VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Marek Kwitek

BUILDING A FOUNDATION FOR SMART ENERGY MANAGEMENT SYSTEM

Technology and Communication

2013

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Degree Programme in Information Technology

ABSTRACT

Author	Marek Kwitek
Title	Building a Foundation for Smart Energy Management System
Year	2013
Language	English
Pages	80
Name of Supervisor	Smail Menani

The purpose of the thesis was to design and develop a data model, communication protocol and human machine interface for in-home energy management system. The design and implementation are user-centered. The designed system allows both vertical and horizontal integrations. The data model was approached with adaptability and flexibility coming first. The proposed communication protocol was designed at high-level of abstraction and its translations to given underlying technologies can be specified when required. Both data model and communication protocol were set to be versatile and can be continuously adjusted to changing environment. The Qt C++ and QML were used to implement the human machine interface of the energy management system. A simulator was also designed and implemented in the scope of this thesis in order to test the proposed solution.

Objectives of the thesis were fulfilled and solid foundation for further development was created. Implementation of the human machine interface helps in visualization of the system potential as well as in communication between engineers and users.

Keywords Energy management system, smart grid, human machine interface, data model, communication protocol

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The scope of the thesis is to build a foundation for in-home energy management system which interfaces with the one managed by a utility company and with smart devices present at home. Its goal is to provide benefits for both parties, energy consumers and utility companies, with the main focus on consumers.

Energy consumption per capita in developed countries is much higher than in developing ones and the vast majority of the world's population lives in developing countries. Current power generation, distribution and consumption are not sustainable. Additional move towards renewable energy sources introduces even more challenges. Scheduled blackouts are normal for hundreds of millions of people, partially due to the lack of energy production and partially due to inefficient energy usage. Smart energy management system for the utility and in-home is needed. The thesis concerns the latter one.

The project took place in Vaasa, the city which is the Nordic leader in energy technology. Therefore, there are a lot of possibilities for it to have actual impact on the energy sector.

1.1 Objectives of the Thesis

The objective of the thesis was to create data model, protocol and human machine interface (HMI) for an in-home energy management system. The developed data model introduces general logic which could be used in the implementation of highly extensible solutions. The HMI allows the user of the system to control energy related features and home automation from one place. The protocol was defined in order to specify the standardized way of communication between distributed parts of the system, such as gateway, smart devices and human machine interfaces. The translation of the protocol made it possible to adjust the system to existing solutions and technologies. The system aims to fully utilize all data available to it in order to improve energy efficiency, safety, security and comfort of living.

1.2 Problem Definition

The demand for energy is constantly growing along with an ecological awareness. Therefore, many countries are trying to increase energy production by increasing part of the energy generated from the sustainable sources.

Electricity is an extreme example of the Just in Time (JIT) production strategy. It has to be produced at the exactly same time when it is consumed, because electricity storage on a big scale is inefficient. Matching the production to the varying demand is not easy even with centralized power plants and it could be even harder with decentralized ones. A complementary approach to production adjustment is demand management. The need for electricity usually significantly varies by the hour of the day, day of the week, month of the year, as presented in the exemplary figures below.

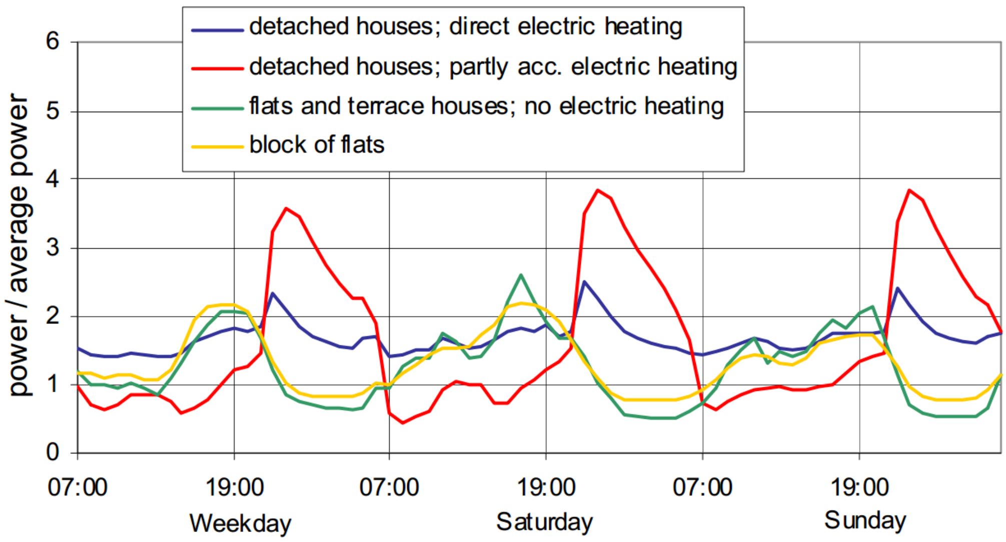

Figure 1. Day indexes relating to the two first weeks of the year for four types of dwellings /1/

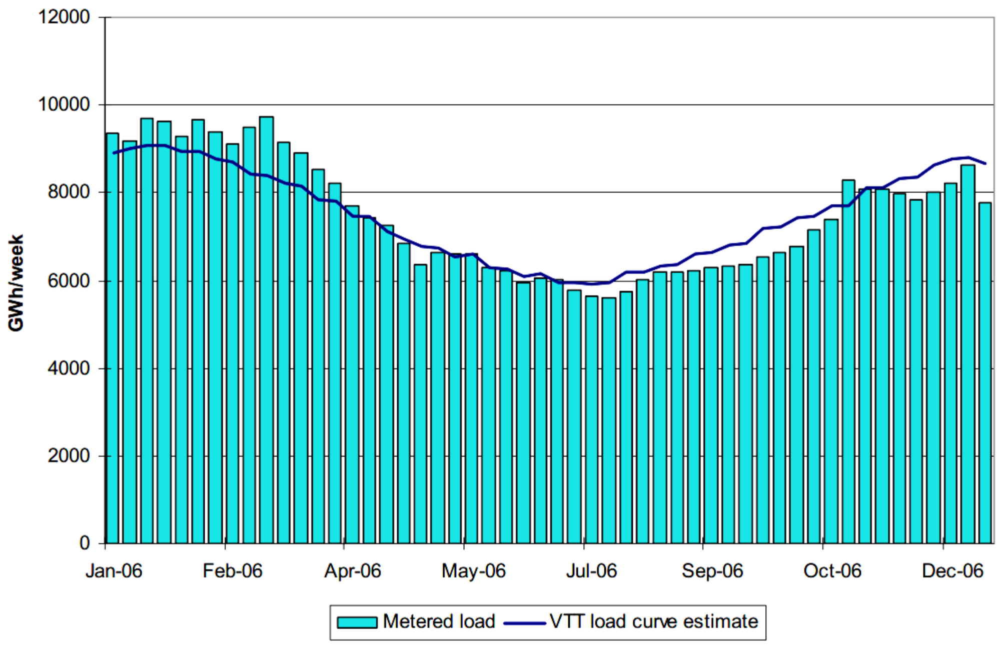

Figure 2. Estimated and metered load curves for the Nordic electricity market area in 2006 /1/

The first step to smooth the daily demand was by introduction of the different tariffs for the day and for the night. Electricity is more expensive at peak demand and less expensive during off peak hours. However, it will not be enough with decentralized power plants; especially in case of wind and solar ones. New types of meters, called smart meters allow for two way communication. Using them energy price could be directly bounded to the energy availability in real time.

Distributed energy production will become common as technology efficiency and cost factor become more and more favorable for small investors. Many houses in the future become energy positive, meaning that those houses will produce more energy than consume. That excessive energy would be sold to the grid.

There are numerous issues rising from the decentralization of energy production and from the move towards energy sources such as wind and solar. In-home energy management systems facilitate effective energy usage and demand management due to its interfaces with the user, smart devices and utility company.

1.3 Proposed Solution

The smart in-home energy management system which seamlessly integrates with the smart grid, building automation and other smart devices will help to increase energy usage efficiency. A single solution with vast functionality ensures higher adaptation ratio among households and frequent user interaction. Presenting benefits in a clear way helps to keep users motivated in their pursue of efficient energy usage. It makes a difference to be precise instead of vague. For example, by providing more exact information on how household economy would be affected in each month by decreasing the room temperature by two degrees. Figure 3 below depict basic overview of proposed system.

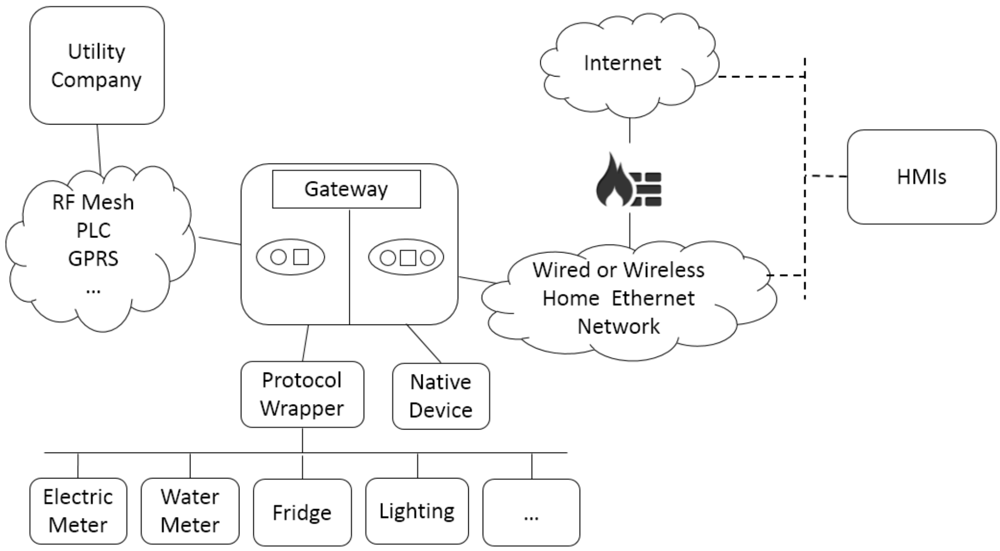

Figure 3. Proposed solution block diagram

2 SYSTEM DESCRIPTION

Following chapters provide an introduction to functionalities of the system as well as shortly describe its environment with the focus on system limitations.

2.1 Functionalities of the System

Actual functionalities of the system constantly evolve as ones are introduced to the user and are re-evaluated with each review cycle. Functionalities listed below serve as a starting point for further iterations in lean software development.

2.1.1 Improved Communication

Two way communications between the utility and consumers mean more than just sending of the current tariff and meter readings. Two way communications could include for example:

- In direction from the utility to consumers
  - Current energy tariff
  - Information on planned and not planned maintenance
    - Maintenance schedule
    - Maintenance reason
  - Consumed energy sources
  - General information on energy quality and availability
  - Control/Event signals, e.g. anti-blackout signal
- In direction from consumers to the utility
  - Detailed but anonymous information on energy consumption
  - Contact requests and messages

2.1.2 Energy Source Selection

The consumer should be able to configure preferable energy sources, e.g. to use energy generated from renewable sources, such as wind or sun. Environmentally aware consumers may want to pay a bit extra for eco-friendly energy. Information on which consumers were using energy from which source would be based on consumer preferences, their actual consumption and energy production information on given time. The utility would also learn about customers' preferences and would be able to take that into account while considering future investments. Consumers would have bigger impact on the positive development of the energy market.

2.1.3 Load Management

A changeable tariff is a simple and basic solution for load management. Energy would be more expensive when less of it is available and cheaper otherwise. Energy price would be bound to energy availability in real time. Smart devices will try to optimize their energy usage to be more economical for the owners. For example, the fridge will drop temperature a bit below the set temperature while electricity is cheap and will allow it to rise a bit when electricity is expensive. Thus, it will use more energy when the tariff is low and will try to consume less on peak hours.

2.1.4 Prosumer

A prosumer in that context means producer-consumer. It describes the case in which the consumer is not only buying energy but is also selling it. In that vision it is closely related to an energy positive house. Modern houses of the future would be an evolution of the passive house which is not only optimized for energy efficiency by proper isolation and lighting but which could also produce and store energy. For example, it would have its own photovoltaic (PV) solar panels, small wind turbine, MicroCHP and an electric car in the garage. The house would be able to produce its own energy, as well as to buy and store when cheap and sell when expensive.

2.1.5 Avoiding Blackouts

The concept of smart grid introduces many possibilities for the avoidance of energy delivery disruption. Some of them are listed below:

- Smart devices with anti-blackout interface implemented could be switched off when needed.
- Energy could be borrowed from consumer, e.g. from electric cars
- Changeable tariffs help to shape demand. Some consumers may consider doing certain activities when more and cheaper energy is available, for example, washing clothes or dishes, taking sauna, heating home/water, etc. These would include mostly activities which do not have to be performed exactly at a given time and postponing of which do not have significant impact on quality and comfort of living.

2.1.6 Customer Needs Understanding

Utility companies could improve consumer services based on better understanding of consumer needs and their behavioral patterns. This functionality concerns customers privacy and ultimately it should be their decision what is made available to the utility. Some additional incentives for sharing would probably help to improve the participation ratio. Information shared with utility could include following:

- Energy usage by device and time
- List of devices at home with their energy classes
- Preferable energy sources
- Preference as for example putting energy cost over comfort or the other way around

2.1.7 Automatic Failure Notification

The system is automatically monitoring its condition and would send warning or alarm when appropriate. Warnings are sent when a deviation from the expected quality is not yet significant but it also could not be defined as a good quality. Alarms are sent when a deviation from expected quality is significant and indicated for a fault in the system. Messages coming from different locations along with attached log data would significantly help to determine the root cause of the problem and would significantly decrease the downtime period.

2.1.8 Current and Historical Readings of Energy Usage

The system provides access to current and historical readings of the energy usage. It is not only limited to the functionality provided by smart meters but the system independently stores usage information from other sources as well. For example, it can show historical usage of the electricity and water for a dishwasher.

2.1.9 Control Lighting

Lighting could be controlled with physical switches or via any available interfaces, such as touch screen panels, mobile phones, tablets and web interfaces.

In order to save energy, lights could be automatically turned off after specified inactivity time is detected. The activity could be detected by:

- Alarm System Sensors
- Usage of Smart Devices
  - Changing TV channels
  - Working on computer
  - Opening/Closing Fridge
  - Etc.

