AN IMPLEMENTATION OF THE PLANNING PHASE OF TRIANA USING THE FLEXIBLE POWER APPLICATION INFRASTRUCTURE

by

STERGIOS DAGIOGLOU

A THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

SUSTAINABLE ENERGY TECHNOLOGY

Faculty of Engineering Technology
(Construerende Technische Wetenschappen - CTW)

UNIVERSITY OF TWENTE
The Netherlands

August 2014

to my family

Acknowledgments

This thesis was conducted in the period September 2013 - August 2014, partially at the University of Twente (Faculty of Electrical Engineering, Mathematics and Computer Science, Departments of Computer Science and Applied Mathematics) and partially during the internship conducted in the period 3 February -2 May 2014 at the facilities of TNO located in Groningen, the Netherlands and within the department of Business Information Services − BIS. I would like to thank my supervisors at the University of Twente, Professor Johann Hurink and Ph.D. candidate Hermen Toersche as well as my supervisors at TNO, Wilco Wijbrandi and Dennis Krukkert for their valuable advice and guidance while carrying out this assignment.

Abstract

Over the last decades, the impacts induced from human activities on the planet are a multiple of the impacts the human species induced since its existence. The technological advancements of the first and second Industrial Revolution, led to a rapid growth of world population and life expectancy and to a subsequent growth of world energy consumption. Human technology is still improving very fast and predictions show that world population and energy consumption will further increase over the coming decades. Although all these technological advancements improved the quality of life for a large part of the world population, they also had tremendous impacts on the environment that threaten the ability of future generations to meet their own needs. The greenhouse effect, deforestation, drought and potable water shortage, biodiversity loss, are only some of the problems that the future generation will inherit as a result of the human activity of the last centuries.

The energy used to produce the electricity consumed nowadays, is approximately one third of the energy consumed world-wide. But only less of the half of the energy used to produce electricity is finally converted to electricity, as a result of the losses that occur during the production of electricity [34]. After electricity is produced 5-10 % of this energy is lost during the transmission and distribution of electricity to the end-users [2]. The largest part of the electricity consumed today, is still produced by large power plants using fossil fuels like coal and oil and is delivered to the consumers using the transmission and distribution grids that result in the losses mentioned above. The efficiency of these power plants is also relatively low unless used for the parallel production of heat and electricity (Combined Heat and Power-CHP).

The aforementioned evolutions lead governments and other administrative institutions to the establishment of measures that protect the environment and improve the overall energy production chain. An example is the “20-20-20” target of the European Union that aim to 3 objectives by 2020: reducing the EU greenhouse emissions by 20% compared to 1990, produce 20% of the energy consumed using renewable sources and improve energy efficiency by 20% [28].

As far as electricity is concerned, the evolution of the above grid to a Smart Grid is considered a necessity in order to achieve goals like the ones mentioned. Although it is risky to give an explicit definition of the Smart Grid, we could generally argue that the Smart Grids are systems which use modern technologies to create more efficient energy systems compared to the present ones. A Smart Grid usually combines decentralized energy production (often from renewable energy sources), smart metering, information technology, energy storage devices, load and generation monitoring and control and other modern technologies to produce and distribute electrical and thermal energy in the most efficient way possible [12].

Some of the challenges the electric utilities face today are meeting the rising electricity demand without a parallel increase in greenhouse emissions, improving the efficiency with which energy is produced, transmitted and distributed and increasing the grid reliability so that an uninterrupted energy supply can be made possible to a larger extent. Regarding the latter one, the electricity supply is frequently threatened during the hours of peak electricity consumption. Utilities must always be prepared to meet the maximum electricity consumption expected. An interrupted or disturbed (regarding the voltage and frequency levels) electricity supply has major negative effects on industries whose operation depends on a stable electricity supply but also to commercial consumers as electricity is necessary to meet basic needs nowadays. Furthermore, meeting the large peak demands have a negative financial impact to electricity producers as they can use cheaper energy sources to cover the base and intermediate demand and have to use more expensive sources to cover the peak demand. One method to reduce the peak electricity demand is to shift demand from peak hours to hours with lower demand. By this the need for expensive peak production units is decreased as well as the cost of the electricity production. Till now the production

was driven by the demand of consumers. Demand Side Management (DSM) technology is one the Smart Grid aspects and aims to shift the energy consumption from hours of high consumption to hours of lower consumption. By doing so, the electricity production can in a certain extent reduce its dependence on the electricity demand. That would allow the energy producers to use cheap sources to cover a larger percentage of the electricity consumed and increase the grid reliability. DSM and the improvements in forecasting the electricity produced by renewable sources may also lead to a larger integration of electricity produced by renewable energy sources.

DSM usually includes an application which can communicate with the devices whose energy consumption is desired to be controlled. An important problem that holds up the implementation of DSM in large scale is that devices use different communication protocols to send messages or data to other devices or controllers making the communication between devices and energy applications difficult. The Flexible Power Application Infrastructure (FPAI) is a platform developed by the Netherlands Organisation for Applied Scientific Research (TNO) which aims to solve this problem. The purpose of FPAI is to create an intermediate platform that is able to connect to a variety of devices and also to support different DSM systems [24]. Furthermore if a user wants to substitute its current DSM system with another, nowadays the installation of a new device that contains the hardware and software of the new system would be necessary. With FPAI the user just needs to uninstall the previous system and install a new one.

One of the many DSM methodologies developed during the last years is Triana; a DSM technique developed at the University of Twente. The goal of the Triana methodology is given in [4] and it is "*to manage the energy profiles of individual devices in buildings to support the transition towards an energy supply chain which can provide all the required energy in a sustainable way*". Triana has three main stages: Forecasting, Planning and Real-time Control. The ultimate goal of Triana is to exploit the flexibility of devices in order to achieve a goal related to the energy consumption of these devices. A typical goal is to achieve an aggregated energy consumption profile for a number of devices that is as flat as possible. In the Forecasting step, Triana tries to predict the flexibility offered by devices for a specific time horizon. Flexibility denotes the range of controllability of every device; how much the consumption pattern of a device can be altered in order to achieve a certain goal for a number of devices. The flexibility of devices can be estimated by determining parameters that influence the operation of a device. For example a useful parameter for a domestic device that consumes electricity and produces heat, would be to predict the heat demand of the house (using information like weather data). The Planning phase takes into account the flexibility of a number of devices and determines the operation of devices for a planning horizon, in order to achieve a global goal. In Real-time Control a replanning process might need to take place if we find that the forecast which formed the base for the Planning did not lead to the desired goal. Hereby, the replanning process takes into consideration new forecasts which use more recent data related to the behavior of the devices.

The research question of the current thesis is to explore if it is possible to implement the Planning phase of Triana using the FPAI platform. It was considered useful for both the developers of FPAI and Triana to search if FPAI can indeed provide a platform on top of which Triana can be executed, as FPAI had previously been used by only one other DSM system, the Powermatcher. During the implementation, incompatibilities between FPAI and Triana had to be detected and by using specific tools provided by FPAI or by introducing additional methods the gap between FPAI and Triana had to be bridged.

In order to answer this research question, a software programme has been implemented that executes the Planning of Triana on top of FPAI. Using this software, a number of simulations was done to examine if the Planning results were the desired, thereby proving that the implementation of the Planning phase of Triana using FPAI is indeed possible.

This thesis contains material that was written during a literature research that was done as a preparation of the thesis as well as material from the internship that was conducted in TNO facilities at Groningen and includes the development of the software that executes the Planning phase of Triana using the FPAI.

The first chapter of the thesis describes the relation of the topic of the thesis with the master programme of Sustainable Energy Technology. The reader can find there information about how this topic is related to the aspects of sustainability. The second chapter describes in more detail the Triana methodology. The three steps of Triana are discussed in detail. It also includes information about the Powermatcher system and a brief description of other energy control methodologies. The largest part of this chapter was written within the literature research which was made as a preparation assignment for the thesis. The third chapter contains a description of FPAI where we focus mainly on the aspects of FPAI which are needed for the implementation of this assignment. The fourth chapter presents in detail how the implementation of the Planning phase of Triana using FPAI was done; how certain tools provided by FPAI were used, the incompatibilities between FPAI and Triana and how they were overcome and the specific implementation for the different device types within FPAI. The fifth chapter presents the results of the simulations that were run to see if the code developed produced the results that were expected, namely if certain goals related to the energy behavior for a number of devices were achieved using Triana. Finally in the sixth chapter we present our conclusions about the results of this thesis and recommendations for future work that can be done.

Acronyms

CHP Combined Heat and Power
COP Coefficient of Performance

DP Dynamic Programming
DSM Demand Side Management

ECN Energy research Center of the Netherlands

FAN Flexiblepower Alliance Network
FIT Feed In Tariff
FP Flexible Power
FPAI Flexible Power Application Infrastructure

ICT Information and Communication Technology
IDDP Iterative Distributed Dynamic Programming

MAS Multi-Agent System

PHEV Plug-in Hybrid Electric Vehicle

RAI Resource Abstraction Interface
RAL Resource Abstraction Layer

SDM Supply-Demand Matching
SoC State of Charge

TOU Time of Use

VPP Virtual Power Plant

WSN Wireless Sensor Networks

List of Figures

Contents

1 Introduction

Numerous events related to the energy production and consumption, have created the need to explore new ways with which energy is produced and supplied to the consumers. The increase in energy consumption is accompanied with a parallel increase in greenhouse gas emissions as the main energy sources the humanity exploited first were fossil fuels. Regarding the electricity production, fossil fuels are still the dominant source too. Environmental concerns and the rapid depletion of fossil sources led to the development of technologies that produce electricity from renewable sources. Electric utilities face a series of additional challenges apart from the substitution of fossil fuels with the intermittent renewable sources. One of those challenges is the reduction of peak consumption occurring during certain hours of the day. Peak consumption results in higher electricity cost as more expensive sources are used to cover this demand due to their fast responsiveness. Peak demand might affect the stability of the grid as the voltage and frequency levels might be disturbed. These problems can be partially solved using Demand Side Management (DSM); a technology which aims to the manipulation of the operation pattern of energy consuming devices so that the overall energy consumption is distributed more evenly during the day.

Over the last years, several DSM methods have been developed one of them being Triana; a DSM methodology developed by the University of Twente [4]. The implementation of DSM in large scale is among others hindered by the communication difficulties between the devices and the DSM applications as there is a large number of communication protocols that devices use to send messages and information to other systems. The Flexible Power Application Infrastructure (FPAI) has developed an intermediate platform between devices and DSM applications in order to solve that problem. FPAI has specific components responsible for receiving information related to the energy behavior of devices. This information is shaped in a form that can be used by energy control systems like DSM systems. For example Triana needs such information to schedule the operation of a fleet of devices according to a goal set, which could be a flat aggregated energy consumption for a specific time horizon. Triana achieves this goal through the execution of three steps: Forecasting, Planning and Real-time Control.

The goal of this thesis is to implement the Planning phase of Triana using FPAI. Triana normally uses information related to the energy consumption of devices from the Forecasting step, which predicts energy parameters that determine how much the consumption pattern of a device can be modified. As FPAI does not have a yet a proper tool that can make such forecasts, the input used to implement the Planning phase of Triana is taken from the so-called Control Space; information provided by a Resource (Device) Manager bound to every device. The Control Space also provides information that can be used by Triana to execute the Planning phase although in a form that is different from the information provided normally by the Forecasting phase of Triana. This and other problems faced during the implementation of the Planning phase of Triana using FPAI are discussed in this report.

In this first chapter we discuss the importance for new technologies that may lead to a more efficient energy supply chain frequently referred to as Smart Grid. One of the aspects of Smart Grid is also DSM. Furthermore, the contributions of these new technologies to the different sustainability dimensions are discussed.

1.1 The necessity for a Smart Grid

The world energy consumption is rapidly increasing the last 200 years [8] (figure 1) as a result of the increase in world population [29] (figure 2) due to the technological advancements since the era of Industrial Revolution. The largest part of world energy production is still based on fossil fuels, mainly oil, coal and natural gas.

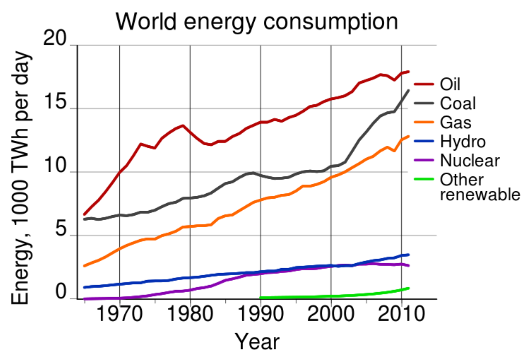


Figure 1: World energy consumption by year

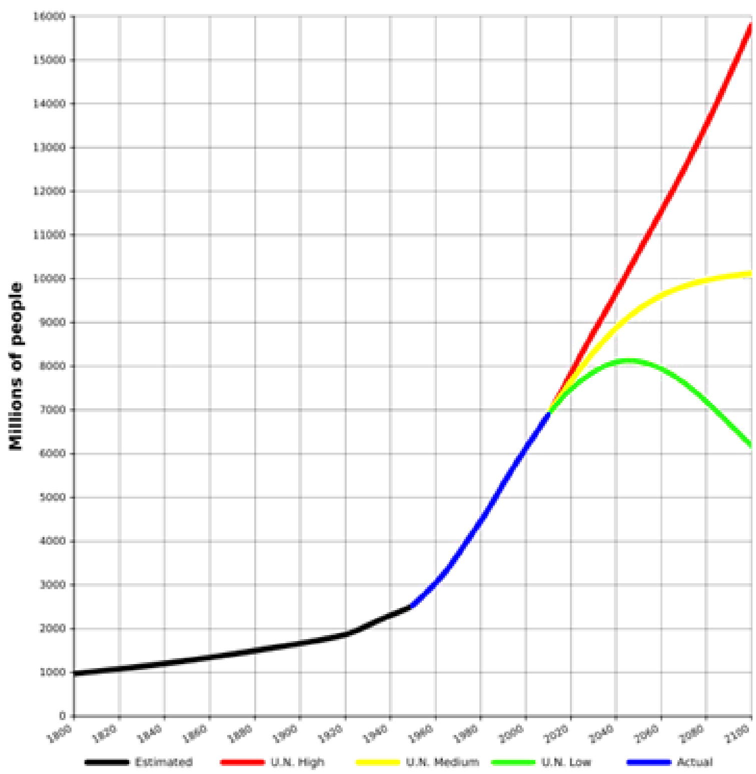
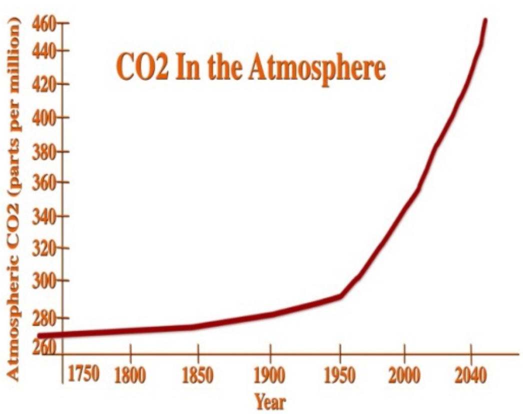


