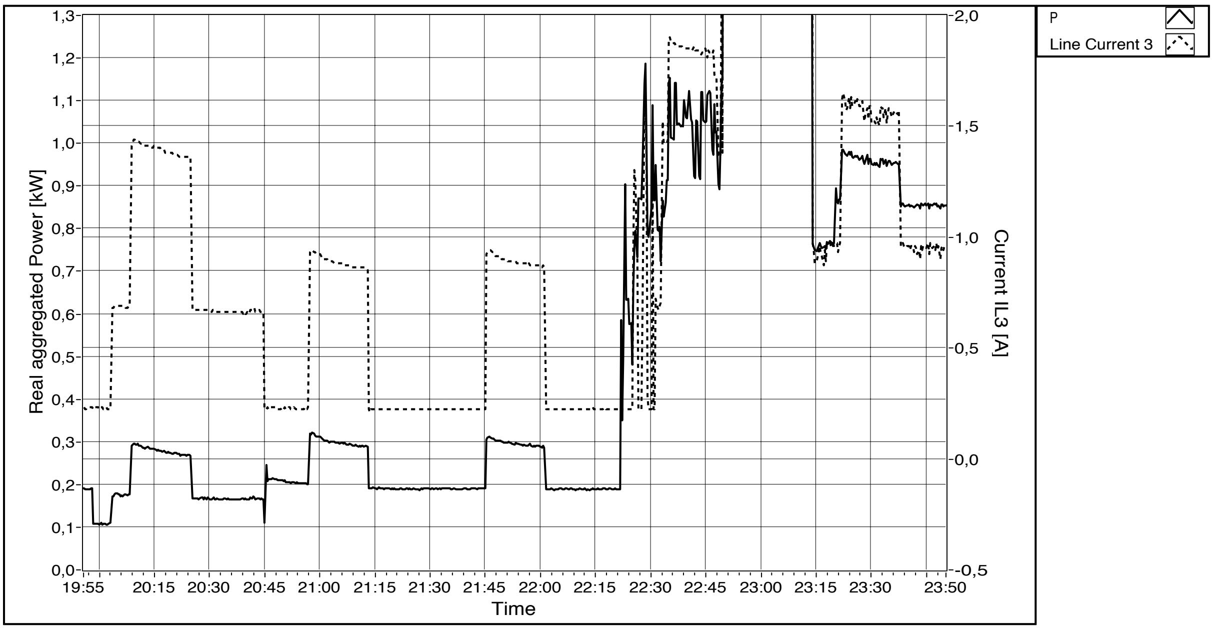

Figure 5.17: Freezer load profile characteristic in aggregated load profile. Source: author

Figure 5.17 shows a zoomed section for the real power and line three current from the aggregated load profiles illustrated in Figure 5.5 and 5.8. As a template for the recognition of the freezer the profile shown in Figure 9.6 can be taken. At a human intuitive consideration of the template and the aggregated profile, the freezer device can be matched at 20:08, 20:56, 21:44 and 23:20. For an automatic edge based algorithm the corresponding detection rule can be realized as: If a rising edge happen in real power of 120 W and at the same time a rising edge of 0.647 A in the line current three profile and after a time period of 17 minutes a falling edge of 100 W and 0.615 A, the freezer can be identified. At a more detailed consideration of the activation times (rising edges) there is a leak between 21:44 and 23:20. The time distance between 20:08, 20:56 and 21:44 is constantly 48 minutes and the distance between 21:44 and 23:20 is 96 minutes (exactly 48 times 2). So theoretically there must be an activation of the freezer at 22:32 (21:44 plus 48 minutes). An indication for the activation of the freezer at this time is the negative exponential progress of line current three for 17 minutes. The real power consumption for this period of time is overlapped with other appliances and totally unstructured. As a result of the analyzation and matching of the freezer template three facts can be pointed out for the automatic identification of such automatically and cyclic activated appliances:

- For an edge based algorithm the hight of the rising and falling edges independent of the offset level is important. For single phase loads there must be equivalent edges at the power profile and at the same times at one of the line currents.
- The time difference between rising and falling edge (on and off event) is a characteristic parameter of an appliance.

- The cyclic period time of the activation is characteristic value of automatically activated devices.
- The progress and shape of the signal (real power or current) is characteristic for an appliance type (i.e. negative exponential progress indicates a cooling appliance).

Another detection method of appliances, which are not periodically activated (i.e. the water pump) will be a edge based $\Delta P - \Delta Q$ plane algorithm. The actual state of real power P and reactive power Q will be calculated continuously. If a positive edge in P and Q values happen (for the pater pump example ΔP is 540 W and ΔQ is 460 VAR calculated from Figure 5.15) and this states will be held for a defined steady time the appliance will mark a point in the $\Delta P - \Delta Q$ plane. This could be compared with every template appliance registered in the load database. If the matching probability is greater than a threshold value the load will be identified. This results in the following facts for the identification algorithm design:

- For the identification of non cyclic activated appliances the $\Delta P - \Delta Q$ space method can be used. A rising edge in the real and reactive power profile, and if the new states are held for a defined threshold time indicates a switch on event of a load.
- With a $\Delta P - \Delta Q$ based algorithm devices with the same amount of real power consumption can be distinguished (See the example with two TVs in Section 5.4).

6 System Architecture and Implementation

This chapter describes the architecture and design of the implemented soft- and hardware components of this project, that fulfill the goal of realizing an integrated energy feedback and load identification system (see scope description in Section 1.2). The realized system is divided into several sub-tasks (i.e. read smart meter data, storage and interpretation of data, load identification algorithms, graphical user interface, etc). The sub-tasks, their interaction and data transfer with other modules will be mentioned and described. The main focus for the system is on the interpretation and analyzation of load profiles (see Section 5.5), and realizing an accurate load identification architecture.

6.1 Requirements on the System

The main parts of the implemented system are acquirement and storage of measurement data, identification of loads in a measured aggregated load profile and presentation of smart meter and load identification information. Each part must fulfill common design aspects and properties. The following requirements and property definitions are valid for the whole system design. Detailed functional requirements for sub-modules will be specified at the first paragraphs of the detailed descriptions.

- ***Modularity:*** All sub-modules must be executed and realized in different processes. This enables independent development, debugging and testing of the parts. If module tests are accomplished successfully the integration and data interaction of the parts can be tested. The permanent developement, test, and integration of the specified functions (top-down design) leads finally to the desired application.
- ***Expandability:*** The update of modules and addition of new functions (i.e. new load identification methods) in the design must be possible. Therefor the framework must be scalable in an appropriate way for managing the new modules. Data and control information must be buffered and shared for acting as in- or output data at the new implemented functions.
- ***Platform and System Independency:*** Software implemented in this project must be designed operating system portable. Standard libraries and tools must be used which are available on different state of the art operating systems (Linux, MacOS, Windows - 32 and 64-bit versions). An additional requirement is target platform independency. The

application must be executable in in state of the art embedded platforms. Target dependent compilers build a binary for embedded implementations. (i.e. Embedded Linux, Windows Embedded, etc).

- ***Applicability:*** The realized system must be applicable in standard households which are equipped with a state of the art smart meter device (see Section 3). Load identification functions must deliver appropriate results independent of the data source. Standard smart meters with an IEC 62056 conform digital interface connection must be applicable for the system.
- ***Usability:*** According to Section 4 and Illustration 4.1 the system must be an easy readable, well arranged and an informal platform for customers. The user interface must provide functions for system configuration and presentation of smart meter data and load identification results.

6.2 Overview and Architecture

To meet common requirements and fulfill the scope of the project a system architecture illustrated in Figure 6.1 is chosen.

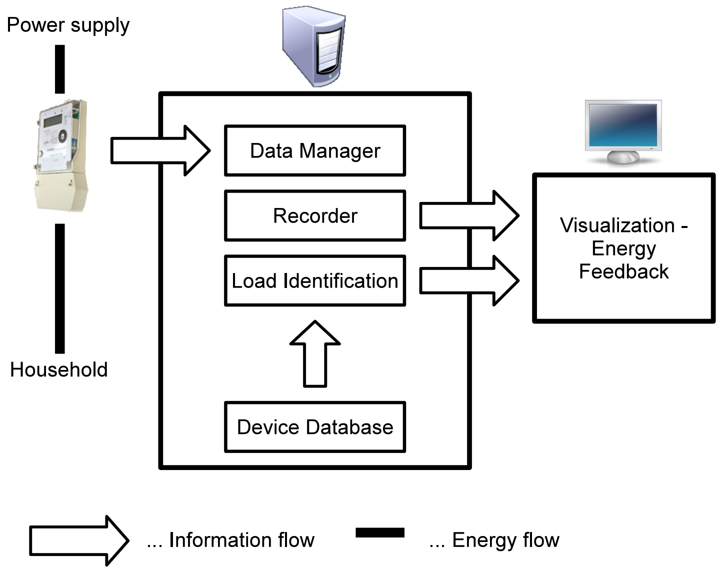

Figure 6.1: System architecture and design overview. Source: author

The system is divided into three parts. The first part is the smart meter which acts as a data source is an AMIS device. The second part is a collection of software modules, which are responsible for reading data, storage and load identification. The third part is the visualization of all results and measurement values at an appropriate graphical user interface. The smart meter has a data link to a software-implemented data manager module. The received smart meter information will be stored in a file (see Section 6.3.3) and actual consumption values will be be transferred to the visualization unit for energy feedback. With the help of stored data information a load identification software module tries to find activated individual appliances out of the aggregated load profile. The load identification module requires a device database file which holds modeled load templates for matching and compare algorithms. The results of the load identification process will be also displayed at the GUI. All software sub-modules are implemented in separate processes. So they can be executed and tested totally independently. This enables the possibility that software modules especially the GUI need not to be executed on the same system. Recorded values and load identification results can be tunneled via ethernet to an other computer system. The realization of software modules in processes requires a system for inter-process communication. The software architecture and its communication channels are illustrated in Figure 6.2.

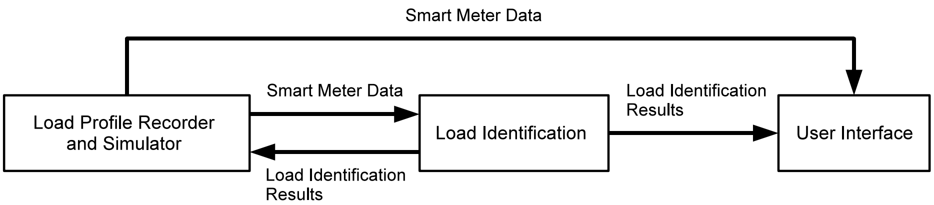

Figure 6.2: Software modules and communication channels. Source: author

The communication between the modules is realized with queues from the standard Portable Operating System Interface for Unix (POSIX) library [21]. The queue elements offers functions for buffered transfer of user typed data-blocks between two processes. The Load Profile Recorder and Simulator module is able to read data from a recorded load profile file or simulates a test load profile. These smart meter data is transferred via two queues, to the User Interface module for presentation and to the Load Identification module for nonintrusive load detection. The results of the load identification are transferred via queues back to the Load Profile Recorder and Simulator for an analysis of the identification results, and again to the User Interface for the presentation of detection results.

The usage of wireless technology for transferring smart meter and individual power measurement data to a base station, like states in the scope description (Section 1.2) is not mandatory for fulfilling the requirements of energy feedback and electrical load identification. The distribution of sensor nodes in the household and a wireless transfer of smart meter data to the system (left arrow of Figure 6.1 is a wireless data link) moved to a future work of this project. The realized system in this work is implemented according to Figure 6.1. The development of the system happened on a standard PC with Linux (kernel version 2.6.35.6-45.fc14.i686, fedora FC14 distribution) as operating system. The POSIX library is used in version 8.2.3. All software modules are developed in C++ and compiled on the system with the GNU C compiler (GCC) version 4.5.1. Due to the implementation in C++ and the usage of standard libraries which are available for each state of the art operating system the software could be ported to other platforms and other operating systems. The conversion of the software into an embedded platform version moved also to a future work activity. The usage of tablet and industrial PCs offers performances of standard desktop i.686 architectures and requires no cross compilation of the application and libraries. So a pre-compiled PC binary of the application can directly be executed.

6.3 Aggregated Load Profile Measurement

The analyzation of load profiles described in Section 5.3, 5.4 and 5.5 requires reading and recording datapoints of the AMIS smart meter. Data transfer between PC and meter device is realized with a serial link via an optical reading head. The reading software on the PC must handle the serial connection at the standardized IEC 6256-21 protocol and the storage of read data.

6.3.1 Functional Requirements

This section describes functional requirements on the system for reading data from the smart meter. This is a description of functions that must be implemented for a successful read and storage of data in an appropriate way. The modularization of the functions and its implementation in soft- or hardware will not be defined in this section.

1. The read system must be compliant with the IEC 62056 - part 21 standard which is a protocol for data exchange for meter reading, tariff and load control [EN602]. Protocol data must be sent and received via a serial link between PC and smart meter device.

2. The identification and analyzation of data points must be conform to the Object Identification System (OBIS) for smart meter devices. The OBIS standard which is part 61 of the IEC 62056 standard describes the labeling code of the received values [EN607].

3. The Application must read full smart meter data sets which include every information also available at the display of the meter.

4. The recording of values must be configurable for a flexible storing of different types of read values. The identification of value descriptions must be solved with OBIS codes.

5. A lookup table for OBIS code entries must be implemented to get a human readable text for each type of value.

6. For the timing of the reading operation a maximum cycle time of 15 seconds must not be exceeded.

7. In the case of communication errors or a connection cut off of between meter and computer the application must not lock in a undefined or waiting state. Every action on the serial port must be limited in execution time by a timeout to avoid locks at communication errors.

8. The storage of the data must use a well defined file format and the creation of a rolling file appender system is required (avoid large data files).

In the next sections the parts of the read our system implementation will be described in detail and how to fulfill these requirements is discussed.

6.3.2 Hardware Overview and Setup

The measurement setup for recording aggregated and individual load profiles is illustrated in Figure 5.4 and 5.16. This section will describe the connection between optical interface of the smart meter and recorder software in more detail. For reading data in this project the optical D0 interface of the AMIS smart meter is used (see Section 3.2). The physical properties of this interface are defined in the IEC 62056-21 standard [EN602, p. 12]. An optical reading head can be magnetically attached over the transmitter and receiver of the meter (left side of Figure 6.3). An infrared diode is connected with the transmit signal, and an infrared detector with the receive signal of the serial communication line. Optical receiver and transmitter must be positioned opposite to each other for a successful data communication. The alignment of the reading head and the avoidance of parasitic environmental light are two important factors for minimizing communication errors. In this project the receive and transmit signals leads to an additional electronic component between smart meter and PC. An integrated circuit (MAX231CWE) handles signal levels and is used as drivers for the receiver and transmitter semiconductors (see Figure 6.3). The synchronization between PC and the signal level transformer box is realized with ready to send (RTS) and cleared to send (CTS) synchronization lines. Also optical reading heads without this additional level drivers and synchronization lines are available. In this case hardware flow control must be managed by the communication software on the PC. This setup fulfills the physical connection of the IEC 62056-21 standard which is stated in application requirement one.

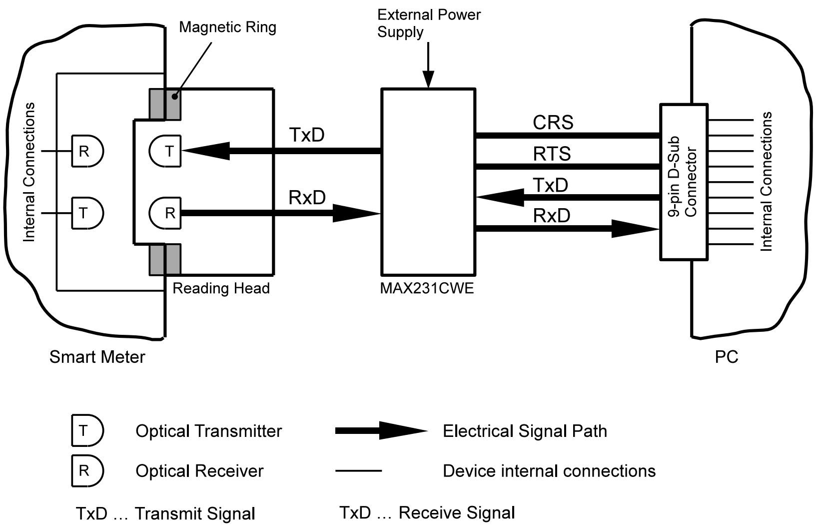

Figure 6.3: Connection between smart meter and PC. Source: author

At this setup all standard data transmission speeds (300, 600, 1200, 2400, 4800, 9600) up to 19200 bit per second are possible. Transmission of signals is half duplex and the character format on the data lines is realized according to IEC 1177:1985 standard. The format for one data frame is defined that it contains of 7 data bits, 1 start bit, 1 stop bit and 1 parity bit.

6.3.3 Load Profile Measurement Software

For this project the measurement software is implemented in the graphical programming language G [1]. The company National Instruments offers LabVIEW [19] as an integrated development environment for the G language. The software could be also implemented in any other programming language but for a following analyzation of data LabVIEW offers powerful diagram, measurement and file storing functions. The application consists of three parts: A read algorithm which handles the data communication via the serial port. An identification part which formats the read values and adds a human readable OBIS description. And a file writer part which is responsible for a formatted file writing of measurement values.

Figure 6.4 illustrates the sequence of operations executed by the data read software algorithm. The serial connection will be initialized at 300 baud with seven databits, one start-, one stopbit, an even parity bit and RTS/CTS flow control parameters. If no error occurred a request message

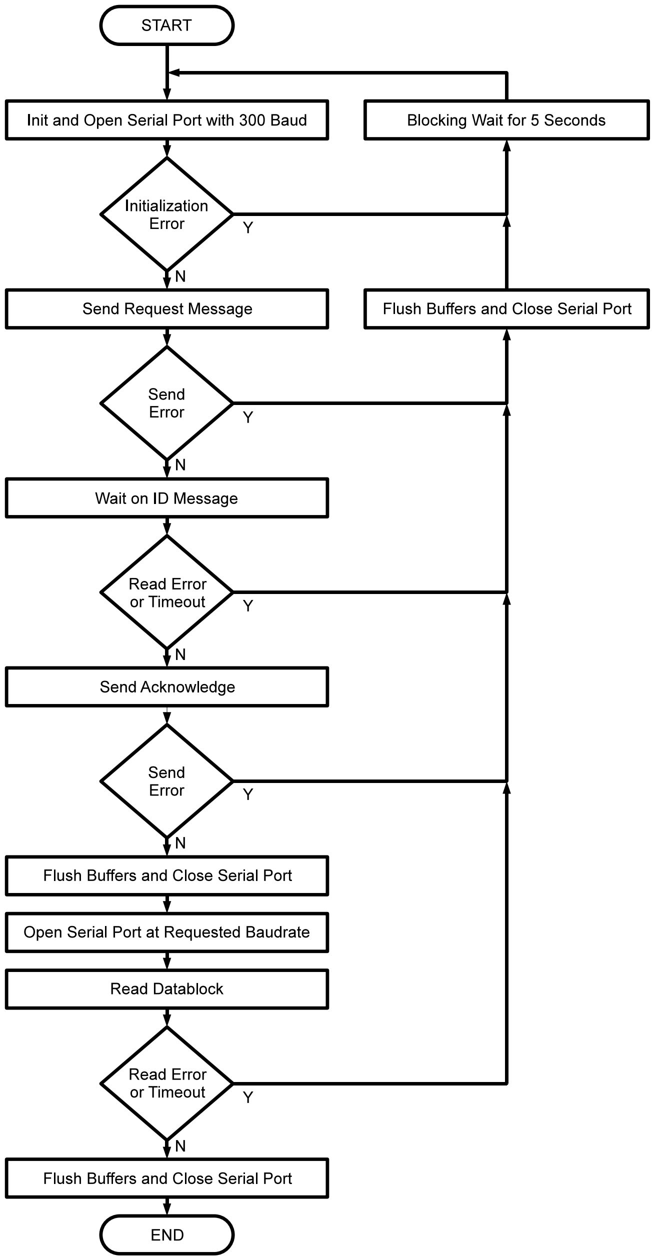

Figure 6.4: Program flow chart of the read alotithm. Source: author

will be sent to the smart meter. After an successful send process the read application waits for an identification (ID) message of the meter. The ID message contains a baud rate identification field for baud rate switching. This ID message must be acknowledged and then the smart meter changes its serial baud rate according to the code of the baud rate identification field. If no read timeouts at the receive process and any send error at the transmit process occurred the PC close the serial interface and reopens it at the arranged baudrate. Normally the arranged baudrate is the maximum specified speed of 19200 bit per second. Now a full smart meter data block with all available information will be transmitted at the high data rate. If no read errors returned and the data block has a correct format, the serial buffers will be flushed and the port will be closed. If any error or timeout in this send and receive action occurred, buffers will be flushed the port will be closed and the read of data will be tried again. This behavior fulfills requirement Nr. 7. At 19200 baud a full datablock, which has a size of 16.4 kilobyte needs approximately 7 seconds to be transmitted, depending on the number of reading errors. Practical measurements have shown that at an established connection the transmission of a full datablock never took longer than 10 seconds, which fulfills requirement Nr. 6. For detailed information about the coding and fields of the messages the IEC 62056-21 [EN602] standard page 18 will be referenced.