The system also allows for creation of the lighting schemes. Besides obvious control of lamps it would involve also, for example control of blinds, projector, electric fire place, etc.

2.1.10 Control and Monitor HVAC

The system easily allows to control and monitor humidity, ventilation and air conditioning (HVAC). The input of the process includes:

- Desired rooms humidity and temperature
- Preference as for example putting energy cost over comfort or the other way around
- Weather forecast
- Energy forecast

- Current HVAC state
- Calendar to determine when residents are at home

The system utilizes the whole potential of the home as for example controlling blinds in order to control energy exchange between inside and outside space. When available and desired it helps to utilize the fireplace as an additional heating source.

2.1.11 Estimations for Future

The mobile application advices with the purchase of new devices. The system learns about our behavior and is able to predict how a new device would affect home economy. For example, while buying a new fridge it would be able to say how much we would save per month/year if we decide to go with the given fridge. Simply it creates estimates for future usage based on past behavior.

2.1.12 Water Leakage Detection

The system can apply a variety of logical conditions for water leakage detection and damage limitation. Further damage is limited by turning off the main water valve when the leakage is detected. Some logical conditions which could be used for the leakage detection are listed below:

- Alarm system is engaged, i.e. no one at home. However, the water meter is detecting continuous water flow. The washing machine or dishwasher does not take water continuously for any longer period of time. The shower, water taps, etc. are assumed not to be running while no one at home.
- Residents at home. The system detects continuous water flow. It is assumed that the system is sensitive enough to detect even small water flow. It could be achieved with new water meters. In case the inhabitants are at home, much longer period of time needs to be awaited before the leakage is assumed to take place, for example, someone could be

taking a long shower. However, if there is some continuous water flow for a period of one hour or more, it could be assumed that a leakage occurs.

2.1.13 Moisture Detection

A building could be equipped with a number of humidity sensors allocated at strategic places. These sensors could be built-in into the main building construction, for example:

- Inside wall, attached to the main beams
- Inside concrete floors, at least in critical spots as for example a bathroom
- Building foundations. Additional benefit of allocating the sensors within concrete is during the construction to control humidity of the concrete which is an important factor. Construction work should not be continued till concrete humidity is at certain level.
- Under the roof
- Within outer cover of the water pipe. New water pipes contain outer cover and inside pipe through which water actually flows. In case of the main pipe leakage, water will flow within outer pipe.

Besides controlling humidity in order to prevent building construction damage, it is also an important factor from the perspective of the human health. High humidity means potential problems with mold which in turn could trigger health issues, such as asthma or allergic reactions.

2.1.14 Smart Devices Status Monitoring and Maintenance

The system communicates with smart devices which are capable of their own condition monitoring. These devices are able to detect potential problems and advice on required steps. They also notify when a periodic maintenance is required. For example, a car could indicate necessity for:

- Oil change
- Increase tire pressure
- Brake pad replacement

2.1.15 Suggestions for Improvements in Energy Consumption

The system learns about the energy consumption of the user. It knows also about devices at home and their characteristics. Based on that, the system is able to suggest more economical usage of the energy. For example, the system can notify the user how much money would be saved on electricity if an old energy inefficient fridge was replaced with a new one. Another example, if there is an electric and gas cooker, it could advice which is best to use at the time being.

2.1.16 Learn to Use Energy Efficiently and Economically

The system learns to use energy in a more efficient and economical way. It learns based on historical data gathered. It is used to adjust a different configuration. The user can override it but in case if it was not done then the system tries to optimize itself, for example, to know when the inhabitants are at home, when they sleep, in which areas usually they stay at given part of the day, etc. For example, it would be used to optimize heating and cooling.

2.1.17 Tariff Forecast

Utility companies should provide a short term tariff prediction. These predictions are used then by devices and consumers to make wise decisions. Energy production and consumption levels could be determined for example based on pattern learned and weather forecast. Knowing expected cloudiness and wind speed, the amount of energy produced in a given area from solar and wind power plants could be predicted. Rarely, it could happen that the actual value will not match the predicted one. However, it should be limited to incidental cases and on daily basis. As with any forecast, if it is highly inaccurate, then no one will check it. The tariff forecast is beneficial for both sides. The consumer knows when energy is expensive and can try to limit its consumption. For utility, it helps to move part of the load out of high demand hours.

2.1.18 Usage Limits

In order to promote rational usage of the energy, limits are introduced. The unit price of the energy rises each time the amount of consumed energy reaches threshold of the next level. The limits could be daily, weekly or monthly. For example, it could be used to promote rational usage of the water. The limits could also be based on the number of inhabitants, season of the year, location, etc. The main point is to come with rational limits which do not reduce comfort of the life but which promote efficient usage of the resource.

The purpose of the system is to help users in rational resource usage. It could be achieved by:

- Clear visualization of the current usage in reference to limits
- Gentle notification when the resource usage per task or activity takes more than planned or is not properly executed. E.g. a gentle sound or light notification when the shower is prolonged or when watering the yard in middle of the day.
- Advising on hardware purchase. E.g. when noticing that the low flow showerhead is not used.

2.1.19 Utility Using Consumer Goods as an Energy Storage

Utility company can use electric car when it is not used and in return it provides cheaper energy for loading. Some kind of schedule would have to be agreed between utility and owner. Battery would be used by utility during time when car suppose not to be used. Depends on agreement level it will allow for more or less frequent out of schedule trips. Car should never be totally drained so that it cannot be driven. There always need to be energy for given kilometers to be driven.

2.1.20 Creation of the Energy Plan

The user can create an energy plan for a given period, e.g. a year or a month. The plan is done based on the cost or energy amount. The system helps to create the plan using its knowledge of the house and inhabitants. For example, if the house is electrically heated, it will allocate more energy to be used during the winter months than summer. The system presents to the user a number of suggestions for energy saving. After creating the plan the user is guided through its execution.

2.1.21 Heating Pole

Heating poles used to warm up cars engines during the winter could be remotely controlled. For example, the user needs to leave earlier than planned. With an old ordinary solution it would require going outside and manually changing timer. With the heating pole as a smart device everything could be done via any available user interfaces, such as mobile phone application.

2.2 Vision of the Whole System

The system can be divided into internal and external in perspective to the smart home in which it is installed. The internal refers to the parts of the system which are within the home and over which the user has the greatest control. The external refers to the outside world, to the parts of the system which create a bigger part and expose attributes and functionalities; which could be monitored and controlled remotely to home and by other entitled entities than the inhabitants of the smart home. Some parts of the system belong to both parts. However, attributes and functionalities exposed could depend on where they are accessed. A smart electricity meter could serve as an example. Its readings should be available to both, utility company and home inhabitants. However, changes in the tariff should be only available while accessed from the utility company. The reason for dividing and differentiating between internal and external comes mostly because of users' privacy aspect. The system is intended to make our lives better. However, in order to do it so it has to learn a lot about users and it has to have a vast control over the building automation and appliances. Therefore, it should be the users' decision how much of that they want to expose to the outside world. The users need to have and to know that they have control over their privacy. Otherwise they may not be interested in or even can be actively against the system.

We can identify following parts of the system:

- Gateway – the heart and brain of the system. The gateway communicates with all devices at home and control exposer to the outside world. It is able to communicate and control a vast number of devices. The gateway provides translation between the native protocol used and any other one. It is able to make a smart device out of dummy one and it takes care of the whole logic of the system.
- Human Machine Interfaces – for all of them, the main functionality is to provide user friendly access to the system. All brilliant functionalities mean nothing if the user does not understand or do not want to use them. Therefore, the user interface has to be appealing, intuitive and easy. It could be e.g.:
    - Wall Mounted Touch Screen
    - Web Interface
    - Mobile Device Interface
- Smart Device – a device capable to communicate and which exposes at least a single attribute.
- Dummy Device – not a smart device
- Protocol Converter – a device capable to convert between different protocols. It is placed on the boundary between two different protocols and it allows for interoperability between them, for example, the device that is able to convert between the native protocol of this system and IEC 61850. From the perspective of the IEC 61850, the system would be seen as Intelligent Electronic Device (IED). On the other hand, from the perspective of the system, IEC 61850 would be seen as a single or multiple smart device. The purpose of the protocol converter is to take care of interoperability and make it work seamlessly from both perspectives.
- Smart Meters
- Utility Company

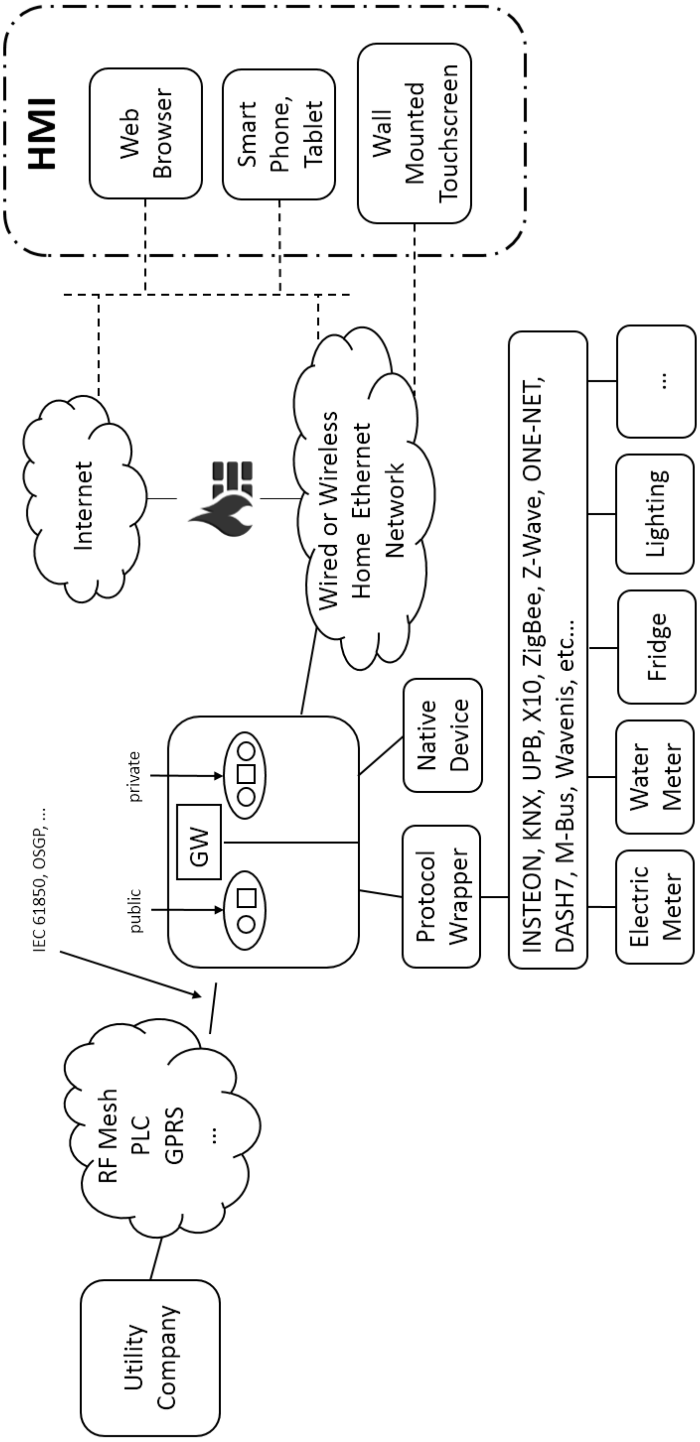

Figure 4. System overview

2.3 Concepts Definitions

Short definitions of basic concepts used within the thesis are provided below.

2.3.1 In-Home Energy Management System

The In-Home Energy Management System is the key element of efficient energy usage. For that purpose it is able to bring all elements together and make them work in harmony with each other. It tries to fully utilize resources available in smart home to improve energy efficiency and comfort of living. It communicates also with energy management system of the utility company in order to provide benefits for both parties, consumers and utility companies.

2.3.2 Smart Grid

The Smart Grid is about bringing together power generation and distribution with information and communication technology. It is about improving the efficiency and sustainability and other important factors, such as economics or reliability.

2.3.3 Smart Home

The Smart Home is a home designed in a way to enhance its security, usability, energy efficiency and comfort of living by the vast use of building automation and information and communication technology.

2.3.4 Smart Meter

The Smart Meters are a new generation of metering which allows for two way communication between the meter and the utility company. The smart meters should allow for the frequent logging of specific medium usage (electricity, water, gas, etc.).

Smart meters should allow for:

- More frequent logging of utilities usage so that e.g. billing can be based on the consumption time.

- Automatic Meter Reading (AMR). Statistical data regarding medium consumption could be read remotely by the utility company.
- Remote control so e.g. electricity could be turned off/on remotely from the utility company office (lowering operation costs)
- Detection of illegal usage
- Detection of service outage

Smart meters should help with:

- Balancing of medium consumption through applying different pricing based on the demand.
- Understanding of consumers' usage pattern e.g. by learning when the highest and lowest.

Smart meters are already installed for millions of consumers and during the time they will replace old meters. Regulations in many countries are forcing utility companies to use smart meters.

2.3.5 Smart Device

Smart Device is any object with at least single attribute and capable to communicate in at least single direction. Most often it would be device capable of two way communication which can perform actions as for example changing states between ON and OFF.

2.4 Limitations

The following chapter lists a number of identified limitations which need to be taken into consideration while working on the solution. Some of them are valid and probably should remain and some should be adjusted to the changing world.

2.4.1 Legislation

Certain functionalities cannot be controlled remotely, for example, turning the sauna on. This is due to fire hazard. Controlling it remotely would likely cause more frequent incidents. For example, the child could left a flammable object on a heater and the parent remotely turning sauna would cause a fire. It serves just as an example and more of them should be identified.

2.4.2 Insurance Companies

Insurance companies are important stakeholders. Most of the potential system users are likely to have a building insurance. The remote or automatic control of smart devices may not get an initial approval. Therefore, insurance companies should be involved in development of the product. They should be assured that system is not to be seen as hazardous, but as a positive development in the improvement of buildings safety. Potential aversion should be replaced with an opportunistic vision as the system can limit damages caused by water or fire.

2.4.3 Customers

Potential customers can be categorized into:

- New technology followers – ones who love to have the latest technology gadgets. They are willing to pay more and can accept imperfections easier as they are aware that they are dealing with some kind of prototype.
- Average – definitely the biggest group among the three. They are harder to please in comparison to the previous group. In order to target this group, the product needs to be finished. The advantages of using the product needs to be clear to this group.
- Technology scared – ones that do not like any changes.