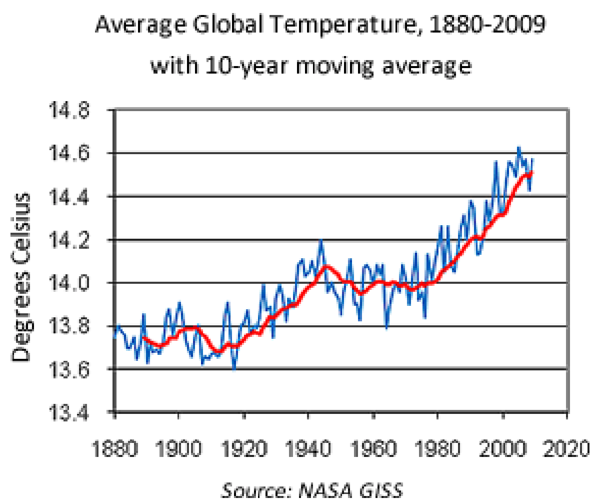
Figure 2: World population by year

At this point it is considered important to mention the inequity regarding the energy consumption per person in different parts of the world. The Ecological Footprint is a size that shows the amount of ecologically productive land and sea area required to cover the consumption needs of a given population and to assimilate associated waste. The unit that is used to express the Ecological Footprint is the global hectare (gha) [20][27]. By dividing the amount of biologically productive land and sea water of a region with its population, we find the biocapacity of a region per person [26]. As an example the Ecological Footprint of a citizen in the U.S.A. is 8.00 gha and the biocapacity of U.S.A. is 3.87 gha/person. The Ecological Footprint of a Citizen in Indonesia is 1.21 gha and the biocapacity of Indonesia is 1.35 gha/person [25]. The world average Ecological Footprint is 2.2 gha whereas the estimated earths biocapacity in 2008 was 1.79 gha/person showing that humanity nowadays consumes more than the earth can sustainably produce [20, 26].

A series of events are forcing the humanity to change the means with which energy is produced, transported and consumed. Environmental concerns, related to global temperature rise, as a result of greenhouse gases are perhaps the most important of these events. The increase of CO2 levels in the atmosphere from the industrial revolution up to present is in line with an increase of the average global temperature [1, 15] (figure 3) . Moreover, the political instabilities in oil producing countries over the last decades have changed the energy policies of many governments which want to secure the energy supply of their countries. Hence more and more countries are exploiting their energy resources and try to become, in the largest possible grade, independent on energy imports. Renewable energy sources have gained a larger share in the market due to these two reasons.



(a) CO_2 in atmosphere by year



(b) Average global temperature by year

Figure 3: CO_2 in atmosphere by year and average global temperature by year [1], [15]

Furthermore, the present electricity production consists of large power plants whose fuel is primarily coal and natural gas. However, the construction of such new plants demands huge investments which are hindered by the current economic context world-wide. Decentralized, small-scale production from sustainable resources is promoted with subsidies or other regulations. An increasing number of consumers has started producing energy at domestic level, making the role of the so called “prosumers” an important variable in the energy systems of the future.

In [18] the evolution in electricity grids is presented and it is sketched how this evolution led to the need for energy control methods. In the same paper the importance of distributed generation is highlighted and it is stated that it will play an increasingly important role to the energy production. Distributed generation will be integrated to the current electricity networks following the next three stages.

1. *Accommodation* → Distributed generation sources run free and control is centralized only.
2. *Decentralization* → The share of distributed generation increases thanks to the use of ICT systems; again the control is centralized.
3. *Dispersal* → In this last stage, local, low voltage networks are formed, where demand is supplied by local resources or by small amounts of energy provided by external (with respect to the local network) energy producers of the grid. There is not a central control of the grid anymore rather than a central coordination of the local networks.

Furthermore, the aforementioned centralized electricity production has resulted in a transmission network which comprises very long transmission lines, resulting in relatively large building cost, energy losses and high maintenance costs. The same design is also found in e.g. natural gas transportation and distribution. Finally, technological improvements in the field of Information Technology and smart metering have led to improved abilities in monitoring and control of energy load and generation.
The smart grid has risen as a result of all the above mentioned events. Though there is no specific definition for the smart grid, all methodologies that have been developed for the implementation of smart grids, include some common characteristics. The smart grid could be described as a system which uses modern technologies to create more efficient energy systems compared to the present ones. The smart grid usually combines decentralized energy production (often from renewable energy sources), smart metering, information technology, energy storage devices, load and generation monitoring and control and other technologies to produce and distribute electrical and thermal energy in the most efficient way possible.

1.2 Technical aspects of the Smart Grid

As mentioned above, smart grids use the latest advancements in many technological fields. Some of these evolutions are briefly discussed in the next paragraphs.

Demand Side Management (DSM)

Demand Side Management exploits the flexibility offered by the devices to achieve specific goals related to the energy production and consumption. In most of the electricity networks nowadays, the production is determined by the demand; production should in every moment be equal to the demand and demand is determined only on consumers will to turn on or off a device. DSM uses the scheduling flexibility of consuming devices to determine their energy pattern. DSM systems use this flexibility to achieve certain goals. One possible goal is creating a demand curve that is more flattened, thus reducing the need for expensive energy during peak hours. During the implementation of the goals set, the demand will have to be reduced in the peak hours and the energy that would be consumed during these hours is aimed to be consumed in off-peak hours creating one, as much as possible, flat demand curve. In other cases the goal might not be to create a flat curve but shifting the demand in hours when energy from renewable sources is available. In cases like the previous two, devices that were usually consuming energy in peak hours, are changing their consumption pattern for the goal to be achieved. This should not result in discomfort of the residents and preferably these changes should not even be noticeable by the consumers. The smart grid project described in [16] has shown that shifting the consumption of specific devices (like clothe dryers in the specific example) to achieve a certain goal is possible while creating hardly any discomfort to the residents. Of course not all devices have the same scheduling freedom to be part of such a management of demand; e.g. lighting should always be available when user demands for it and thus there is no chance offered for transposing its operation.

Micro-grid

Though there is not an explicit definition for microgrids, a microgrid consists of a group of loads, placed within a neighborhood or a broader area, whose energy needs can be covered by micro generation units and storage devices which can communicate and create a local distribution network. The microgrid is usually coupled with the main grid. In cases where the energy production inside the microgrid is equal to the energy demand of the devices in the microgrid, the microgrid can even be decoupled from the rest of the grid and operate autonomously (islanding.

Integrated Communications

Smart grid implementation includes the collection, processing and storage of (a large amount of) data that needs to be communicated between different components of the smart grid. This data can be coming from e.g. smart meters that have to send their data to a local controller or data that need to be sent from a grid controller to local house controllers and backwards. These actions demand the existence of protocols and standards which will allow these devices to communicate efficiently.

Storage

Within the effort to alter the energy consumption pattern in a manner that was previously described, energy storage devices can play a very important role. An obvious use of storage devices is the storage of energy during off-peak hours in order to use this energy during peak hours. In the simulations conducted in [4] the interaction of micro-CHP units and heat buffers results in maximizing the profit from the electricity produced by the micro-CHP units as the production is shifted in high-price hours. Storage devices can also be used within a smart grid to minimize the problem resulting from intermittent energy production from renewable resources.

VPP Virtual Power Plant

In the aforementioned example of [4], it was shown that a large group of micro-CHP units can be controlled and the electricity production can follow a plan decided by a controller. As a result of the large number of micro-CHPs that can be controlled, the final amount of energy produced can be comparable to the energy produced by a power plant. An advantage of a Virtual Power Plant (VPP) compared to a conventional power plant is that it usually consists of micro generators that can reach their nominal power very fast (compared to a coal plant for example) thereby providing the ability to cover the electricity demand in peak (high-price) consumption periods.

PHEVs Plug-in Hybrid Electric Vehicles

The need of reducing the consumption of fossil fuels, the advancements in battery technologies and the high efficiency and controllability of electric motors are some of the reasons that have resulted in an increased number of electric or hybrid vehicles over the last years. This evolution is of course leading to an increased electricity demand. If PHEVs were charged "on-demand" this could lead to an overload of the grid as most people would charge their cars in peak hours in the evening when they return back home from their jobs. Research is conducted in [30] to develop algorithms to optimize the utilisation of PHEVs in electricity grids. Furthermore the use of PHEVs in a smart grid could also be reversed when the vehicles could supply certain amount of energy in the grid during peak loads (vehicle-to-grid technology).

1.3 Smart grids and sustainable development

This report treats Demand Side Management (DSM) technologies since it examines the implementation of Triana using the FPAI platform. As this report is also part of the study programme of the master Sustainable Energy Technology provided by the University of Twente, the relation of smart grids and DSM technologies to the master programme is explained in the following section.

1.3.1 Technology and sustainable development

The importance of developing sustainable technologies nowadays is profound due to a series of problems, related to human activities on the planet, like the increase of CO2 emissions. In order for a technological development to be characterized as sustainable, it has to contribute to all dimensions of sustainability (figure 4): It has to be 1) environmentally responsible, 2) economically viable for both the producer of the technology and the users and 3) it must support the functioning of a society in general by e.g. providing employment seats and/or fulfilling a need of the society. In the next paragraphs it is made clear how the DSM technologies and generally Smart Grid technologies are related to the three sustainability dimensions.

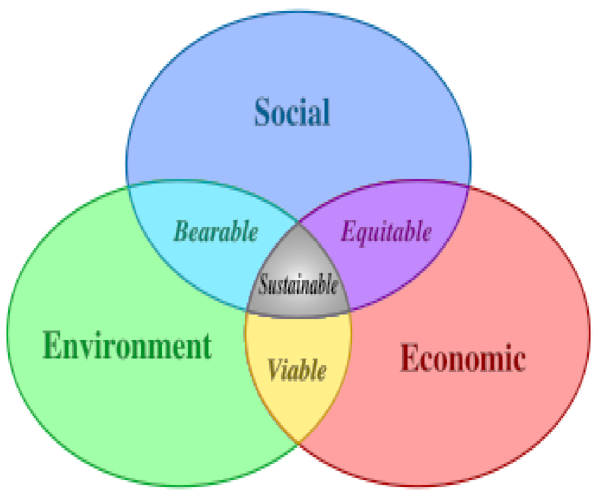


Figure 4: The three dimensions of sustainability

1.3.2 Contribution to environmental aspects

CO_2 emission reduction

Electricity and heat generation and the transportation sector are the largest CO_2 emission producers (figure 5). As it is shown later in this report, the introduction of DSM technologies can lead to a larger integration of renewable energy sources and to a more efficient exploitation of them. Reducing the CO_2 emissions by decreasing the electricity produced by fossil fuels is a central environmental and energy-policy goal worldwide and the integration of renewables plays a decisive role towards this direction.

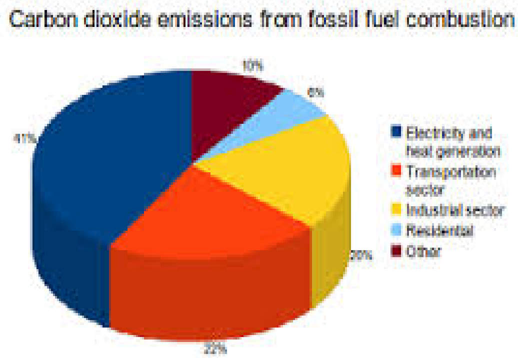


Figure 5: CO_2 Emissions from Fuel Combustion (2012), Source: International Energy Agency

Moreover, a larger integration of renewables and the introduction of Electric Vehicles (EVs) will reduce the CO2 emissions produced by transportation means. Another promising concept related to DSM and electric vehicles is Vehicle-to-Grid (V2G), according to which vehicles charge their batteries in low consumption-price hours and could offer it back to the grid when they are parked in peak hours. The substitution of heaters that use fossil fuels with heat pumps using electricity coming from sustainable sources is another modification that could reduce CO2 emissions.

Energy consumption decrease by increasing the energy efficiency

Specific Smart Grid systems aim at the maximum possible exploitation of electricity produced locally, mostly because of the economic benefits it brings to the end-users and also because most of this energy usually comes for renewable sources (environmental concerns). By using energy produced at a local level we reduce the energy wasted due to transportation and distribution losses. Furthermore, voltage gradually decreases along a transmission and/or distribution line. Utilities are sometimes forced by that fact to provide excessive voltage to the consumers during their effort to deliver a minimum amount of voltage. This issue can also be addressed by a smarter control of the voltage levels and thus by eliminating the need of electric utilities to provide excessive voltage [21].

1.3.3 Contribution to financial aspects

Reducing the energy production and consumption cost

One important target of DSM systems is to reduce the peak electricity demand (figure 6). During peak demand hours, energy sources are used that are more expensive compared to the sources used to cover the base and intermediate demand; thus decreasing peak demand directly saves money concerning electricity companies as less use of peak plants is done. On its turn, this should have a positive impact to the consumers electricity bill. DSM methodologies like Triana have also as a local goal to minimize costs for the energy consumed during the operation of a device. Furthermore during peak hours the stability of the grid, regarding the voltage and frequency levels, is put under stress, threatening the uninterrupted energy supply to the customers.