At the next step the received data block must be interpreted. The format of the block is string oriented and consists of OBIS code, value, unit and each line is finally closed by an newline and carriage return control sequence. OBIS is standardized in IEC 62056 part 61 and is a standard coding system for all types of smart meter data in different business fields. Full OBIS code tables are available at [17]. Only a selection of the received data values will be recorded. The selection in this project minimally consists a full smart meter block like stated in Table 6.1. Additional OBIS data codes could be attached into the recording list. For the OBIS codes in the record selection value and unit will be searched in the received data block and a human readable description of the OBIS code will be indexed in a look-up table.

Table 6.1: OBIS codes, description and unit for a minimal smart meter data record

OBIS Code	Description	Unit
0.9.1	System time	String formatted HH:MM:SS[1]
0.9.2	System date	String formatted DD:MM:YYYY[1]
31.7	Line 1 current	Ampere
51.7	Line 2 current	Ampere
71.7	Line 3 current	Ampere
91.7	Neutral conductor current	Ampere
32.7	Line 1 voltage	Volts
52.7	Line 2 voltage	Volts
72.7	Line 3 voltage	Volts
1.7.0	Active power import	Kilowatts
1.8.0	Energy import	Kilowatthours
81.7.4	Angle information $arg(I(L1)) - arg(U(L1))$	Degree
81.7.15	Angle information $arg(I(L2)) - arg(U(L2))$	Degree
81.7.26	Angle information $arg(I(L1)) - arg(U(L1))$	Degree

The last part of the smart meter data reader software is an Extended Markup Language (XML) file writer. The datapoints selected for recording will be temporary stored in a structure element. This structure consists of the OBIS code the readable description of the code, value, unit and a validation flag. The collected datapoint structures are appended to the XML file as a new data set at each recording cycle (A recording cycle consists of serial reading, interpretation and storing of data). Exactly the same structure can be found at the XML data file (see Listing 6.1). The upper tag is named "Profile" with the recording date as an attribute. The "Profile" contains

[1] HH:MM:SS stands for 2 digits hour, 2 digits minutes and 2 digits seconds (i.e. 12:34:23). DD:MM:YYYY stands for 2 digits day, 2 digits month and 4 digits year (i.e. 05.03.2011).

"Data Sets". A data set contains all "Data Points" stored at one recording cycle. The XML file format can be opened, edited (only a text editor application is required) and computed at any state of the art operating system and target platform. A property of the file writer module is that each day a new XML file will be created. The files can be indicated by the date property at the "Profile" tag or by the naming of the crated file ((YEAR)(MONTH)(DAY).xml).

Listing 6.1: XML recording file structure

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Profile date="20.03.2011">

<DataSet>
  <Timestamp>20:00:45 -20.03.2011</Timestamp>
  <TimeStampValid>TRUE</TimeStampValid>

  <DataPoint>
    <PointValid>TRUE</PointValid>
    <ObisCode>31.7</ObisCode>
    <Description>L1 current</Description>
    <Value>1.57</Value>
    <Unit>A</Unit>
  </DataPoint>

  <DataPoint>
     ...
  </DataPoint>
   ...
</DataSet>

<DataSet>
   ...
</DataSet>
 ...
</Profile>
```

6.4 Load Database Structure and Access

The load identification functionality requires templates for matching. These modeled template loads are stored in a database, which must be available at each load identification request. If availability is not given, events in load profiles can be detected then, but not interpreted and matched with an appliance. The load database in this project is not a scheme like known form business or warehousing systems, which requires a high amount of computing resources and data is distributed and backed up on multiple servers. The requirements on the load database are that devices and its parameters can be stored in an appropriate way and the stored items can be searched and its parameters can be read. To meet these requirements in this work a XML file in combination with a C++ access class is realized. XML is a platform independed document format and set of rules for encoding information into a machine readable form. The design of XML increases simplicity and generality for the distribution of information [7]. A further advantage of XML is a structured storing of data. Different data in this case load templates can be stored and sorted into types and available load information. The structured storing requires the acceptance of XML rules for the architecture of the file, and the satisfaction of syntax rules provided in the XML specification [7]. The interface for accessing the XML file will be described in Section 6.4.3.

6.4.1 File Structure

The XML data file is structured into categories for a better overview readability and faster searching of appliances. For this work the different categories are kitchen appliances, household appliances, other appliances and test appliances . Each appliance registered in the database has a unique identifier. Categories holds an identifier offset XML-property element [7]. This offset enables an easy insertion of new appliance entries in each category without renumbering of identifiers. A second property entry holds the name of the category (see Listing 6.2). The first line of the file is a standard header definition of the XML specification [7]. It defines that XML version 1.0 is used, the character encoding is UTF-8 which stands for 8 bit Unicode Transformation Format and the XML file is in standalone mode (usage without an additional datatype definition file). The type of the document is defined as a LoadDatabase file. The root element of the file is named `LoadDatabase` and the element properties give additional information (actual version, date of the actualization, an authors and editors field and a change log entry field) about the database file. All described fields and property elements described can be seen in Listing 6.2.

Listing 6.2: Structure of the XML database file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE LoadDatabase>

<LoadDatabase Version="12" Date="16.8.2011"
   Author="MRathmair" ChangeLog="none">

  <KitchenAppliances ID_offset="100" Name="Kitchen Appliances">
  </KitchenAppliances>

  <HouseholdAppliances ID_offset="50000" Name="Household Appliances">
  </HouseholdAppliances>

  <OtherAppliances ID_offset="800000" Name="Other Appliances">
  </OtherAppliances>

  <TestAppliances ID_offset="1000000" Name="Test Appliances">
  </TestAppliances>

</LoadDatabase>
```

6.4.2 Template Modeling and Storage

One of the most important activities that are required for a load identification system is the parameterization of the templates. Inappropriately defined template parameters will result in a bad identification performance (low probabilities returned) and appliances could not be detected. In this project load parameters are extracted manually from individually measured appliances load profiles (see Section 5.4 and Appendix Section 9.1). These parameters are written in defined structured element sections in the XML database file. With the help of the extracted parameters, a model of the load is defined and stored in the database.

Appliances stored in the database have fields which are common, independent of parameters for the implemented sub-algorithms (see Section 6.5.2). An example for a full template description is shown in Listing 9.1 at Appendix Section 9.2. The common fields are `LoadDescription` (line 2), `Events` (line 9) and `ElectricalInformation` (line 65). The description field holds some information about the load: a human readable explanation, some electrical data and user editable notes. The event section contains descriptions of identification system events for the detection of the load. And finally the electrical information field includes some data about the electrical connection of the device the device type and detailed electrical load parameters.

Depending on the detection algorithm, load specifications in the database hold different additional fields. For the pre-identification sub algorithm (see description in Section 6.5.3), which works with rising edges in the load profile `opening events` must be defined (see line 10 in Listing 9.1). An opening event models a rising edge in the load profile and is of the type `RISING_EDGE_EVENT`. A event source field and the delta value describes the edge event. The tolerance value which is a parameter of the `Delta` node element is used for matching tolerances between sampled edge event and the template. The same event structure also exists for falling edge events summerized under section `closing events` (see line 28 in Listing 9.1).
For a combined edge and timing parameter detection (see algorithm description in Section 6.5.3) in addition to opening and closing events timing parameters must be stored. Section `TimingInformation` (line 57 in Listing 9.1) contains data about the activation time and the timing period of the activations. A parameter for combined detection is the time of activation of an appliance. Therefor timing windows can be defined in the template (see line 60 in Listing 9.1). If the sampled appliance is activated at one of the defined template windows the algorithm will results a positive matching.

For the ΔP - ΔQ identification algorithm appropriate template points in the ΔP - ΔQ space must be stored in the database (for algorithm description see Section 6.5.4). The section `DPQEvents` in the XML file contains information about the rising and falling edges in real and reactive power of the modeled device (see line 46 in Listing 9.1). The open event defines a turn on action of the device and a close event the turn off action. The definition of an event type is not required. A positive sign of the value indicates a rising- and a negative sign a falling edge.
Common parameters are compulsory for each registered device. All other parameters for the implemented load identification sub-algorithms are optional. In other words if it is not desired that a load is detected with ΔP - ΔQ method the corresponding parameter field does not exist.

6.4.3 File Access

Accessing the XML file is realized with the C++ version of the standard libxml2 library [15]. For reading the load database at a higher level an access class is realized at application level. This access class offers methods for direct reading of load categories, searching loads by its ID, walk through registered loads, direct request of parameter sets for each template load, etc. The interface methods use basic functions from the libxml2 library (i.e. reading node contents, reading property entries, searching tags, etc). The access class can be instantiated in multiple processes. The libxml2 library is not totally thread safe [12]. Tis means that not all modules of the library are implemented in thread safe style. If read and write operations are required the access must be designed thread safe with offered locking elements. But in this project only read actions will happen on the XML file. The parsing unit of the library is a thread safe implementation part of the library[12]. So the parsed XML catalog information (a binary equivalent of the file in the main memory) exists uniquely, independent of multiple parsing calls. This causes that the document pointer points to a constant memory location, and the catalog can be used for read operations from several independent processes. For detailed information about the integration of the load database in the object oriented load identification framework see Section 6.5.2.

6.5 Object oriented Load Identification Functionality

The load identification algorithm implemented in this work is an object oriented approach. The architecture and main functioning of the implementation is to enable an integration of different sub-algorithms. Sub-algorithms implemented in this work are variations or combinations of the identification methods described in Section 5.2. The object oriented idea behind the system is that load identification functionality is divided into different software objects. Each object is assigned to solve a sub-task of the identification work. The implemented collection of load identification methods works independent and creates their own detection results. Finally the implemented sub-algorithms of the object oriented framework delivers different probabilities and solutions for the identification of the activated load. With this architecture the dynamic changing requirements over time, and customer dependent requirements on the detection of appliances can be fulfilled and maximized detection probabilities should be delivered.

6.5.1 Requirements on the Identification System

The main task of the load identification system in this project is the non intrusive detection (see Section 5.2) of activated electrical loads out of an aggregated smart metering data set (see Table 6.1). Therefor the requirement of maximizing the detection rate has highest priority. An appropriated detection rate $r_{detected}$ in percent is defined according to Equation 6.1 where $n_{detected}$ is the number of successful detections and $n_{activated}$ is the number of load activations.

$$r_{detected} = \frac{n_{detected}}{n_{activated}} \cdot 100\% \tag{6.1}$$

A second indicator for the performance of load identifications is the probability value for matching a detected load to a registered load template in a data base. This value defines the accuracy of the detection system and should be maximized for each load type. This depends on the implementation of the matching algorithm and has a significant impact on the precision of the system (under precision the summarized detection accuracy of all available loads can be understood). Loads can be detected in different ways. Different implemented objects can observe events on the aggregated load signal and an independent calculation of the load identification should happen. So every type of load can be detected with a maximum value of detection probability by using the best qualified identification algorithm. All implemented algorithms and methods are collected in an object oriented framework. Each algorithm in the system calculates its detection probability and sends the results to a common analyzation unit. The sub-algorithm delivering the maximum detection probability is significant for the presentation of the result at the graphical user interface. A maximum of system flexibility should be reached by a customized configuration of the algorithms. In the view of system extensibility a new detection algorithm for a type of appliance must be capable of being integrated. Each algorithm contains of one, or a set of interconnected software objects. Modularity and a clearly structured design of the architecture is a significant requirement for implementing the system in a customer specific way (specific activation and deactivation of functions) and target platforms.

6.5.2 Common Algorithm Implementation

This section describes the load identification model used in this work and its implementation in C++ in a more detailed way. The common architecture of the framework is described, not the implemented sub algorithms. The object oriented concept is shown in Figure 6.5. The load identification functionality is realized as a standalone software process and can be executed independent of the visualization or data reading software. The `main` function, which is the entry point of the software dynamically, creates all other objects. (All classes are instantiated with the `new` operator so their object location in the main memory is not statically fixed before execution.)

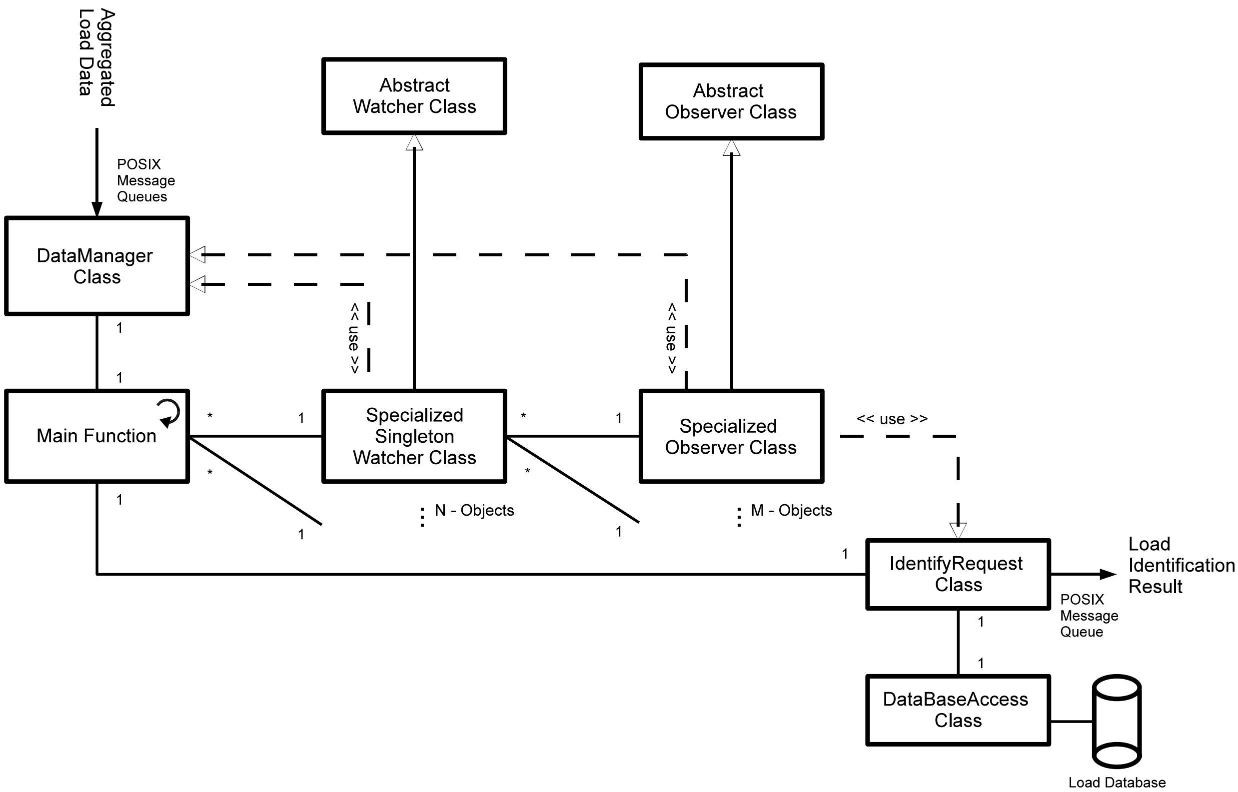

Figure 6.5: Load identification framework object overview. Source: author

For receiving input data a inter process queue channel is created between read and load identification process. The queue is realized with the standard POSIX library [22]. The object responsible for this task is the `DataManager`. First this module tries to open the load identification input message queues which also buffers data transmissions between read process and load identification module. If the opening fails the whole module cannot be updated with current data information and will be terminated immediately. At the second step the constructor of the `DataManager` initiates the creation of two circular buffer modules. One for the received data and one for calculated average input values. The circular buffers are implemented as a template class [Lou98, p. 881] and based on `deque` elements [KS99] of the C++ standard template library (STL). The `DataManager` scans the queues continuously with the help of the `select` function [13] and inserts the received data elements to circular buffers. `select` is a very powerful function and enables the concept of synchronous I/O (input/output) multiplexing. In other words it allows to check file descriptor sets whether they are read- or writable. After a successful initialization the module offers interface functions for getting actual datapoints, history datapoints, average datapoints and history average datapoints. The relation between `main` function and `DataManger` is that it exists definitely one `DataManager` object in the system and a cyclic function of the `DataManager` will be called continuously. (see Figure 6.5)

Common objects of the load identification sub-algorithms itself are so-called watchers. According to Figure 6.5, watcher objects are also created by the main function. There existing number N of different watcher objects is known before execution time. All watchers are specialized from an abstract watcher base class which implements watcher common variables and methods [Lou98, p. 881]. For example all watchers must implement a cyclic callable function which is used for different timing analyzation functions and periodic calculations of the object. An identifier indicates the type of the watcher and the activity of the object. The watcher's objective is to observe all available input waveforms. They can request an actual data set by calling the `GetData` method at the `DataManager` object (for the call of the method of an other object standard message passing notation of C++ is used). Different types of watchers scans the actual data set for events they are resistive on. For example an edge-watcher can be specialized to scan the input signal for rising or falling edges. To avoid duplicated detections watcher objects are implemented as singleton classes using the C++ standard singleton pattern design [2]. Dynamically created events are the results of watch processes. An event is an object which includes some properties of the event and an unique identifier (see Listing 6.3). Each event is a specialization of an abstract event class listed in 6.3. The creation and management of the `uuid` typed identifier is solved with the Open Source Software Project library OSSPuuid+ [20]. `eventID` characterizes the type of the event, and the `eventSource` variable contains an enumeration value for the input wave the event occurred on. The public interface functions in Listing 6.3 offers access to the event parameters.

Listing 6.3: Structure of an event

```
class event {
public:
        string EventToName (enum SMEventSources_e e);
        enum SMEventSources_e getSMEventSource ();
        int getSMEventID ();
        std::string GetUUId ();

protected:
```

```
        uuid identifier;
        int eventID;
        enum SMEventSources_e eventSource;
        time_t occurenceTime;
};
```

Each watcher can create an observer object if an event is detected. The creation of an observer is not mandatory. Different types of observers for specialized jobs are existing. An observer takes a set of events from the watcher and observes these events for further actions and creates state information based on the watcher events. For example a one step load profile watcher takes some rising edge events (real power and current rising edges), observes the levels, counts a timing register for staying at a constant level and matches the observation with falling edges from another watcher. All observer objects are also implemented as specializations from an abstract observer base class [Lou98, p. 881] (see Figure 6.5). The number of watcher objects is not known before execution because the objects are created depending on the occurred events. All dynamically created observers are registered in a list at the creators watcher object. The relation between watcher and observer is that a watcher can register different number M of observers and one observer is registered and managed by one watcher (see Figure 6.5). For the actualization of data information all created observer objects are enabled to call the `DataRequest` functions at the `DataManager`. A cyclic function in each observer object is implemented for timing measurement of the held levels in power and current. If the periodic call of this cyclic function fails timing parameters are wrong which results in low detection result probabilities.

The `IdentifyRequest` object is used to match observer results to load database entries. According to Figure 6.5 every created observer or watcher can place different types (with different arguments and object configurations) of load identification requests at the `IdentifyRequest` object. According to the watcher or observer an appropriate function in the `IdentifyRequest` module must be implemented. The module itself will be instantiated by the `main` function. Exactly one module of this type exists. The identify request functions for an watcher or observer object are called via C++ message passing. All results of the matching algorithms are delivered back to the calling object and written to a POSIX queue. A threshold value permits only result elements thats matching probability is greater or equal to a predefined level. A result element has the type illustrated in 6.4 and consists of an unique load identifier, a description of the detected load, the probability of the detection, some additional flags, and a timing value which indicates how long the detected appliance has been turned on. Results are returned to a GUI module and a simulation process which statistically analyzes them.

Listing 6.4: Type definition of an identification result element

```
typedef struct LoadIdResult_s {
        unsigned int id;
        std::string description;
        float propability;
        unsigned char flags;
        unsigned int upTime;
} loadIdResult_t;
```

An interface to the load database is given by the `DataBaseAccess` software object. The database itself, is like described in Section 6.4, a XML-based file. This software module implements a XML parsing library for the interpretation of the file. In this project the `libxml2` library is used [15]. The class can be instantiated by any module of the system in multiple ways. At the load identification module the access object will be instantiated once for handling the identification requests (see Figure 6.5). The object offers interface functions for a high performance walk through the XML tree, searching elements, identifying load types, etc. For more details on the implementation and structure of the XML file see Section 6.4.1.