During the design and development the average users should be taken as a target group, while the new technology followers could be used to help with the product development and testing.

3 HUMAN MACHINE INTERFACE

The Human Machine Interface defines part of the system which is responsible for interaction with the user. This chapter is mostly concerned about User Interface design than on software architecture as user experience has higher importance than engineering details.

3.1 User Interface Design

Requirements for User Interface implementation are as follows:

- Targeted for touch screen device
- Easy to understand and follow; minimum learning curve
- Animations should be used when adding on user experience but not too extensively
- Rather simple than complex. Simple and clear design will add on user experience.

3.2 Main Menu Window

The main menu window contains a number of big tiles. Each tile contains an icon and its title. Both should unambiguously indicate for the purpose and content of the underlying sub-window. The content of the sub-window is displayed after clicking onto the tile. Additionally, on the bottom, energy availability is depicted by a number of symbols displayed. They are:

- Lightning for an electricity, more lightnings means cheaper electricity
- Water drop for a water, more drops means that user stays within allocated limits. Crossing each usage limit causes one water drop less displayed.
- Fire flame for a heating, more flames means that heating is done economically.
- Gas cooker for a gas, more cookers means that the user stays within allocated limits. The logic is the same as for water.

On the right side of the window, the following icons can be found:

- Gear for a settings. A click causes the main settings window to appear. It allows for more advanced configuration of the system. As for example creating an energy plan, setting carbon footprint goal, selection of preferred energy sources, etc. Access to the window could be additionally protected.
- Question mark for a help. A click causes the help window to appear. It works as a detailed user manual.
- Lock for locking the screen. A click causes the screen to be locked. When the locked number of images or animations can be displayed, depends how it was configured. Additionally, the displayed content could be linked to carbon footprint or generally to energy usage. The content scenario in that case would be more nature rich when the user's energy usage is more ecological and less otherwise. In order to unlock, the correct pattern on the mesh of dots need to be provided. It is used to protect the interface against unauthorized use, for example by young children.

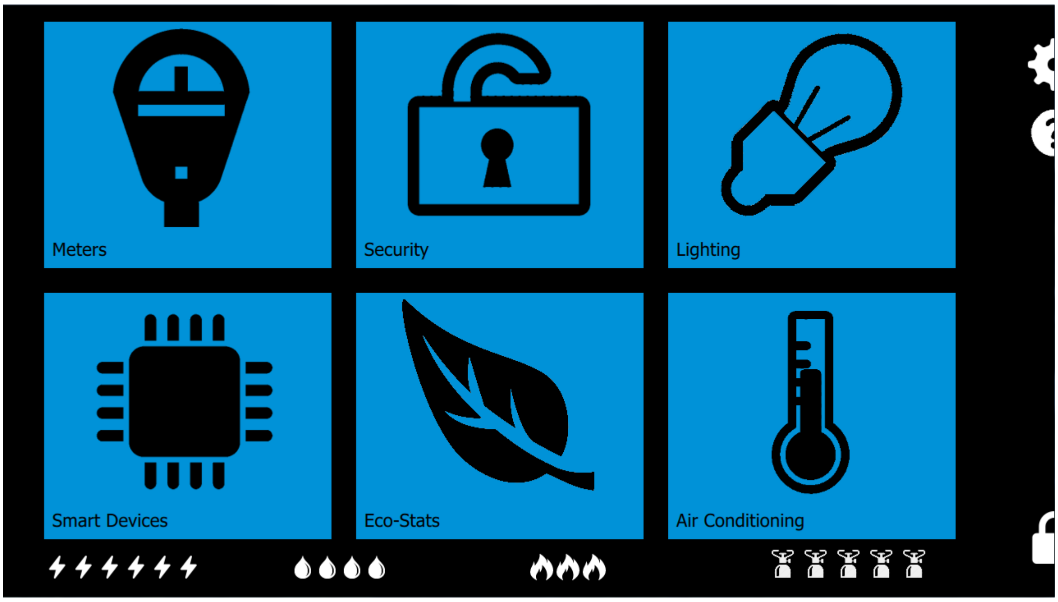

Figure 5. Main menu window

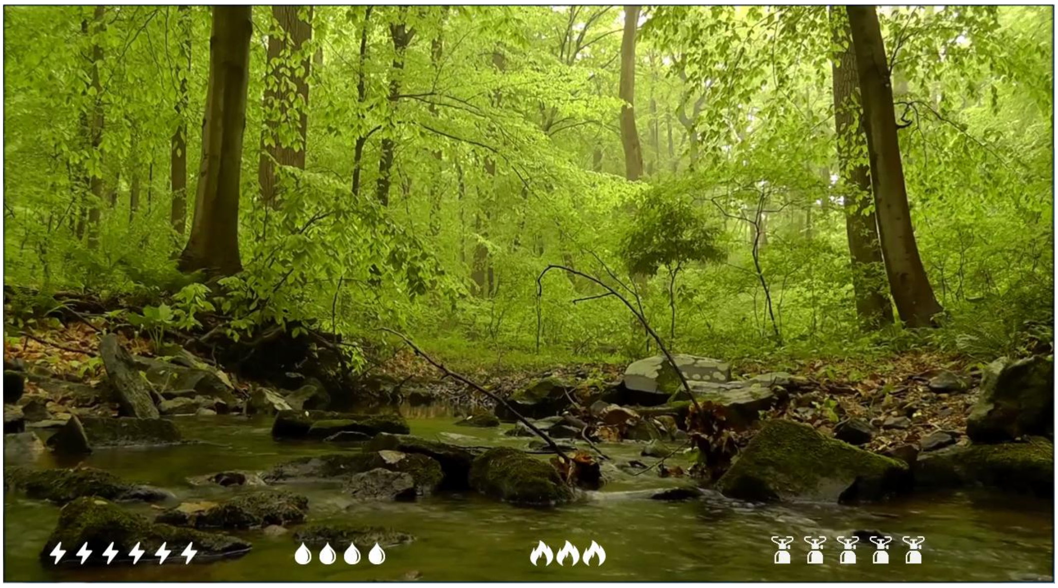

Figure 6. Lock screen window

3.2.1 Animations

Animations are as follows:

1. Wave animation on a tile click

   Clicking a tile triggers an animation of the wave. It aims to provide feedback to the user that its interaction is encountered.

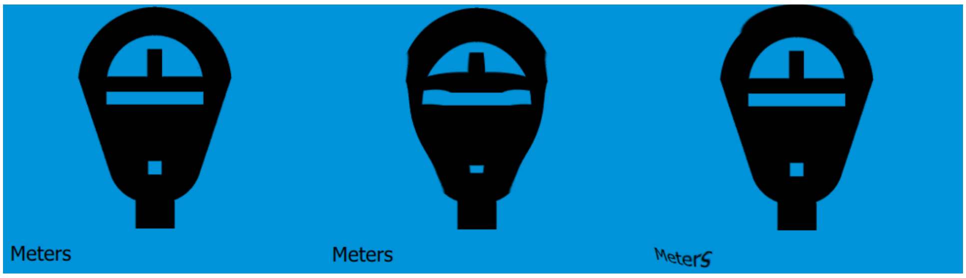

Figure 7. Selected frames of the wave animation

2. Moving the main window to the right and revealing the desired sub-window. This animation helps to understand that the main menu window is on the right side of the sub-window. It is used to navigate back to the main menu window by clicking the right arrow located on the right edge of the sub-window.

3.3 Meters Window

The meters window contains the tiles of smart meters with theirs readings and tariff price displayed. More information is available after clicking on selected meter. At that moment, the menu will slide from under the bottom of the meter and will present the user with two options to select. It is either checking on the tariff forecast or going to the details window.

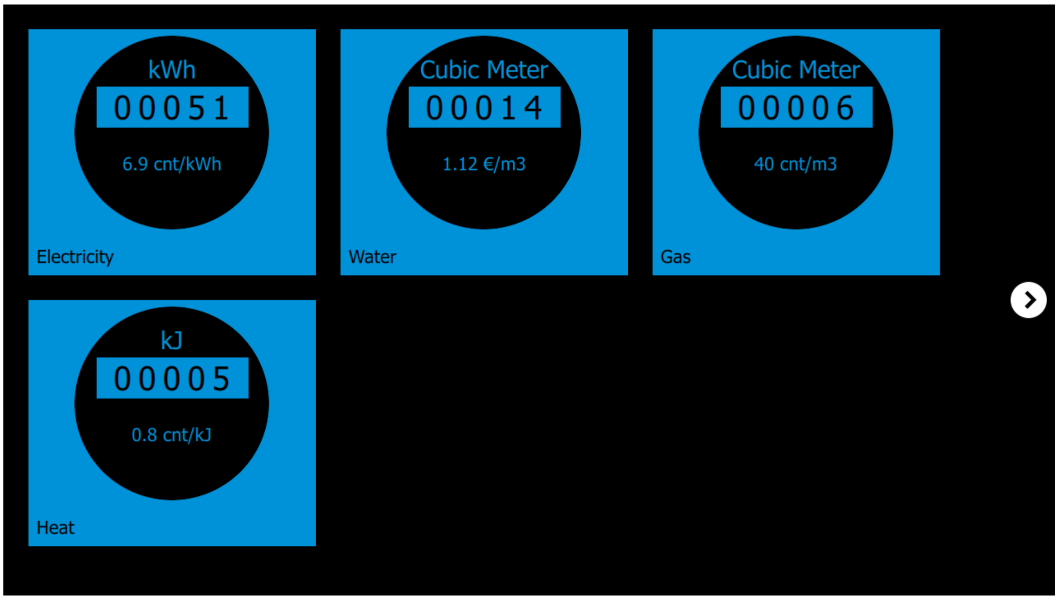

Figure 8. Meters window

3.3.1 Animations

Animations used are as follows:

1. Revealing of the tile menu
   The tile menu is normally hidden in order to create a neater interface. It is displayed only when needed. When the tile is clicked it reveals itself from the bottom of the tile and automatically hides after given timeout.
2. Changing meter value
   An analog counter alike animation is used while the meter value change. The animation introduced in order to improve the perception of the product quality by not too intrusive accent.

3.3.2 Energy Forecast Chart

The energy forecast window shows predictions in the development of the energy price over the given time period.

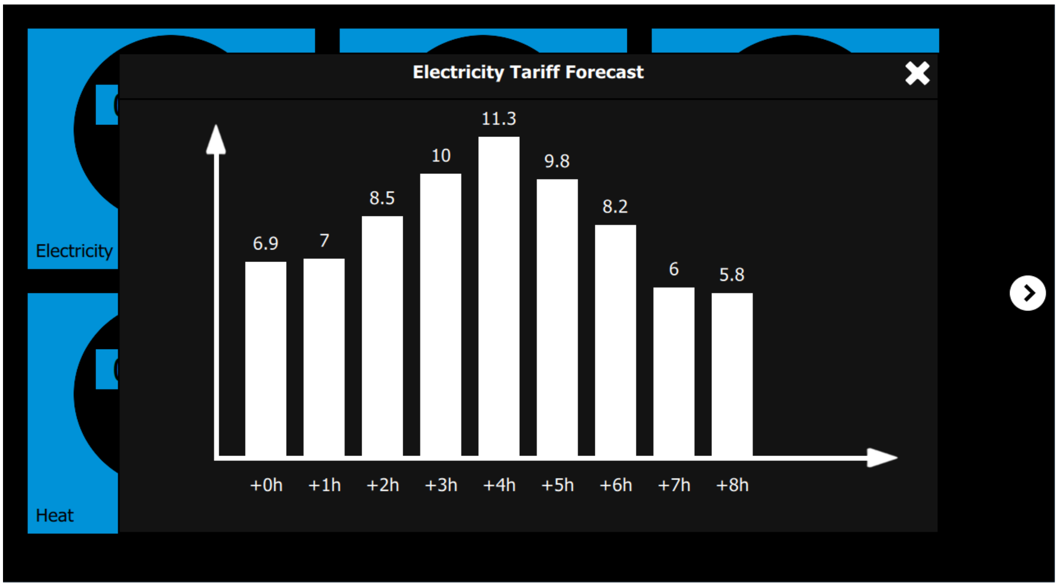

Figure 9. Energy tariff forecast chart.

3.3.2.1 Animations

Chart bars are animated. All bars and values start at the zero level and they rise up to the final level over a short period of time. The animation is introduced in order to improve the perception of the product quality by not too intrusive accent.

3.3.3 Energy Consumption Detail Window

The energy consumption detail window provides detailed information of which device used how much of the energy over the given time period of 24 hours, one week, one month or one year. Maybe even more importantly, it shows how much it costs the user. These have to be indicated separately especially because of changeable energy tariff.

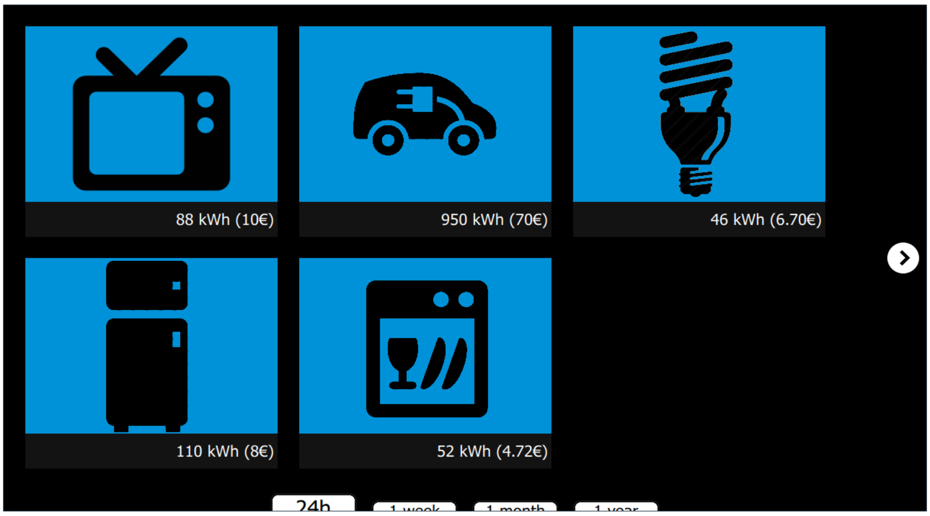

Figure 10. Energy consumption detail for the electricity meter.