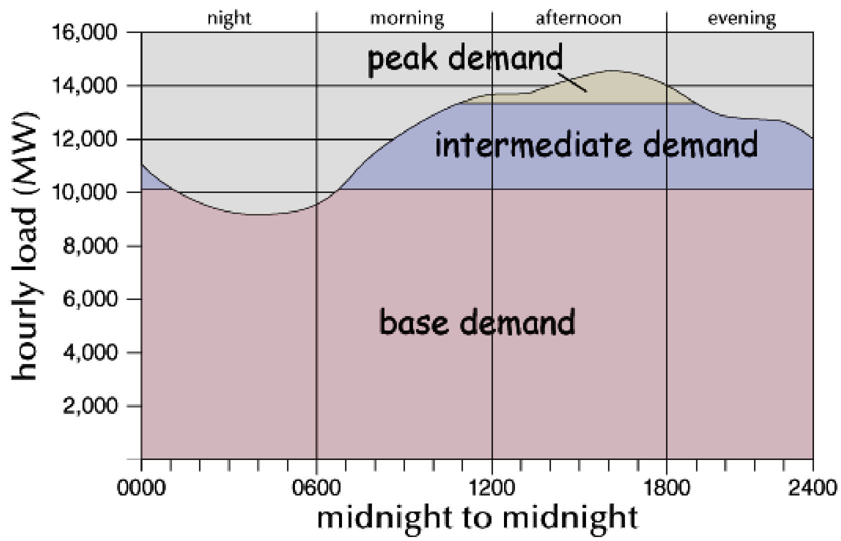


Figure 6: Daily demand curve: DSM technologies aim at shifting the consumption from peak hours to hours of lower consumption

The reduction of the energy cost can also be considered as contribution to the societal need for cheaper and thus affordable electricity for more people.

1.3.4 Contribution to societal aspects

Creating new employment seats

Apart from the jobs related directly with the development of Smart Grids, like people working in the ICT sector or companies working in the smart meter industry and researchers, there is a significant number of jobs created indirectly. That includes installation technicians or people working in the renewable energy sector as a consequence of their growing integration.

Finally, it is reasonable that all benefits posed in this chapter, will increase with a parallel increase of the integration of DSM systems. FPAI is expected to play an important role towards the deployment of DSM systems in large scale as it provides an intermediate platform that makes communication between different devices and DSM systems possible.

In the next chapter a series of Demand Side Management and other energy control technologies are described, thereby giving more details for Triana and Powermatcher. In the third chapter a presentation of FPAI is made giving more emphasis to the Resource Abstraction Layer and Resource Abstraction Interface, the two components whose task is to provide to an energy application the parameters that determine the flexibility of a device.

In the next chapter a series of Demand Side Management and other energy control technologies are described, thereby giving more details for Triana and Powermatcher. In the third chapter a presentation of FPAI is made giving more emphasis to the Resource Abstraction Layer and Resource Abstraction Interface, the two components whose task is to provide to an energy application the parameters that determine the flexibility of a device.

1.4 Research questions

The main research questions of this thesis are answered in the fourth and fifth chapter. The main issues that are researched within this thesis are:

- Investigate if FPAI offers a suitable platform on top of which the Planning phase of Triana can be implemented.
- Which are the incompatibilities that must be overcome for the implementation of the Planning phase of Triana on top of FPAI to become possible?
- Develop the proper software that executes the above mentioned implementation.
- Does the software produce the desired results; is the implementation successful?
- Can Triana be implemented on top of FPAI as a whole? What are the additional components that need to be developed in the future so that all three steps of Triana can be implemented on top of FPAI?

2 Energy control methodologies

In this chapter we give an extended description of Triana, a DSM technology developed by the University of Twente followed by a description of Powermatcher, a control system developed by the Energy Center of Netherlands (ECN) and later by TNO. Finally a brief presentation of other energy control methodologies is made.

2.1 Triana

A detailed presentation covering all aspects of Triana can be found in [4], [22] and [7], the three Ph.D. theses that lead to the creation of Triana. Most of the material used in this section can be found in [4] *"TRIANA: a control strategy for Smart Grids"*, Vincent Bakker, Ph.D. Thesis, University of Twente, 2012.

In [22] a definition of Triana is given. According to that definition the goal of Triana is *"to monitor, control and optimize the domestic import/export pattern of electricity and to reach objectives which may incorporate local but also global goals"*. A local goal could be the use of locally produced electricity and a global goal could be to shift the peak demand load.

The term "local" could refer to the level of a neighborhood or a single building or a device and the term "global" could refer to a country, a city or a part of them. The necessity for communication between the local nodes (e.g. a building) and the global planner (e.g. a Distribution System Operator) leads to a tree structure, the root of which is the global planner and the bottom-leaves of which are the local node controllers (figure 7). Electricity prices sent from the global planner to the local nodes and electricity consumption profiles of devices are sent from the devices to the global planner.

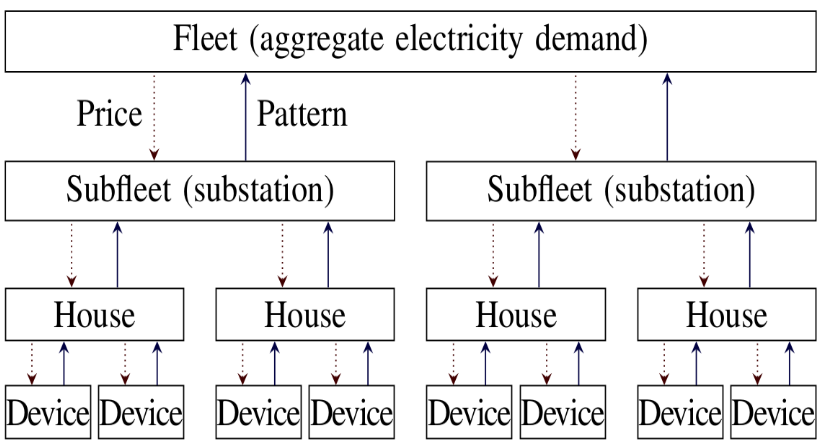


Figure 7: The tree structure of Triana

Triana has three steps, developed to achieve the goal set above. These steps are: offline local forecasting, offline global planning and real-time local control.

During the first step, a forecast is made regarding different parameters related to the energy consumption of a device for a specific time horizon. This information determines the scheduling freedom of a device the extent to which the consumption profile of a device can be shifted in time or altered in order to achieve a certain global goal. An example of a global objective could be to achieve a certain global consumption profile. This goal and the prediction done for every device in the first step, are the inputs for the second step the planning. During that step, the energy consumption of a device for every interval of a time horizon is defined, based on the global goal and the minimum energy cost for the device. If during the operation of a device based on the planning, an unacceptable mismatch is observed between the global plan and the actual given global energy profile, a new (re)planning phase begins, this time using e.g. forecasts resulting from data obtained from the recent history of every device or from weather predictions for the next hours (if e.g. a prediction must be made for a heat producing device). In the end the solution that leads to the closest result regarding the planned goal is chosen or even a new global objective might be formed.

2.1.1 Forecasting

The parameters related to the energy consumption of a device, can vary depending on the device type. Planning the operation of a heat pump, requires the forecast of heat demand, whereas planning the operation of a washing machine requires the forecast of a permitted starting time. A general categorization of device types is done in [4] based on how the devices handle the energy that flows in them as input. Devices can exchange energy with the outside world (Exchanging devices), convert energy of one type to another type (Converting devices), store energy temporarily (Buffering devices) or consume energy (Consuming devices).

Within the software implementation, Triana makes a distinction of 4 different device types according to the scheduling freedom these devices can offer. These types are:

- Buffered Converters (e.g. heat pump + heat buffer): These converters consist of a pair of devices; one which consumes or produces energy and one which can temporarily store an amount of energy.
- Smart Appliances (e.g. a washing machine): The energy profile of these devices cannot be altered, but the starting time can be shifted taking into consideration the preferences of the consumers.
- Standard Appliances (e.g. PV panels or lighting): There is no flexibility offered by these devices.
- Standard Battery (e.g. batteries): The scheduling freedom of these devices is similar to the Buffered Converters but they can also return electricity to the grid.

The scheduling freedom determines the extent of control that a device can accept during its operation by e.g. shifting the operation hours during a day. As mentioned in [6] 50% of the consumption by household appliances can be shifted in time.

As stated before, the planning step requires as input, data resulting from the forecasting step. Forecasting procedure must fulfill certain criteria. Firstly the forecasting should be autonomous and require no input data by the user. Secondly, forecasts must be made for every device in a building (or at least for the devices which have a scheduling freedom). We must also be able to conduct forecasts for different time spans e.g. forecasting the consumption of a device for the next day or the next hours since planning might need to be done for different planning horizons.

Finally, forecasting depends on variables that are dynamic like temperature and forecasting should be able to track the changes of these variables which can be rapid.

Triana uses neural network techniques to implement the forecasting. The forecasting results in a number of output values O, which come from a function F with two variables: the input for the forecast model I and the forecast model parameters Fp: F(I, Fp)$\rightarrow$O. As mentioned above, the input might change over the time. The neural network can adapt these changes thanks to the relearning ability, which is obtained by a training procedure where input and output data are known and the error of the output of the neural network is observed. During this procedure the weights of the neurons change, representing the changes in the input and the connection between the input and the output [4].

A use case: forecasting the heat demand

Concerning one of the use cases where Triana was tested, the heat demand should be forecasted for the upcoming day on an hourly basis. It is considered that the heat demand changes every day of the week due to the influence of the human activity. Therefore a different neural network is used for each day to incorporate the changes during the weekdays. To let the neural network to take into account these differences during the week, historical heat demand are used as input to the neural network. Another important parameter which determines the heat demand is weather. Therefore, predictions from weather stations near the houses are taken as inputs too.

The goal for this case was, as said above, to predict the heat demand of the next day. Historical heat demand of the previous 7 days and also 14 days before, comprise the possible inputs of the heat demand. Minimum and maximum temperatures and wind speeds of the day to be forecasted and also one day earlier of the forecasted day are used. Furthermore, the average forecasted hourly temperatures and wind speeds (both of the day to be forecasted and one day before) complete the possible inputs. The potential inputs are summarized in the next table [4].

Input	Description
H_{-1}	Hourly heat demand 1 day earlier
H_{-2}	Hourly heat demand 2 days earlier
H_{-3}	Hourly heat demand 3 days earlier
H_{-4}	Hourly heat demand 4 days earlier
H_{-5}	Hourly heat demand 5 days earlier
H_{-6}	Hourly heat demand 6 days earlier
H_{-7}	Hourly heat demand 7 days earlier
H_{-14}	Hourly heat demand 14 days before D_n
$W_{0,\mathrm{mm}}$	The forecasted minimum and maximum windspeed
$W_{0,30}$	The average forecasted windspeed (per half hour)
$W_{0,60}$	The average forecasted windspeed (per hour)
$W_{-1,\mathrm{mm}}$	The minimum and maximum windspeeds 1 day earlier
$W_{-1,30}$	The average windspeed (per half hour) 1 day earlier
$W_{-1,60}$	The average windspeeds (per hour) 1 day earlier
$T_{0,\mathrm{mm}}$	The forecasted minimum and maximum temperatures
$T_{0,30}$	The average forecasted temperatures (per half hour)
$T_{0,60}$	The average forecasted temperatures (per hour)
$T_{-1,\mathrm{mm}}$	The minimum and maximum temperatures 1 day earlier
$T_{-1,30}$	The average temperatures (per half hour) 1 day earlier
$T_{-1,60}$	The average temperatures (per hour) 1 day earlier

Table 1: The possible inputs of the forecasting model [4]

An important aspect during the training of the neural network is the use of a sliding window. Usually, in neural network training, random set of heat demand data during the year are chosen. This creates a more general prediction. But heat demand fluctuates a lot between seasons. Thus, for the training of the neural network of each day, only data from the recent 4, 5 or 6 weeks are used. Then these predictions are compared with the ones where the sliding window was not used.

For this comparison to take place, the performance of each forecasting is calculated. The performance is actually measuring how much the forecasted values for the heat demand deviate from the actual values of the heat demand for a number of days of a house.

During training, the heat demand of 4 houses was forecasted. When using as input data the heat demand of the previous day H-1, the heat demand of one week before H-7 and the hourly forecasted temperatures T0, 60 ,it was found that the forecasted values were always higher than the actual values as can be seen in figure 8.

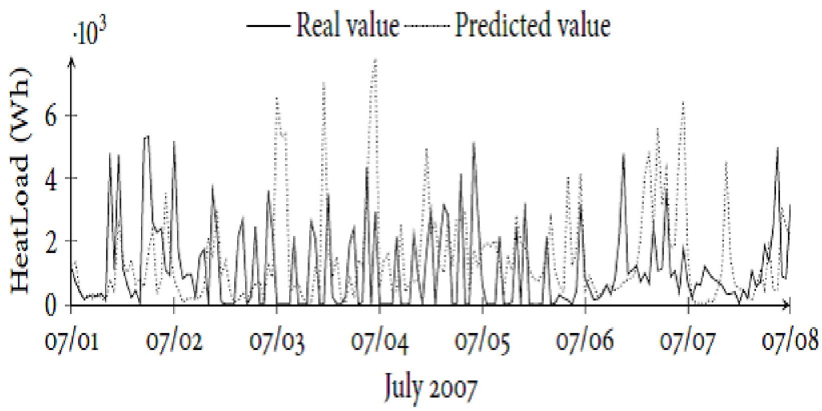


Figure 8: Real and predicted heat demand values [4]

2.1.2 Planning

As a result of the first (forecasting) step, estimations of the production and/or the consumption pattern of all devices present in a building for the next day are given. The second (planning) step uses these forecasts to reach an objective set for the whole grid, a global objective. This could be e.g. a flattened demand curve or balancing demand and supply using as much as possible energy produced by distributed renewable resources.

In contrary to other control methodologies like “Powermatcher”, Triana does not use an agent-based approach of planning where each agent tries to optimize its own profit, but planning is determined by a global objective. On the other hand, Triana like other control methodologies uses a hierarchical planning scheme where results are aggregated in each level of the hierarchy.

The hierarchical structure can be represented by a tree structure the root of which is the global-grid planned. Next there is a number of levels where each node represents the controller of a part of the whole grid, the sub-grid controller. At the bottom level of the tree, the nodes represent the controllers of a single building (see also figure 7).

Two different control approaches are distinguished in [4]. In the first approach, the control is dominated by the goal specified by the global planner (top of the tree). The global planner steers signals to the sub-grid controllers and the sub-grid controllers send these signals to the local controllers (bottom of the tree). After the decisions of the global planner have been executed by the local controllers, the latter send their results to the subgrid controllers and these results are sent back to the global planner (after they have been aggregated by the subgrid controllers). The global planner compares the results with the goal set in the beginning of the planning (which could be for example a specific pattern for the consumption). If the global planner is not satisfied by the mismatch between the goal set and the actual situation, it might send new signals to the subgrid and local controllers based on this comparison and this process may be iterated until the initial goal is approximated in a larger extent.

In the second approach, the global planner takes decisions for the sub controllers planning and the sub controllers are afterwards responsible for determining the planning of the local nodes below them. Thus the load of achieving the global goal is split into smaller sub goals creating a more distributed planning.