All components described above are common for the implemented sub load identification algorithms. The realization of different detection methods and techniques is implemented by putting their algorithmic functionalities into watcher and observer objects. Watchers and observers can interact together and solve the task of load detection at an implemented method. The description of the methods and sub-algorithms implemented in this work, and how they are realized in watcher and observer objects will be topic of the next sections.

6.5.3 Edge Detection and Parameter Matching Algorithm

This algorithm implementation is one of the sub-algorithms realized in the object oriented load detection framework. The presented method is an edge sensitive approach for detecting electrical loads according to Section 5.2.1. In addition to the edge matching functionality characteristic load parameters are extracted out of the load signal and matched with templates. The presented method could also be used to analyze any other time series data which behave similar to electric appliances [BV04].

The input for the detection algorithm is a smart meter data set D including actual aggregated load information (see Table 6.1). The data content of a D set is according to Equation 6.2, power- (P) and energy information (E), currents in line 1 (I_{L1}), line 2 (I_{L2}), line 3 (I_{L3}) and neutral wire current (I_N), voltages of line 1 (V_{L1}), line 2 (V_{L2}), line 3 (V_{L3}), angle information between current and voltage vectors (φ_1, φ_2, φ_3), a time-stamp (τ) and a flag register (f).

$$D = \{P, E, I_{L1}, I_{L2}, I_{L3}, I_N, V_{L1}, V_{L2}V_{L3}, \varphi_1, \varphi_2, \varphi_3, \tau, f\} \tag{6.2}$$

Current and voltage values acquired from the smart meter are RMS values according to Section 2.1, Equations 2.2 and 2.3. Angle information φ between current and voltage vectors are defined according to Equation 6.3 as the subtraction of current argument minus voltage argument.

$$\varphi_{1,2,3} = arg(I_{L1,L2,L3}) - arg(U_{L1,L2,L3}) \tag{6.3}$$

According to [BV04] a time series $\{D_t\}$ of D sets could be defined. This series describes the received smart meter data information over time. The $\{D_t\}$ series could be split into subsets of series with the same physical quantity $\{P_t\}$, $\{E_t\}$, $\{I_{L1,t}\}$, $\{I_{L2,t}\}$, $\{I_{L3,t}\}$, $\{I_{N,t}\}$, $\{V_{L1,t}\}$, $\{V_{L2,t}\}$, $\{V_{L3,t}\}$, $\{\varphi_{1,t}\}$, $\{\varphi_{2,t}\}$, $\{\varphi_{3,t}\}$, $\{\tau_t\}$, $\{f_t\}$. This is shown in Equation 6.4 and sketched in Figure 6.6.

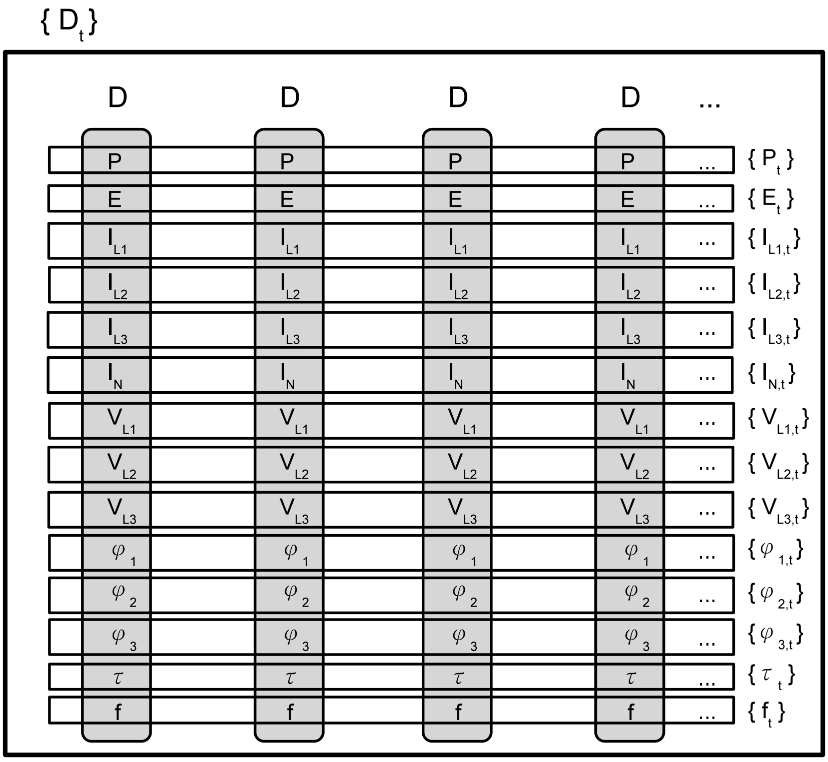

Figure 6.6: Data series overview, according to Equation 6.4. Source: author

$$\{D_t\} = \{P_t\} \cup \{E_t\} \cup \{I_{L1,t}\} \cup \{I_{L2,t}\} \cup \{I_{L3,t}\} \cup \{I_{N,t}\} \cup \{V_{L1,t}\} \cup \{V_{L2,t}\} \cup \{V_{L3,t}\} \cup \{\varphi_{1,t}\} \cup \{\varphi_{2,t}\} \cup \{\varphi_{3,t}\} \cup \{\tau_t\} \cup \{f_t\} \tag{6.4}$$

With respect to the behavior of typical electric household appliances the minimum and maximum values for the elements of sub-series could be defined. Generally it could be said that for this load identification application all electrical values (excepting angle values) are greater or equal to zero [BV04]. For the energy series the elements are monotonic increasing (see Equation 6.6) (meter reading is always increasing). Equations 6.14 to 6.16 state that the angle between current and voltage vector is in a range between $-90°$ - full capacitive and $+90°$ - full inductive. According to Equations 6.5 to 6.18 these definitions are valid for all discrete points in time or in other words for all defined values of t [BV04].

$$P_t \geq 0 \qquad \forall t \tag{6.5}$$

$$E_t \geq E_{t-1} \geq 0 \qquad \forall t \tag{6.6}$$

$$I_{L1,t} \geq 0 \qquad \forall t \tag{6.7}$$

$$I_{L2,t} \geq 0 \qquad \forall t \tag{6.8}$$

$$I_{L3,t} \geq 0 \qquad \forall t \tag{6.9}$$

$$I_{N,t} \geq 0 \qquad \forall t \tag{6.10}$$

$$V_{L1,t} \geq 0 \qquad \forall t \tag{6.11}$$

$$V_{L2,t} \geq 0 \qquad \forall t \tag{6.12}$$

$$V_{L3,t} \geq 0 \qquad \forall t \tag{6.13}$$

$$-90^\circ \leq \varphi_{1,t} \leq +90^\circ \qquad \forall t \tag{6.14}$$

$$-90^\circ \leq \varphi_{2,t} \leq +90^\circ \qquad \forall t \tag{6.15}$$

$$-90^\circ \leq \varphi_{3,t} \leq +90^\circ \qquad \forall t \tag{6.16}$$

$$\tau_t \geq \tau_{t-1} \qquad \forall t \tag{6.17}$$

$$f_t \geq \mathsf{0x00} \qquad \forall t \tag{6.18}$$

The time-stamp τ is a standard UNIX time-stamp formatted in seconds since 1.1.1970 - 00:00 [3]. For the value t increasing discrete time interval indices are stated. For each time index an appropriate time-stamp could be found in the D set. According to Equation 6.19 due to the increasing manner of the index count the corresponding time-stamps series $\{\tau_t\}$ is strictly monotonic increasing. The time difference between two time indices need not to be constant (see Equation 6.20). The time-stamp field in the D series identifies when the smart meter data set was received by the data manager software module. Depending on communication errors and delays a full data set could be received in a periodic time interval of about seven seconds (see Section 6.3.3).

$$\text{for} \quad t \geq t-1 \quad \Rightarrow \quad \tau_t \geq \tau_{t-1} \qquad \forall t \tag{6.19}$$

$$\tau_{t+1} - \tau_t \neq const. \qquad \forall t \tag{6.20}$$

The electrical conditions of a time series $\{D_t\}$ for minimum and maximum values stated in Equations 6.5 to 6.16 could be copied to every single appliance indexed with j of the system.

$$P_{t,j} \geq 0, \quad E_{t,j} \geq E_{t-1} \quad \forall t, j \tag{6.21}$$

$$I_{L1,t,j} \geq 0, \quad I_{L2,t,j} \geq 0, \quad I_{L3,t,j} \geq 0, \quad I_{N,t,j} \geq 0 \quad \forall t, j \tag{6.22}$$

$$V_{L1,t,j} \geq 0, \quad V_{L2,t,j} \geq 0, \quad V_{L3,t,j} \geq 0 \quad \forall t, j \tag{6.23}$$

$$-90^\circ \leq \varphi_{1,t,j} \leq +90^\circ, \quad -90^\circ \leq \varphi_{2,t,j} \leq +90^\circ, \quad -90^\circ \leq \varphi_{3,t,j} \leq +90^\circ \quad \forall t, j \tag{6.24}$$

The total smart meter series $\{D_t\}$ could be described as a sum of single appliances, and the time series $\{D_t\}$ can be divided into three partitions. The first one is a set of devices which are identified by an individual load monitoring system, the second one is a set of detectable loads for this edge detection based identification algorithm and the third one is a set of unknown loads which could not be identified by this algorithm implementation. The part of individually detected loads is not present in this project, but will become relevant if the implemented system is updated to a hybrid system architecture (see Section 4.2). The unknown partition can be summarized as a constant level assigned to the rest of the electrical devices. This partitioning is stated in Equations 6.25 to 6.26 where N_d stands for the number of detectable loads in the system N_i for the number of individually detected loads. The indices i and j are used for the devices in these partitions. In spoken words Equation 6.25 says that the total value of real power consumption at a point in time is a summarization of the power consumption of individually measured devices, edge detectable appliances and undetectable components in the system. This argument is only valid for the electrical parameter in $\{D_t\}$ which are capable to be summarized (power and currents) [BV04].

$$P_t = \sum_{j=1}^{N_i} P_{t,j} + \sum_{i=1}^{N_d} P_{t,i} + P_{rest} \quad \forall t \tag{6.25}$$

$$I_{L1,L2,L3,N,t} = \sum_{j=1}^{N_i} I_{L1,L2,L3,N,t,j} + \sum_{i=1}^{N_d} I_{L1,L2,L3,N,t,i} + I_{rest} \quad \forall t \tag{6.26}$$

$$\tag{6.27}$$

Important for the functionality and the implementation of this edge detection based load identification is to find the detectable subset components $P_{t,j}$ and $I_{L1,L2,L3,N,t,j}$. This is the base knowledge for the identification of loads and input data for the load database search algorithm. The following explanations are valid for detectable components of the $\{D_t\}$ time series.

Changes in the smart meter load measurement stream are handled as switch events. These edge events contains information about all involved electrical appliances. A delta of a switch event can defined according to Equation 6.28 by the difference of values after one step in time. X in Equation 6.28 can stand for any physical value (P, I_{L1}, I_{L2}, I_{L3}, I_N, φ_1, φ_2, φ_3) in the smart meter data set.

$$\Delta X_t = X_t - X_{t-1} \quad \forall t \tag{6.28}$$

For the creation of edge events a threshold value δ for each physical value must be defined. If a change in the load signal (delta value) is grater or equal to this barrier a new event object will be created. According to Equations 6.29 and 6.30 two types of events are valid for this edge detection algorithm. A rising edge event R is created if the delta value is grater or equal than a positive threshold value. A falling edge event F is created if the delta value is grater or equal than a negative threshold value.

$$R_k : \text{if} \Delta X_t \geq \delta \wedge \delta > 0 \tag{6.29}$$

$$F_k : \text{if} \Delta X_t \geq \delta \wedge \delta < 0 \tag{6.30}$$

In Equations 6.29 and 6.30 the index k notes that each event is unique and signed with an event identifier (see Section 6.5.2). The existing event types R and F can be handled as a combined datatype value (i.e. struct value in C programming language). The sets include (as stated in Equation 6.31 and 6.32) an event identifier (id), an event type description (predefined constant enumeration value) an event delta (d) and a source information register (s).

$$R_k = \{id, RISING\_EDGE\_EVENT, d, s\} \tag{6.31}$$

$$F_k := \{id, FALLING\_EDGE\_EVENT, d, s\} \tag{6.32}$$

After each step in time a set of events ψ containing R and F elements according to their occurrence can be defined (see Equation 6.33).

$$\psi = \{R_0, R_1, R_2, \ldots, F_0, F_1, F_2, \ldots\} \tag{6.33}$$

If the set ψ is not empty ($\psi \neq \{\}$) a load identification request for the event set will be started. The set of events ψ and its containing parameters are input data for the load identification.

First step at the identification procedure is an analyzation of the event set and the extraction of additional load parameters. If there is an edge event in one of the line currents and the neutral line current the load is a single phase appliance. If there are edge events in all three current lines the appliance is a three phase load. The special case of edge events in two of three current lines is handled as two single phase loads which will be activated or deactivated simultaneously. This single or three phase load type information will be the first parameter for the database matching algorithm. The second step at the identification procedure is that the event set will be compared with stored template event sets in the load database. The set will be compared with the opening event set if the type of the event is a $RISING\_EDGE\_EVENT$ (pre identification algorithm) and with closing event sets in the case of $FALLING\_EDGE\_EVENT$ type (post identification algorithm). The minimal information for a stored load in the database is a set of opening and closing events according to Section 6.4.2. The third criteria for identification algorithm is the number of events included in the set ψ. The fourth one are the events itself. For each event the type information (rising or falling edge), source and delta (d) will be checked. For matching the delta values a tolerance window of $d_{tmplate} \pm (\frac{tolerance}{2})$ is defined. The matching is successful if $d \leq d_{template} + (\frac{tolerance}{2})$ and $d \geq d_{template} - (\frac{tolerance}{2})$. The last step for the pre- and post identification algorithm that the time of the activation is checked. The load database holds for each stored load one or more timing windows for the typical activations of a device. If the occurrence time of the event set ψ is in one of the defined timing windows a positive matching criteria is detected.

Depending on this four criteria (single or three phase load, number of events, the events itself and the activation time) the pre - and post identification matching probability for each appliance registered in the database can be defined according to Equation 6.34. It can be calculated by dividing the number of parameters which could be matched with a template element in the database by the maximum number of possible matches. The added constant 3 in the denominator of Equation 6.34 is the maximum number of possible matching parameters (single or three phase load, number of events and the activation time). The multiplied factor 3 stands for the matching parameters containing each event (rising or falling edge, source and delta).

$$p_{match} = \frac{\text{number of matches with template parameters}}{3 + \overline{\overline{\psi}} \cdot 3} \qquad 1 \tag{6.34}$$

For combined event detection the time for staying at constant levels is an extra identification criteria. This time will be returned by an appropriate observer software module (see Section 6.5.2). For this time value a defined tolerance is stated in the template description. At the calculation for the matching probability the denominator of Equation 6.34 changes due to the additional timing parameter to a constant value of 4 (see Equation 6.35).

$$p_{match} = \frac{\text{number of matches with template parameters}}{4 + \overline{\overline{\psi}} \cdot 3} \quad {}^{1} \tag{6.35}$$

The implementation of this edge detection and parameter matching algorithm is realized with two watcher and an observer object. Watchers are sensitive for rising or falling edges in the load profile. If a rising edge is detected an identification request and a matching process with templates is started. After this pre identification an observer object is responsible for the timing analyzation of staying at constant power and current levels. If a falling edge occurred the falling edge watcher initiates a post identification request with the detected parameters. If rising and falling edge matches each other the observer will be destroyed and a combined identification algorithm with timing parameters delivered by the observer will be executed. With the help of watcher and observer objects the mathematical description of the edge detection and parameter matching algorithm can be implemented into the object oriented load identification framework.

6.5.4 Detection of Real and Reactive Power Characteristics

A disadvantage of the edge based parameter matching algorithm described in Section 6.5.3 is that mobile loads can be connected to different lines. Mobile loads are appliances that can be plugged in in different sockets in a household. Due to the internal electrical installation of the house sockets and plugged devices are sourced by different lines. This makes a unique parameter extraction for the load impossible. An over-determination with parameter database entries for all three lines, or a under definition with parameter entries for only one line will cause bad identification results. Another disadvantage of the edge based parameter matching algorithm is that an identification and a combined identification with matching of the activation times parameters can only happen if the turn on event of the appliance is successfully detected. These two disadvantages can be eliminated by using a ΔP - ΔQ approach similar to the described one in Section 5.2.1. The algorithm is based on the rising or falling edges in the real and the reactive power plane of the load profile. If a positive or negative edge in the profile occured or in other words if the load profile has a positive or negative value change which is greater or equal than a defined threshold value the increases or decreases in real power ΔP and reactive power ΔQ can be calculated. At a time point t these values are given by the subtraction of actual power values and the values at $t-1$ (see Equations 6.36 and 6.37).

$$\Delta P = P_t - P_{t-1} \tag{6.36}$$

$$\Delta Q = Q_t - Q_{t-1} \tag{6.37}$$

The values for P_t, P_{t-1}, Q_t and Q_{t-1} can be calculated according to Equations 2.5 and 2.6 with 6.38, 6.39, 6.40 and 6.41.

[1] $\overline{\overline{\psi}}$ in Equation 6.34 and 6.35 stands for the cardinality (number of included elements) of the set ψ [8].

$$P_t = V_{x,t} \cdot I_{x,t} \cdot cos(\varphi_{x,t}) \tag{6.38}$$

$$P_{t-1} = V_{x,t-1} \cdot I_{x,t-1} \cdot cos(\varphi_{x,t-1}) \tag{6.39}$$

$$Q_t = V_{x,t} \cdot I_{x,t} \cdot sin(\varphi_{x,t}) \tag{6.40}$$

$$Q_{t-1} = V_{x,t-1} \cdot I_{x,t-1} \cdot sin(\varphi_{x,t-1}) \tag{6.41}$$

The algorithm is implemented for detecting single phase loads. So in Equations 6.38 to 6.41 x must be set for the line an event occured on. If there are falling or rising edges on more than a single phase this algorithm returns an error. But for single line events the calculation results a ΔP and ΔQ value independent of the line the appliance is connected to. The real power values P_t and P_{t-1} of a rising or falling edge can also be read direct from the smart meter (see description of a smart meter data set in Section 6.5.3) without a calculation. A switch in the C++ source code enables or disables the calculation of the real power value (Equation 6.36).