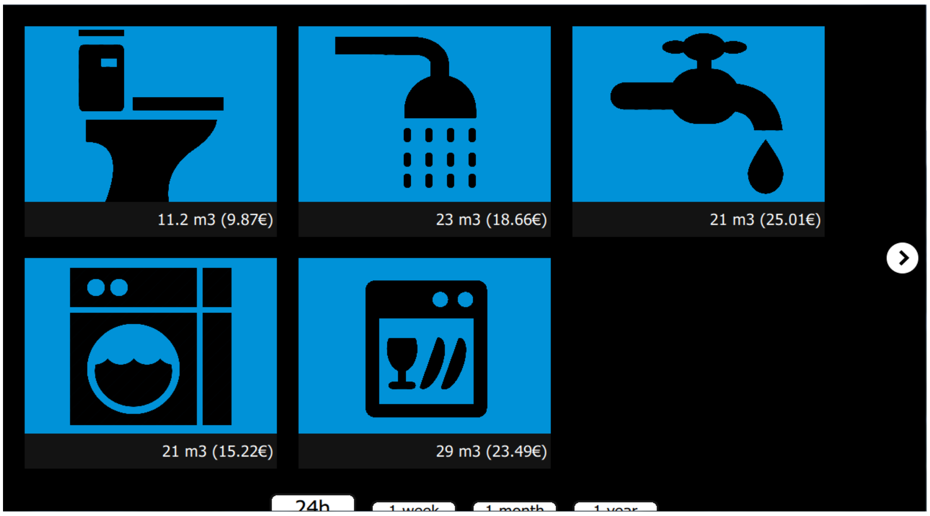

Figure 11. Energy consumption detail for the water meter.

3.3.4 Energy Usage Chart

The energy usage chart shows energy usage over the time period in relation to the energy tariff and energy class. It helps to determine if energy is used efficiently and economically. The energy class serves as a benchmark. It shows a benchmark according to the smart device. If the device is supposed to be “A” class, it will show the benchmark for “A” class, if it is “A++” it will show the benchmark for “A++”, etc. Probably even more important it is to see when energy is used. Is it used economically or not? That could be determined with the analysis of the energy usage in relation to its tariff. Preferably most of the energy consumption should be during off peak hours.

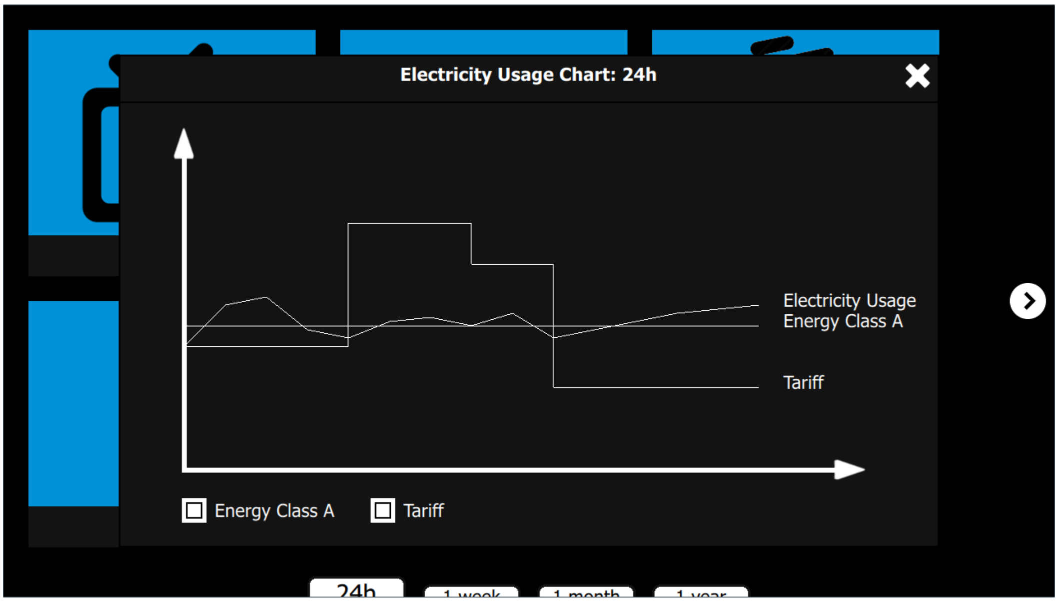

Figure 12. Energy usage chart.

3.3.4.1 Animations

Chart lines are animated. All lines and theirs labels start at the zero level and they rise up to the final level over a short period of time. The animation is introduced in order to improve the perception of the product quality by not too intrusive accent.

3.4 Security Window

The security window is composed of two main parts. On the left the house floor plan is displayed with different sensors marked on it. The sensors are interactive and the user can interact by clicking any of them. The second part is an alarm keypad which is used to arm and disarm the alarm system.

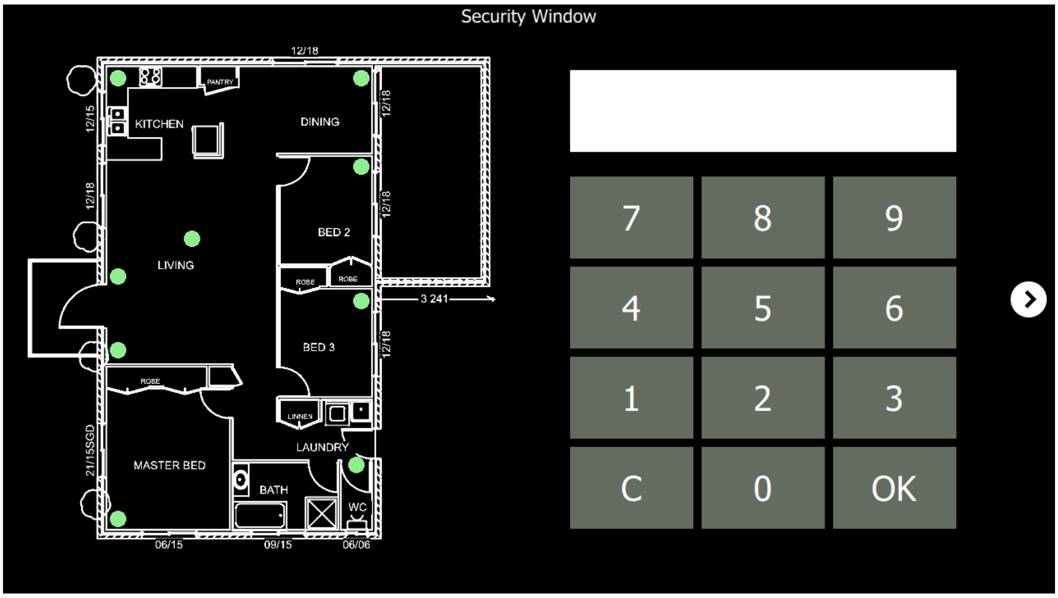

Figure 13. Security window

3.4.1 Animations

Animations found in this window are the following:

1. Blinking sensors

   The sensors are blinking in order to indicate that they are interactive elements. The user can click the selected sensor and she/he would be then presented with some additional information regarding the selected sensor. Different colors are used to indicate specific states

   - Green – everything is OK
   - Yellow – user attention is required
   - Red – something is wrong.

2. Zoom-In of pressed button

   In order to provide the user interaction feedback, the keypad buttons zoom-in when clicked.

3.5 Lighting Window

The lighting window provides control over lighting at home. All remotely controllable light related devices are depicted onto that window. It includes lamps, blinds, electric fire place, etc. Additionally, pre-configured lighting scenarios are available with one click, allowing to turn on/off all lights or to set perfect lighting for reading, watching movie, banquet, cooking, etc. The system is equipped with light sensors and it tries to fully utilize natural light. It will automatically adjust as natural light condition change.

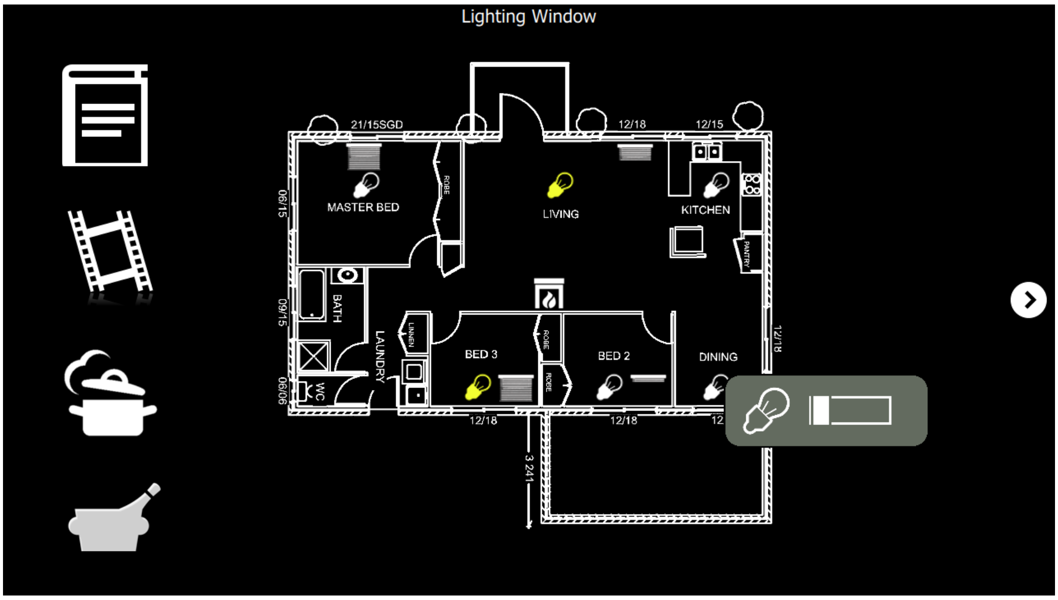

Figure 14. Lighting window.

3.5.1 Animations

Animations in this window are:

1. Pulsating active elements (zoom in and out)

   Active elements pulsate in order to indicate to the user that interaction with them is possible.

2. Light scenarios scrolling
   Icons representing light scenarios are scrolled vertically on the screen in order to display selected part of bigger icons collection.

3.6 Smart Devices Window

The smart devices window lists all smart devices available. Monitoring and controlling of those devices is done from that place. Every device exposes its own set of functionalities to the user. For example the user can:

- Set the washing machine to start when energy is cheap
- Set the fridge temperature and allowed variation
- Check on the progress of the dishwasher
- Check when the car requires maintenance

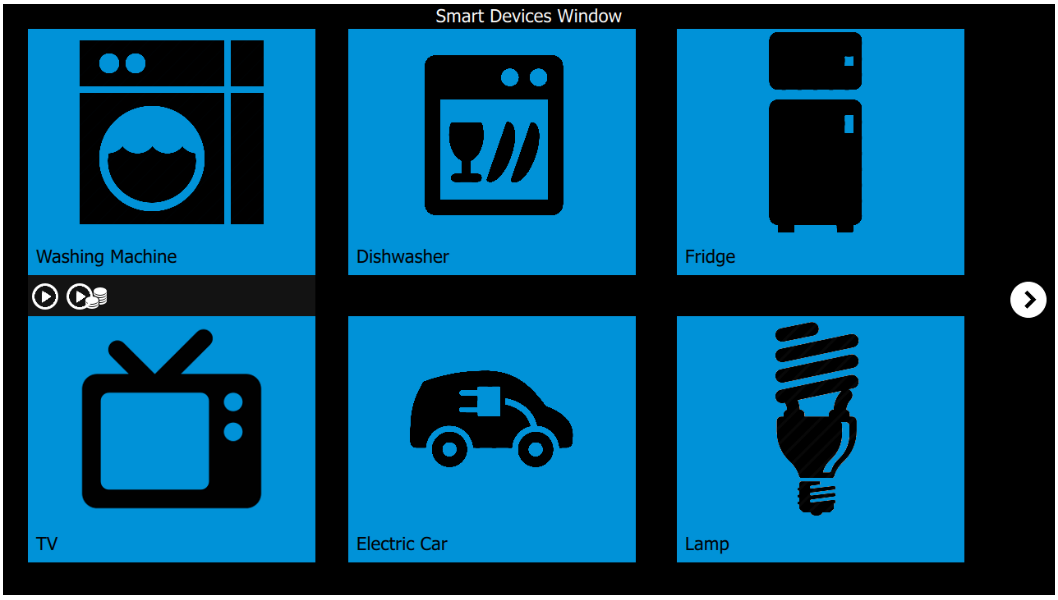

Figure 15. Smart devices window.

3.6.1 Animations

Reveal of the tile menu is animated as it is normally hidden in order to create a neater interface. It is displayed only when needed. When the tile is clicked it reveals itself from the bottom of the tile and automatically hides after the given timeout.

3.7 Eco-Stats Window

The Eco-Stats window shows information related to the ecological energy usage. It includes the energy sources chart and carbon footprint. The purpose of the first is to depict energy sources and to promote renewable ones. The second one shows the user's carbon footprint with relation to the goal set.

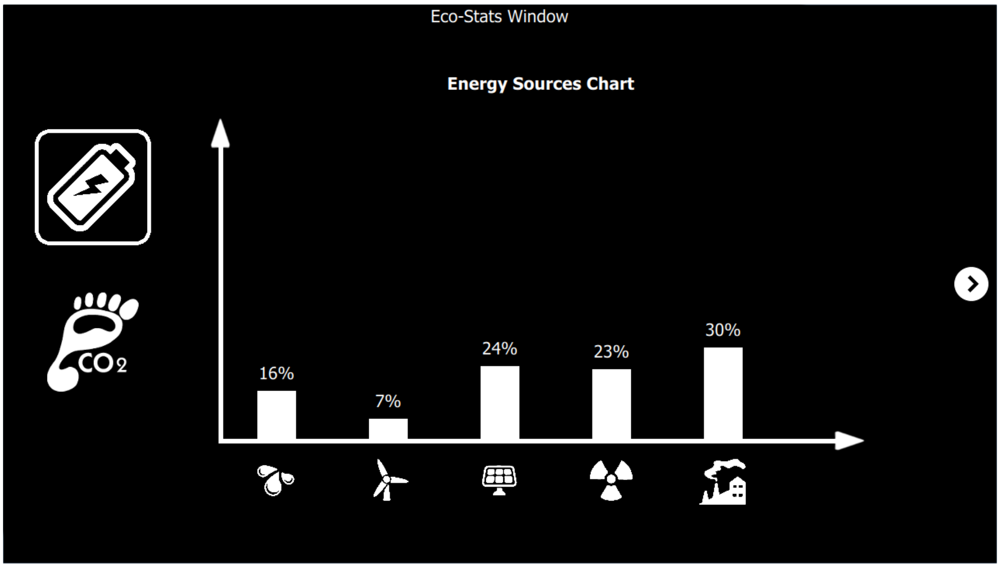

Figure 16. Energy sources chart.