Two use cases. Planning the operation of 50 and 200 freezers

Two examples of how the planning of the operation of 50 freezers (1st example) and 200 freezers (2nd example) can be achieved are presented as an implementation of the planning strategy.

The global objective is to achieve a flat consumption over the planning horizon which is one day. Generally freezers present a high scheduling freedom and they can be switched on or off whenever we want in order to achieve the planning as long as their temperature stays within certain boundaries. In the first example the global planner communicates directly with the house controllers to which the same signals are steered. Thus the tree in figure 7 has only two levels, the root node and 50 children representing the house-device controllers. In the second example 4 intermediate controllers are used each one controlling a group of 50 houses.

In the first simulation of the 50 houses it was found that the objective was only achieved when different price vectors were sent in each house. When the same price vector was used for every freezer, the aggregated consumption had significant demand peaks since all house controllers drove the operation of the freezers in the low cost periods which were the same for every building since they achieved the same price vector. When different price vectors were used for each house, the mismatch M was almost 0 after the 25th iteration of the planning procedure (see figure 9).

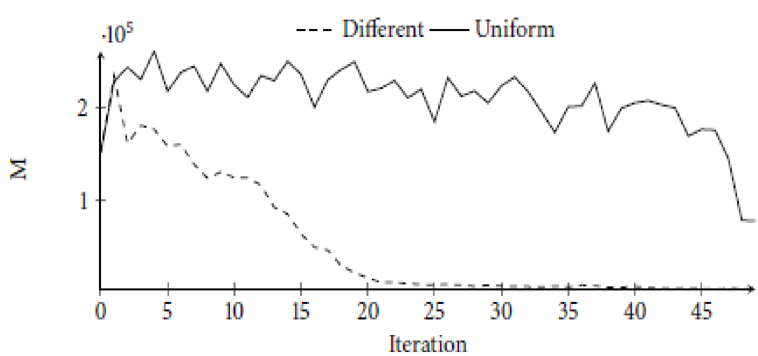


Figure 9: The mismatch of the planning when the same (solid line) of different (dashed line) price vectors are used in the simulation of 50 houses [4]

In the simulation of 200 houses, there are 3 possibilities. Firstly all 200 houses could receive the same price vector. Secondly each group of 50 houses could receive the same price vector from the corresponding subgrid controller (top-down approach). Thirdly all 200 houses could receive different price vectors. In the last case the subgrid controller was responsible to determine the price vectors for the 50 houses it controlled (bottom-up approach). The last case revealed the best results (figure 10). In this case the global planner defines the global objective and determines a pattern for every subgrid. In the end of one iteration, the subgrid controllers aggregate the results of their group and send them back to the global planner which decides if a new pattern should be created for each subgrid or not according to the mismatch of the planning.

Figure 10: Mismatch in the second simulation is less when different price vectors are steered to each house controller [4]

An important parameter within planning is how communication is achieved between the levels of the tree. The communication design determines how much information is exchanged between the nodes of the tree and how fast this happens. It is preferable that the computational needs of the planning are distributed as widely as possible. This is succeeded in the case when different price vectors are determined for each controller. That was experimentally found in [4] after conducting several simulations. Of course, the information load can also be reduced when the simulation interval becomes larger. Another way to reduce data size is to decrease the size of messages that are communicated.

The two previous steps (forecasting and planning) influence in a great extent the real-time control. An unacceptable mismatch between the initial goal and the actual result might occur because of bad forecasts or an extreme goal. In that case, new forecasts are made using more recent data, making them more reliable. A new planning is also made, this time aiming to another goal.

2.1.3 Real-time control

Real-time control is the third and final step of Triana. In contrary to forecasting and planning which are implemented off-line, control is done in real-time. The most important criterion during the implementation of control is to minimize the total cost of devices which are controlled by a controller.

In every time interval there is a specific number of options regarding the operation of a device. An option "o" of a device for each time interval, determines how much energy flows in and out of the device. But there might be different options concerning the source of the energy that flows in a device; electricity from the grid or from a battery for example. Factors like the previous define the cost function for each device. If "tc_d" is the cost function of one device, the cost minimization problem can be expressed by:

$$minimize \sum_{d}^{Dev} tc_d \text{ ,where } tc_d = \sum_{o \in O_d^S} A_o x_o + B_o c_o$$

Equation1: The cost minimization problem in Triana

In the above equation:

Dev is the set of the devices which are controlled,
O_d^S , is a subset of the possible operational options for a device d in a state s,
$A_o x_o$,is the part of the cost which corresponds to the use of an energy stream of volume x_o, $B_o c_o$,is a fixed cost related to certain operation option where $c_o$can be zero or one. It is one if the option o is valid and zero when o is not valid. In the first case the energy flow x_o can take a value between a lower and upper bound.

In figure 11, the cost function of importing or exporting electricity via a grid connection point is depicted. The cost is proportional to the energy import-export. (the meaning of negative cost in figure 11 denotes the gain of exporting electricity).

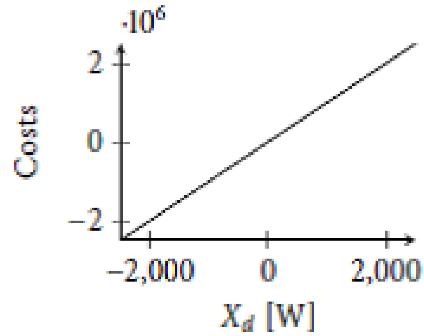


Figure 11: The cost of importing or exporting electricity is linear to amount of energy imported or exported [4]

More enhanced control can be achieved when the controller does not take a decision only based on the current demand, but when also the demand in future intervals is considered using forecasts for the future demand. This can be achieved using Model Predictive Control (MPC). More details around MPC and how it is applied in Triana can be found in [22].

2.1.4 Results Simulations

In chapter 6 of [4], the results of the simulations of 3 different use cases are presented. These results are presented here to give the reader an idea of what the practical implementation of a control methodology in the future might be.

1^{st} Case

In the first case a VPP is created controlling a number of micro-CHP devices. 50 houses are equipped with a micro-CHP device and a heat buffer.

CHPs produce electricity and heat simultaneously. The question is what happens when only electricity is needed in a certain period. In this case the heat that is produced cannot be consumed (since no demand exists) and therefore is stored in a buffer and can be used later. Thus the operation of the CHPs can also be electricity demand driven. When a large number of CHP devices is controlled, a large amount of electricity can be produced creating a VPP. The operation of the VPP has also an impact when the expected energy produced by it is about to be fed in the grid to cover short peaks. Supply of the peak demand is possible due to the fast start-up of the CHPs.

A micro-CHP appliance must be operated at its minimal load for a minimum amount of time, the "minimum runtime" (MR). Moreover it has to stay off for a certain amount of time before it can be started again, the "minimum offtime" (MO). MR and MO are two of the parameters that determine the scheduling freedom of the CHP appliance. Concerning the buffer, it should be provided with heat when the stored heat is below a lower level and it cannot store additional heat when the stored energy has reached an upper level (thus assuming that heat dumping is not allowed).

The CHP units that were used in the simulation had 1 kW electrical and 8 kW thermal nominal power. Two variations were simulated concerning the type of the heat buffer. The first used heat buffers with a 10 kWh capacity and the second buffers with 20 kWh capacity. Two different simulations were also done when the 10 kWh buffers were used. In the first the lower level of the buffer was 1 kWh and the upper 9 kWh. In the second the levels were 2 and 8 kWh respectively. The lower and upper levels of the 20 kWh buffers were set to 4 and 16 kWh.

For the simulation, heat demand data are required since they are going to determine the heat that must be provided by the CHPs. To determine the heat demand of the 50 houses, forecasts were made using real data of heat demand of four houses during 2006 and the heat demand of six houses from October 2009 to February 2010. The forecasts resulted to an error of about +10% from the real heat demand.

During the simulation, the goal set by the global planner was to maximize the revenue from the produced electricity. Thus CHPs were preferred to run in the morning and early in the evening (peak demand).

The results of the simulation when the CHPs tried to follow a given plan, were compared with the results from a simulation were the CHPs produced energy without any reference plan. It was shown that when Triana control was introduced, production was shifted in high-price periods. Furthermore, in the case where the lower and upper bound of the heat buffer were 2 and 8 kWh respectively, the results were improved, meaning that the production was even larger in the targeted hours.

The best results (largest revenues), were nevertheless obtained for the case with the 20 kWh buffers. On one hand less energy was produced compared to the 10 kWh buffer case (with 2-8 kWh limits) but the price per Wh and the total revenue was larger. The results are summarized in table 2.

Buffer size	Field	Realization	Planning (Forecast)	Planning (Actual)
10 kWh	Total production (Wh)	364000	389625	350875
1-9 kWh	Average price (Euro/MWh)	30.39	30.35	31.30
10 kWh	Total production (Wh)	367250	390250	354625
2-8 kWh	Average price (Euro/MWh)	29.90	29.50	30.70
20 kWh	Total production (Wh)	364625	392125	354375
4-16 kWh	Average price (Euro/MWh)	31.68	30.26	31.78

Table 2: The results of the VPP simulation [4]

2^{nd} Case

In the second case, 100 houses were supplied with a heat pump and a heat buffer of 10 kWh. The goal set in this simulation was to flatten the electricity consumption of this group of houses. Results are also compared to a case where no planning is done. As can be seen from the demand curve of figure 12 the morning peak that is present around 9 am is reduced. The load duration curve is also significantly flattened.

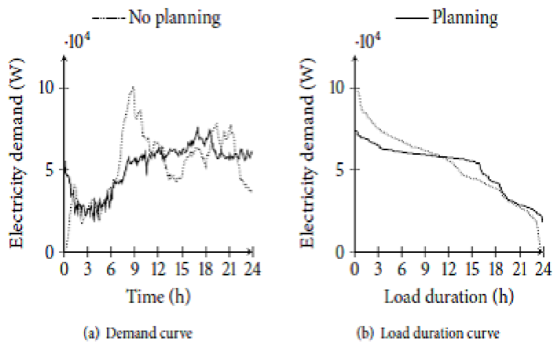


Figure 12: The results of the second use case [4]

<u>2^{rd} Case</u>

In the last case 50 houses comprise a group of buildings, all of them supplied with a micro-CHP appliance of 1 kWe and half of them are supplied with a heat pump which consumes 2 kW. This division is done so that all required power for the heat pumps could be supplied from the CHPs. Again, the global planning is to flatten the electricity demand of the building fleet. As can be seen from the figure 13, the demand is a flattened a bit more (shorter spikes) when planning is used than in the case were no planning was introduced and the highest demand is lowered as well.

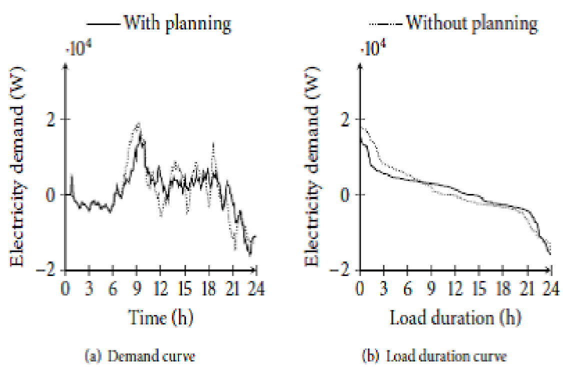


Figure 13: The results of the third case [4]

2.2 Powermatcher

Powermatcher is an energy control methodology developed by ECN and later by TNO. It is the first energy control system that was implemented on top of FPAI.

2.2.1 The need for energy control

According to [18], the Powermatcher (PM hereafter) is defined as *"a market-based control concept for supply and demand matching (SDM) in electricity networks with a high share of distributed generation"*.

In the final stages of the evolution of electricity grids which are presented in [18], communication of multiple systems must be achieved. Central coordination will finally also have to be replaced by distributed coordination to reduce computational complexity. Coordination of the large number of players competing in the energy market demands an ICT infrastructure whose design reflects to that need for coordination. Such ICT architecture should take into account the multiple actors who are involved and who care both for their own goals but also might have to cooperate for a global objective. The ICT system may not be owned by a central player but may be built thanks on the contribution of all actors involved in the electricity system.

Multi-Agent Systems (MAS) are suitable for such a system. In MAS, a large number of agents interact in a certain way based on parameters set by the electricity system like the achievement of common-global goals and the economic principles of electricity market.

The effort of different agents to achieve their goals in a competitive market is called "market-based control". Resource allocation is a type of multi-commodity flow problem, in which a number of participants interact in a market to share commodities (in this case energy) according to a specific function. PM uses the last concept within its implementation.

2.2.2 Powermatcher architecture

PM uses a tree structure. In the bottom level, the devices of the system are controlled by agents who try to optimize economically the devices operation. Hereby, the price which has to be paid for the energy which is consumed or produced by a device is finally determined by the root node, called "Supply-Demand (SD) Matcher". Between the devices and SD matcher, the intermediate SD matchers collect and aggregate the bids of the devices and send them to the root SD matcher (figure 14). The root SD matcher determines the final price of energy for each device. The bids of the devices depend on the function set by the agent of each device. Each device belongs to a category related to its scheduling freedom. These categories are similar to the categories in which devices are clustered in Triana.

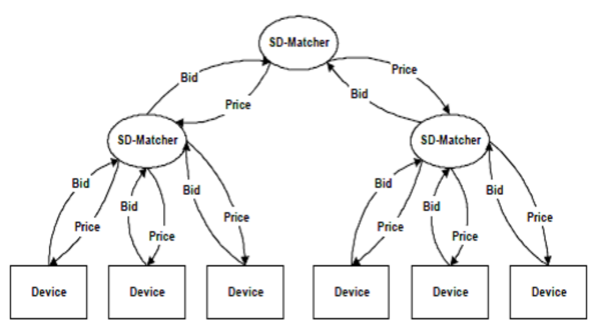


Figure 14: The tree structure of Powermatcher [18]

In [19] a more analytical description of the agents in each level can be found. According to that the agents are distinguished in four categories (figure 15):

- **Local device agent**: each Distributed Energy Resources (DER) device has one local device agent. The agent makes its latest bid known to the auctioneer and receives price updates from the auctioneer to determine how much energy it has to produce or consume. The local device agent correspond to the leaves of the tree of figure 14.

- **Auctioneer agent**: Collects the bids of all agents and computes the equilibrium price; it corresponds to the root of the tree of figure 14.
- **Concentrator agent**: There is one concentrator agent for a cluster of devices. It collects and aggregates the bids from the local agents in its cluster to and sends them to the auctioneer agent. Reversely it communicates the calculated prices by the auctioneer to the local agents. The concentrator agent is the intermediate SD-matcher of figure 14.
- **Objective agent**: The objective agent determines a goal for the cluster of devices with which it is connected. In case a cluster of devices is not connected to an objective agent, then the cluster tries to achieve balance in demand and supply between the devices of the cluster. The objective agent is not depicted in the representation of figure 14.