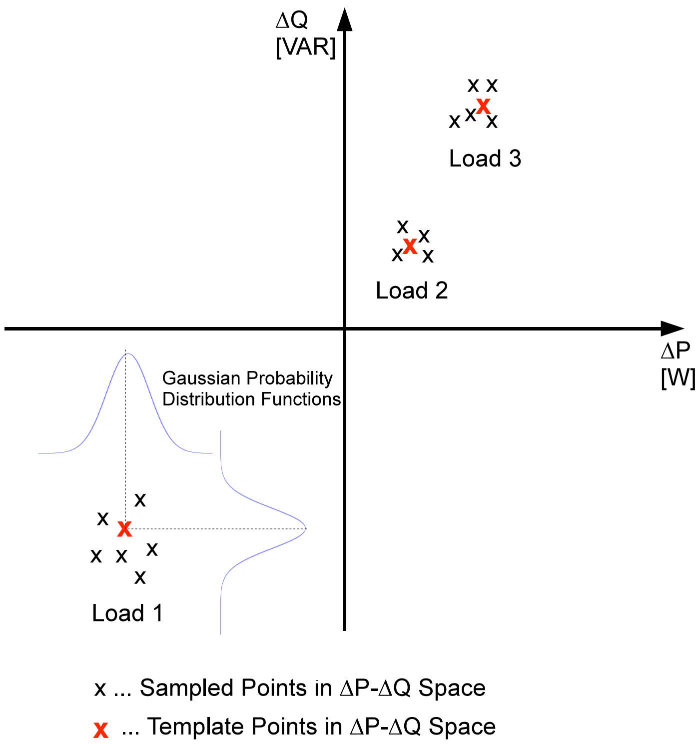

Figure 6.7: ΔP-ΔQ space with example samples, template loads and an illustration of the Gaussian probability functions. Source: author

The evaluated samples of ΔP and ΔQ can be charted in an two dimensional space like illustrated in Figure 6.7. Sampling points which can be assigned to an specific load, forms a cloud around the template sample which is marked with a red bold cross in the figure. A sample has defined distances in ΔP and ΔQ direction between the recorded point and the template. This distance is significant for the matching probability of a sample with a database registered template device. The greater the distance the lower the matching probability for the template load. The distribution function of the probability is a Gauss function in both (ΔP and ΔQ) directions (see Figure

6.7) The maximum of the Gauss function is 1 and is located at the templates ΔP and ΔQ value. The mathematical equation of the used Gauss function is stated in Equation 6.42 [BÖ5, p. 183].

$$f(x) = \frac{1}{0.3989 \cdot \sqrt{2 \cdot \pi}} \cdot exp\left(\frac{-1}{2} \cdot \left(\frac{x - \epsilon}{\beta}\right)^2\right) \tag{6.42}$$

Where ϵ is the reference value (template) and β the width of the Gaussian curve. The numerical value of 0.3989 results in the fact that the factor multiplied with the exponential function, which corresponds to the hight of the Gaussian peak, must be 1 to get a probability of 1 at the reference value. According to Figure 6.7 two independent probability values for ΔP and ΔQ can be calculated (see Equation 6.43 and 6.44).

$$p_{\Delta P} = \frac{1}{0.3989 \cdot \sqrt{2 \cdot \pi}} \cdot exp\left(\frac{-1}{2} \cdot \left(\frac{\Delta P_{sample} - \Delta P_{template}}{\Delta P_{template}/10}\right)^2\right) \tag{6.43}$$

$$p_{\Delta Q} = \frac{1}{0.3989 \cdot \sqrt{2 \cdot \pi}} \cdot exp\left(\frac{-1}{2} \cdot \left(\frac{\Delta Q_{sample} - \Delta Q_{template}}{\Delta Q_{template}/10}\right)^2\right) \tag{6.44}$$

It can be seen that the width parameter β of Equation 6.42 is set to $\Delta Q_{template}/10$ and is dependent from the distance between the sample point and the origin. At this definition large loads with high power consumption have an expanded tolerance window for the detection. Experiments showed that $\Delta Q_{template}/10$ is a practicable width value (see Section 7 for more details). The tolerance window is defined by the width parameter of the Gauss probability function, and a threshold value for the probability which must be grater or equal that the detection result will be written in the load identification result queue. According to Equation 6.45 a result probability for a sample at the ΔP - ΔQ detection algorithm can be calculated by returning the average value of $p_{\Delta P}$ and $p_{\Delta Q}$

$$p_{result} = \frac{p_{\Delta P} + p_{\Delta Q}}{2} \tag{6.45}$$

A disadvantage of returning the average value of the $p_{\Delta P}$ and $p_{\Delta Q}$ probabilities according to Equation 6.45 is that if one of the probabilities is low the resulting probability is low. If the resulting probability could be mathematically held high in this case a successful detection of a load can happen with one spike probability. This characteristic can be fulfilled by calculating p_{result} with Equation 6.46.

$$p_{result} = \sqrt{{p_{\Delta P}}^2 + {p_{\Delta Q}}^2} \tag{6.46}$$

A calculation according to Equation 6.46 requires other maximum values for the Gaussian distribution functions. The value of 1 is a still valid maximum constraint for the calculation of p_{result} but for $p_{\Delta P}$ and $p_{\Delta Q}$ this results in Equation 6.47.

$$\text{with} \quad max(p_{result}) = 1 \Rightarrow \quad max(p_{\Delta P}) = \sqrt{\frac{1}{2}} \quad \text{and} \quad max(p_{\Delta Q}) = \sqrt{\frac{1}{2}} \tag{6.47}$$

The maximum values of the Gaussian probability distribution function must be set to $\sqrt{1/2}$. The multiplication factors before the exponential function in Equation 6.43 and 6.44 corresponds the maximum values of the function. So these equations changes to 6.48 and 6.49 for the calculation of the resulting probability with Equation 6.46.

$$p_{\Delta P} = \sqrt{\frac{1}{2}} \cdot exp\left(\frac{-1}{2} \cdot \left(\frac{\Delta P_{sample} - \Delta P_{template}}{\Delta P_{template}/10}\right)^2\right) \tag{6.48}$$

$$p_{\Delta Q} = \sqrt{\frac{1}{2}} \cdot exp\left(\frac{-1}{2} \cdot \left(\frac{\Delta Q_{sample} - \Delta Q_{template}}{\Delta Q_{template}/10}\right)^2\right) \tag{6.49}$$

The realization of this algorithm at the object oriented framework a watcher (`DPQWatcher`) is responsible for the detection of the ΔP and ΔQ events. No additional observer is required because the reached levels in power, current and angle profiles need not to be monitored over time. At this method the activation time is no parameter for the detection probability. With detected ΔP and ΔQ values the watcher can place an identification request at the `identifyRequest` object. For the event detection the power signal of an received data set will be monitored. If a rising or falling edge is detected the appropriate line current for single phase load will be evaluated. Finally the values for ΔP and ΔQ will be calculated according to Equations 6.36 and 6.37. The identify request checks all available ΔP - ΔQ template points of the database. An appropriate probability value for each template device according to the Gauss functions of Equations 6.43 and 6.44 or Equations 6.48 and 6.49 will be calculated. The method for the calculation of the resulting probability (Equation 6.45 or Equation 6.46) can be set with the help of a flag in the source code. If the probability is greater or equal than a user defined threshold value a load identification result element, according to the structure stated in Listing 6.4 will be created and sent to the load identification result queue.

6.6 User Interface

For an accurate presentation of energy feedback data and results of the load identification algorithm an user interface is required. The design of high performance graphical user interfaces (GUIs) in the view of application usability and clear presentation of all information is a complex challenge, and not main topic of this work. In this project the GUI software is based on a state of the art graphic toolkit and uses standard widgets for the presentation of information. A more complex and full customized look and feel of the widgets at the GUI could be a future work of this project. Nevertheless the representation and design must follow the following requirements.

6.6.1 Interface Requirements

As mentioned in the introduction of this section a clear and non complex graphical user interface for the presentation of the results is needed. A main requirement for the implementation of the software module is to use a sate of the art platform independent graphic toolkit. The toolkit must provide functions for a quick and easy understandable creation and setup of standard windows and widgets. In the view of the general look and feel of the widgets and windows, two requirements must be fulfilled. First all buttons, expanders, panels, etc. must be large enough to be operated with fingers if the GUI will be executed at a touch panel display. In the case of the execution on a standard desktop PC with a mouse pointer device this requirement is fulfilled automatically. And the second main requirement is that the GUI must be executable on systems with different screen resolutions. In other words the size (hight and width) of the GUI window must be modifiable by changing two software parameters. This will enable to run the GUI on different target devices (also embedded devices) with different connected display types. An additional requirement for the text contents and labels in the GUI is that the software must be realized multi-langual. Standard languages (in this project English and German) must be implemented. But the software should be ready to add more languages by handling the translations with the help of external files. For the presentation of all non standard characters included in different languages text labels must be encoded in standard UTF-8 unicode format. As a summarization of the requirements on the GUI software, it could be said that the interface should be held simple, the widgets must be large enough for being operated on touchscreen displays, a reconfiguration of the window sizes for the execution on displays with different resolutions should be possible and text labels must be implemented in a way to present them in different languages.

To fulfill the mentioned requirements state of the art open source software tools are used. The user interface is implemented with the free GNU Image Manipulation Program Toolkit (GTK+). It is available for developing GUIs at several different platforms. To meet the requirement of building a multi-langual software the GNU gettext localization toolkit is used. In the next section the general features of these toolkits, their interaction and the global software structure of the GUI application will be described.

6.6.2 Tools and Implementation

GTK+ offers a complete set of standard widgets for information presentation. It is written in standard C but has been designed to support a wide range of other programming languages, not only C/C++. Using the GTK+ toolkit with languages such as Perl and Python provides an effective method of rapid and customized application development. GTK+ has been involved in many projects, some big platforms and is the standard primary library used to realize user interfaces in Linux GNU Object Model Environment (GNOME) applications [5]. The main Features according to [5] and key points for using the GTK+ toolkit for this project are:

- Stability: GTK has been developed and updated over a decade of years by a large community and has a core maintainer team from large software companies such as Red Hat, Novell, Intel, etc.
- Language Bindings: GTK+ is supported for attractive state of the art programming languages.
- Cross Platform: In its original implementation GTK+ was developed for X Window System but has grown over years to run on other standard windowing systems. GTK software can be compiled for GNU/Linux and Unix, Windows (32-bit) and 64-bit and Mac OS X platforms.
- Accommodating: The GTK+ toolkit includes a number of features that today's developers are looking for: native look and feel, thread safety, object oriented approach, UTF-8 support, documentation, etc. GTK+ depends on the GLib library package. This offers additional functions which can be used independently from the GTK library package: object

and type system, timer support, memory allocator, threaded queues, string utilities and charset handling, etc.

- Standard Interface Elements: The GTK+ toolkit offers standard graphical user interface elements like different windows, buttons, numerical and text boxes, labels, combo boxes, sliders, expanders, etc. Also the creation of user specific widgets by deriving new classes from the `GObject` base class is possible.

For a successful build of GTK+ applications several package dependencies must be considered. For a detailed list see [Kra07, p. 22]. In addition to the standard GNU C compiler (GCC) the Linux pkg-config application could be used to resolve include path and library dependencies. `pkg-config --cflags gtk+-2.0` returns all required preprocessor compiler flags and include fags for GTK+ version 2.0 applications. `pkg-config --libs gtk+-2.0` return all library flags required for linking GTK+ code.

The execution and function call structure of the GTK+ application in this project can be described within the following points:

- Initialize the GTK+ toolbox for its usage with the `gtk_init()` function.
- Create a window or a dialog box and configure it. Different types of windows and dialogs are available (see GTK+ documentation [18] chapter 4, subsection windows). The main importance in using windows is to decide what GTK+ controls and what the system built in window manager controls [Kra07, p. 19].
- Create a container for adding widgets. At this step the layout of the widgets will be set up and the arrangement of decorations will be configured. Examples for so called layout containers used in this GUI application are: horizontal and vertical boxes, tables, expanders, notebooks, etc. For detailed information see [18] or [Kra07].
- Create widgets. Standard widgets can be created by calling the corresponding `gtk_new` function. For setting the properties of widgets two type of property setup methods are implemented in this application. The first one includes application global configurations which are set up in an external resource widget style file. The second method is used for constant setting of the GUI look. Font descriptions, image spacings, button styles, etc. are direct configured in the source code.
- Add one or more widgets to a container. This could be done by calling the `gtk_container_add` function or by using a layout container function.
- Add the containers to the top level window or dialog box.
- Connect signals and callback functions. A signal in GTK+ is a notification to your application that the user has performed some action. The signal can be caught and a callback function will be executes. For the linkage between a signal occurrence, and a defined callback function these two components must be connected by calling the `g_signal_connect` function. User specific data, containing any type of information can be transferred by using `gpointer` parameters in the argument list of the callback function [Kra07].
- Run the `gtk_main` loop function or in case of a dialog the `gtk_dialog_run` function. For a continuous actualization of data view labels the timer functions of GLib will be used. A continuous running timer will be configured and a callback function is registered which will

be executed at every timer overflow event. In the callback function, visible text labels will be updated with new values. (i.e. Time view in the lower left corner of the main window).

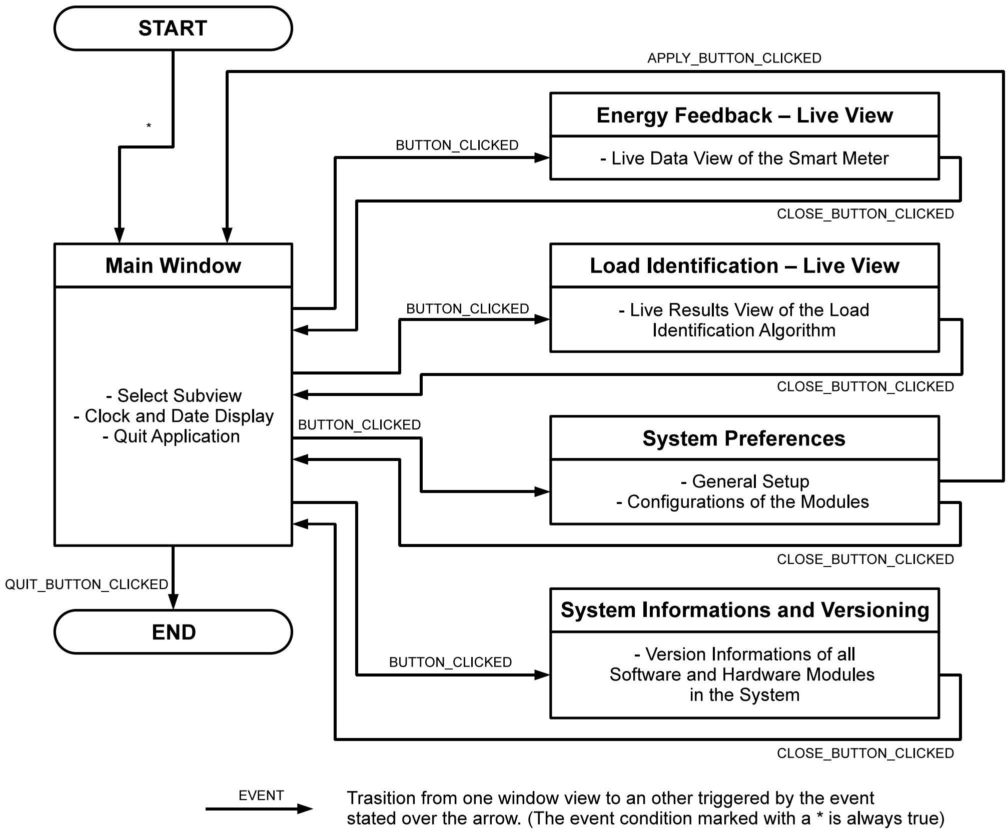

Figure 6.8: Overview over the architectural design of the GUI application. The execution starts at a START state which is automatically followed by loading and displaying the main window. Four sub-windows can be loaded by clicking the appropriate button at the main window view. At a change of system preferences the GUI must be reloaded. The whole program could be terminated by the main windows QUIT button. Source: author

The architecture and all possible window sequences are illustrated in Figure 6.8. The program execution starts at the START state and changes directly to the presentation of the main window, without waiting on any event condition. First action the START state is that the GTK+ library will be initialized. As a second procedure an independent POSIX thread will be started [21]. The main loop function of the thread handles the inter process communication with all other software modules. It tries to read data information out of the process communication queues. If any data is available the thread writes the actual values to a shared circular buffer. The access to the circular buffer must be handled thread safe with the help of mutual exclusion locking variables [21]. If a live view window is displayed the corresponding timeout overflow function for the actualization of the values can thread protected read the circular buffer. Once the main window is initialized (all widgets created, signals connected, properties set, etc.) the `gtk_main` loop will be entered and the application is waiting for GUI events. According to Figure 6.8 four different sub-windows can be called or the application can be terminated by clicking the QUIT button. At this case the program execution ends in the END state and the main window will be closed. With the four main buttons the view can be changed to: Energy Feedback - Life View window, Load identification - Live View window, System Preferences window or the System Informations and Versioning window. The Energy Feedback window contains a life data view of the connected smart meter device. Actual real and reactive power consumption, currents, voltage levels, meter reading, etc. will be displayed. Due to the circular buffer data structure the call and view of history values is an easy manageable feature. The Load identification - Live View offers results of the load identification algorithm implemented in this project. The system preferences window enables a custom configuration of system parameters. General configurations like the language of the GUI, and configurations of the systems software modules can be performed at this view. The System Informations and Versioning window offers an overview over all implemented and installed soft- and hardware modules. Versioning information, release date, persons developing and maintaining the modules are displayed. Like illustrated in Figure 6.8 each of the four main sub-windows can be closed by clicking the CLOSE button. For the preferences window an additional APPLY button exists. This confirms the settings and restarts the GUI main window view (i.e. in the case of a language change the reload and redraw of the main window with new selected language labels is necessary).

At the programmers view the main window is set up as a toplevel window, and the sub-windows as dialog boxes. All windows and dialogs have standard decorations and are controlled by the window manager of the system. Widget style configurations are like mentioned above configured in an external **.gtkrc* file which will be parsed at the application startup.

For meeting the localization requirement on the GUI software the free software tool GNU gettext is used [11]. Gettext is an important part of the GNU Translation Project and helps developers solving localization in their software in a comfortable and high performance way. The tool is based on external message catalogs which are located and named in an appropriate and well-arranged way. An integrated runtime library for supporting translated strings and a stand-alone programs for the creation of catalogs and messages are supported [11]. At the view of the GUI software developer `libintl.h` must be included and libintl library must be linked with the source code. The gettext module is configured with the help of environment variables. The main steps at the initialization are selecting and binding the textdomain, setting up the path to the catalog files and setup of the translation language (see software documentation for more details). The usage of the tool is to write all text labels in english language and call the main function of gettext (i.e. `printf(gettext("Hello World"))`). The gettext function will return a translated string in the set up language. If no catalog for the selected language is available or the translation could not be found the original parameter string will be returned [10]. New languages could be added by configuring new translation catalogs. At the translators view high performance editors for gettext catalogs exist (i.e. Poedit [14]). These editors offers well structured and easy to use graphical interfaces for entering and editing translations.

7 Results and Discussion

The system for energy feedback and load identification implemented in this thesis was tested at real measured load profiles. This chapter is divided into two sections. First the direct energy feedback methods of the system will be discussed. At the second step several results of the implemented load identification framework will be stated. This section shows which identification performance and detection probabilities can be reached at real measured load profiles and different algorithm configurations.

7.1 Energy Feedback Results

In this project direct energy feedback methods are implemented (see Section 4.1). The user interface which illustrates all smart metering and load identification results is displayed on a standard computer monitor. It has a default size of 700 x 450 pixels, and is ready for being converted into different other screen resolutions. For data presentation a combined solution with a in home display and a ambient display style is realized (see Section 6.6.2). The user interface is a standalone executable software process based on a state of the art GUI toolkit, and can be compiled for different operating systems (for toolkit availability see [5]). All embedded widgets are realized in an accurate size which increases readability of texts and values and enables using fingers for touch screen inputs.

The testability of energy feedback systems always requires customer tests over a longer period of time. Studies explained in Section 4.3 have a minimum duration of three weeks (optimally two to three month) and a lower limit of 24 test households. The effectiveness of energy feedback also depends on several other factors like weather, social reasons, system acceptability of customers, etc. Such a long test period and the equipment of real customers with the system is not feasible within this project. So the results of the energy feedback part of this work are that presented information and offered functions are nearly equal to the systems of the described studies and so similar efficiency rates could be reached. The impact of the additional factors for a test in Austria and acceptance of the system at customers side could be a future work of this project.

7.2 Load Identification Results

For the evaluation of the implemented load identification framework, according to Section 6.5, a benchmark load profile is created. This test load profile is used for rating load identification results at changing algorithm configurations. The detection and identification probability results of fixed test loads is a clear defined factor for the performance of the implemented object oriented algorithm.

Creation of the test load profile

The test load profile is a summarization of real measured individual load profiles (see Figure 9.19). Some of the appliances included in the test profile can be found at the collection in Appendix Section 9.1. The XML files from the individual load profile measurement can be converted into tables with the help of a software tool. The load profile tables of the appliances are summarized and overlapped in random manner. This results in a one hour long test load profile. A property of the profile is that it is a constant set of test data created from real measured appliances. The switch on and off times and all electrical parameters of the individual devices included in the test load profile are exactly known. A further advantage in manual overlapping of individual device profiles is that for special simulations loads can be excluded from the aggregated test profile. This enables better debugging probabilities and the influence of configuration parameters can be pointed out in better clarity. The test load profile and its parts are illustrated in Figure 9.19. Individual Appliances included in the test load profile are: (For IDs numbering see Section 6.4.1)

- **Freezer- ID 50001:** A standard freezer device activated two times. It is a load with a constant duty cycle duration and invariable electrical load parameters. Peak load = 120 W.
- **Water pump- ID 800001:** An inductive three phase motor which is activated three times in the test load profile. The duration of the activation is constant but the activation times are totally unpredictable. Peak load = 2000 W.
- **Hoover- ID 50002:** The vacuum cleaner is a large load with unstructured times and durations of activations. The device is a mobile device which means that at the first activation the appliance was connected to a socket which is internally sourced by L3 and at the second activation the hoover was connected to L2. Peak load = 550 W.
- **Halogen light- ID 50003:** This is the smallest load of the test load profile. The halogen light has a nominated power consumption of 25 W. Peak load = 25 W.
- **Plasma display- ID 50004:** A state of the art TV. The device has due to its internal parts a capacitive characteristic. The turn on event of the first activation is not part of the test load profile. A high noise level is present at the power profile during activation. Peak load = 110 W.
- **Water boiler- ID 101:** This is an appliance which will result short but high peaks in the aggregated load profile. The water boiler is a large resistive heating element at its internal structure. Peak load = 1500 W.
- **Motor- ID 800002:** This is an inductive electric motor device. After activation the power consumption rises due to an increase of mechanical load. The hight of the falling edge in the power level is due to a non constant mechanical load unpredictable for this device. Peak load = 1000 W.
- **Toaster- ID 102:** The toaster is a standard kitchen appliance with two resistive heating elements. At the first activation it is sourced by L3 and at the second by L2. Peak load = 760 W.
- **Electric heater- ID 50005:** A 1 kW additional heating element with a fan. The turn off event of the heater is not part of the test load profile. Peak load = 1050 W.