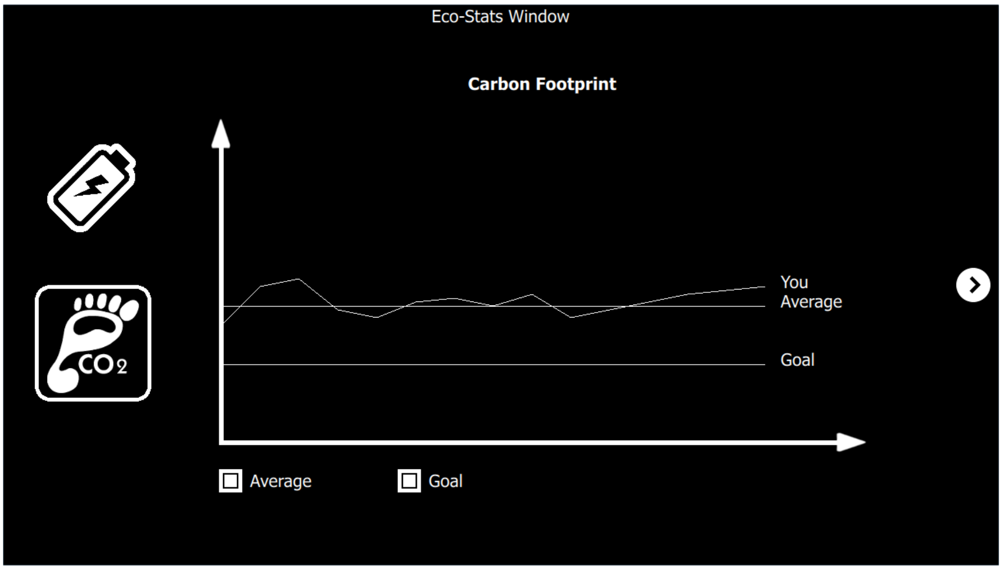

Figure 17. Carbon footprint.

3.7.1 Animations

The change between the charts is animated. The energy source chart comes from the top of the screen and the carbon footprint from the bottom. The charts could be also changed by a swipe of the finger on the chart area adequately to the logic direction.

3.8 Air Conditioning Window

The air conditioning window lets the user to control the temperature in rooms. The system will try to do it in an energy efficient way. Additionally, on the left side it shows the weather forecast for coming days. Colors are used to indicate temperature to be:

- Optimal – green
- Higher than optimal – red
- Lower than optimal – blue

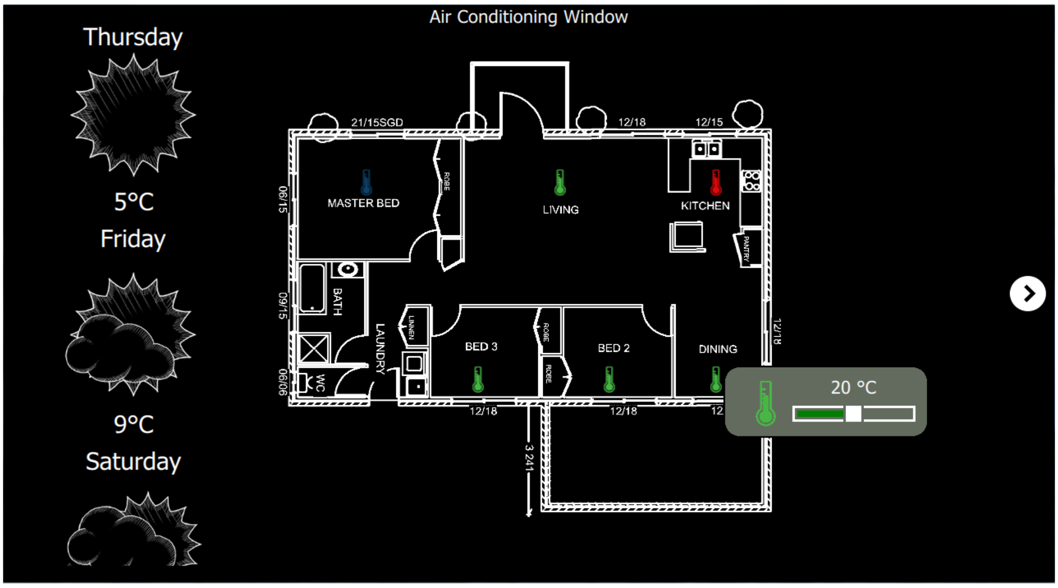

Figure 18. Air conditioning window.

3.8.1 Animations

Animations in this window are:

1. Pulsating active elements (zoom in and out)
   The active elements pulsate in order to indicate to the user that interaction with them is possible.
2. Weather forecast scrolling
   Icons and labels, which represent days of forecasted weather, are scrolled vertically on the screen in order to display selected part of bigger forecast period.

4 MODEL

Model is one of the key elements of the solid and versatile system. Its main details are provided in the content of this chapter. However, it is not meant to be full specification and some in-depth details are omitted.

4.1 Definition of the Model

The model could be defined in many ways. In the context of this thesis it means a set of concepts and rules which are used to represent real world entities for a computer system in order to allow and simplify operations on them.

4.2 Objectives for the Model

Objectives of the model are as follow:

- Conceptually simple and versatile
- Not limited to the smart meter, smart grid or any other scope
- Limits could come from the model implementation but are not introduced by the model itself

4.3 Concepts and Rules

The model was developed with extensibility as a first priority over resource usage. It is assumed that the model at its full capabilities is implemented by the devices which are able to handle additional overhead coming from the versatile approach.

The model could not be highly optimized because it does not incorporate any constraints as to its scope, type of objects, data types, supported attributes and functions. With many modern devices the additional overhead of the model should be insignificant. However, it is also assumed that the full implementation of the model will not be used at every step. Translations to other models should be implemented. It most likely would be required while communicating with simple embedded devices, such as for example sensors.

The basic element defined by the protocol is an “Object”. The capabilities of the object are defined by the interfaces which it implements. The model defines interfaces and their mandatory functionality which needs to be implemented by each object which claims to implement the specific interface.

The object is very general and it is not limited to any specific category of devices. It does not have to be device, either. It could describe something that is physical as well as abstract. The object would usually have a number of attributes and functions associated with it.

The devices are not grouped because it would be very difficult or probably even impossible to define groups for devices which could be anything. As a result some devices would belong to many groups. Using the idea with interfaces is simpler and therefore more versatile. Every object/device could implement as many interfaces as it needs.

4.3.1 Object

The object is very general and virtually could be anything. The object is described by interfaces which it implements and optionally by number of other member objects. Every object has to implement at least “Object Interface”.

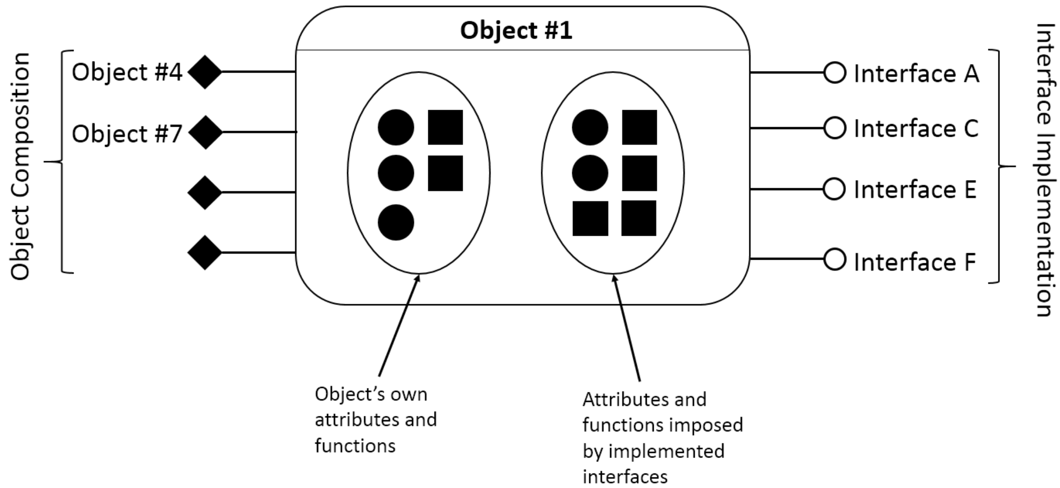

Figure 19. Object.

4.3.2 Interface

Every interface defines a set of mandatory as well as optional attributes and functions. For example "On Off Interface" requires "State" attribute and "ChangeState" function to be implemented by each object which claims to implement it. It is not necessary to know the object and its full functionality but by knowing which interfaces it implements we know how to interact with it. Within the interface, the attribute number zero contains the description of the interface itself.

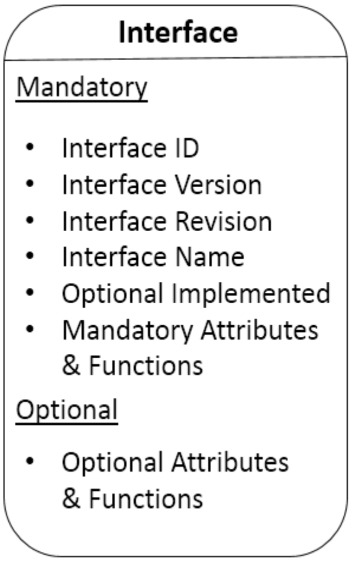

Figure 20. Interface.

4.3.3 Attributes

Attributes are part of the interface. They are used to represent physical and intangible properties of the object which implements the given interface.

Meta-attributes are normal attributes which are used to describe other ones. They can be divided as follows:

- Mandatory
  - ID
  - Name

  - Data type
- Optional
  - Min, max and default values
  - Unit
  - Scale
  - Enumeration

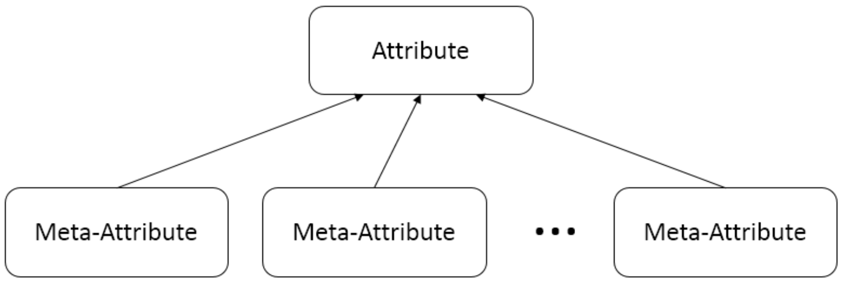

Figure 21. Attribute and its meta-attributes.

4.3.4 Functions

The function is in fact modeled as an attribute with a set of meta-attributes accompanying it. The function is called with a set operation on a function attribute, if a required number of function arguments (meta-attributes) should be set prior to call. The execution state meta-attribute can be read at any moment. If the function execution finishes normally, then the return value is set. Otherwise, the error value is set.

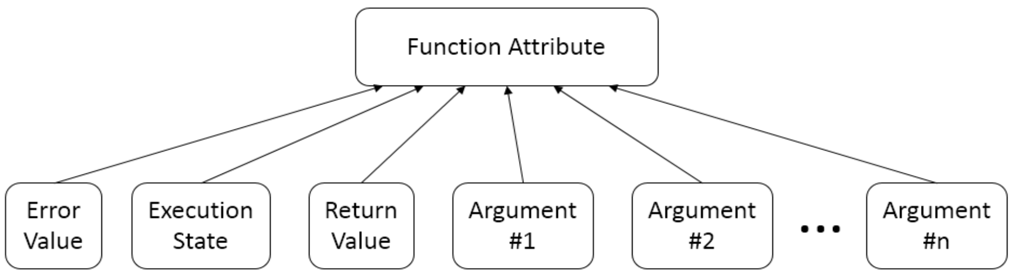

Figure 22. Function and associated meta-data.

4.3.5 Data Types

Primitive data types as well as composite ones are supported. The composite data types are constructed using the primitive ones. The composite data types could be nested in each other without the limitation on the depth. An array of any primitive data type values is supported as well.

The following is the list of primitive data types supported:

- Boolean
- Byte
- Unsigned Integer
- Signed Integer
- Unsigned Long
- Signed Long
- Double
- Binary
- String

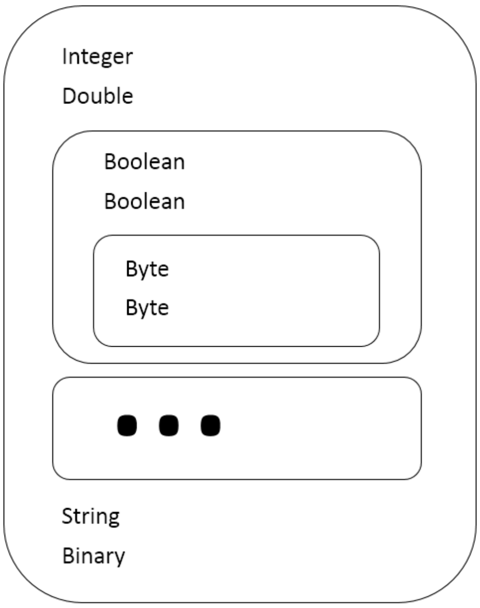

Figure 23. Composite data type.

4.4 Interfaces Definitions

The following interface names and descriptions serve as examples. It is not their specification.

Interfaces used in smart grid and home solution:

- Object Interface – Interface used to describe object itself.
- Widget Interface – Interface to be used with objects which are having their visual representation implemented. The HMI uses that interface to display custom widgets which are provided by the manufacturers. It gives manufacturers an opportunity to customize users' experience. With custom widgets it is possible to utilize all capabilities of the device. The custom widget can provide access to all features, not only mandatory ones.
- Meter Interface – Interface to be used by metering devices, for example the electricity smart meter. The interface gives access to the meter readings. An access to the current meter reading is mandatory but optional devices could support history readings as well.
- Location Interface – Location coordinates and textual description. Coordinates are accompanied with the reference to the map. The location could be as:
    - Pixels – x and y on a map image
    - Longitude and latitude – in order to position an object anywhere on the earth
    - Sector – row and column
- Tariff Triggable Interface – Interface which would be generally implemented by devices of which operation could be affected by the energy tariff. For example, a dishwasher could be set to start when electrical energy is low.
- On Off Interface – A simple interface to the state which could be represented in binary logic, 0 or 1, on or off.
- Adjustable Interface – Interface to be used with properties whose values could be adjusted within a specified range and at a specified step, for example blinds and their light passing percentage.