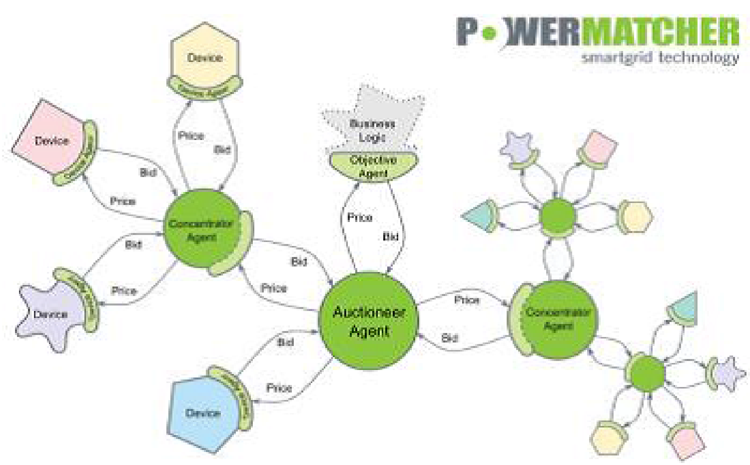


Figure 15: The structure of Powermatcher [19]

A number of simulations were conducted in [18] and [17] to test the reliability of Powermatcher. Some of these simulations are presented in the next paragraphs.

2.2.3 Tests and results

The PM methodology was tested in [18] in a simulation which included 40 houses connected in a low-voltage grid. Half of them use heat pumps as heat source and the other half use micro-CHP units, while all of them are equipped with batteries for electricity storage. Every house has a “Home Energy Management - HEM” box connected to all home devices. HEM is actually the intermediate SD matcher of figure 14. The root-node in this case is represented by an exchange agent which is on its turn connected to the medium-voltage grid. The group of the 40 houses is fed with energy by the medium-voltage grid in case the energy produced within it is not enough and delivers energy to the rest grid when a surplus of energy is observed.

The results of the simulation are compared to the results in the case where no control is applied during the operation of the house appliances. The main results are:

1. A decrease of the peak consumption and a 30% decrease of the imported energy by the external grid. Due to the control strategy, devices consumed less during peak periods and production was increased because of the high price of electricity during that period. The use of batteries during that period also decreases the need for external energy.

2. The consumption curve is more flattened and so are the curves both for the local energy production and the energy fed from the MV grid.

In [17], two real world test cases and their results are presented. One of them was conducted within the European programme "CRISP" distributed intelligence in Critical Infrastructures for Sustainable Power. It examines how the problems related to the intermittency of wind power production can affect the operation of a network and how these problems can be dealt partly with PM. Except for the wind parks, the field test also comprised an emergency generator, small CHP devices, a cold store and a place where control actions are executed (ECN).

Each of the above parts of the field test had an agent who was responsible to predict the energy that each device produces so that the field test configuration can participate in the Amsterdam Power Exchange (APX) where predictions of cluster consumption and production are placed 12 hours ahead of a 24 hour horizon. The predictions conducted by the agents are possible when models related to the operation of the devices they control, like wind forecasts for wind turbines, are provided to the agents. The predictions determine the bids that the agents offer during the day in the APX. After all bids are placed, the PM defines the prices of energy and agents act according to them.

Wind turbines provide by far the largest amounts of energy in the examined configuration. When predictions concerning their power production contain an error, the remaining devices have to compensate this error. The errors in wind power production result in errors in the total power production by the cluster of the devices. By compensating the errors, the other devices aim to minimize the deviation from the prediction of the energy produced by the cluster.

The problem of overproduction is solved by switching off the CHP units which are normally continuously on. In underproduction problems, the emergency generators may be used but are not enough to solve the problem since they produce power of some kW which are negligible to the MW missing from the wind turbines underproduction. Heat pumps and thermal energy storage mediums provide do not provide large energy storage to significantly reduce the problem too.

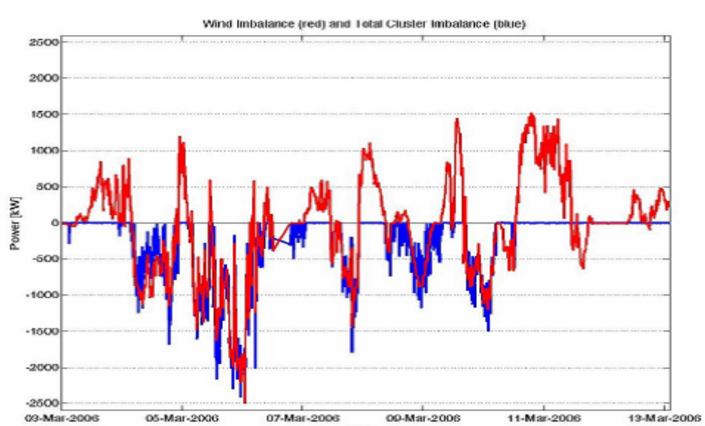


Figure 16: Wind imbalance (red) and cluster imbalance (blue) [17]

The graph in figure 16 indicates that when wind power imbalance is positive (overproduction), the cluster is able to reduce the total imbalance. However when underproduction occurs, the cluster has an imbalance too.

In figure 17, the heat pump behavior related to the energy price is depicted. In periods with high price, heat pumps stop working and the water temperature drops. The opposite happens in low-price periods. Of course there is a certain range in which water temperature must be kept. When no control takes place, the temperature changes in time from the maximum to the minimum allowable temperature creating a saw-tooth profile in time.

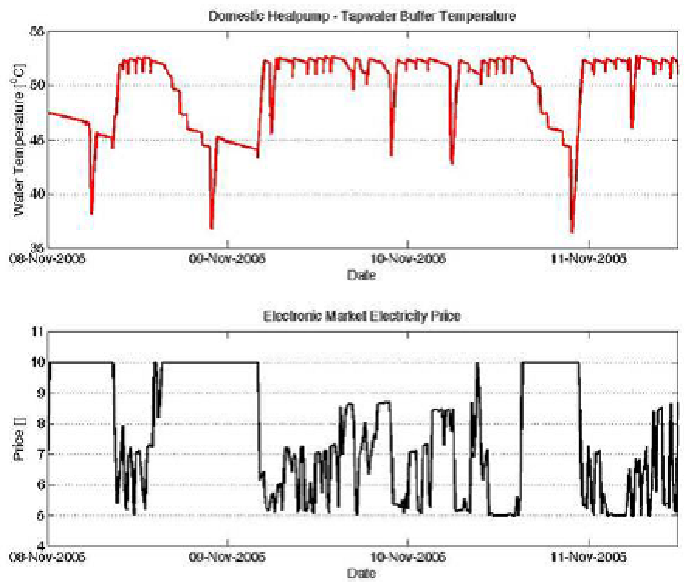


Figure 17: The response of the heat pump to the electricity price [17]

2.3 Other energy control methodologies

The next paragraphs describe other energy control methodologies developed by universities or research institutes. A useful insight into several aspects of the Smart Grid can be gained by exploring the different approaches that are followed by different researchers.

2.3.1 Game Theoretic Methods for the Smart Grid

In [31] it is shown how game theoretic methods can be applied for the realisation of certain sub-systems of the smart grid, namely micro-grid systems, demand-side management and communications. Thus, the work in [31] does not refer to a complete control strategy (like Triana) rather than to specific aspects of it.

A smart grid is defined in [31] as "a power network composed of intelligent nodes that can operate, communicate, and interact, autonomously, in order to efficiently deliver power and electricity to their consumers". The nodes represent the players to which game theory is applied. The nodes are the parts that compose the smart grid and vary from smart meters and warehouse appliances to micro-grids and prosumers.

The basic game theory classes and strategies are described in [31] are presented schematically in figure 18.

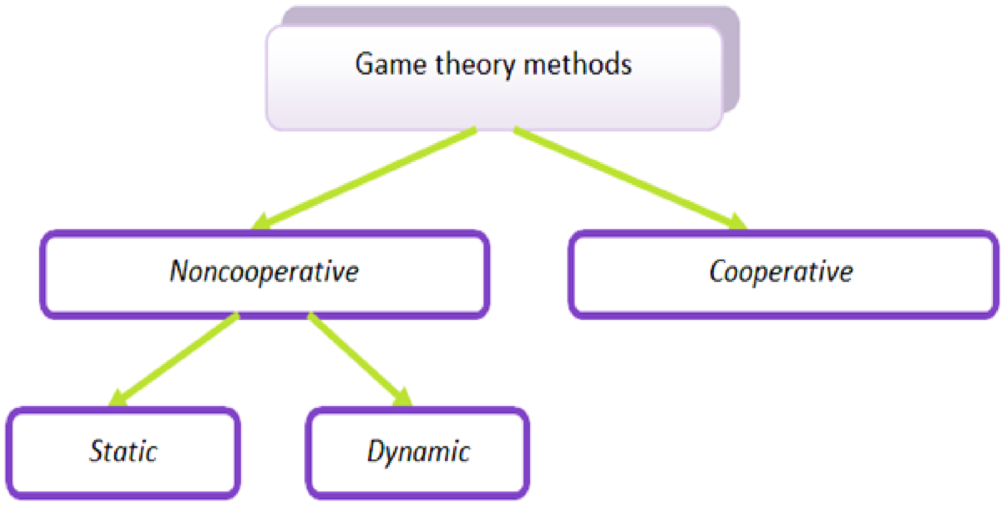


Figure 18: Game theory methods-strategies[31]

In cooperative theory, the players can communicate and decide to cooperate with each other. It is applied to the integration of power, communication and network technologies. An example of such a situation is the use of relays in a smart grid network to improve the efficiency of the communication links between the smart grid elements.

In noncooperative game theory the players have to some extent conflicting interests over the outcome of the decision process and also no communication and coordination between the players exist. It is applied to aspects like demand-side management and real-time monitoring.

Noncooperative theory strategies are further distinguished in static and dynamic. A short definition of a static method is given in [31] and can be summarized as follows. It consists of three components:

1. A set of players N,
2. Action sets $A_i \ \forall \ i \in N$ and
3. Utility functions $u_i \ \forall \ i \in N$

Every player i wants to choose an action $a_i \in A_i$ so as to optimize its utility function $u_i(a_i, a_{-}i)$ which depends both on i's player action choice a_i but also on the actions of the other players $a_{-}i$.

On the other hand, in dynamic strategies each player has some information related to the other players choices. Furthermore it is of critical importance the time at which a decision is made.

2.3.2 Adaptive Stochastic Control for the Smart Grid

*"Stochastic control or stochastic optimal control is a subfield of control theory that deals with the existence of uncertainty either in observations of the data or in the things that drive the evolution of the data" [35]

The rationale given in [3] is that the smart grid is a system that consists of a large number of stochastic variables. The problem of the long term management of the smart grid now is decomposed in a series of short term problems by using dynamic programming. In the paper a set of five components is defined, which can describe a stochastic, dynamic system. These are sets of variables and functions that are involved in the management of the smart grid. On their turn these are used in two specific applications:

1. "Load and source control" describes the flow of energy from different sources to serve different loads. The sources in that case are the renewable energy sources as well as any storage inside the smart grid. The loads of the smart grid should be supplied with energy in an uninterruptable way.
2. The problem of energy storage. Here the research question is when to store energy to a battery and when not to. The decision is received taking under consideration variables like variations in wind, load demand, electricity prices and charging state of the battery.

The mathematical equations that are formed in [3] are solved using the simplex algorithm.

Finally the authors enumerate the criteria that their control method should fulfill in order to be successful and give a detailed reference of the advanced components and other aspects of the smart grid they consider. These components are:

1. Advanced Metering Infrastructure (AMI) and Home Area Network (HAN)
2. Power electronics
3. Photovoltaics and solar heating
4. Recharging electrical vehicles
5. Microgrids
6. Energy storage
7. Distributed generation
8. Storm management
9. Massive solar thermal and wind generation facilities
10. Nanotechnologies

2.3.3 Control method for multi-microgrid systems in smart grid environment-stability, optimization and smart demand participation

The interaction of the main grid with micro grids is examined in [32]. More specifically the hierarchy and the decision-making process are described, in order to achieve stability in the main grid when instabilities occur in the micro grids.

A conceptual description of the algorithms used to implement the specific methodology, can be found in [32]. An important issue there is how a microgrid can influence the operation of the main grid especially in transient phenomena. In case a microgrid causes problems to the main grid, a decision has to be taken whether this microgrid should stay connected with the rest of the (main) grid or not (islanded). The main core of the paper is defining the hierarchy with which these decisions should be taken. The main grid operator which is the supervisor of the SCADA systems is on the top of the pyramid and Distributed Energy Resources (DER) are on the bottom. Thus the main grid operator makes the final decision if a micro grid should be islanded from the rest of the grid or not (figure 19).

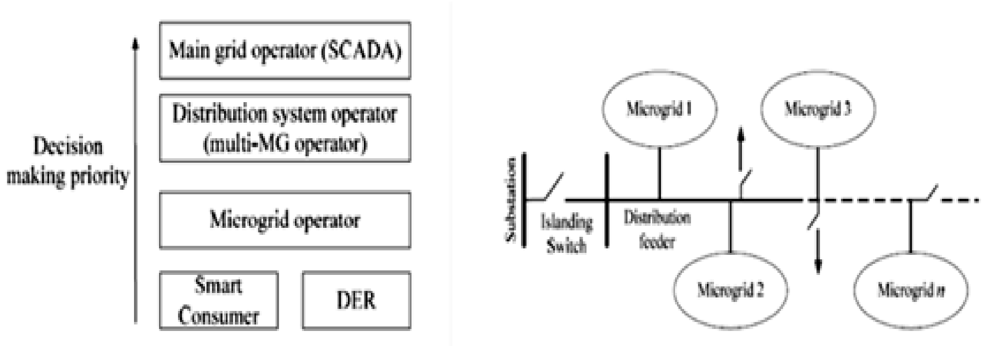


Figure 19: The decision making priority that is proposed in a grid consisted of micro grids [32]

2.3.4 Optimization and Control Theory for Smart Grids

The goal of the research conducted in [10] is to optimize 1) the Grid Design, 2) the Grid Control and 3) the Grid Stability by using emerging technologies like renewables, storage, meters and PHEVs (plug-in hybrid electrical vehicles). These three aspects are described next.