Simulation report file:
The test load profile described above and illustrated in Figure 9.19 is the basis for the simulations described in the following sections. A simulation includes the execution of the entire system (load detection, database matching and presentation at the graphical user interface) and the test load profile is fed as input data. Parameters of a full simulation cycle are the configuration setting of the load identification module and the load database file for matching samples. New data are simulated at a cycle time of ten seconds according to the timing stored in the test load profile table. This is important because the timing parameters of activation and deactivation events are additional matching factors for the identification of loads (see Section 6.5.3). If simulation samples are forced faster than stated in the load profile table, the simulation is not consistent to a real load identification situation and wrong timing parameters will result in low identification probabilities.

For the analyzation of the results delivered by the load identification the simulation module will print a result report file (identification results are transferred back to the simulation source according to Figure 6.2). A full report file is printed in Listing 9.2. The header of the file holds information about the tester, simulation time and date, and the input file location containing the test load profile table. The event log section is a chronologic listing of events. The symbol `<<<` marks an event of the simulation whether an appliance is turned on or off. `>>>` marked lines are identification results returned from the load detection module. These labels are followed by the simulation time in seconds and a description of the event. Finally at a summary report all activations are listed and the corresponding average identifications probabilities for each implemented algorithm are shown.

Database and algorithm configurations
The database XML file used for the simulation of the test load profile contains entries for each test appliance. Table 7.1 gives an overview which parameter are configured for each device. Configured parameters are marked with a X in the table. So devices can be detected by one or more implemented load detection sub-algorithms depending on the definition of template parameters.

Table 7.1: Parameter setup in the load database for the simulation of the test load profile. Configured parameter sets for a load are marked with a 'X'. Source: author

Appliance	ID	Pre Identification	Post Identification	Combined Identification	ΔP-ΔQ Identification
Freezer	50001	X	X	X	
Water pump	800001	X	X	X	
Hoover	50002	X	X		X
Halogen light	50003	X	X		X
Plasma display	50004				X
Water boiler	101	X	X	X	X
Motor	800002	X	X	X	X
Toaster	102	X	X	X	X
Electric heater	50005				X

The configurations of the algorithms realized in the object oriented identification framework can be found in a header source file of the load identification module code. First the most important configuration parameters for the pre- and post identification algorithms are threshold values for power, currents, voltages, and angles (see Equations 6.29 and 6.30). A rising or falling edge in the load profiles will be detected if the change is greater or equal these configured threshold parameters. For the ΔP-ΔQ identification algorithm such threshold values only exists for power and current values. A second main parameter is the lower limit configuration for load identification result probabilities. If a algorithm calculates an identification probability which is less than the configured value the result will not be returned to the output queue of the software module. For the simulation of the test profile threshold parameters of the pre- and post identification were set to 40 W for power, 200 mA for current 50 V for voltage and 10 degree for angle changes. For the ΔP-ΔQ algorithm a threshold values of 70 W and 300 mA are configured. An identification result must have a minimal probability of 0.7 (set for each algorithm) for getting into the result queue. This probability limit could be also a parameter which can be set by the customer at the GUI. 70 % minimum detection probability and the threshold values stated above are defined reference simulation parameters for the test load profile. These configurations result in a minimum number of identification errors.

Test load profile simulation

In the next itemization a chronologic description and discussion of simulation results stated in Listing 9.2 will happen. The test load profile and its included devices is stated in Appendix Figure 9.19. The most important simulation timing points will be explained and identification errors will be discussed. Basis of the simulation results is the test load profile according to Figure 9.19, a configured database file and the algorithm configurations as mentioned in the last subsection. (The numbers stated at the itemizations are simulation times in seconds.)

- **150, 170:** Water pump and freezer appliances are turned on and correct pre identified. The probability for the identification is 1. This means that the appliances were detected uniquely out of the the registered templates with the sample parameters of the rising edge in the load profile. The probability of 1 is reached at this rising edges because the parameters for the manual configuration of the database are taken from these edges.
- **270, 280:** The water pump is turned off and the post- and combined identification algorithm returns an unique identification of the load. For the water pump no parameters for the ΔP-ΔQ algorithm are configured (see Table 7.1). So this algorithm is completely inactive at this time.
- **410, 450:** The toaster and the water boiler device are turned on. These loads have parameters for all identification algorithms in its database entries. The devices have a resistive electrical characteristic and a small change in the angle profile (2 degree for the boiler and 5 degree for the toaster). These small changes will not be interpreted as an appropriate events because the angle change is smaller than the configured threshold value of 10 degrees. This results to detection probabilities of 0.5. The results of the pre identification are not returned to the result queue (limit parameter is 0.7). That is why the 0.5 value is also not visible in the report file (Listing 9.2). To solve this problem the angle threshold value can be set down to 1 degree. But this results in lower probabilitied at other identification cycles. In this case the toaster and water boiler devices are detected by the ΔP-ΔQ algorithm at probabilities of 0.83 and 0.7.
- **530, 540, 570, 580:** The toaster and the water boiler are turned off at this times and also successfully detected by the ΔP-ΔQ algorithm.

- **970, 980:** At this time the turn off event of the freezer, switched on at 170 seconds occurred. Rising and falling edges are successfully matched and the post- and combined identification algorithm returns an detection probability of 0.75.
- **990:** The hoover is switched on. For this event pre- and ΔP-ΔQ identification returned appropriate probability greater than the threshold limit of 0.7. It can be seen that in this case the ΔP-ΔQ algorithm is the best qualified algorithm for the detection of the hoover load (identification probability of 0.93).
- **1050:** The hoover is switched off at this time. Due to a non matching falling edge event of the current, the probability of the post identification algorithm is only at 0.57 and is not visible in the report file. A combined identification with timing parameters is not configured and meaningless for the hoover (see Table 7.1), due to completely undesirable activation times. The ΔP-ΔQ algorithm returns an adequate identification probability of 0.84.
- **1350:** The smallest load of the aggregated test load profile is the halogen light which is turned on at this simulation time. It has a nominal power consumption of 25 W. This low consumption results in no detection from any algorithm because the threshold values are 40 W for preidentification and 70 W for ΔP-ΔQ identification. A solution of the problem is to decrease the threshold values. Listing 7.1 shows the according section of the report file if the threshold limits are set down to 10 W for power and 100 mA for currents. The load will be detected (1350 in Listing 7.1) but the the high power noise value of the hoover appliance will cause several error identifications during its activation (see 3420 to 3550 in Listing 7.1).

Listing 7.1: Detection of the halogen light and error detections due to lowering threshold values for power and current. Error detections at 3470, 3530 and 3540.

```
...
<<< 1350: Turn Load 50003 (Halogen Light) ON
>>> 1350: Preidentification Result -> ID: 50003 p = 1
>>> 1350: PQ Method turn on Result -> ID: 50003 p = 0.997349
...
<<< 3420: Turn Load 50002 (Hoover) ON
>>> 3420: Preidentification Result -> ID: 50002 p = 0.948394
>>> 3470: Preidentification Result -> ID: 50003 p = 0.833333
>>> 3530: Postification Result -> ID: 50003 p = 0.714286
>>> 3530: Pre- and Postidentification Result
          -> ID: 50003 p = 0.595238
>>> 3540: Preidentification Result -> ID: 50003 p = 0.833333
<<< 3550: Turn Load 50002 (Hoover) OFF
...
```

- **1700:** At simulation time 1700 the motor appliance is turned on. The pre identification returns a probability of 1 due to the configuration of the database parameters according to this rising edge. The ΔP-ΔQ method returns a relatively low identification probability of 0.71. Several measurements showed, that the motor device is an appliance with extremely wide parameter variations. The configuration of large tolerance windows at the the pre identification algorithm will cause several error detections. So the ΔP-ΔQ algorithm with its Gaussian probability calculation function will deliver more accurate results and is the more qualified detection method for this load.

- **2300, 2310:** The turn off event of the motor causes three detection results. The post- and combined identification algorithm returns the correct identification load (ID 800002) at a probability of 0.75. The ΔP-ΔQ algorithm returns a wrong load detection with ID 102 (toaster appliance) at a detection probability of 0.70. In this case the post- and combined detection algorithm have detected the turn off event of the motor correct. This corresponds in a higher detection probability than the result of the ΔP-ΔQ algorithm. The detection probability of the ΔP-ΔQ algorithm for load ID 800002 which would be the correct one is lower than the threshold value at 0.61.

The final report (Listing 9.2) inform that at the pre identification an unique detection of load 800001 (Water pump), 50001 (Freezer) and 800002 (Motor) is possible. Accurate parameterization of load database entries makes this possible. The ΔP-ΔQ algorithm delivers for turn on events the highest number of matches with accurate average matching probabilities between 0.7 and 0.9. These average probability results underlines the different properties of the algorithms. If all sample parameters of a load are in the tolerances stated at the database file the the appliance is detected with a probability of 1. The problem of this method is if an other device has approximately the same template parameters the tolerance window for the identification must be defined very tight to avoid errors (double identifications). If a sample is not exactly located in the templates tolerance window a wrong detection will occur, and only the incorrect result will be returned (see Figure 7.2 b). The definition of overlapping tolerance windows is possible, but then the property of a unique detection is lost. Due to the Gaussian distribution probability function this problem does not exist in ΔP-ΔQ method. In the case of two approximately equalizing samples both devices will be returned at a high probability (see Figure 7.2 a).

Figure 9.20 and 9.21 in Appendix Section 9.2 illustrates the described identification probabilities over time. In the figure for each device included in the test load profile the detection probability for the defined identification algorithms is shown. The grayed areas are the real time intervals for the activation of the appliance. For example the motor device which probabilities are illustrated in Figure 9.20 and Figure 7.1 is activated between 1700 and 2300. But the diagram shows that there are several more detections of the motor device with a probability value less than the defined threshold of 0.7. The pre identification algorithm detects the motor at 150, 170, 410, 950, 990, etc. but always at a low probability. This equals to the changes of the probability values in the pre identification curve. The ΔP-ΔQ algorithm returns due to the falling edge in the power profile at 530 a relatively high turn off detection probability of 0.6. According to this result it can be seen that the defined detection threshold value of 0.7 is an appropriate limit for reaching minimal detection errors. If the threshold will be decreased to a value of less or equal than 0.6 the detection at 530 would cause an error identification. At the real activation of the motor device (simulation time 1700) the pre identification results a probability of 1 and the ΔP-ΔQ algorithm 0.707. At the turn off event (simulation time 2300) the post- and combined identification method identifies the motor at a probability of 0.75 and the ΔP-ΔQ algorithm at 0.614. The diagrams in Figure 9.20 and 9.21 contains full simulation information including all probability rates for all implemented algorithms of each device.

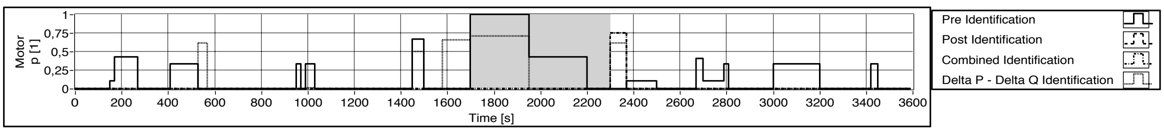

Figure 7.1: Identification probability results for the motor device. Source: author

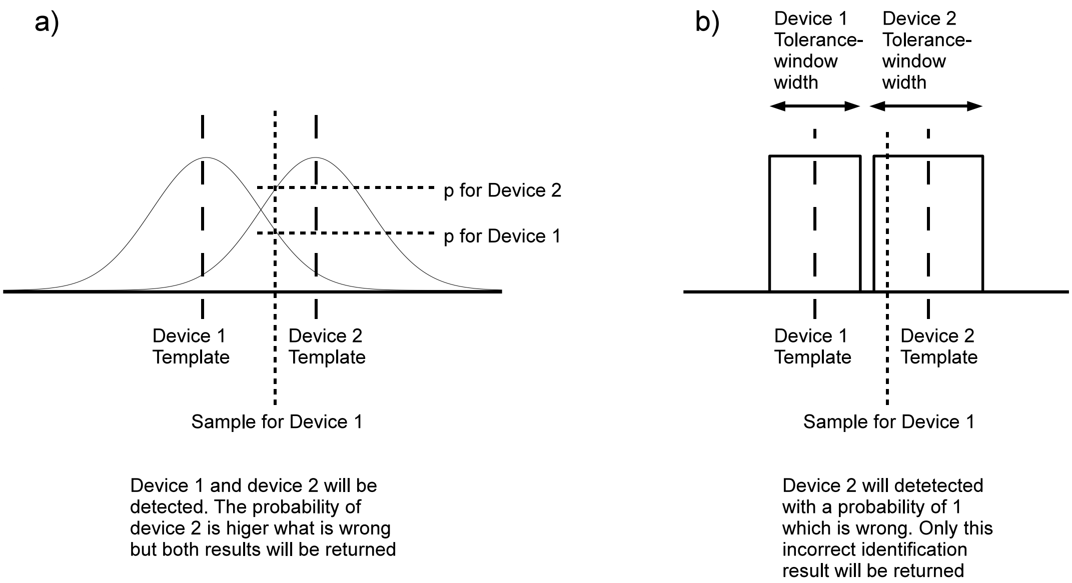

Figure 7.2: Example probabilities of the ΔP-ΔQ algorithm and the edge detection and parameter matching identification algorithm. Source: author

Noise overlapping experiment

For this experiment, noise signals for power, currents and angles are generated. The maximum amplitude of the noise signals are calculated from a predefined noise factor parameter according to Equations 7.1 to 7.3.

$$\hat{P}_{noise} = f_{noise} \cdot 1W \tag{7.1}$$

$$\hat{I}_{noise} = \frac{f_{noise}}{230} \cdot 1A \tag{7.2}$$

$$\hat{\varphi}_{noise} = \frac{f_{noise}}{3} \cdot 1^\circ \tag{7.3}$$

$\hat{\varphi}_{noise}$ in Equation 7.3 is stated in degree. In Equations 7.1 to 7.3 the dimensionless factor f_{noise} must be multiplied by appropriate units. Samples are calculated by using a random function which returns uniformly distributed values between zero and one (Equation 7.4). Power, current and angle samples can be calculated by multiplying the random function with the maximum amplitudes and subtracting the half of the maximum value to get a bipolar noise signal (Equations 7.5 to 7.7).

$$0 \leq random() \leq 1 \tag{7.4}$$

$$P_{sample} = \left(\hat{P}_{noise} \cdot random()\right) - \left(\frac{\hat{P}_{noise}}{2}\right) \tag{7.5}$$

$$I_{sample} = \left(\hat{I}_{noise} \cdot random()\right) - \left(\frac{\hat{I}_{noise}}{2}\right) \tag{7.6}$$

$$\varphi_{sample} = \left(\hat{\varphi}_{noise} \cdot random()\right) - \left(\frac{\hat{\varphi}_{noise}}{2}\right) \tag{7.7}$$

The random noise load signals are overlapped with the test load profile and the resulting profile is the input data-set for the load identification algorithm. This test shows the number identifications and error identifications dependent on the noise factor of the test load profile.

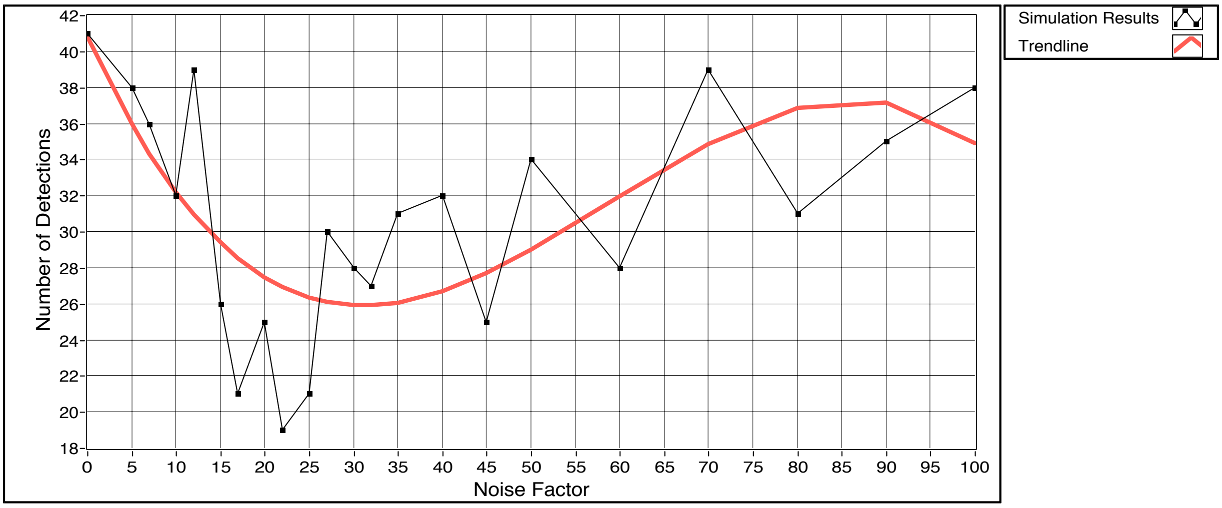

Figure 7.3: Number of identifications dependent of the noise factor. Source: author

Figure 7.3 illustrates the number of detections depending on the noise factor f_{noise}. The black line connects the resulting number of identifications points of the simulation. The bold red curve is a trendline based on the simulation results. The correct number of detections is 41, reached at a noise factor of 0. Up to a noise factor of about 22, which corresponds a power noise amplitude of 22 W the number of identifications drops to a minimum value of 19 detections. This rapid decrease is caused by tight defined tolerance windows of the parameter matching load detection algorithm. Beginning at a noise factor of 5, tolerances are already violated which results in decreased number of identifications. The decrease is definitely not caused by the ΔP - ΔQ algorithm (the ΔP - ΔQ algorithm performance will be discussed below). At a higher noise factor than 30 there is an increasing trend in the number of identifications. The increase is caused by error identifications. The load detection module returns results, but with a wrong device ID. For the edge detection and parameter matching algorithm this means that noise overlapped samples are in the tolerance windows of incorrect loads and so wrong identified.

The number of detection errors depending on the noise factor is illustrated in Figure 7.4. In this diagram also the marked points shows the simulation results, and the bold red line is the trend of the results. For a noise factor of 0 no detection errors occurred. Like mentioned above up to a noise factor of 30 errors are caused by missing matches between samples and templates. The dominating reason for identification errors are a high number of wrong template detections. The number of wrong device identifications has a saturation for high noise factors. This is caused by the reason that tolerance windows for large loads ($\geq$ 1000 W) have a larger with. At a noise amplitude of about 65 W and higher, the load templates with a large tolerance window will always be detected.

At small noise factor values up to about 30, which corresponds to an overlapped maximum power noise amplitude of 30 W, the edge detection and parameter matching algorithm results the highest counts of detection errors. The algorithm with the lowest detection errors, independent of high noise factors is the ΔP - ΔQ detection algorithm for turn on events. The number of identification errors of this algorithm is illustrated in Figure 7.5. There is a saturation at about 5 detection

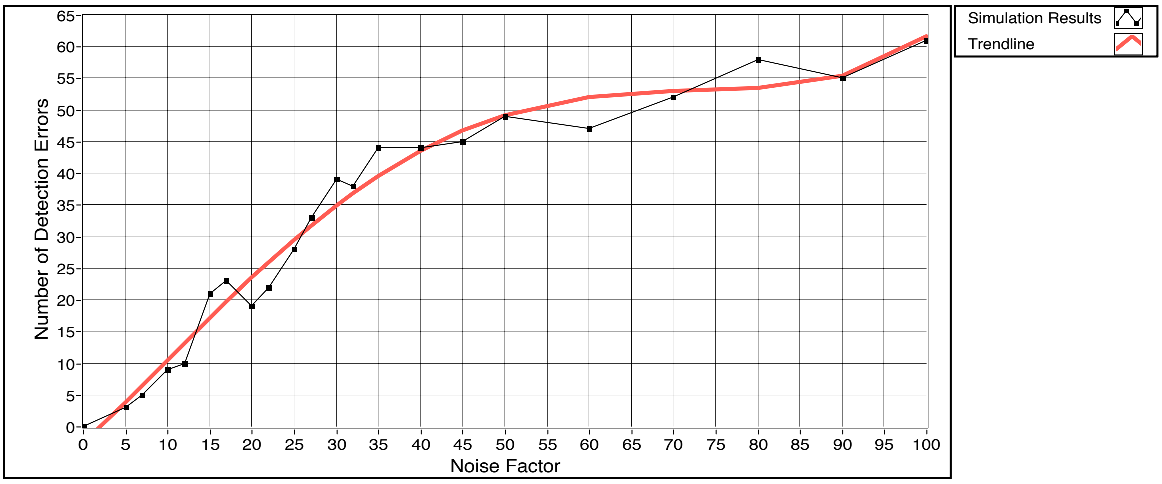

Figure 7.4: Number of identification errors dependent of the noise factor. Source: author

errors. At high noise factors the probabilities of the detections are decreased to about 0.75 but still a high number of correct identified loads is delivered.