- Battery Interface – Interface to be used with devices equipped with batteries. It provides basic information about the battery as for example its SOC (State of Charge).
- Energy Consumer Interface – Interface to be used by devices which are aware of their energy consumption. For example, the washing machine could provide information about its energy consumption over the given period of time.
- Heat Regulator Interface – Interface to be used with devices which control temperature, for example a thermostat.
- Anti-blackout Interface – Interface to be used by devices which are aiming to help to avoid blackouts. Devices which implement that interface could be turned off in case of energy shortage. Some of them can even provide energy to the grid. For example it could be a case with an electrical car. The users of those devices most likely would be given some kind of incentives.
- Date Time Interface – Interface to be used by devices which have an internal clock. They are mostly used to synchronize the time of all of those devices to one. The list of devices which can be equipped with internal real time clock:
    - Wall clock
    - Alarm clock
    - Computer
    - Mobile phone
    - Dishwasher
    - Washing machine
    - Coffee machine
    - Etc.

  The time synchronization is an important issue as many processes are controlled based on it.
- Alarm Interface – Interface to the alarm system. It is used to control and monitor it.

5 PROTOCOL

Protocol constitute basis for communication between distributed elements of the system. Chapter specifies details of the native protocol which is used by In-Home Energy Management System.

5.1 Definition of the Protocol

The protocol creates a common means for communication between entities. It defines rules for message exchange as well as message formats itself.

5.2 Objectives for the Protocol

Objectives of the protocol are:

- The protocol is not limited to, but is meant to be used for communication between devices in the system like the gateway and human machine interfaces.
- Extensible and versatile. The protocol is designed in the way that new functionalities could be easily added to existing ones with backward compatibility supported.
- Provide querying functionality, for example, to get a list of devices which meet the specified criteria.

5.3 Basic Protocol Overview

The protocol uses XML in order to fulfill the main objectives like extensibility and versatility. The protocol can be split vertically into two layers. The top one which creates a common abstract protocol layer and the bottom one translates between the common abstract layer and the specific underlying third party protocols and technologies. The protocol utilizes concepts introduced by the model and provides the means of communication with devices which implement it.

The following messages are provided by the protocol:

- Get Single Attribute
- Get Single Attribute Response
- Set Single Attribute
- Set Single Attribute Response
- Get Multiple Attributes
- Get Multiple Attributes Response
- Set Multiple Attributes
- Set Multiple Attributes Response
- Call Function
- Call Function Response
- Get Call Function Response
- Execute Query
- Execute Query Response
- Secure Message
- Error Response

5.3.1 Get Single Attribute

The Get Single Attribute is the message used to read the attribute of a single object.

The message format is:

```
<GetSingleAttribute
path="objID:intfID:intfVer:intfRev:attrID"/>
```

5.3.2 Get Single Attribute Response

The Get Single Attribute Response message is sent as a response to Get Single Attribute one.

The message format is:

```
<GetSingleAttributeResponse
path="objID:intfID:intfVer:intfRev:attrID">
   <!-- Exact content depends from attribute's data type -->
   <Integer>23</Integer>
</GetSingleAttributeResponse>
```

5.3.3 Set Single Attribute

The Set Single Attribute is the message used to set the attribute of a single object.

The message format is:

```
<SetSingleAttribute path="objID:intfID:intfVer:intfRev:attrID">
   <!-- Exact content depends from attribute's data type -->
   <Integer>23</Integer>
</SetSingleAttribute>
```

5.3.4 Set Single Attribute Response

The Set Single Attribute Response message is sent as a response to Set Single Attribute one.

The message format is:

```
<SetSingleAttributeResponse
path="objID:intfID:intfVer:intfRev:attrID">
   <!-- Exact content depends from attribute's data type -->
   <!-- Returns attribute's actual value for comparison  -->
   <Integer>23</Integer>
</SetSingleAttributeResponse>
```

5.3.5 Get Multiple Attributes

The Get Multiple Attributes is the message used to read multiple attributes which are not limited to particular object.

The message format is:

```
<GetMultipleAttributes>
   <Attribute path=”objID1:intfID1:intfVer1:intfRev1:attrID1”/>
   <Attribute path=”objID2:intfID2:intfVer2:intfRev2:attrID2”/>
   ...
   <Attribute path=”objIDn:intfIDn:intfVern:intfRevn:attrIDn”/>
</GetMultipleAttributes>
```

5.3.6 Get Multiple Attributes Response

The Get Multiple Attributes Response message is sent as a response to Get Multiple Attributes one.

The message format is:

```
<GetMultipleAttributesResponse>
   <Attribute path=”objID1:intfID1:intfVer1:intfRev1:attrID1”>
      <!-- Exact content depends from attribute’s data type -->
      <Structure>
         <Integer>12</Integer>
         <Boolean>True</Boolean>
      </Structure>
   </Attribute>
   <Attribute path=”objID2:intfID2:intfVer2:intfRev2:attrID2”>
      <Double>25.123</Double>
   </Attribute>
   <!-- Partial error response -->
   <ErrorResponse path=”objID3:intfID3:intfVer3:intfRev3:attrID3”>
      <ErrorCode>55</ErrorCode>
      <ErrorMessage>Error textual description</ErrorMessage>
   </ErrorResponse>
   ...
```

```
   <Attribute path="objIDn:intfIDn:intfVern:intfRevn:attrIDn">
      <Byte>24</Byte>
   </Attribute>
</GetMultipleAttributesResponse>
```

5.3.7 Set Multiple Attributes

The Set Multiple Attributes is the message used to set multiple attributes which are not limited to particular object.

The message format is:

```
<SetMultipleAttributes>
   <Attribute path="objID1:intfID1:intfVer1:intfRev1:attrID1">
      <!-- Exact content depends from attribute's data type -->
      <Structure>
         <Integer>12</Integer>
         <Boolean>True</Boolean>
      </Structure>
   </Attribute>
   <Attribute path="objID2:intfID2:intfVer2:intfRev2:attrID2">
      <Double>25.123</Double>
   </Attribute>
   ...
   <Attribute path="objIDn:intfIDn:intfVern:intfRevn:attrIDn">
      <Byte>24</Byte>
   </Attribute>
</SetMultipleAttributes>
```

5.3.8 Set Multiple Attributes Response

The Set Multiple Attributes Response message is sent as a response to Set Multiple Attributes one.

The message format is:

```
<SetMultipleAttributesResponse>
   <Attribute path="objID1:intfID1:intfVer1:intfRev1:attrID1">
```

```
    <!-- Exact content depends from attribute's data type -->
    <Structure>
      <Integer>12</Integer>
      <Boolean>True</Boolean>
    </Structure>
  </Attribute>
  <Attribute path="objID2:intfID2:intfVer2:intfRev2:attrID2">
    <Double>25.123</Double>
  </Attribute>
  <!-- Partial error response -->
  <ErrorResponse path="objID3:intfID3:intfVer3:intfRev3:attrID3">
    <ErrorCode>55</ErrorCode>
    <ErrorMessage>Error textual description</ErrorMessage>
  </ErrorResponse>
  ...
  <Attribute path="objIDn:intfIDn:intfVern:intfRevn:attrIDn">
    <Byte>24</Byte>
  </Attribute>
</SetMultipleAttributesResponse>
```

5.3.9 Call Function

The Call Function is the message used to call the function of a single object.

The message format is:

```
<CallFunction path="objID:intfID:intfVer:intfRev:attrID">
  <Argument number="1">
    <!-- Exact content depends from attribute's data type -->
    <Structure>
      <Integer>12</Integer>
      <Boolean>True</Boolean>
    </Structure>
  </Argument>
  <Argument number="2">
    <Double>25.123</Double>
  </Argument>
  ...
```

```
   <Argument number="n">
      <Byte>24</Byte>
   </Argument>
</CallFunction>
```

5.3.10 Call Function Response

The Call Function message is sent as a response to Call Function one.

The message format is:

```
<CallFunctionResponse path="objID:intfID:intfVer:intfRev:attrID">
   <CallState>Success | Executing | Error | Ready</CallState>
   <!-- If CallState is Success then return value follows. -->
   <ReturnValue>
      <Integer>78</Integer>
   </ReturnValue>
</CallFunctionResponse>
```

5.3.11 Get Call Function Response

The Get Call Function Response is the message used to explicitly request Call Function Response one.

The message format is:

```
<GetCallFunctionResponse
path="objID:intfID:intfVer:intfRev:attrID"/>
```

5.3.12 Execute Query

The Execute Query is the message used to submit and execute query.

The message format is:

```
<ExecuteQuery>
   <Query>SET ATTRIBUTE 1:1:1:1:1 TO 1
```

```
          WHEN ATTRIBUTE 1:1:1:1:1 == 0
          RETURN BOOLEAN = ATTRIBUTE 1:1:1:1:1 == 1
   </Query>
</ExecuteQuery>
```

5.3.13 Execute Query Response

The Execute Query Response message is sent as a response to Execute Query one. Its content depends from executed query.

The message format is:

```
<ExecuteQueryResponse>
   <Boolean>TRUE</Boolean>
</ExecuteQueryResponse>
```

5.3.14 Secure Message

The Secure Message is used to encapsulate other messages and to provide verification of integrity and authentication. A digital signature is created by first hashing message with SHA-1 and then encrypting it using a private key. When received, the digital signature is decrypted using the public key paired with the private key used for encryption. Then SHA-1 hash is calculated separately for payload message and compared with the decrypted one. Integrity and authentication is verified if both hashes are the same. Additionally, if confidentiality of the payload message is required, it can be also provided by encrypting it with the same private key as hash. It is an option because asymmetric encryption requires much computing power to be performed and it makes the private key more vulnerable. Encrypting only the hash instead of the whole message improves performance. The alternative is a hybrid cryptosystem which first sends a session key encrypted in a payload using the public key and then it uses it for much faster symmetric algorithm to encrypt and decrypt the payload. The digital signature always uses the public-key cryptosystem.

The message format is:

```
<SecureMessage>
   <Payload encrypted=”false” keyType=”public-key | session-key”>
      <GetSingleAttribute
      path=”objID:intfID:intfVer:intfRev:attrID”/>
   </Payload>
   <Origin>Utility Co.</Origin>
   <DigitalSignature>9A3C129B…</DigitalSignature>
</SecureMessage>
```

5.3.15 Error Response

The Error Response message is sent to indicate and to describe an error situation occurred due to a prior request.

The message format is:

```
<ErrorResponse path=”objID:intfID:intfVer:intfRev:attrID”>
   <ErrorCode>55</ErrorCode>
   <ErrorMessage>Textual description of the error</ErrorMessage>
</ErrorResponse>
```

5.4 Concurrency Control

Concurrency control ensures the correctness of operation result when more than one entity can access the shared data simultaneously. Locking is one of the major methods for concurrency control. The implementation of locking does not require any special messages to be defined. It is implemented based on the interface with an attribute working as a lock. Concurrency control could be done for:

- Entire device
- Entire object
- Interface implementation
- Single attribute or function
- Group of attributes and/or functions

5.5 Transactions

Transaction means providing indivisible operations. Everything within the transaction fails or succeeds as a unit. The implementation of the transaction does not require any special messages to be defined. It is implemented based on the interface with attributes providing support for the transaction.

5.6 Access Control

The access to the attributes and functions can be limited depending on the user's role. It can be done for:

- Entire device
- Entire object
- Interface implementation
- Single attribute or function
- Group of attributes and/or functions

5.7 Query Language

The query language is used with the ExecuteQuery message. It provides powerful extension to the standard functionalities provided by other messages. It allows to:

- Improve system performance and throughput
- Reduce network usage
- Enhance functionality

With the query language the following tasks could be accomplished:

- Return the computed attribute from an expression that can use ordinary attributes, functions, mathematic operators and constants.
- Return of custom message
- Conditional set operations

The syntax and semantics of the query language are not part of this thesis and they would need to be specified. The query presented below is provided as an example.

```
SET ATTRIBUTE 1:1:1:1:1 TO 1
WHEN ATTRIBUTE 1:1:1:1:1 == 0
RETURN BOOLEAN = ATTRIBUTE 1:1:1:1:1 == 1
```

6 PRACTICAL PART

The main applications developed as practical part of the thesis are depicted in Figure 24 below. Each of them uses different technologies. Out of all applications, the human machine interface is the biggest one. However, every single one had its own challenges which helped to improve technical skills. More detailed information can be found in the chapters which follow.

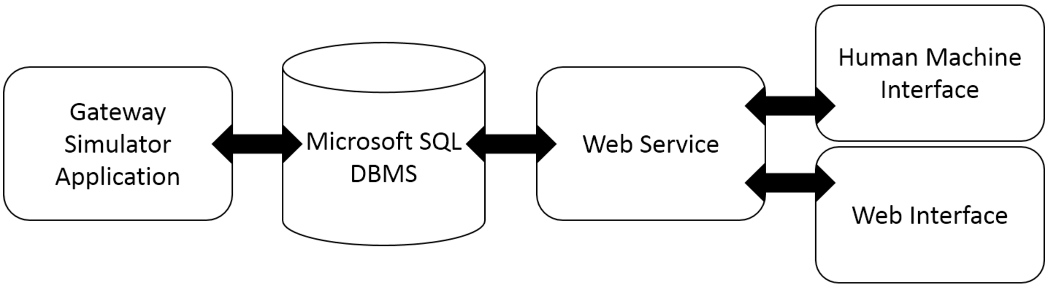

Figure 24. Components of the thesis's practical part implementation.

6.1 Human Machine Interface

The Human Machine Interface was developed to serve as a main interface to the system. It is intended to run on a touch screen device. Besides implementing the functional requirements it has also a number of non-functional ones, such as:

- Intuitive
- Suitable for learning
- Comprehensible
- Graphically appealing
- Responsive

Key technologies used are:

- Qt C++
- QML

The Human Machine Interface communicates with the Web Service, which works as a single access point to the system, in order to perform various actions on it. Additional information about human machine interface can be found in Chapter 3.