Grid Design

One issue that needs to be addressed when designing a grid is where to place new transmission, generation and storage facilities. Here, simulations are carried out where the grid is represented by a graph in which is composed of nodes (that correspond to generators, storage devices and loads) and lines (that correspond the transmission lines between the nodes) (see figure 20). The optimization problem is a cost minimization problem where the cost refers to the power (Ohm) losses from the lines and the cost of installing new transmission lines. The cost function includes one matrix with the conductivities of each node and one with the currents that are injected by or consumed at each node. The cost function is:

$$J^T \cdot K^{-}1(y) \cdot J + \sum c_i \tilde{y}_i^{\gamma}$$

Equation2: The cost function which represents the Ohm losses in the grid

In this equation the variable y_i represents the conductivities which range from 0 to $y_i^m ax \cdot \tilde{y}_i$ are the normalized conductivities and $c_i$is the effective cost per line. The sum of the products of the second term of the equation represents the cost of building the transmission lines. In the first term, J is a matrix which includes the currents that are injected or consumed by the nodes and $K(y)$ is the matrix of conductivities which depends on grid topology. Although it is not mentioned explicitly in the paper, the strategy followed in [10] is similar to an agent-based method in which a control parameter $0 \leq \gamma \leq 1$ is changed in iterative simulations.

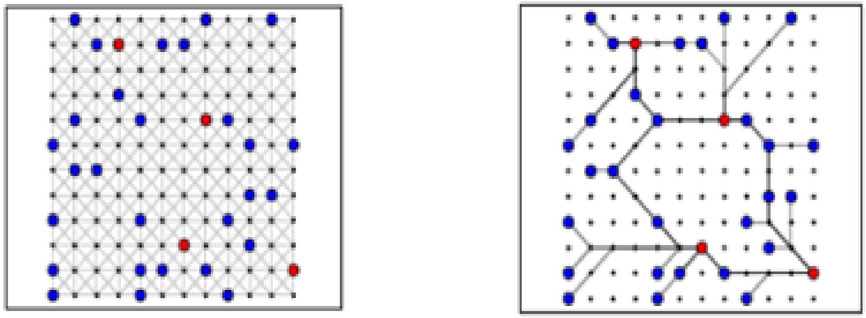


Figure 20: The grid (red nodes-generators and blue nodes-loads) and a solution for $\gamma = 0.1$ [10]

Grid Control

The core task in the grid control is to balance generation and consumption. The grid performance would improve if the peak demands (in the afternoon) are trimmed and valleys (in the night) are filled. The tool to achieve this is the optimized charging of PHEVs using methods from queuing theory.

Grid Stability

During the stable operation of the grid, structural faults (like generator or transmission lines loss) might occur, that lead to unstable situations which require an adjusting of the consumption loads or the generating capabilities or might even require a transmission line to be cut out of the grid in order to reach to another stable point where demand and supply are balanced. Heuristic algorithms are developed to detect an efficient way for the transition between these two states.

2.3.5 Using Neural networks to create a new control methodology for smart grids.

In [33] a quite different method compared to all others found in literature is presented. Unfortunately no research papers exist for the specific method but only a general description in the official site and other sites related to smart grids.

In this method, neurons grown in dish plates are"taught" how to respond to voltages and speed signals from a simulated power grid. The results of such experiments are incorporated in bio-inspired artificial neural networks (BIANNS). The authors mention: "*As power-systems control becomes more and more complex, it makes sense to look to the brain as a model for how to deal with all of the complexity and the uncertainty that exists*".

2.3.6 Agent based control of Virtual Power Plants

In [13] the control methodology that is described is based on Multi Agent Systems (MAS). The scheme in figure 21 presents the realization of the proposed MAS.

This scheme involves three levels of agents which can partly communicate among each other. The bottom "Field" level is related to the control of single production units or controllable loads. It should be clarified that groups of agents in each level form a MAS. The second "Management" level involves agents that are responsible for information exchange between the MAS of the first level in order to achieve cooperation between the MAS of the first level. Finally the third “Enterprise” Level includes agents that try to form larger MAS to participate in the energy market (figure 21).

The energy market is based on the scenario that hourly prices are announced 12 hours in advance. In such an energy market, the goal of each production unit or load is to maximize its profit by the participation in the market.

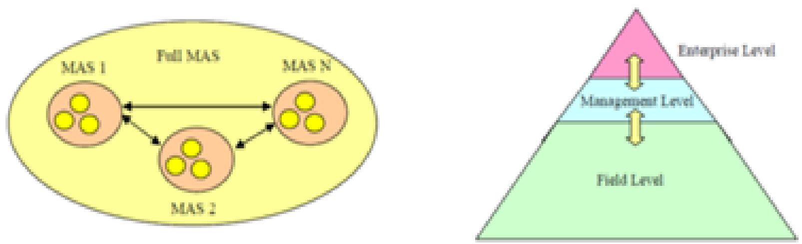


Figure 21: The organization and the levels of MASs [30]

2.3.7 Optimisation algorithm for a virtual power plant operation

In [9] it is mentioned that, optimisation algorithms for a virtual power plant might range from economic optimisation to technical optimisation like the reliability and the power quality of the VPP. Within [9] the VPP tries to integrate a large number of distributed generators of heat and power and provide energy in the most economical manner. The VPP depending on the variable costs (besides fuel and Operation and Maintenance costs) decides which type of generator should be dispatched. More specifically the VPP controls a number of heat and power generators as well as a number of CHP units. All of them have a cost function and the output of each function determines the cost of each source in $C.U./h$ (currency unit per hour). These cost functions are derived for each source in [9]. The objective function which is minimised is the sum of the cost function which is related to a heat source, the cost function of a power source, the cost function of a CHP unit and finally the cost function of the electric power exchanged with the grid.

A common assumption in [13] and [9] is that the VPP is able to take decisions on the basis of the generation costs of Distributed Generators.

2.3.8 A Stochastic Dynamic Programming Model for Optimal Use of Local Energy Resources in a Market Environment

The “Optimal Use of Local Energy Resources” in the title refers to the optimization of the operation of a micro grid. More analytically, as it is stated in [11]: “*This involves optimization of the production of the local microsources and storage and the exchange with the main distribution grid subject to market conditions.*” In other words, the aim is to optimize the operation schedule of the micro grid so that the microsources, the storage devices and controllable loads operate in a way that minimizes the cost of imported energy from the main grid. A stochastic approach of the problem is preferred due to the uncertainties during the operation of the microgrid.

The resulting scheduling problem comprises a plethora of functions that are related to the generation of power by the micro sources, the cost of supplying or not power to controllable loads, the cost of operating storage devices at a certain level and the cost of importing-exporting power from the main grid. Dynamic programming techniques are used to solve the scheduling problem (implemented in Matlab). Generally the problem looks very similar to the problem of optimally introducing a number of power stations in a power grid using dynamic programming but is more complicated due to the number of parts of a microgrid and the interaction with the main grid.

2.3.9 Wireless Sensor Networks in the Smart Power Grid

The main idea of the methodology presented in [14] is that Time Of Use (TOU) tariffs, namely promoting energy consumption in off-peak hours, together with Wireless Sensor Networks (WSN) can significantly reduce the amount of load in peak hours. An Energy Management Unit (EMU) creates an operation schedule for the flexible devices (the start time and the duration of operation) based on TOU and the availability of local energy sources. The WSN carry the information between the EMU and the devices. Further decrease of the load in peak-hours can be achieved when local energy resources and storage units are available in home which makes it adaptable to a Feed In Tariff (FIT) market. The results of the research done in [14] are represented in figure 22. In that figure, the top line shows that when no energy management occurs, then 30% of the demand of the appliances takes place during peak hours. When TOU and EMU are used for the same appliances, then only 10% of their demand is during peak hours. Finally when local energy sources and storage units are available in the home and TOU is implemented, then only 1% of the demand of the controlled appliances happens in peak hours.

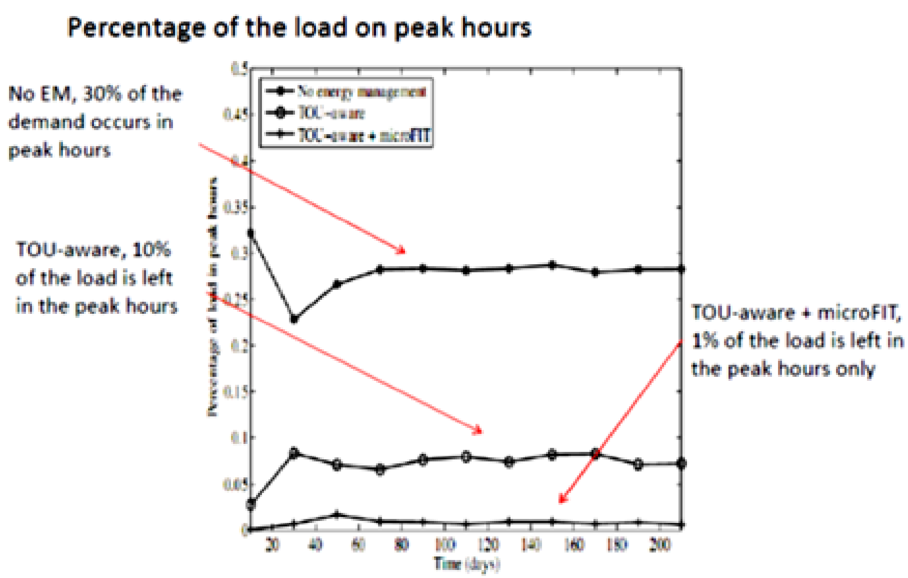


Figure 22: Simulation results of [14]. The peak load is reduced introducing TOU tariffs with WSN.

2.3.10 Coordinated micro-generation and load management for energy saving policies

In [5], a platform based on existing software and hardware tools is built to achieve energy savings by monitoring the load of a microgrid and by optimal management and control of the available energy resources.

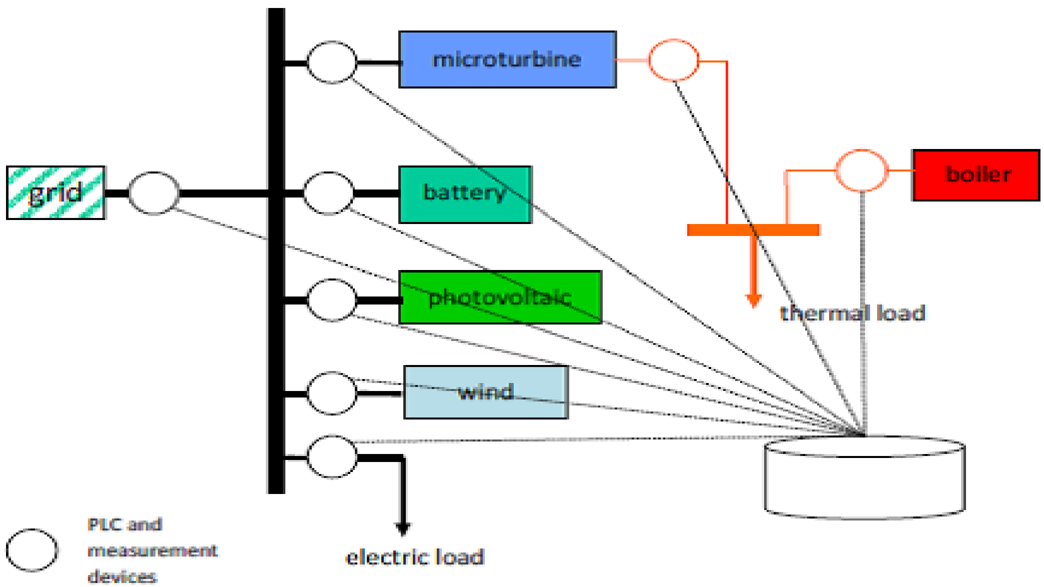


Figure 23: The concept of the proposed strategy in [5]

A local controller receives real-time measurements concerning both the (thermal and electric) load and the generation (figure 23). The controller, based on the data and existing software, takes decisions for the optimal generation and load management.

The monitoring of the distributed energy resources is implemented via the tool “Nextep”, a platform built out of three major components: a) sensors (meters), b) a data transmission network and c)an information architecture. The information architecture refers to a number of data loggers which collect information from the different energy sources and make them available to the users via the Internet.

For the forecasting of day ahead load the researchers used data of power consumption and min/max temperatures that were available for each day of the years 2006 and 2007. They used the data of every day of 2006 to predict the load of the corresponding day of 2007. Then they compared their predictions with the real data they had. They ended up with an average error of 9.72%.

The methodology was tested in three microgrids in Italy. During the tests different anomalies were detected and corresponding actions were taken to face them. For example a saw-tooth night consumption was detected in a site due to the air condition systems which were activated for 15 minutes and then switched of for the rest of the night. A new programme for the operation of air conditions was designed and this resulted in a more flattened night consumption curve and a decreased overall consumption.

Summary

This chapter focused mainly on the description of Triana and Powermatcher. Nevertheless, other energy control methodologies were briefly discussed. There is a large variety of such methodologies, each one of them following another strategy. Most systems use mathematical optimization techniques or agentbased programming (or both) to achieve specific goals regarding the aggregated energy consumption of a fleet of devices.

3 Flexible Power Application Infrastructure FPAI

This chapter contains an extensive description of FPAI. The purpose of FPAI is explained before going in more details regarding the different components and functions of FPAI.

To explain what the FlexiblePower Application Infrastructure (FPAI) is, we begin by giving the definition-goal of the FlexiblePower Alliance Network (FAN) which developed it. "*Flexiblepower Alliance Network (FAN) is an open industry alliance for the development and promotion of semantic (de facto) standards, with respect to communication of and communication with energy consuming and producing devices for end users. These standards will facilitate the emergence and use of energy services, on a uniform, accessible and cost -effective manner*" (FAN-General Documents) [24]. The aforementioned standards are the actual content of the FPAI. The official site of FAN (FlexiblePower.org) provides several documents in which all FPAI details are presented. In this chapter the most important aspects of FPAI are described.

3.1 The goal of Flexible Power Application Infrastructure (FPAI)

During the last years many different DSM methodologies have been developed. The software and hardware components of a system that implements such a method are usually referred together as an Energy Management System (EMS). Furthermore, devices use different communication protocols to send messages and data to controllers or other devices they need to communicate with. These two facts create an interoperability problem between the devices and the DSM methods. In an ideal case all DSM approaches should be able to handle all devices. This is the actual goal of FPAI (see also figure 24). "The FPAI aims to create an interoperable platform that is able to connect to a variety of appliances and support a host of SDM approaches" [24]. Hereby, SDM stands for Supply and Demand Matching, one kind of DSM whose purpose is to balance supply and demand. DSM describes a broader range of energy management techniques, a subset of them being SDM. The SDM term indicates that controlling the supply is also necessary to achieve a certain goal. Initially the FPAI was tested using the Powermatcher which is a SDM technique.