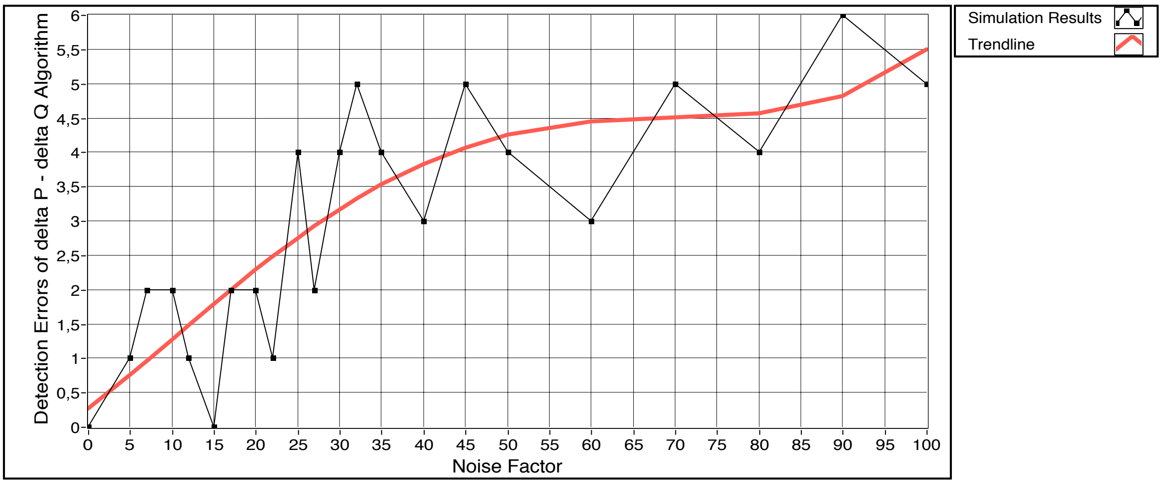

Figure 7.5: Number of identification errors of the ΔP - ΔQ detection algorithm for turn on events. Source: author

The result of this experiment is that level variations in the load profile causes identification errors. Depending on their amplitude, loads can not be detected or results with incorrect loads are delivered. The number of identification errors is also depending on the tolerance values defined in the load database template file. Especially for the implemented edge detection and parameter matching algorithm the definition of appropriate tolerance windows can help to keep identification errors at a minimum level. The width definition of windows will become a hard challenge for a high number of template loads if a unique detection is required (see Section Test load profile simulation above and Figure 7.2). The implemented ΔP - ΔQ algorithm is more robust against overlapping noise signals. A high number of correct identified loads is delivered, also for overlapped power noise amplitudes of about 50 W. The results of this experiments showed that user customized database entries with adequately defined tolerance windows are required for identifying devices at a minimum detection error rate.

8 Conclusion and Future Work

In this thesis a load identification system for energy feedback is realized. Data source for all implemented algorithms and visualizations are smart meter information. Smart meters offers an interface for serial reading of display data. The data manager software module is connected to the meter via an optical reading head. Read information are presented on a graphical user interface at a PC display. The visualization of the current energy consumption state of a household and historical values are available. Aggregated power consumption real- and reactive power, currents and voltage levels of each line are presented. For a better overview the main window contains a quick view section. There the actual power consumption level is displayed non textual in traffic light style. According to the current power consumption the red, yellow or green light is turned on. This combination of textual energy feedback with historical values and the ambient display style traffic light view should motivate customers to use the available energy more effective. An additional increase of knowledge is, if the user can determinate whether an individual appliance is turned on or off, and what is the actual consumption of this device. In other words a division view of the aggregated power consumption in individual appliances is required. Therefor a non intrusive load detection algorithm is implemented.

Load detection is based on an object oriented extensible framework, so that future algorithms can be realized within the framework. The idea behind the framework is that each object can handle a sub-task of the load identification. Data and results are transferred between the objects via message passing. An advantage of this approach is that different sub-algorithms can be implemented. The sub-algorithm objects work independent and deliver their identification results. The highest detection probability is the final result of the load identification. All other identifications are also presented to the user if the identification probability is greater or equal to an user defined threshold value. So each load type can be optimally identified by different algorithms. In this work an edge based parameter matching algorithm and an algorithm based on the real and reactive power consumption of devices are implemented. The first mentioned parameter matching algorithm detects rising and falling edges in the input load profile. Parameters (amplitude, timing, source of the edge, etc.) are matched with modeled template load profiles. The second implemented algorithm is based on different real and reactive power consumptions of devices. According to these device characteristics a turn on or off event can be detected in the aggregated real and reactive load profile. For sample matching a load database is required. Database entries are modeled loads with parameter entries which are used of the different sub-algorithms. In this project a manual setup of the load database is required. Identification errors and detection of wrong devices depends on the tolerance windows stated in the database templates. The detection result and an appropriate probability is also presented at the graphical user interface.

The combination of energy feedback and a non intrusive identification of individual loads is an effective method to change the users behavior in appliance usage, because customers know which appliances are turned on and how much energy they consume. This change in alliances usage flattens demand load profiles, decreases the consumption of electrical energy and finally reduces customers energy bills.

Future Work

To improve the functionality of the system implemented in this work soft- and hardware modules can be added. Due to the modularization and distribution of tasks into processes the function update can be accomplished by fulfilling the interface requirements. A detailed knowledge of all modules is not required. Update and replacement of hardware (i.e. target system) depends on the availability of compilers and tools for the creation of binaries. At scaleability aspects the system is ready to serve smart meter data for new functionalities, host several more entries in the load database, offers new functions and views at the GUI, etc. This itemization holds some ideas which can be realized and added to this project at a future work.

- The connection between data manager and smart meter is realized with with the D0 optical interface of the AMIS smart meter (see Section 6.3.2). This interface is reserved for maintenance and service works of energy suppliers. For the connection of energy feedback systems the AMIS smart meter offers a M-bus interface (see Section 3.2). The data transfer between smart meter and data manager can be replaced with a communication using this M-bus interface.
- The connection between data manager and measurement devices which are smart meter and individual load measurement nodes, can be realized with wireless technology. In this case additional hardware components for radio communication are required (i.e. usage of ZigBee modules). A fact that must be taken under consideration at a wireless connection solution is communication security. Electrical consumption data is private and confidential information.
- At the load identification framework new sub-algorithms can be implemented. Model based and pattern recognition approaches can be added and tested. The edge detection and parameter matching approach (see Section 6.5.3) can be updated with new matching algorithms and additional rule sets for reaching higher detection rates at an increasing number of template loads.
- The graphical user interface, which is the user's software view of the implemented system can be designed with modern style widget objects to get a full customized GUI solution (color styles, specific icons, special menu bars, etc). Additional functions can be graph and tabular views of the consumption data.
- Consumption data and the current energy state of a household can be transferred to mobile devices. State of the art computer technology and internet connection at any platform enables the transfer of load identification and smart meter data to any mobile device. The graphical presentation of data, connection and protocol management, user authentication, etc. will be the challenge of this functional extension.
- The system realized in this project is ready for the implementation on state of the art embedded systems. Cross compilation of the application and libraries will deliver an executable for embedded platforms. In combination with a touch screen display the implemented system will become a PC independent solution for energy feedback and load identification.

9 Appendix

9.1 Load Profiles of Home Appliances

This Appendix section holds load profiles of typical household appliances. The profiles are recorded with a measurement setup described in Section 5.4. This section also informs about typical characteristics of some load profiles stated below. Commonly the left chart shows the real power and as broken line the reactive power profile of the device under test. The right figure illustrates the current profile and as broken lines real power information in the same chart. With these two diagrams all characteristic parameters for a full entry in the load database for the automatic identification of the load can be determined.

Fan Heater (Figure 9.1 and 9.2)

A standard domestic heating fan for indoor use. Turned on and off manually at about 35 and 85 seconds. The heating element is a dominating resistive load and has a high amount of real power consumption (1770 W).

Halogen Lamp (Figure 9.3 and 9.4)

A standard 12 V, 20 W halogen lamp. The lamp was turned on at 30 and off at 80 seconds. The labeled 20 W bulb has a real power consumption of 25 W real and 8 VAR reactive power during operation.

Freezer (Figure 9.5 and 9.6)

The curves shows the load profile of a standard freezer unit. The device will be turned on an off controlled by an thermostat. The device under test is adjusted on freezing level 2 of 7. The on-time for one cycle is 895 seconds which equals about 15 minutes. The period time of activation is 2769 seconds which equals about 46 minutes.

Coffee Machine (Figure 9.7 and 9.8)

The device under test is a full automatic coffee machine. The power profile shows a heat up period from 30 to 90 seconds. The profile shows also 2 operations for making coffee. The first at 180 seconds which was a large cup and the second at 320 seconds which was an espresso. As a result of the two operations it could be mentioned that the structure of the load profile is the same not depending on the size of the coffee. Only the period of activation for the water pump differs from 75 seconds for a large cup to 50 seconds for a small cup.

Plasma TV (Figure 9.9 and 9.10)

A state of the art plasma TV. The TV was turned on after 150 seconds from standby mode into normal operation, and at 550 seconds back to standby. The first mentionable fact is a standby power level of 20 W which is extremely high for a modern appliance. The second fact is that the device has a capacitive characteristic in standby and operation mode and is a source of reactive power.

Washing Machine (Figure 9.11 and 9.12)

The load profile of this 3 years old washing machine shows the washing program at 30 degree celsius. The program took 3881 seconds which equals about one hour and five minutes.

Laundry Dryer (Figure 9.13 and 9.14)

The laundry dryer is like the washing machine a large electrical load with maximum peaks of 2600 W in the power load profile. The level was constantly held for 26 minutes, and a second high level of 1600 W was held for 38 minutes. That is why the laundry dryer has the largest energy consumption in the household.

Stereo Sound System (Figure 9.15 and 9.16)

The load profiles illustrates playing a compact disc in a stereo sound system. After 60 seconds the device was powered up from standby mode and the playback has been started. Up to 180 seconds the volume was at a comfortable level. After the 180 seconds mark the volume has been increased continuously and also the power level increased from 22.5 W to 23.5 W.

Cathode Ray Tube TV (Figure 9.17 and 9.18)

A 10 years old cathode ray tube TV. The device was turned on after 60 seconds from standby mode to normal operation. A key fact of this TV is that it has a positive phase angle during operation. This inductive characteristic is caused by internal coils of the cathode ray tube.

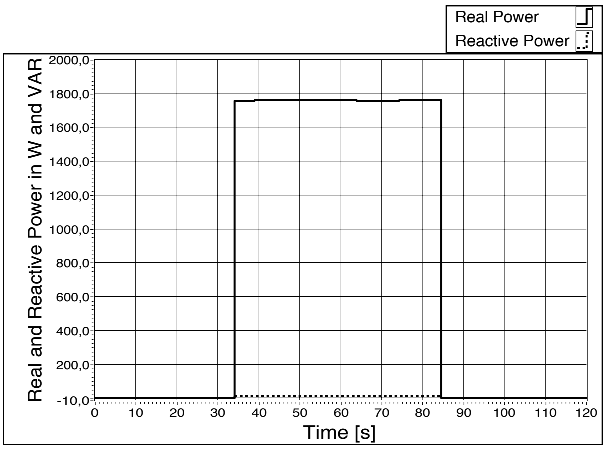

Figure 9.1: Real and reactive power profile of the fan heater device. Source: author

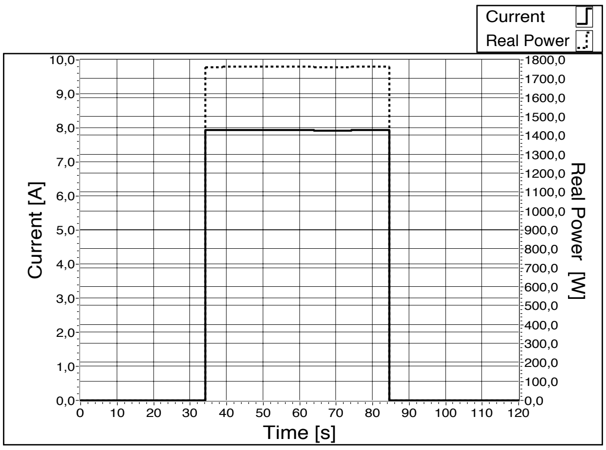

Figure 9.2: Real power and current profile of the fan heater device. Source: author

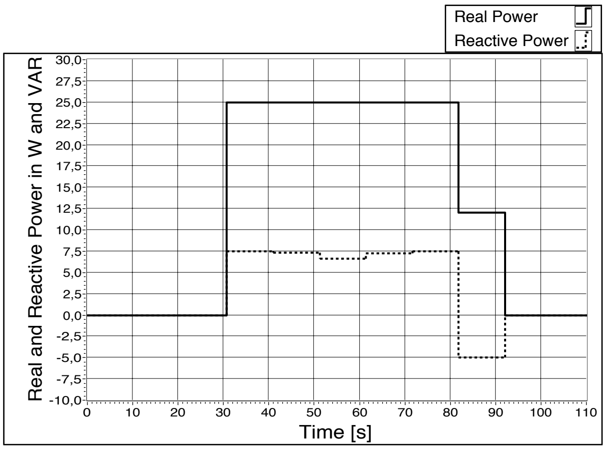

Figure 9.3: Real and reactive power profile of the halogen light. Source: author

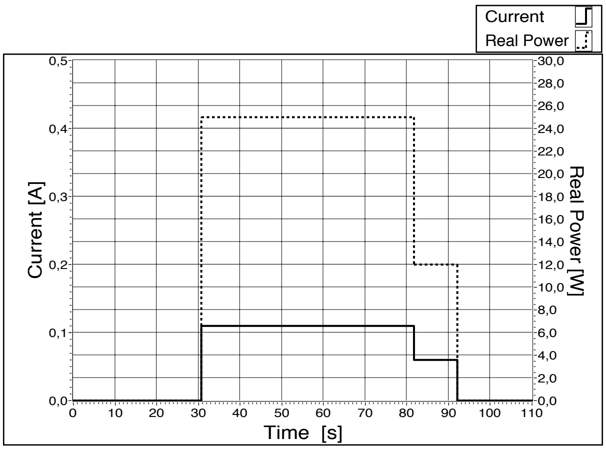

Figure 9.4: Real power and current profile of the halogen light. Source: author

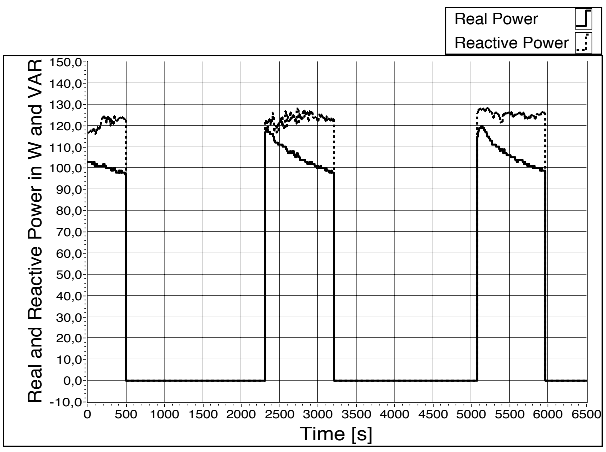

Figure 9.5: Real and reactive power profile of the freezer. Source: author

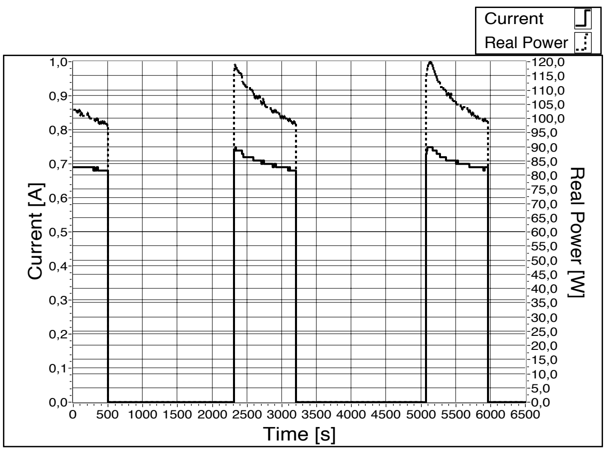

Figure 9.6: Real power and current profile of the freezer. Source: author

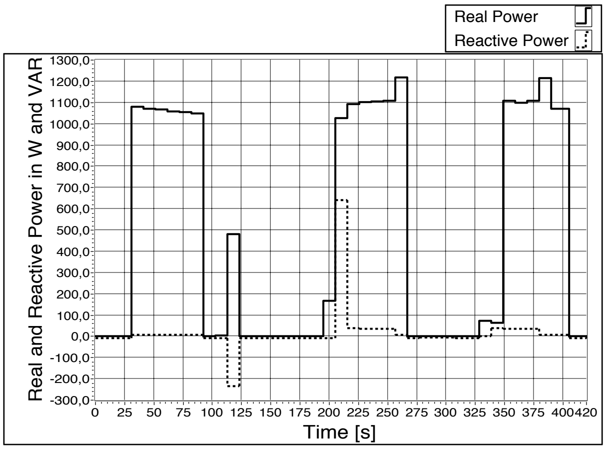

Figure 9.7: Real and reactive power profile of the coffee machine. Source: author

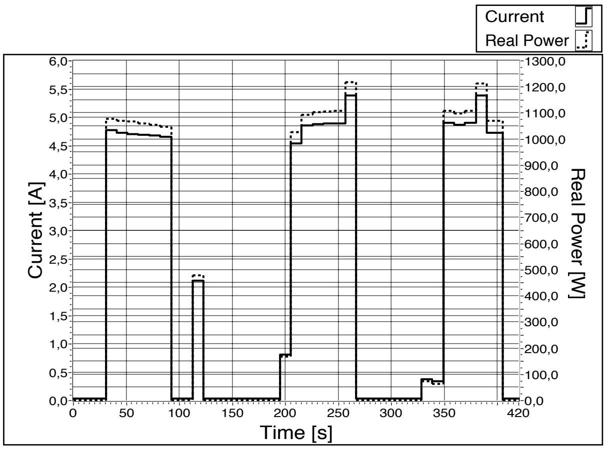

Figure 9.8: Real power and current profile of the coffee machine. Source: author

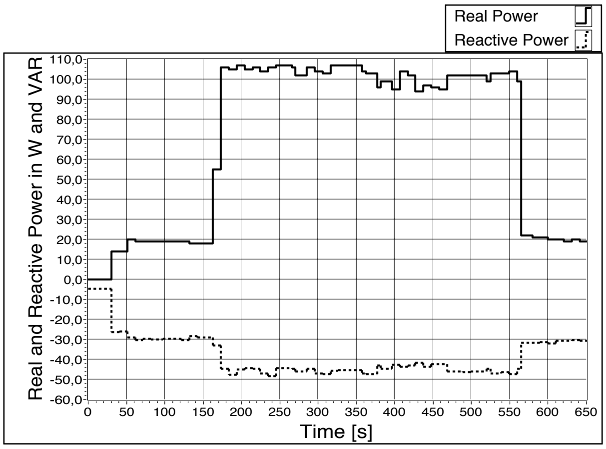

Figure 9.9: Real and reactive power profile of the plasma TV. Source: author

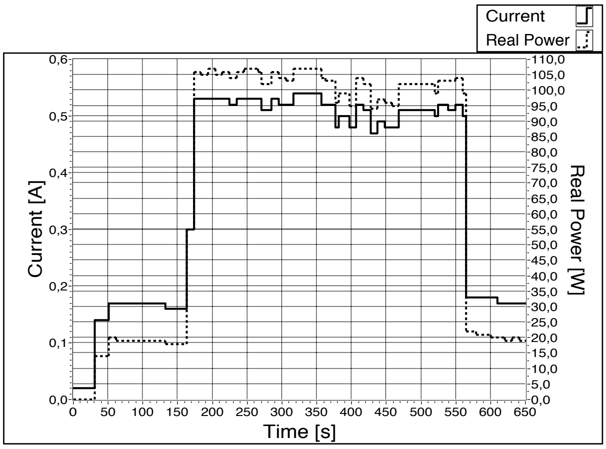

Figure 9.10: Real power and current profile of the plasma TV. Source: author

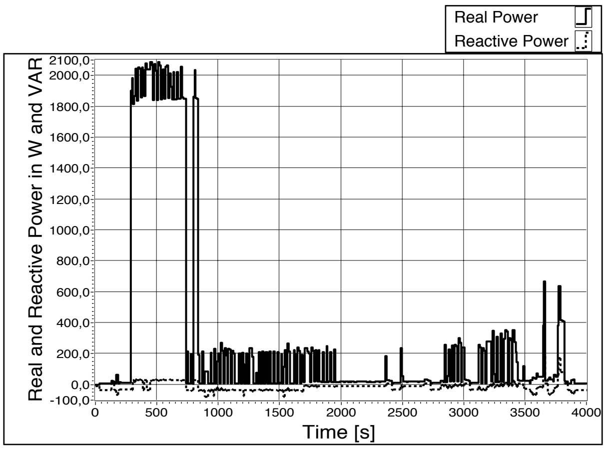

Figure 9.11: Real and reactive power profile of the washing machine. Source: author