6.1.1 Introduction to Qt C++

Qt is an application framework which uses standard C++ for development of applications with the Graphical User Interface (GUI). However, it is not only meant to be a widget toolkit but it provides also a vast variety of libraries which are useful for general application development. One of the most important characteristics of the Qt is it being a cross platform. It means that the same code without changes can be built for different ones.

The Qt applications can run on following platforms/2/:

- Windows Desktop
- Windows CE and Windows Mobile
- Linux/X11
- Embedded Linux
- Mac OS X
- Symbian phones
- Nokia N9 smartphone

The Qt libraries provide a vast variety of operations. Therefore, they are divided into modules. This approach lets the developer include only the ones which are required by the application and the ones that reduce the footprint of the framework. It is especially important for embedded devices where resources are usually much more limited than on desktop platforms. Only the subset of the Qt libraries which are really used by application needs to be present on the machine which is running the Qt application. The list of the Qt essential modules and the Qt add-ons are listed in tables below.

Table 1. Qt essentials modules /3/

Module	Description
Qt Core	Core non-graphical classes used by other modules.
Qt GUI	Base classes for graphical user interface (GUI) components. Includes OpenGL.

Qt Multimedia	Classes for audio, video, radio and camera functionality.
Qt Network	Classes to make network programming easier and more portable.
Qt QML	Classes for QML and JavaScript languages.
Qt Quick	A declarative framework for building highly dynamic applications with custom user interfaces.
Qt SQL	Classes for database integration using SQL.
Qt Test	Classes for unit testing Qt applications and libraries.
Qt WebKit	Classes for a WebKit2 based implementation and a new QML API. See also Qt WebKit Widgets in the add-on modules.
Qt WebKit Widgets	WebKit1 and QWidget-based classes from Qt 4.
Qt Widgets	Classes to extend Qt GUI with C++ widgets.

Table 2. Qt add-ons /3/

Module	Development Platforms	Description
Active Qt	Windows	Classes for applications which use ActiveX and COM
Qt Concurrent		Classes for writing multi-threaded programs without using low-level threading primitives.
Qt D-Bus	Unix	Classes for inter-process communication over the D-Bus protocol.
Qt Graphical Effects	All	Graphical effects for use with Qt Quick 2.
Qt Image Formats	All	Plugins for additional image formats: TIFF, MNG, TGA, WBMP.
Qt OpenGL		OpenGL support classes. Note: Provided to ease porting from Qt 4.x. Please use the QOpenGL classes in QtGui for new code.
Qt Print Support	All	Classes to make printing easier and more portable.
Qt Declarative	All	Qt Declarative is provided for Qt 4 compatibility. The documentation is available through the Qt 4.8 Qt Quick documentation.
Qt Script	All	Classes for making Qt applications scriptable. Provided for Qt 4.x compatibility. Please use the QJS* classes in the QtQml module for new code.
Qt Script Tools	All	Additional components for applications that use Qt Script.
Qt SVG	All	Classes for displaying the contents of SVG files.
Qt XML		C++ implementations of SAX and DOM. Note: Deprecated, please use QXmlStreamReader and QXmlStreamWriter for new functionality.
Qt XML Patterns		Support for XPath, XQuery, XSLT and XML schema validation.

6.1.2 Introduction to QML

QML is a declarative language which is used for both design and development. It is meant to ease the learning curve for non-programmers and to improve cooperation between the designer and developer. QML simplifies and accelerates the development of User Interface (UI), both appearance and behavior. For simple solutions, the application can be written entirely in QML. However, it does not suit more complex projects in which case QML and Qt C++ are used together. More advanced UI functionalities and business logic are written in Qt C++.

The following is a simple start up QML file created by the Qt Creator:

```
import QtQuick 2.0

Rectangle {
    width: 360
    height: 360

    Text {
        anchors.centerIn: parent
        text: "Hello World"
    }

    MouseArea {
        anchors.fill: parent
        onClicked: {
            Qt.quit();
        }
    }
}
```

6.1.3 Interconnection between Qt C++ and QML

Qt C++ and QML can communicate through the signal / slot paradigm. It is possible to extend QML with additions written in Qt C++ as well as it is possible to work with QML components from the Qt C++ level. QML makes the development of smooth UI easier and faster but is not as good for advanced business logic as Qt C++. /4/, /5/

A few possible integration approaches are listed below:

- Using signals and slots
- Running and controlling QML from within Qt C++ code
- Embedding Qt C++ object into QML
- Extending QML with implementation of the new QML component in Qt C++

6.1.4 Licensing

While developing with Qt, a few licensing options are available. At least difference between two major ones, which are listed below, should be understood. /6/ /7/, /8/

- GNU Lesser General Public License (LGPL)
- Commercial licensing (Qt Commercial)

Both of them allow for proprietary closed code development. The application which is linked against Qt libraries does not have to be an open source. However, in case of LGPL any changes to the Qt library itself have to be distributed along with the application. Licensing should be chosen wisely at the beginning because there is no possibility of switching from LGPL to Commercial license.

The most characteristic difference between LGPL and GPL is linking. LGPL allows to link close code against LGPL licensed libraries. GPL license do not allow for it. Meaning, any application using GPL licensed libraries have to be an open source.

6.2 Web Service

The Web service exposes the functionalities of the gateway to the outside world over the network. It provides interoperability between different entities. In the case of practical implementation it provides means of communication between the gateway and both of the human machine interfaces implemented using native protocol messages defined within the thesis. The key technology used was SOAP.

6.2.1 Introduction to SOAP

SOAP (Simple Object Access Protocol) is a communication protocol which is mainly used by Web Services for information exchange. SOAP can use any transportation protocol but most commonly it uses HTTP (Hypertext Transfer Protocol) or SMTP (Simple Mail Transfer Protocol). /9/

Advantages of SOAP are:

- Based on XML and thanks to it is highly extensible
- Uses HTTP/HTTPS for communication. It is widely supported and mostly not blocked by firewalls. Using other protocols may require additional firewall configuration and usually it is not needed with HTTP/HTTPS.
- Well known protocol and widely supported. It is recommended by W3C.

Disadvantages of SOAP are:

- Bandwidth – data is not optimally packed; therefore more bandwidth is required to send packets over the network.
- CPU and memory – the parsing of XML message requires more CPU and memory than it would be required in case of hardcoded protocol.

6.3 Gateway Simulator

The gateway simulator application was implemented in order to facilitate the development and testing of the system. It communicates with the web service through the database where it manipulates entries representing the model defined. The gateway simulator allows for the interfaces and objects of the model to be added to the database. It also allows editing the attribute contents of the model.

The key technologies used are:

- .NET Framework
- Model-View-ViewModel (MVVM) Pattern

6.3.1 Introduction to .NET Framework

The .NET Framework /27/ is a software framework that facilitates the development and execution of applications targeted mostly for the Microsoft platforms. Two main components of the framework can be identified, namely:

- Common Language Runtime (CLR) – a virtual machine component which manages .NET application during its execution and provides core functionalities, such as memory management, thread management, security, exception handling, etc.
- Base Class Library – the main framework library that provides a vast number of reusable types which in turn help to accomplish tasks, such as database connectivity, IO operations, XML parsing, data collections management, etc.

Applications which are executed by CLR are called to be managed. Ones which are executed without CLR are called to be unmanaged.

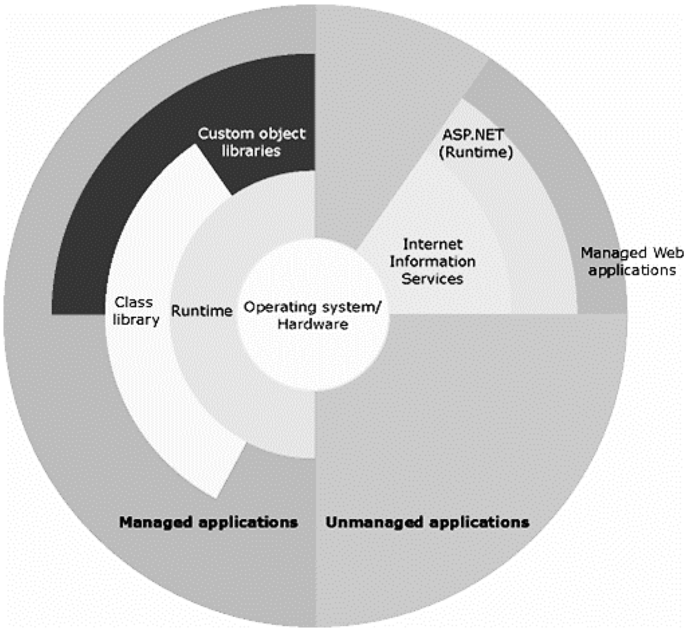

Figure 25. Main features of the .NET Framework. /27/

6.3.2 Introduction to Model-View-ViewModel Pattern

Model-View-ViewModel is a software design pattern targeted at event-driven extensively graphical UI based applications. The main goal of the pattern is to provide clear separation between the user interface and business logic. MVVM is a specialization of the more general Presentation Model pattern in order to fully utilize Windows Presentation Foundation (WPF) features like data bindings. /28/, /29/

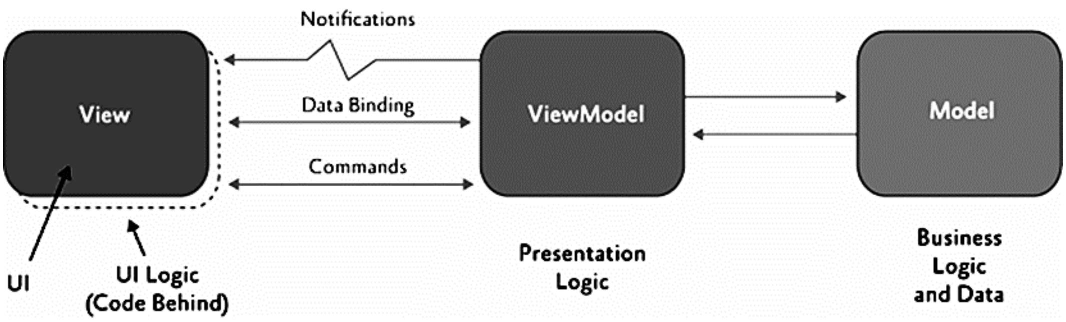

Figure 26. Relations between View, Model and ViewModel. /29/

6.4 Database

The database stores the model representation and serves as an interface between the Gateway simulator and Web Service. The main task related to the database implementation was to create relational model mapping of the abstract model defined within the thesis.

The key technology used was the Microsoft SQL DBMS.

6.5 Web Interface

A simple Web Interface was developed in order to present extensibility of the system by an example of how easily other user interfaces can be added. It is accomplished thanks to the translation of the native protocol to standard technologies, such as Web Services and SOAP.

The key technology used was Model-View-Controller (MVC) Pattern.

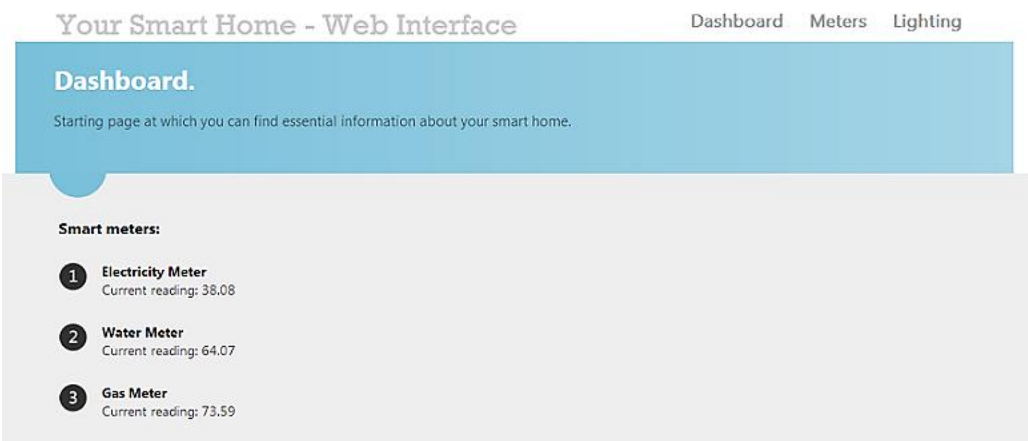

Figure 27. Screenshot of very simple web interface.

7 EXISTING STANDARDS AND TECHNOLOGIES

Chapter provides an overview of technologies and standards which are already adapted and with which In-Home Energy Management System may need to communicate in order to accomplish its tasks.

7.1 Standards Related to Electric Grid and Smart Metering

Following chapters describe common standards which are in use for substation automation and smart metering.

7.1.1 IEC 61850

The IEC 61850 is a global standard for design of electrical substation automation. It is divided into several parts /10/:

- IEC 61850-1 – Communication networks and systems in substations –
  Part 1: Introduction and overview
- IEC 61850-2 – Communication networks and systems in substations –
  Part 2: Glossary
- IEC 61850-3 – Communication networks and systems in substations –
  Part 3: General requirements
- IEC 61850-4 – Communication networks and systems in substations –
  Part 4: System and project management
- IEC 61850-5 – Communication networks and systems in substations –
  Part 5: Communication requirements for functions and device models
- IEC 61850-6 – Communication networks and systems for power utility automation –
  Part 6: Configuration description language for communication in electrical substations related to IEDs
- IEC 61850-7 – Communication networks and systems in substations - Basic communication structure for substation and feeder equipment:
    - IEC 61850-7-1: Principles and models
    - IEC 61850-7-2: Abstract communication service interface (ACSI)

  - IEC 61850-7-3: Common Data Classes
  - IEC 61850-7-4: Compatible logical node classes and data classes
- IEC 61850-8 – Communication networks and systems in substations - Specific Communication Service Mapping (SCSM):
  - IEC 61850-8-1: Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3
- IEC 61850-9 – Communication networks and systems in substations - Specific Communication Service Mapping (SCSM):
  - IEC 61850-9-1: Sampled values over serial unidirectional multidrop point to point link
  - IEC 61850-9-2: Sampled values over ISO/IEC 8802-3
- IEC 61850-10 – Communication networks and systems in substations - Part 10: Conformance testing

Its goal is to provide interoperability between intelligent electronic devices (IEDs) from different manufactures which are used for protection and substation automation /11/. An additional advantage of the standard is Standard Configuration description Language (SCL) which ensures the integrity of the data entries and which can be used directly by engineering and design tools to simplify process of design and commissioning. At the same time SCL serves as the documentation of the Substation Automation (SA) system.