As mentioned above a DSM software accompanied by the hardware which is necessary for its execution is usually described as Energy Management System. Without the interoperable platform that FPAI provides, a user would have to buy new hardware every time she/he would like to change the current DSM system she/he uses with a new one. FPAI solves that problem since the substitution of a DSM by another would just require the software installation of the new one on the existing hardware and software platform.

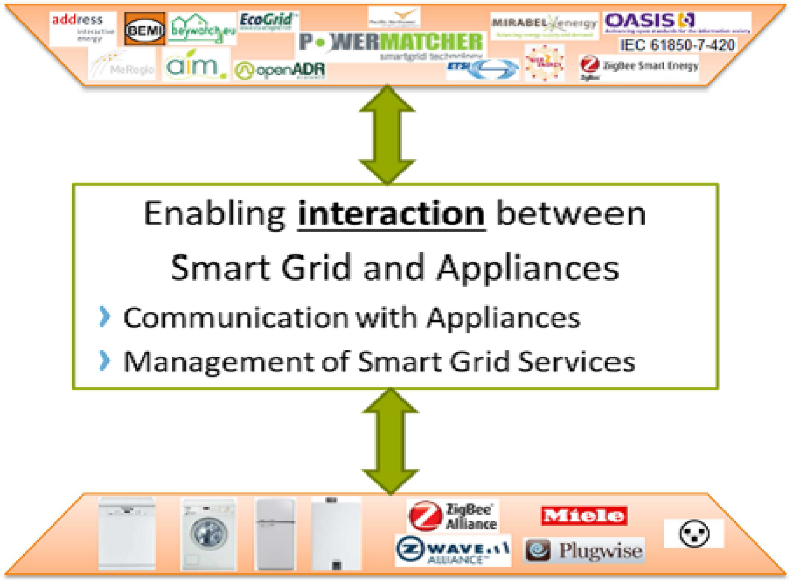


Figure 24: FPAI: creating an interoperable platform for a variety of appliances and EMS [23]

3.2 Presentation of FPAI

In this subsection the different FPAI components are described, focusing more on the components that are relevant to this assignment.

<u>Notation:</u> To make a distinction between the different terms of FPAI and other entities, all FPAI terms are written in Italics.

3.2.1 Introduction

FPAI has three major components:

1. The *FP HomeBox*
2. The *FP Management Center*
3. The *FP AppStore*

This assignment deals with aspects that are related to the *FP HomeBox* and more specifically to the *FP Runtime* component; the software installed on the *FP HomeBox* which in its turn refers to the hardware. *FP Node* is a term that is commonly used to denote *FP HomeBox* and *FP Runtime* as one entity. We initially give a description of *FP Runtime* and in the end of the chapter we give a brief presentation of the *FP Management Center* and *FP AppStore*.

3.2.2 The Resource Abstraction Layer (RAL) and Resource Abstraction Interface (RAI)

The three most important components of *FP Runtime* are 1) the *Energy Application* (e.g. a DSM system like Triana, Powermatcher), 2) the *Resource Abstraction Layer (RAL)* and the *Resource Abstraction Interface (RAI)*.

On its turn, the RAL consists of two different components specified for every device (a device is distinguished by the other devices via a unique identifier that is given): the *Resource Manager (RM)* and the *Resource Driver (RD)*. The *RM* is the connection link between the devices and an energy application. It receives data by the *RD* related to the operation of a device such as energy consumption, temperature etc. and also information regarding the preferences of the end users. These inputs are translated by the *RM* to the energetic flexibility represented by the *Control Space* of a device.

The *RAI* provides to an energy application the information of a device that specifies its energetic flexibility. *RAI* makes a distinction of devices in 5 categories according to the type of flexibility they can offer. After an energy application has received the flexibility of a device, it should analyse it according to its internal algorithm developed within the energy application and return an *Allocation*; the energy profile of a device as determined by the energy application. The *Control Space* of a device shows to the energy application what it can do with that device to achieve a certain goal and the *Allocation* gives information of how we finally exploit that *Control Space* to achieve that goal. Following the opposite direction, the *RM* receives the *Allocation* and is responsible to translate it into actions that should be applied by the RD. Figure 25 summarizes graphically the FPAI architecture.

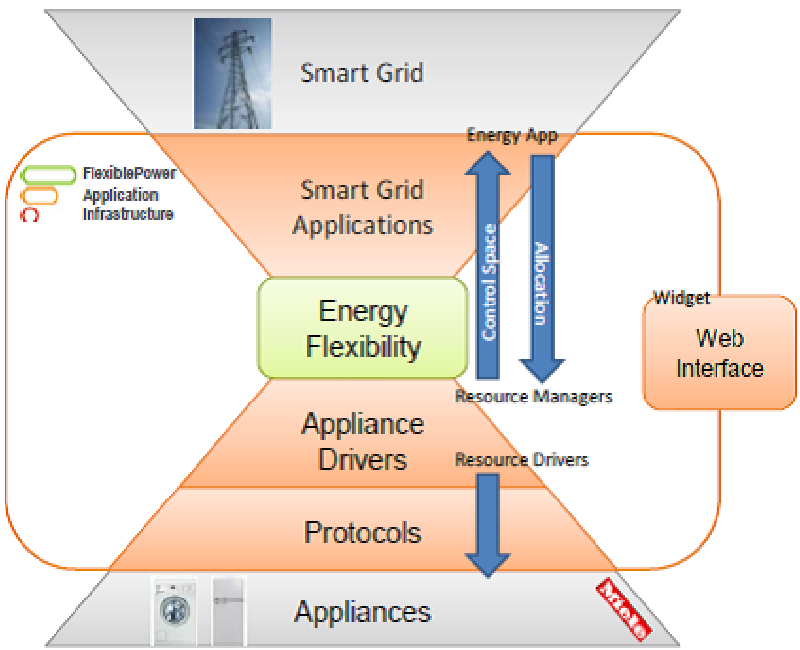


Figure 25: The FPAI architecture [23]

Next we present the attributes that describe the flexibility *Control Space* of every device type and the attributes that determine the *Allocation*. It should be mentioned that there is not yet any documentation describing the attributes of *Controlled* devices *Control Space*.

Control Space

From a software perspective, the *ControlSpace* is a super class for all device types ControlSpaces. The most important variables of the *ControlSpace*, common for all ControlSpaces are:

- *id*, *resourceId* and *resourceManagerId*: each *ControlSpace* can be recognized by a unique id number.

Similarly, the *resourceId* represents the appliance to which a *ControlSpace* refers to and *resourceManagerId* the *Resource Manager* that produced the *ControlSpace*.

- *timestamp*: the creation time of a *ControlSpace*.
- *validFrom* and *validThru*: each *ControlSpace* determines the flexibility of a device for a specific period of time. *validFrom* is the starting point of that period and *validThru* is the end point of that period.

The five different device types that lead to the corresponding Control Spaces are summarized in figure 26.

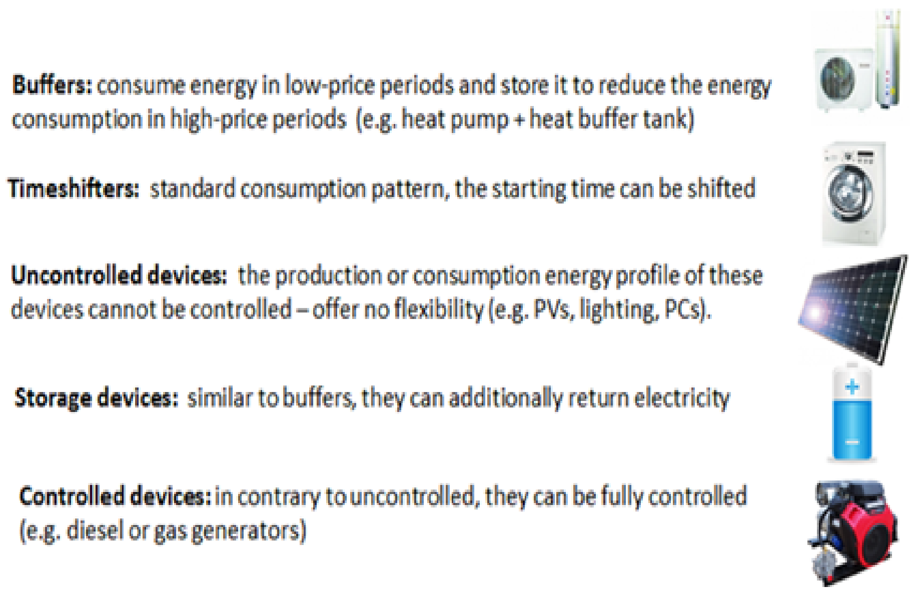


Figure 26: The five device types according to RAI

Buffer Control Space

Buffers can temporarily store an amount of energy during specific periods in order to use it in periods when consuming or storing energy is more expensive or generally less beneficial. Examples of Buffer devices are fridges, freezers and heat pumps accompanied by a heat buffer. The *RAI* of the *FPAI* uses a number of attributes to describe the energy flexibility of a buffer which are listed below.

- *totalCapacity*: it indicates the total amount of energy that a buffer can store or generate (in kWh in our problem but this can be translated to any energy unit using the Measurable Interface, see next chapter).
- *stateOfCharge*: this attribute shows how much of the *totalCapacity* of a buffer is currently used. It can range between 0 (empty buffer) and 1 (full buffer).
- *targetTime* and *targetStateOfCharge*: when a user wants to have a specific amount of energy (e.g. hot water) available at a specific time (*targetTime*) then the buffer should have a specific amount of energy stored (*targetStateOfCharge*) at that time.
- *selfDischarge*: denotes the losses a buffer has due to its finite insulation (in watts in our problem).
- *minOnPeriod* and *minOffPeriod*: if a buffer is turned on, it must operate for a period at least equal to *minOnPeriod* and once it is turned off it must remain turned off for a at least *minOffPeriod*.

- *chargeSpeed*: is the power that a heat pump consumes in order to charge a buffer. A number of single power steps as well as a range within the heat pump can operate is defined using the *PowerConstraintList* class (see section 4.3).

Timeshifter Control Space

Devices that belong to this group can shift their operation for a specific period of time. The *Timeshifter-ControlSpace* is defined by the next three properties:

- *energyProfile*: the energy profile of a device describes the energy consumption pattern of a device. It is a sequence of tuples visualized by a bar graph, each bar having a height representing the amount of energy and a width representing the time during which this amount of energy will be consumed or produced
- *startAfter*: the time after which the device can be started
- *startBefore*: the time before which the device must be started

Storage Control Space

Storage devices can store energy and offer it later to other devices. The difference with the Buffer devices is that they can offer back electricity to the grid. They share all characteristics of buffers and thus the *StorageControlSpace* class is a sub class of the *BufferControlSpace*. Additionally it contains the following variables:

- *dischargeSpeed*: the output power of a storage device is distinguished in a number of power levels. The *PowerConstraintList* is used again to define the different power outputs of a storage device.
- *chargeEfficiency*: represents the losses during the charging process; when charging a storage device a small percentage of the energy consumed is wasted and not used to raise the state of charge of a battery. A parameter that can decrease the charging efficiency is the high temperature of the storage medium.
- *dischargeEfficiency*: represents the losses during the discharge process; correspondingly to *charge-Efficiency* , when a storage medium supplies energy to a device, a small portion of the energy flown out of the storage medium is wasted.

Uncontrolled Control Space

These devices cannot change their schedule. Like in the *Time Shifters* there is an *energyProfile* given for these devices too, which is a prediction of the energy they are going to consume in a future time period. However, in contrary to *Time Shifters*, these devices dont have a *startAfter* and *startBefore* attribute, but only a *startTime* attribute which is an indicator for the starting time of the *energyProfile*.

An energy application uses the *Control Space* of a device to determine the energy profile of this device in order the goals of the energy application to be achieved. The returned energy profile is called *Allocation*.

Allocation

The *Allocation* is the output produced by an energy application and it describes how the flexibility of a device specified in the Control Space should be used. Two of the most important parameters that shape an *Allocation* are the:

- *energyProfile* (as described in the previous paragraph)
- *startTime*: defines the start time of the *energyProfile*

In the next sections the *FP Management Center* and the *FP Appstore* are briefly discussed. As can be seen from figure 27 the *FP HomeBox* interacts with them in a way that is later explained in detail.

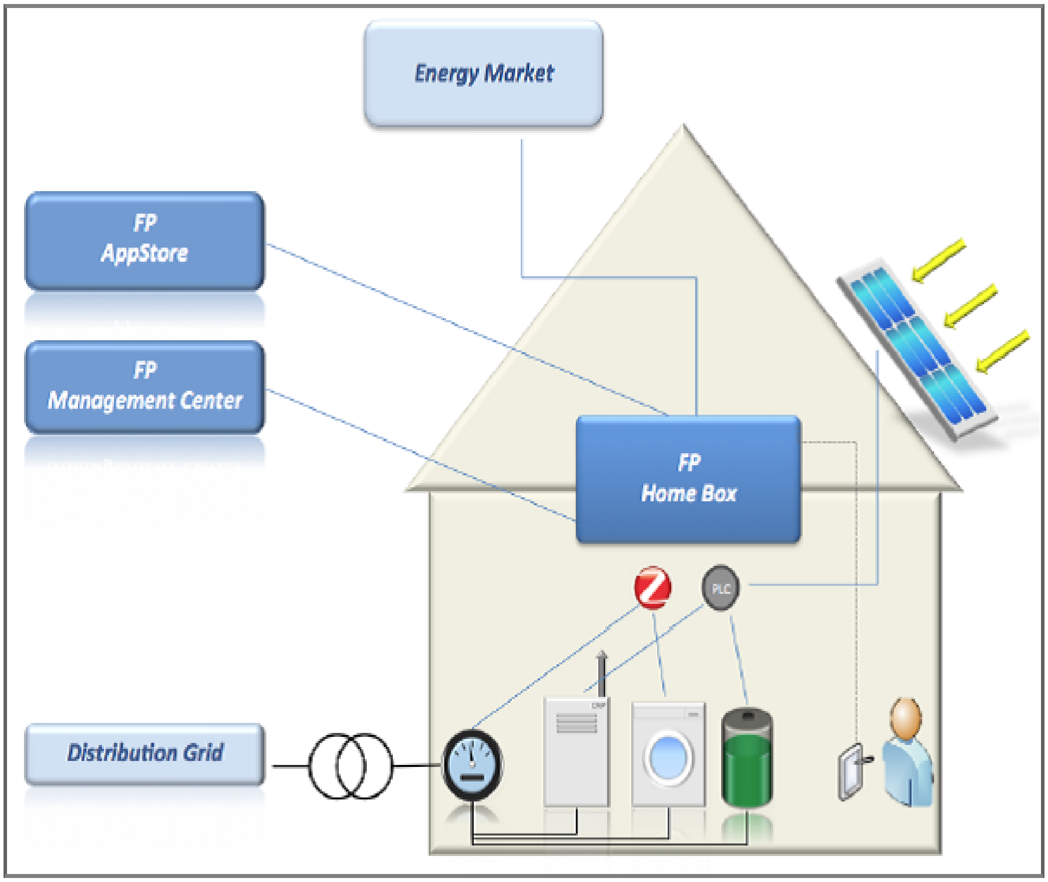


Figure 27: The basic FPAI components: Home Box, Management Center, Appstore [24]

3.3 FP Management Center

A *Flexible Power Network* is a collection of *FP Nodes*. Each *FP Node* communicates with the *FP Management Center* according to a client (*Node*) - server (*Management Center*) scheme. The latter is responsible for a number of tasks:

- Provide updates of the *FP Node* software (*FP Runtime*)
- Provide an authentication id to each Node which verifies it as a part of the *FP Network*
- Provide synchronization services to all *Nodes*
- Maintain a library where all software versions, plugins, protocols and support components are kept
- Monitor the health of the *FP Nodes*

- Keep all security information related to the safe operation of the *FP Network*.