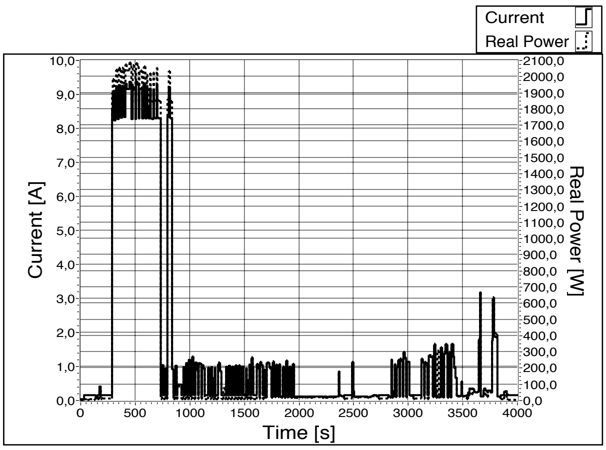

Figure 9.12: Real power and current profile of the washing machine. Source: author

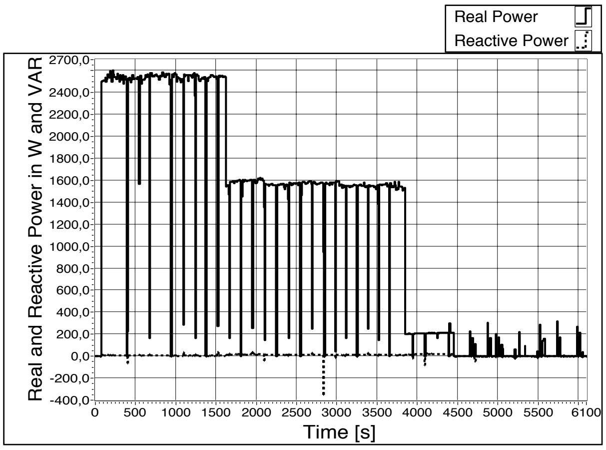

Figure 9.13: Real and reactive power profile of the dryer. Source: author

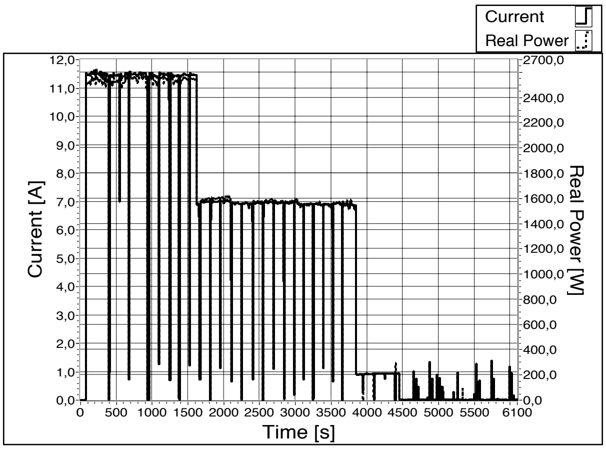

Figure 9.14: Real power and current profile of the dryer. Source: author

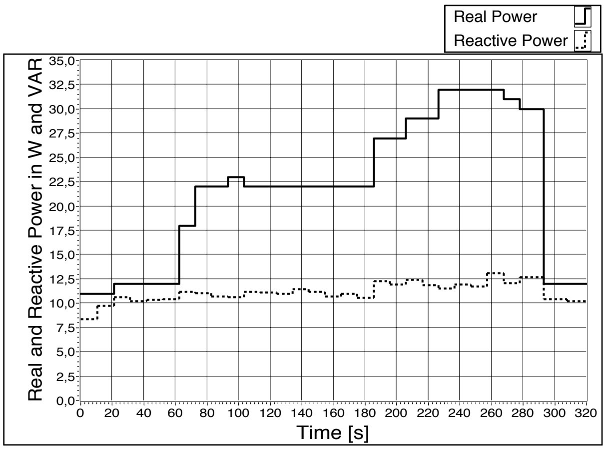

Figure 9.15: Real and reactive power profile of the stereo sound system. Source: author

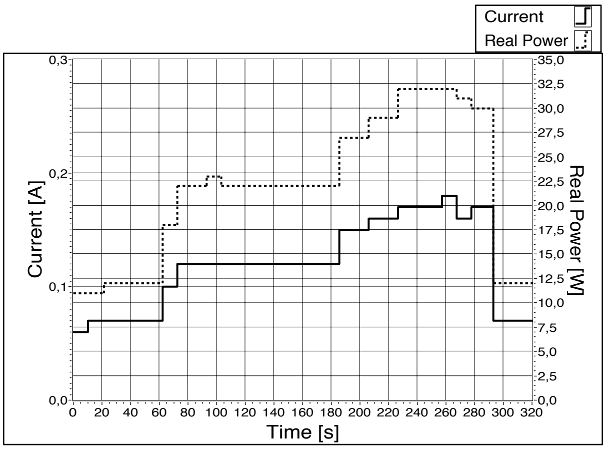

Figure 9.16: Real power and current profile of the stereo sound system. Source: author

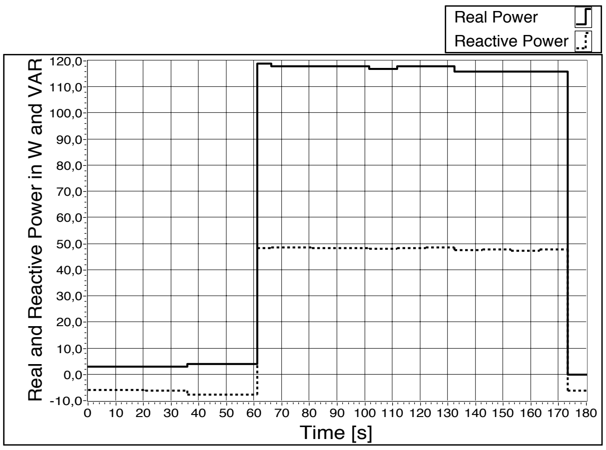

Figure 9.17: Real and reactive power profile of the cathode ray tube TV. Source: author

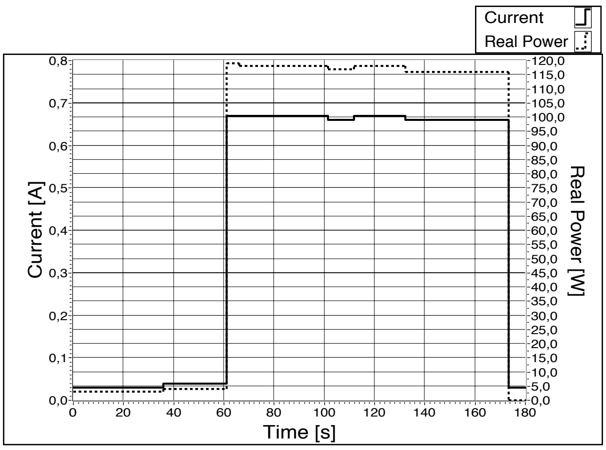

Figure 9.18: Real power and current profile of the cathode ray tube TV. Source: author

9.2 Program Listings and Graphs

Listing 9.1: Full entry of the load database for a halogen light device. The device entry holds parameters for each identification sub algorithms realized in this work (description in Section 6.4.2)

```
<Load_One_Step_Profile Type="ONE_STEP_PROFILE_LOAD" ID="50003">
   <LoadDescription>
    <Description>Halogen Light</Description>
    <ElectricSupplyData>25W,230V,
    cos(phi)=0.9612 </ElectricSupplyData>
    <UserText>Halogen desk light </UserText>
    <Paths>not specified</Paths>
   </LoadDescription>
   <Events>
    <OpeningEvents NumberOfEventEntities="4">
     <Event Type="RISING_EDGE_EVENT">
      <Source>EventOn_Power</Source>
      <Delta Tolerance="0.04">0.025</Delta>
     </Event>
     <Event Type="RISING_EDGE_EVENT">
      <Source>EventOn_I_L2</Source>
      <Delta Tolerance="0.05">0.11</Delta>
     </Event>
     <Event Type="RISING_EDGE_EVENT">
      <Source>EventOn_I_N</Source>
      <Delta Tolerance="0.05">0.11</Delta>
     </Event>
     <Event Type="RISING_EDGE_EVENT">
      <Source>EventOn_P_L2</Source>
      <Delta Tolerance="5.0">16.0</Delta>
     </Event>
    </OpeningEvents>
    <ClosingEvents NumberOfEventEntities="4">
     <Event Type="FALLING_EDGE_EVENT">
      <Source>EventOn_Power</Source>
      <Delta Tolerance="0.04">-0.025</Delta>
     </Event>
     <Event Type="FALLING_EDGE_EVENT">
      <Source>EventOn_I_L2</Source>
      <Delta Tolerance="0.05">-0.11</Delta>
     </Event>
     <Event Type="FALLING_EDGE_EVENT">
      <Source>EventOn_I_N</Source>
      <Delta Tolerance="0.05">-0.11</Delta>
     </Event>
     <Event Type="FALLING_EDGE_EVENT">
      <Source>EventOn_P_L2</Source>
      <Delta Tolerance="5.0">-16.0</Delta>
```

```
      </Event>
     </ClosingEvents>
     <DPQEvents>
      <Event Type="DPQ_OPEN">
       <dP>0.025</dP>
       <dQ>0.00697</dQ>
      </Event>
      <Event Type="DPQ_CLOSE">
       <dP>-0.025</dP>
       <dQ>-0.00285</dQ>
      </Event>
     </DPQEvents>
    </Events>
    <TimingInformation>
     <UpTime Tolerance="20">500</UpTime>
     <PeriodTime>2430</PeriodTime>
     <TypicalDayActivationTimes NumberOfTimeWindows="1">
      <TimeWindow From="0" To="86400"
      DayFlagRegister="254"></TimeWindow>
     </TypicalDayActivationTimes>
    </TimingInformation>
    <ElectricalInformation>
     <Type>ONE_PHASE_LOAD</Type>
     <COS_PHI>0.9612</COS_PHI>
    </ElectricalInformation>
</Load_One_Step_Profile>
```

Listing 9.2: Full print of a laod identification report file for the test laod profile illustrated in Figure 9.19

```
-------------------------------------------------------------------
LOAD IDENTIFICATION REPORT FILE
MRathmair
23.08.2011 - Simulation start time: 16:08:01
Driving Test file: /home/mrathmair/EFLISOFT/TestLoadprofiles/
Test_LoadProfile_freezer_waterpump_hoover_light_plasma_boiler
_motor_toaster_electricHeater-FINAL.csv
-------------------------------------------------------------------


Event Log:
<<< 150: Turn Load 800001 (Water Pump) ON
>>> 150: Preidentification Result -> ID: 800001 p = 1
<<< 170: Turn Load 50001 (freezer) ON
>>> 170: Preidentification Result -> ID: 50001 p = 1
<<< 270: Turn Load 800001 (Water Pump) OFF
>>> 280: Postification Result -> ID: 800001 p = 1
>>> 280: Pre- and Postidentification Result -> ID: 800001 p = 1
<<< 410: Turn Load 102 (Toaster) ON
>>> 410: PQ Method turn on Result -> ID: 102 p = 0.831665
<<< 450: Turn Load 101 (Hot water boiler) ON
>>> 450: PQ Method turn on Result -> ID: 101 p = 0.707091
<<< 530: Turn Load 102 (Toaster) OFF
>>> 540: PQ Method turn off Result -> ID: 102 p = 0.83885
<<< 570: Turn Load 101 (Hot water boiler) OFF
>>> 580: PQ Method turn off Result -> ID: 101 p = 0.705835
<<< 950: Turn Load 101 (Hot water boiler) ON
>>> 950: PQ Method turn on Result -> ID: 101 p = 0.706967
<<< 970: Turn Load 50001 (freezer) OFF
>>> 980: Postification Result -> ID: 50001 p = 0.75
>>> 980: Pre- and Postidentification Result -> ID: 50001 p = 0.75
<<< 990: Turn Load 50002 (Hoover) ON
>>> 990: Preidentification Result -> ID: 50002 p = 0.833333
>>> 990: PQ Method turn on Result -> ID: 50002 p = 0.929376
<<< 1030: Turn Load 101 (Hot water boiler) OFF
>>> 1040: PQ Method turn off Result -> ID: 101 p = 0.700211
<<< 1050: Turn Load 50002 (Hoover) OFF
>>> 1060: PQ Method turn off Result -> ID: 50002 p = 0.83697
<<< 1350: Turn Load 50003 (Halogen Light) ON
<<< 1450: Turn Load 102 (Toaster) ON
>>> 1450: PQ Method turn on Result -> ID: 102 p = 0.700065
<<< 1500: Turn Load 50004 (Plasma TV) OFF to standby
>>> 1500: PQ Method turn off Result -> ID: 50004 p = 0.9586
<<< 1580: Turn Load 102 (Toaster) OFF
>>> 1590: PQ Method turn off Result -> ID: 102 p = 0.933942
<<< 1700: Turn Load 800002 (Motor) ON
>>> 1700: Preidentification Result -> ID: 800002 p = 1
```

```
>>> 1700: PQ Method turn on Result -> ID: 800002 p = 0.707044
<<< 1950: Turn Load 50004 (Plasma TV) ON from standby
>>> 1950: PQ Method turn on Result -> ID: 50004 p = 0.893928
<<< 2200: Turn Load 50004 (Plasma TV) OFF to standby
>>> 2200: PQ Method turn off Result -> ID: 50004 p = 0.763929
<<< 2300: Turn Load 800002 (Motor) OFF
>>> 2310: Postification Result -> ID: 800002 p = 0.75
>>> 2310: Pre- and Postidentification Result -> ID: 800002 p = 0.75
>>> 2310: PQ Method turn off Result -> ID: 102 p = 0.703527
<<< 2370: Turn Load 800001 (Water Pump) ON
>>> 2370: Preidentification Result -> ID: 800001 p = 1
<<< 2500: Turn Load 800001 (Water Pump) OFF
>>> 2510: Postification Result -> ID: 800001 p = 1
>>> 2510: Pre- and Postidentification Result -> ID: 800001 p = 1
<<< 2670: Turn Load 50001 (freezer) ON
>>> 2670: Preidentification Result -> ID: 50001 p = 1
<<< 2700: Turn Load 800001 (Water Pump) ON
>>> 2700: Preidentification Result -> ID: 800001 p = 1
<<< 2790: Turn Load 50005 (Electric heater) ON
>>> 2790: PQ Method turn on Result -> ID: 50005 p = 0.737143
<<< 2810: Turn Load 800001 (Water Pump) OFF
>>> 2820: Postification Result -> ID: 800001 p = 1
>>> 2820: Pre- and Postidentification Result -> ID: 800001 p = 1
<<< 3000: Turn Load 101 (Hot water boiler) ON
>>> 3000: PQ Method turn on Result -> ID: 101 p = 0.706967
<<< 3200: Turn Load 101 (Hot water boiler) OFF
>>> 3210: PQ Method turn off Result -> ID: 101 p = 0.706541
<<< 3420: Turn Load 50002 (Hoover) ON
>>> 3420: Preidentification Result -> ID: 50002 p = 0.833333
>>> 3420: PQ Method turn on Result -> ID: 50002 p = 0.71574
<<< 3450: Turn Load 50001 (freezer) OFF
>>> 3460: Postification Result -> ID: 50001 p = 0.75
>>> 3460: Pre- and Postidentification Result -> ID: 50001 p = 0.75
<<< 3550: Turn Load 50002 (Hoover) OFF
>>> 3560: Postification Result -> ID: 50002 p = 0.857143
>>> 3560: Pre- and Postidentification Result -> ID: 50002 p = 0.71428
>>> 3560: PQ Method turn off Result -> ID: 50002 p = 0.926684
------------------------------------------------------------

 Final Summary:
 Driven loads:
ID: 800001 number of activations: 3
ID: 50001 number of activations: 2
ID: 102 number of activations: 2
ID: 101 number of activations: 3
ID: 50002 number of activations: 2
ID: 50003 number of activations: 1
ID: 800002 number of activations: 1
```

```
ID: 50004 number of activations: 1
ID: 50005 number of activations: 1

Pre detected loads:
ID: 800001 number of detections: 3 pAV: 1
ID: 50001 number of detections: 2 pAV: 1
ID: 50002 number of detections: 2 pAV: 0.833333
ID: 800002 number of detections: 1 pAV: 1

Post detected loads:
ID: 800001 number of detections: 3 pAV: 1
ID: 50001 number of detections: 2 pAV: 0.75
ID: 800002 number of detections: 1 pAV: 0.75
ID: 50002 number of detections: 1 pAV: 0.857143

Combined detected loads:
ID: 800001 number of detections: 3 pAV: 1
ID: 50001 number of detections: 2 pAV: 0.75
ID: 800002 number of detections: 1 pAV: 0.75
ID: 50002 number of detections: 1 pAV: 0.714286

Delta P delta Q method turned on detected loads:
ID: 102 number of detections: 2 pAV: 0.765865
ID: 101 number of detections: 3 pAV: 0.706998
ID: 50002 number of detections: 2 pAV: 0.822558
ID: 800002 number of detections: 1 pAV: 0.707044
ID: 50004 number of detections: 1 pAV: 0.893928
ID: 50005 number of detections: 1 pAV: 0.737143

Delta P delta Q method turned off detected loads:
ID: 102 number of detections: 3 pAV: 0.794961
ID: 101 number of detections: 3 pAV: 0.704782
ID: 50002 number of detections: 2 pAV: 0.881827
ID: 50004 number of detections: 2 pAV: 0.861265
```

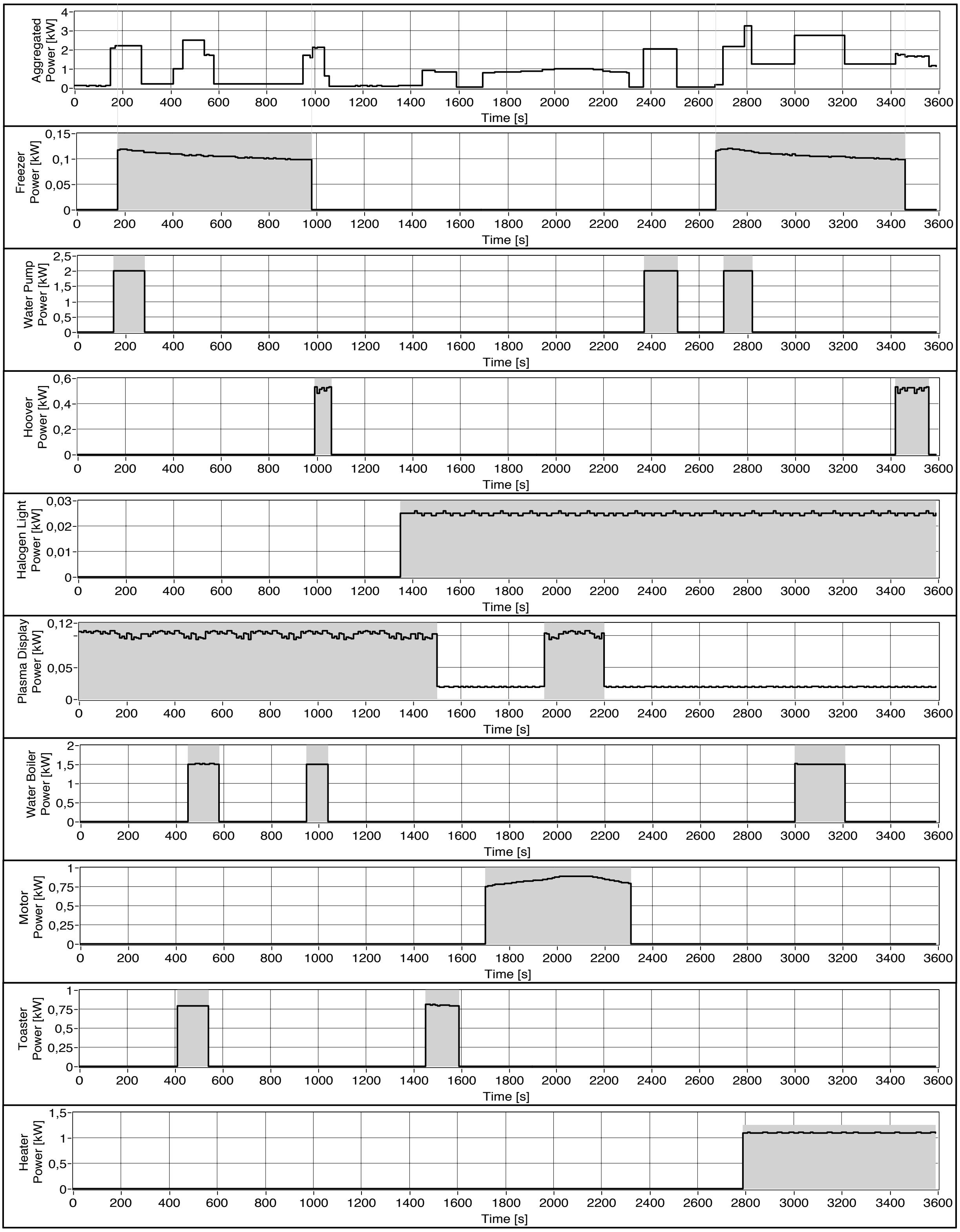

Figure 9.19: Test load profile and its parts for the evaluation of the load identification algorithm. The grayed areas are the activation time periods of the device. Source: author