7.1.2 Generic Object Oriented Substation Events (GOOSE)

The advantages of the GOOSE come in big part are due to its using Ethernet. Using Ethernet helps to simplify design, commissioning and maintenance processes. Prior to GOOSE many signals between devices were hard wired. Therefore, by reducing the amount of hard wiring required we have following benefits: /12/

- Reduction in material and direct labor cost
- Simplified design diagrams which could be easier reused
- Simplified connection test

7.1.3 Open Smart Grid Protocol (OSGP)

The Open Smart Grid Protocol (OSGP) is a set of specification related to the smart grid applications and published by European Telecommunications Standards Institute (ETSI). It utilizes other open ISO/IEC and IEEE standards, for example ISO/IEC 14908 which is used for control networking in smart grid applications and ETSI TS 103 908 PowerLine Telecommunication (PLT) as a physical layer. OSGP focuses on efficient and reliable networking and management of smart grid devices, such as smart meters, gateways, load control modules and other smart grid devices. Currently there are over 3.5 million OSGP capable devices deployed worldwide. /13/

7.1.4 IEC 61107

IEC 61107 (former IEC 1107) /14/ /15/ is a simple and widely used communication protocol for smart meters. It is published by the International Electrotechnical Commission (IEC) and now suppressed with IEC 62056 which also in IEC 62056-21:2002 specify direct local data exchange mainly over an optical port. IEC 61107 sends simple ASCII data either over modulated light, using inexpensive LEDs and photodiodes, or a pair of wires and signal modulated according to the EIA-485 standard.

7.1.5 IEC 62056 (DLMS/COSEM)

IEC 62056 is a set of standards for use in Automatic Meter Reading published by the International Electrotechnical Commission (IEC). It is based on DLMS / COSEM specification maintained by the DLMS User Association. /16/, /17/, /18/

COSEM or Companion Specification for Energy Metering defines an object-oriented interface model of the energy metering devices. The COSEM model represents a physical device as a set of logical ones which contain interface objects and those in turn contain a set of attributes and methods. The access level to the interface objects of logical devices is constrained based on association objects.

Following security levels are defined:

- No security – public access without limitation
- Low level security – password protected
- High level security – mutual identification through challenges

DLMS or Device Language Message Specification defines a messaging layer used to turn information from the model to a series of bytes which can be transported using a number of lower level protocols listed below: /19/, /20/

- TCP-UDP/IP
    - Ethernet
    - GPRS
- G3 OFDM PLC
- PRIME OFDM PLC
- S-FSK PLC
- B-PSK PLC
- Y PLC
- Z PLC
- EURIDIS
- M-Bus
- HDLC

OBIS or Object Identification System is used by DLMS/COSEM to uniquely identify all data provided by the meter. Thanks to it, interoperability of data exchange with meters from different vendors is reached. Standard OBIS codes for following meters are defined in so called Blue book published by DLMS UA:

- Electricity
- Cold and Hot Water
- Heating and Cooling
- Gas

7.1.6 M-Bus (EN 13757-2, EN 13757-3) and Wireless M-Bus (EN 13757-4)

The Meter-Bus (M-Bus) is a European standard for distance reading of consumption meters as electricity, heat and gas ones. EN 13757-2 specifies the physical and link layer which utilizes two wires for cost effective communication. EN 13757-3 specifies the application layer protocol which actually is based on another EN 1434-3 standard for data exchange with heat meters. M-Bus has low power consumption and it allows connecting many meters on a single cable. A wireless alternative is raised and it is standardized by EN 13757-4.

7.2 Related to Home Automation

Chapters below describe communication standards which are popular in home automation solutions.

7.2.1 ZigBee

ZigBee is a specification of a low-power wireless communication protocol. Communication nodes often use other intermediate nodes in order to reach more distance ones. ZigBee is suitable for solutions which require low power consumption, long battery life and secure connection. ZigBee baud rate is rather low
(250 kbit/s). Therefore, it is mostly used for solutions that by its characteristics do not require high volume of data to be transferred. Low power consumption comes also from its fast startup time. The ZigBee node can go from sleep to active in under 30 milliseconds. It allows for the communication module to be in active mode only when needed and sleep all other time.

Typical ZigBee application areas include /21/:

- Building Automation
- Energy Management and Efficiency
- Consumer Electronics
- PC & Peripherals

- Home Control
- Telecom Services
- Industrial Control
- Personal Health Care

In order to facilitate the creation of interoperable products by different vendors the ZigBee Alliance publishes profiles. One of them is ZigBee Smart Energy Profile, which includes following features /22/:

- Metering Support
    - Real time and historical readings
    - Multiple units of measurement
    - Measuring of both consumption and production
    - Measurement of multiple properties
    - Anti-tampering
- Demand Response & Load Control
    - Targeting of individual or group of devices
    - Support for customer override
- Pricing
    - Block tariff
    - Prepayment
    - Multiple currencies
    - Support for price ratios/price tiers
- Text Message
- Sample Devices
- Security
    - Allows for consumer only, utility only and shared networks

7.2.2 Z-Wave

Z-Wave is wireless communication protocol. It has similarities to the ZigBee, for example low-power, low transmit duty cycle and utilization of the wireless mesh networking. The Z-Wave network can have up to 232 nodes. It operates in less crowded frequencies below gigahertz as compared to more crowded 2.4 GHz which is used for example by Wi-Fi, Bluetooth and ZigBee. A lower frequency means a better propagation of the signal through obstacles as walls but it also means a lower bandwidth of up to 100 kbit/s.

7.2.3 BACnet

BACnet is a communication protocol for Building Automation and Control Networks. This simple wired protocol is used mostly in the area of heating, ventilation and air conditioning (HVAC).

7.2.4 X10

X10 is a communication protocol for building automation which uses power line for signaling and control. It was developed in 1975 and thanks to its long time on the market it is still widely available although newer options with higher bandwidth are available. It is very simple and it has very low bandwidth. A single bit is sent at zero crossing of the alternating current waveform. Taking into account 50 Hz grid, the sending of messages twice for elimination of false signaling, retransmissions and line control we are getting bandwidth of 20 bit/s. Communication can be one or two way. Simpler devices can only listen for commands. A high carrier frequency of 120 kHz onto which digital data is encoded do not allow for signal to pass through a transformer or across phases of the multiphase system. If it is required, repeaters need to be used. It is also possible to block signal propagation by use of the inductive filters. /23/

The protocol is very simple. It consists of a four bit house code which is followed by one or more four bit unit codes and it is ended with a four bit command.

Basic implementation of the X10 protocol allows for simple operations as control of lighting and turning device on and off.

7.2.5 INSTEON

INSTEON is a proprietary communication protocol for home automation that utilizes power line, radio frequency or both for data transfer between devices. It is designed as a successor of the aging X10 protocol. INSTEON devices which communicate over the power line can be compatible with X10 ones. Every INSTEON device works also as a repeater, meaning they listen and forward all messages thus increasing the network range. Messages are repeated at a precise time. Therefore, collisions amplify signal and do not attenuate it. /24/

INSTEON defines two types of fixed length messages. The standard one is 10 Bytes in length and Extended Message is 24 Bytes in length. An extended Message allows for 14 Bytes of the user data.

Table 3. INSTEON standard message. /24/

INSTEON Standard Message – 10 Bytes				
3 Bytes	3 Bytes	1 Byte	2 Bytes	1 Byte
From Address	To Address	Flags	Command 1, 2	CRC

Table 4. INSTEON extended message. /24/

INSTEON Extended Message – 24 Bytes					
3 Bytes	3 Bytes	1 Byte	2 Bytes	14 Bytes	1 Byte
From Address	To Address	Flags	Command 1, 2	User Data	CRC

Usable data rates for the power line vary based on the number of hops, acknowledgment requirement and message type selected (standard or extended message type). It is somewhere in the range between 130 bit/s and 1698 bit/s. In case of the radio frequency message is not split but sent as whole and data rate reaches 38400 bit/s.

7.2.6 KNX

KNX is a standard for communication in building automation which is recognized by the following standardization authorities/25/:

- European Standard (CENELEC EN 50090 and CEN EN 13321-1).
- International Standard (ISO/IEC 14543-3).
- Chinese Standard (GB/Z 20965).
- US Standard (ANSI/ASHRAE 135)

KNX Association has over 100 member companies with reach a portfolio of more than 7000 products from wide range of application domains, such as:

- Lighting and blind control
- Heating, ventilation and air conditioning (HVAC)
- Water control
- Monitoring
- Alarming
- Energy Management
- Household appliances
- Metering
- Audio and video

KNX protocol is also versatile with communication media used. The standard specifies several ones, among them /26/

- Twisted Pair
- Power line
- Radio frequency
- Ethernet

8 CONCLUSIONS

The thesis fulfilled its objectives. It provides a foundation for in-home energy management system and for future developments. The model, protocol and human machine interface were designed, developed and successfully tested. The system is seen to be the key element of the successful smart grid implementation. The importance of consumers can never be over-emphasized. Therefore, the whole process of design and development was user centered.

The scope of the thesis was very broad. Therefore, it was not aimed to go into details of every aspect. Guidelines were drawn but details should be specified as well. Some issues which are left for future development are listed below:

- Working even more with end users; assessing the validity of proposed approach on bigger scale and adjusting it according to lean software development.
- Security
- Protocol translations and its query language syntax and semantics

Energy production, distribution, consumption and metering have remained virtually unchanged for past decades; especially when compared to technological advance in other sectors, such as information and communications technology. However, it is now a recognized fact and changes are pushed by political actions as well as by environmentally aware consumers. It creates a lot of potential for new products and companies to emerge.

REFERENCES

/1/ Göran Koreneff, Maija Ruska, Juha Kiviluoma, Jari Shemeikka, Bettina Lemström, Raili Alanen & Tiina Koljonen. 2009. Future development trends in electricity demand. VTT. Accessed 15.04.2013. http://www.vtt.fi/inf/pdf/tiedotteet/2009/T2470.pdf.

/2/ Supported Platforms. Accessed 15.04.2013.

http://qt-project.org/wiki/Platform_Support.

/3/ All Modules. Accessed 15.04.2013.

https://qt-project.org/doc/qt-5.0/qtdoc/modules.html.

/4/ Property Binding. Accessed 15.04.2013.

http://qt-project.org/doc/qt-4.8/propertybinding.html.

/5/ Writing QML Extensions with C++. Accessed 15.04.2013.

http://qt-project.org/doc/qt-4.8/qml-extending-tutorial-index.html.

/6/ License Comparison. Accessed 16.04.2013.

http://qt.digia.com/Product/Licensing/License-Comparison/#.Ua-vp7VvCNg.

/7/ GNU Lesser General Public License. Accessed 18.04.2013.

http://www.gnu.org/licenses/lgpl.html.

/8/ GNU General Public License. Accessed 18.04.2013.

http://www.gnu.org/licenses/gpl.html.

/9/ Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, Anish Karmarkar, Yves Lafon. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). 2007. http://www.w3.org/TR/soap12-part1/.

/10/ Core IEC Standards. Accessed 22.04.2013.

http://www.iec.ch/smartgrid/standards/.

/11/ Peter Rietmann, Tetsuji Maeda, Petra Reinhardt. 2007. Applying IEC 61850 to Substation Automation Systems.

http://www05.abb.com/global/scot/scot221.nsf/veritydisplay/034f355dd2492538c12574f900509190/$file/GRIDTECH%202007%20Applying%20IEC%2061850%20to%20SAS%20ABB.pdf.

/12/ Hakala-Ranta, Rintamäki, Starck. 2009. Utilizing possibilities of IEC 61850 and GOOSE. http://www05.abb.com/global/scot/scot229.nsf/veritydisplay/3ff2dbcff3a10556c12575e60043b35f/$file/ABB_whitepaper_CIRED_2009_0741.pdf.

/13/ ETSI GS OSG 001 V1.1.1 (2012-01). Open Smart Grid Protocol. Accessed 01.05.2013. http://www.etsi.org/deliver/etsi_gs/OSG/001_099/001/01.01.01_60/gs_osg001v010101p.pdf.

/14/ IEC 1107 – What is it all about? Accessed 01.05.2013. http://www.abacuselectrics.com/iec1107.htm.

/15/ DLMS/COSEM Brings Interoperability. Accessed 01.05.2013. http://www.dlms.com/news/043dlmscosembringsinteroperability.html.

/16/ What is DLMS/COSEM. Accessed 01.05.2013. http://www.dlms.com/information/whatisdlmscosem/index.html.

/17/ Gyozo Kmethy. DLMS/COSEM – a smart standard for smart metering. 2008. http://www.dlms.com/downloads/dlmscosem2008asmartstandardforsmartmetering.pdf.

/18/ Gyozo Kmethy. An overview on AMR standardisation. 2005. http://www.dlms.com/downloads/amrstandardisationstockholmv3gk050414.pdf.

/19/ Transport. DLMS UA Images. Accessed 06.05.2013. http://dlms.com/DLMSimages/Transport.pdf.

/20/ Excerpt from DLMS/COSEM Architecture and Protocols. 2009. http://dlms.com/documents/Excerpt_GB7.pdf.

/21/ Bob Heile. Smart Metering: It’s Not Just for Meter Reading Anymore. 2010. http://docs.zigbee.org/zigbee-docs/dcn/10-6056.pdf.

/22/ ZigBee Smart Energy Features. Accessed 06.05.2013. http://www.zigbee.org/Standards/ZigBeeSmartEnergy/Features.aspx.

/23/ Transmission theory of X-10 signals. Accessed 06.05.2013. http://www.eurox10.com/Content/X10SignalTheory.htm.

/24/ INSTEON. Whitepaper: The Details. 2013. http://www.insteon.com/pdf/Insteondetails.pdf.

/25/ KNX The Standard. Accessed 06.05.2013. http://www.knxuk.org/index.php?option=com_content&view=article&id=6&Itemid=111.

/26/ Communication Media. Accessed 06.06.2013. http://www.knx.org/knx-standard/communication-media/.

/27/ Overview of the .NET Framework. Accessed 01.06.2013. http://msdn.microsoft.com/en-us/library/zw4w595w.aspx.

/28/ Josh Smith. WPF Apps with the Model-View-ViewModel Design Pattern. 2009. http://msdn.microsoft.com/en-us/magazine/dd419663.aspx.

/29/ Implementing the MVVM Pattern. Accessed 02.06.2013. http://msdn.microsoft.com/en-us/library/gg405484(v=pandp.40).aspx.