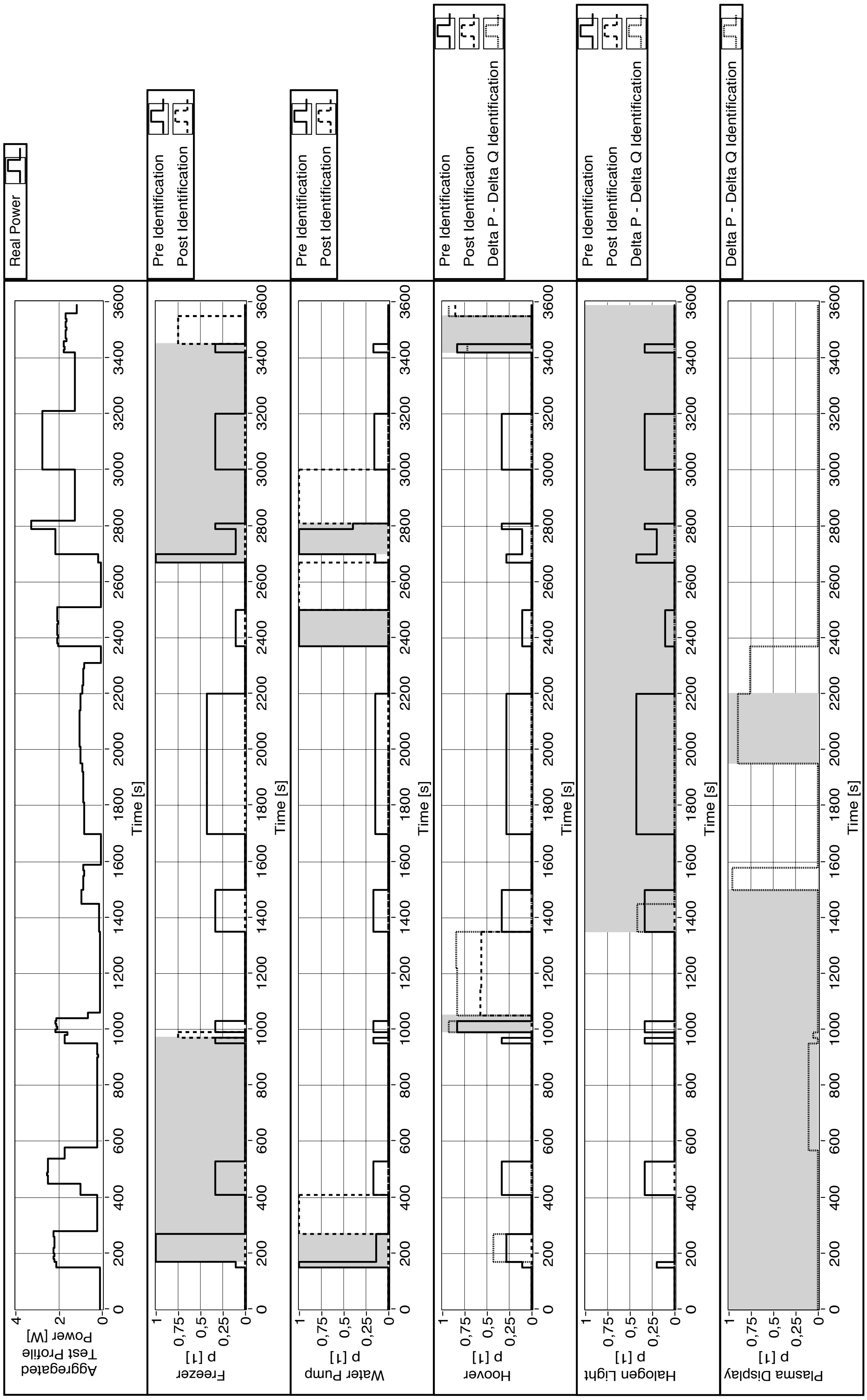

Figure 9.20: Graphical presentation of detection properties. The grayed areas are time intervals of the real activation of the appliance. Part 1: freezer, water pump, hoover, halogen light and plasma display appliance. Source: author

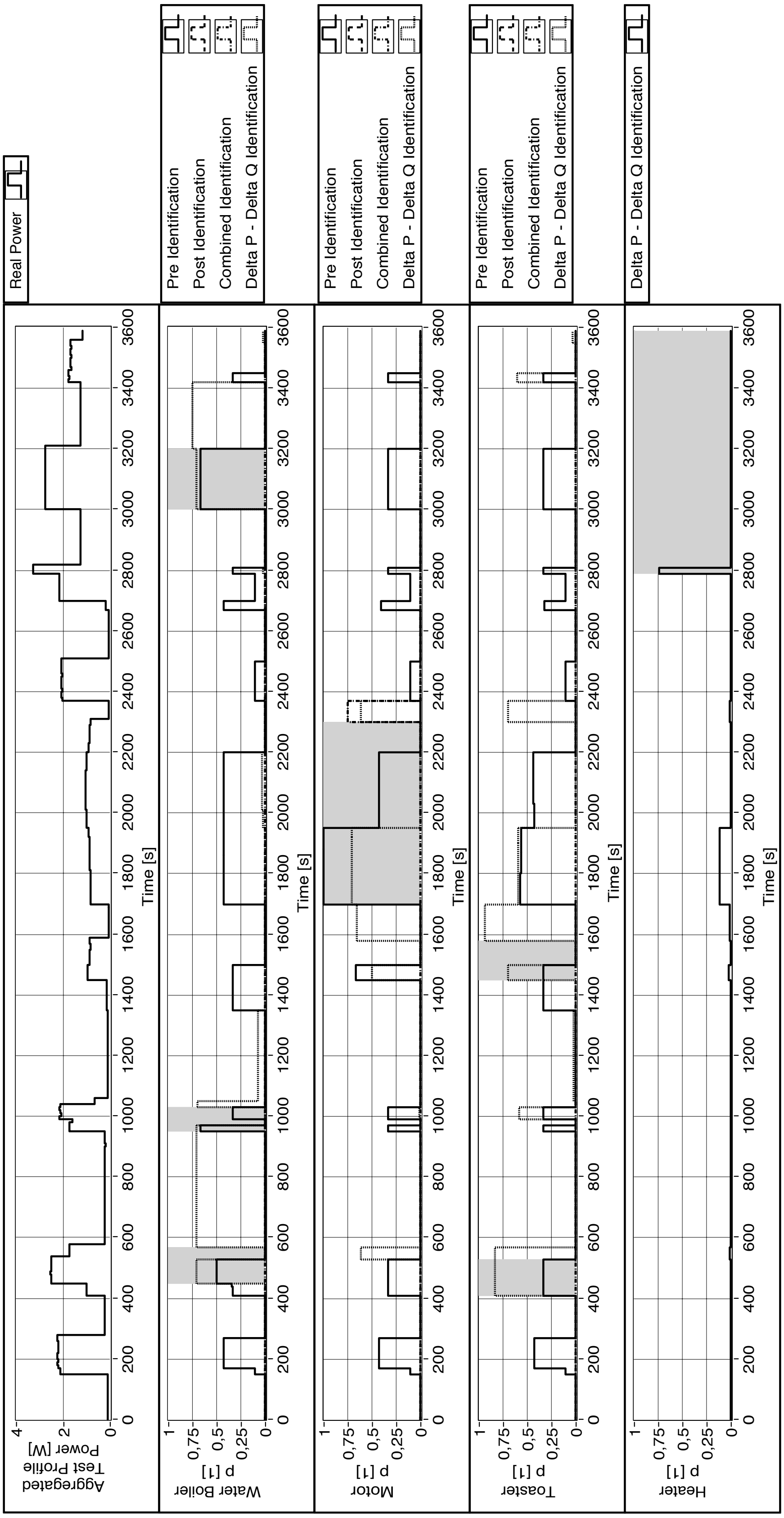

Figure 9.21: Graphical presentation of detection properties. The grayed areas are time intervals of the real activation of the appliance. Part 2: water boiler, motor, toaster and heater appliance. Source: author

Literature

[Ari09] Arivus , iAd: *OFDM-Based High-Speed Narrowband PLC Approved for Smart Metering and Smart Grids*. 1. 2009

[B05̈] Bärwolff, Günter: *Höhere Mathematik für Naturwissenschaftler und Ingenieure*. 1. Spektrum Akademischer Verlag, 2005. – ISBN 3–8274–1136–9

[bti11] bticino: *My Home - Visualizzazione consumi e gestione caichi energetici*. 2011

[BV04] Baranski, Michael ; Voss, Jürgen: Genetic Algorithm for Pattern Detection in NIALM Systems. In: *IEEE International Conference on Systems, Man and Cybernetics*, Faculty for Electrical Engineering, University of Paderborn, Germany, 2004. – ISBN 0–7803–8566–7/04, P. 3462 – 3468

[BXZ09] Bijker, Arend J. ; Xia, Xiaohua ; Zhang, Jiangfeng: Active Power Residential Non-intrusive Appliance Load Monitoring System. In: *IEEE AFRICON 2009*, 2009. – Nairobi, Kenya. – ISBN 978–1–4244–3918–8

[Chr10] Christ, Peter: *Smart Metering in Deutschland*. 1. EW Medien und Kongresse GmbH, Published by G. Fenchel and M. Hellwig, 2010. – Technik, Kommunikation und Prozesse für Elektrizität, Wasser, Wärme und Gas. – ISBN 978–3–8022–1051–8

[CKP+09] Choi, Tae-Seop ; Ko, Kyung-Rok ; Park, Seong-Chan ; Jang, Young-Sik ; Yoon, Young-Tae ; Im, Sang-Kug: Analysis of Energy Savings using Smart Metering and IHD (In-Home Display). In: *IEEE TD Asia 2009*, 2009

[Dar06] Darby, Sarah: The effectiveness of feedback on energy consumption, Environmental Change Institute, University of Oxford, April 2006. – A review for defra of the literature on metering, billing and direct displays

[DD09] Deconinck, G. ; Decroix, B.: Smart Metering Tariff Schemes Combined with Distributed Energy Resources. In: *ESAT/ELECTA (Belgium)*. Linköping, Sweden : Proc. 4th CRIS Int. Conf. on Critical Infrastructures (CRIS-2009), 28-30 April 2009. – Geert.Deconinck@esat.kuleuven.be

[DK99] Drenker, S. ; Kuder, A.: Nonintrusive Monitoring of Electric Loads. In: *IEEE 0895-0156/99*, 1999, P. 47 – 51

[EN602] EN62056-21:2002: *Deutsche Norm - Electricity Metering, Data Exchange for Meter Reading, Tariff and Load Control - Part 21: Direct local data exchange*. 2002

[EN607] EN62056-61:2002: *Deutsche Norm - Electricity Metering, Data Exchange for Meter Reading, Tariff and Load Control - Part 61: Object identification system (OBIS)*. 2007

[Ene09] Energie AG Oberösterreich Data: *AMIS Automatic Metering Information System - Customer Guide*. 1. 2009

[Ene11] Energie AG Oberösterreich: *Handbuch AMIS*. 1. 2011. – Automatisches Metering-und Informations-System

[Fri00] Fritz, Helmut: Zentralisierte contra dezentrale Automation - Reif für die Insel Sigmatek Process Framework, Plastverarbeiter 51. Jahrg. (2000) Nr. 5, 2000

[HKR95] Harrigan, M. ; Kempton, W. ; Ramakishna, V.: Empowering customer energy choices; a review of interaction and feedback in energy efficiency programs. In: *Alliance to Save Energy*. Washington DC, 1995

[Hue09] Huemer, Anton: *Dokumentation AMIS-Zähler*. 1. Energie AG Oberösterreich , Abteilung Metering Services , Team MS-Prozesse und Daten, 2009. – Automatisches Metering und Informations-System

[KP00] Keogh, E. ; Pazzani, M.: Scaling up Dynamic Time Warping for Data mining Applications. In: *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000, P. 285 – 289

[Kra07] Krause, Andrew: *Foundations of GTK+ Development*. Apress, 2007. – Build sophisticated graphical applications using one of the world's most powerful cross-platform toolkits. – ISBN 987–1–59059–793–4

[KS99] Kuhlins, Stefan ; Schader, Martin: *Die C++-Standardbibliothek, Einführung und Nachschlagewerk*. Springer Verlag, 1999. – ISBN 3–540–65052–0

[KS09] Koehler-Schute, Christiana: *Smart Metering*. 1. KS-Energy-Verlag, 2009. – Technologische, wirtschaftliche und juristische Aspekte des Smart Metering. – ISBN 978–3–9813142–0–5

[Lou98] Louis, Dirk: *C/C++ Kompendium*. Markt & Technik, 1998. – ISBN 3–8272–5386–1

[Mou01] Moulin, Etienne: *Energy Data Converters Reduce Cost and Simplify Meter Design*. 1. Analog Devices: Analog Devices, 2001

[Mou06] Mountain, D.: The impact of real time feedback on residential electricity consumption: The Hydro One pilot, Mountain Economic Consultiong and Associates Inc., Ontario, 2006

[NF94] Nagura, Hidetoshi ; Fujino, Kenji: Correction Method for a Single Chip Power Meter. In: *IEEE THPM 4-3* IMTC 94, 1994. – Hamamatsu, P. 1313 – 1316

[PI10] Pedro, Albertos ; Iven, Mareels: *Feedback and Control for Everyone*. 1. Springer-Verlag Berlin Heidelberg, 2010. – ISBN 978–3–642–03445–9

[Sch98] Schmusch, Wolfgang: *Elektronische Meßtechnik*. 4. Vogel Fachbuch, 1998 (Elektronik 6). – ISBN 3–8023–1769–6

[SDB78] Seligman, C. ; Darley, J.M. ; Becker, L.J.: Behavioral approaches to residential energy conservation. In: *Energy and Buildings 1*, 1978, P. 325 – 337

[SR08] Schlabbach, Jürgen ; Rofalski, Karl-Heinz: *Power System Engineering - Planning, Design, and Operation of Power Systems and Equipment*. Wiley-vch Verlag, 2008. – ISBN 978–3–527–40759–0

[UIST05] Ueno, T. ; Inada, R. ; Saeki, O. ; Tsuji, K.: Effectivenes of displaying energy consumption data in residential houses. Analysis on how the residents respond. In: *European Council for an Energy-efficient Economy - paper 6.100*, 2005

[WN06] Wood, G. ; Newborough, M.: Energy-use information transfer for intelligent homes: Enabling energy conversation with central and local displays. In: *Energy and Buildings 39 (2007)*. Edinburgh, Scotland EH14 4AS, UK : School of Engineering and Physical Sciences, Herriot-Watt University, 2 June 2006 2006, P. 495 – 503

[You93] Young, De: Changing behaviour and making it sick: the conceptualistion and management of conservation behaviour. In: *Environment and behaviour 25*, 1993, P. 485 – 505

[Zia10] Zia, Tehseen: *Recognizing Power Consumption Patterns of Household Appliances*, Vienna University of Technology, Diss., October 2010

[ZXM10] Zhou, Long ; Xu, Fang-Yuan ; Ma, Ying-Nan: Impact of Smart Metering on Energy Efficiency. In: *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, IEEE Qingdau, 11-14 July 2010*, Energy System Group, School of Engineering and Mathematical Sciences, City University London, EC1 V 0HB, UK, 2010. – ISBN 978–1–4244–6527–9/10, P. 3213 – 3218

Internet References

[1] *G Programming Reference Manual*, 1998. http://digital.ni.com/manuals.nsf/websearch/8603C072BB0FAAE98625665E006358A6.

[2] *Singleton Pattern and its implementation with C++*, March 2002. http://www.codeproject.com/KB/cpp/singletonrvs.aspx.

[3] *Unix Time Stamp description*, 2002-2011. http://www.unixtimestamp.com/index.php.

[4] *Datasheet of the MCP3909 power measurement chip sensor*, 2006. http://ww1.microchip.com/downloads/en/DeviceDoc/22025a.pdf.

[5] *The GTK+ Project*, 2007-2011. http://www.gtk.org.

[6] *Datasheet of the ADE7569 power measurement chip sensor*, 2008. http://www.analog.com/static/imported-files/data_sheets/ADE7116_7156_7166_7169_7566_7569.pdf.

[7] *Xtensible Markup Language (XML) 1.0 (Fifth Edition)*, 2008. http://www.w3.org/TR/REC-xml/.

[8] *Einige Begriffe und Definitionen der Mengenlehre*, 2009. http://homepage.ruhr-uni-bochum.de/Stephen.Berman/Semantik/Mengenlehre.html.

[9] *Datasheet of the MCP3901 power measurement chip sensor*, 2010. http://ww1.microchip.com/downloads/en/DeviceDoc/22192c.pdf.

[10] *GNU gettext utilities - Manual*, 2010. http://www.gnu.org/software/gettext/manual/gettext.html.

[11] *Introduction to gettext*, 2010. http://www.gnu.org/software/gettext/.

[12] *LibXML2 Thread safety*, October 2010. http://xmlsoft.org/threads.html.

[13] *Linux man page - select function*, 2010. http://linux.die.net/man/2/select.

[14] *Poedit - cross-platform gettext catalogs editor*, 2010. http://www.poedit.net/.

[15] *Reference Manual for libxml2*, October 2010. http://xmlsoft.org/html/index.html.

[16] *Bticino Österreich*, 2011. www.bticino.at.

[17] *Full Object identification system (OBIS) code tables*, 2011. http://www.dlms.com/documentation/listofstandardobiscodesandmaintenanceproces/index.html.

[18] *The GTK+ Reference Manual - GNOME (GNU Object Model Environment) dev center*, 2011. http://developer.gnome.org/gtk3/stable/.

[19] *National Instruments*, 2011. http://www.ni.com.

[20] *Open source software project libraries*, 2011. http://www.ossp.org.

[21] *Portable Operating System Interface (POSIX) standardization*, 2011. http://www.knosof.co.uk/posix.html.

[22] *POSIX Queues programming reference*, 2011. http://www.users.pjwstk.edu.pl/~jms/qnx/help/watcom/clibref/mq_overview.html.

[23] *Reparatur Tiefkühltruhe Kältemittel*, 2011. http://forum.electronicwerkstatt.de/phpBB/search.php?su=stirlingmotor%20wirkungsgrad%20k%E4ltemittel.

[24] *ZAMG - Weather for April 2011*, 2011. http://www.zamg.ac.at/klima/klima_monat/wetterlagen/?jahr=2011&monat=04.

[25] IMC. *Application Note - 400, The Basic of Power Meter*. IMC, January 2011. www.imcpower.com/prodigit/an_400.htm.

[26] VDE. *VDE Verband der Elektrotechnik Elektronik Informationstechnik e.V.*, February 2011. www.vde.de/de/fnn/arbeitsgebiete/messwesen/seiten/muc.aspx.

[27] WiscOnline. *Wisc online - Learning activities help people succeed - find your succeed here!*, February 2011. www.wisc-online.com/Objects/ViewObject.aspx?ID=ACE7903.

Software

Software developed as a part of this thesis is available via E-Mail at Michael_Rathmair@yahoo.com or at the Institute of Computer Technology, Vienna University of Technology.