The communication protocol that is used for the *Management Center Node* communication is TR-069. According to FPAI structure, a network could be partitioned in smaller networks each one of them having a *Management Center Operator*, responsible for the operation of its *Management Center*.

3.4 FP AppStore

The *FP AppStore* offers a series of applications that are available to every node. Like in the *Management Center*, there is an *AppStore* Operator responsible for the operation of the *AppStore*.

Within *AppStore* someone can make a separation of exclusive entities the most important of them being (see also figure 28):

- *Clientinfo*, collects information about the accounts of each Node who accesses the *AppStore*. This information is provided by the *Management Center*.
- *AppsCatalog*, contains all available applications.
- *ApprovalCenter*, approves (or not) an application before it is moved in the *AppsCatalog*.
- *Library Store*, consists of three application categories. *CandidateApps*, received from developers, *ApprovedApps* and *InactiveApps* which can be e.g. old *ApprovedApps* which are no more used by anyone.

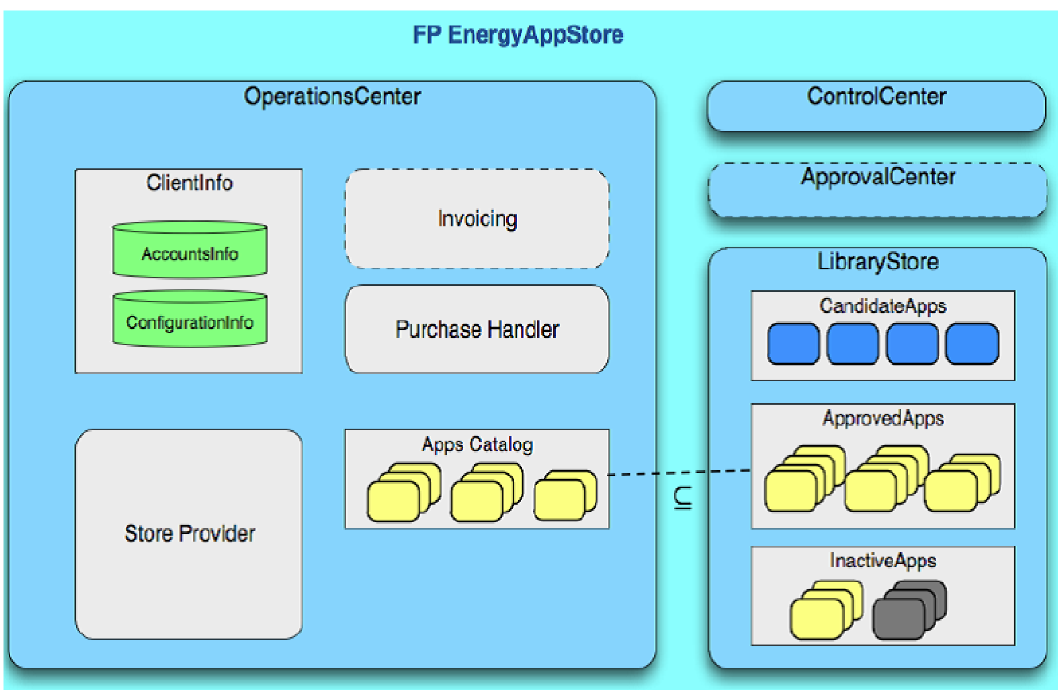


Figure 28: The FP Appstore [24]

A distinction between the different application groups can also be made in:

- *Energy Management Apps*: a user can choose the desired energy application

- *Resourcing Apps*: *ResourceManager* apps to represent a resource and *ResourceDriver* apps that provide device-specific information

- *Interaction Controller Apps*: provide functions for the Graphical User Interface. For example when purchasing a *ResourceDriver* app, a corresponding controller of the device is offered.

4 Implementing the Planning phase of Triana using FPAI

This chapter describes all steps that were followed to implement the Planning phase of Triana on top of FPAI. Firstly some details are given, related to the planning of the operation of a fleet of devices according to Triana. Further information is given to explain how Planning is achieved for a certain device type. Then, the modifications done within this assignment regarding the Planning of specific device types are presented, followed by a description of the incompatibilities between Triana and FPAI that had to be overcome. Finally some details regarding the structure of the developed software are given.

4.1 Triana Planning phase Algorithm description

4.1.1 The Planning phase of Triana

Planning is the second phase of Triana. The Planning process takes into account the forecasts done in the first phase and a global objective to determine the energy profile of each device for a given time horizon, by performing a cost minimization for the device operation. A common time horizon that is used for the planning phase is 24 hours split in intervals of 15 minutes.

Within this thesis the Planning phase of Triana built on FPAI, was implemented for three of the device types that are defined in Triana (see chapter 2):

- “Buffers” of RAI corresponding to the Buffered Converters of Triana
- “Time Shifters” of RAI corresponding to the Smart Appliances of Triana
- “Uncontrolled” devices of RAI corresponding to the Standard Appliances of Triana

In the following sections we describe the algorithm followed by Triana to implement the Planning phase for a fleet of devices and followed by details regarding the Planning for specific device types.

4.1.2 Planning for a fleet of devices

In section 2.1 (see also figure 7) it was shown that Triana applies its control methodology to a fleet of devices using a tree structure which is used to transfer information from the Global Planner (root) to single devices (leaves) and vice versa. The information sent by the Global planner is the electricity price and the information sent from the devices to the Global Planner is the energy profile of the device for a specific time horizon. In the next paragraphs of this section, the planning for a fleet of Buffer devices is presented, giving also information about the changes done within this assignment one of them being that electricity prices are now calculated by the so-called Triana controller; a local controller bound to a single device.

During the planning of the operation of a fleet of Buffers that covers the domestic heat demand of different houses, an initial price (equal for all devices) is defined in a method that is executed by the local controllers (*setInitialPriceValues* method) which are responsible for implementing the Planning for the device with which they are associated. This initial price and all prices in the algorithm are just artificial prices that represent the electricity price. The local controller executes its Planning algorithm taking into account the heat demand of the house and other parameters like e.g. the normalized state of charge of the heat buffer which must always stay between 0 (empty buffer) and 1 (full buffer). The energy profiles of the Buffers that result from the planning are aggregated by the Global Planner to observe in which extent the global objective is achieved. In our case the global objective is to reach a flat aggregated energy profile. In every iteration the final tasks of the global planner are to calculate the mismatch and also send a command to the local controllers to calculate a new energy price for the device they control. This process is repeated until we reach a convergence of the mismatch to a certain value. Then, the profile shaped for every device is the one that is returned to the RM who is responsible to translate into actions to be executed by the RD. The process is depicted in figure 29. An overview of these steps is given below.

1. Initiate-create Triana controllers (executed by the Global Planner). A Triana controller is set responsible to control the Buffer device of a house which has a specific heat demand.
2. Receive the Control Space of the Buffer and the heat demand (for Buffers) for the planning horizon. As soon as a Triana controller is initiated, it reads the Control Space of the device and the heat demand of a house accessing a data file.
3. Receive Initial Price values. Triana controllers are responsible for setting the initial price vector executing a method (*setInitialPriceValues*).
4. Triana controller executes Trianas local Planning algorithm.
5. Triana controller returns the energy profile of the controlled device.
6. Triana controller updates the price vector based on the aggregated energy profile calculated by the Global Planner using a price-revision technique.
7. Repeat 3-6 for a number of iterations (e.g. until the mismatch is adequately small).
8. Return an energy profile to the resource manager.

The main difference regarding the traditional Triana Planning and the current implementation is that the task for calculating the new price (which is different for every device after the 1^{st} iteration) is placed locally to the Triana controller of the device (step 6) and not to the Global Planner.

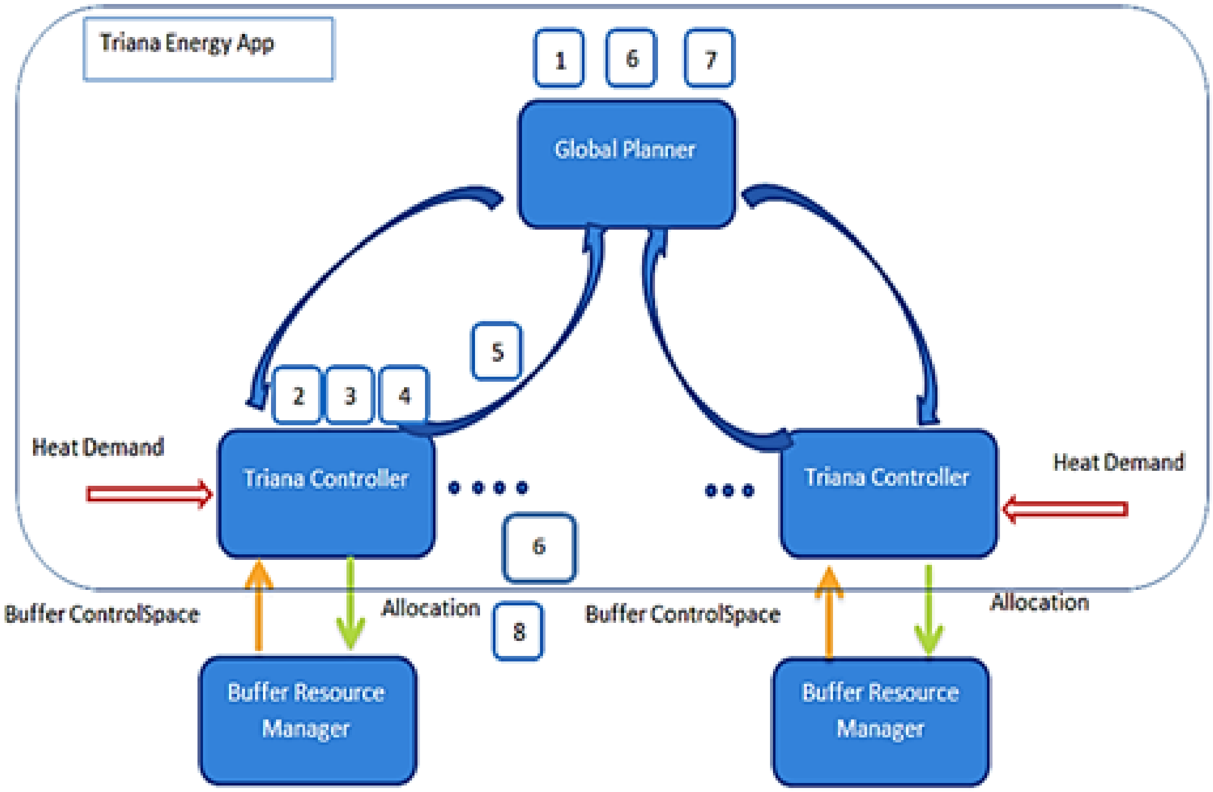


Figure 29: The algorithm steps labeled according to the previous description of the algorithm developed in this thesis

4.1.3 Buffer devices

An example of a Buffer device is a heat pump connected to a heat buffer. A heat pump is a device that uses electricity to produce heat and a heat buffer is a device that can store energy in the form of hot water. When the heat demand of a house is covered using a heat pump and a heat buffer, the flexibility is created by storing hot water in the buffer to be used in periods of high demand-electricity prices. Before giving more information about the planning of the other device types, we begin by giving more details about the mathematical techniques Triana uses to for Planning applied specifically to Buffer devices.

4.1.4 Dynamic Programming in Triana

The Planning phase of Triana uses Iterative Distributed Dynamic Programming (IDDP) as the optimization technique that leads to the achievement of the global goal. Dynamic Programming is extensively used in optimization problems and in [7] various forms of how Dynamic Programming can be used in energy control problems are presented. The next paragraphs show how Dynamic Programming is used to calculate the electricity price of a Buffer device in order to achieve a global goal for a set of devices.

In figure 30, every circle represents a State of a Buffer device. The State contains information about the SoC of the buffer, the energy consumed during the transition to this State from the previous, the total cost until that interval, the charge speeds of the buffer (heat pump power modes), the total capacity of the buffer and a reference to a State useful for the back tracing done in figure 30-c). Once we know the initial State of the Buffer we can calculate the potential States of the next time interval based on the number of the buffer charge speeds and by the SoC of the State (which must range between zero and one). During the transition of the State of a device from a specific time interval to the next one, the device consumes an amount of energy that determines the transition cost between two consecutive time intervals. The total energy cost at a time interval n is defined as the total cost till the time interval n-1, plus a cost related to the last transition and which is calculated by the next equation which has two terms: the first one represents the energy cost and the second is a quadratic term which imposes an artificial penalty to high consumption intervals.

$$cost = \underbrace{energy\_consumption(t) * energy\_price(t)}_{\text{energy cost}} + \underbrace{\beta * energy\_consumption(t)^2}_{\text{local cost}}, 0.1 \leq \beta \leq 0.5$$

Equation3: The total cost associated with the operation of a device for one interval

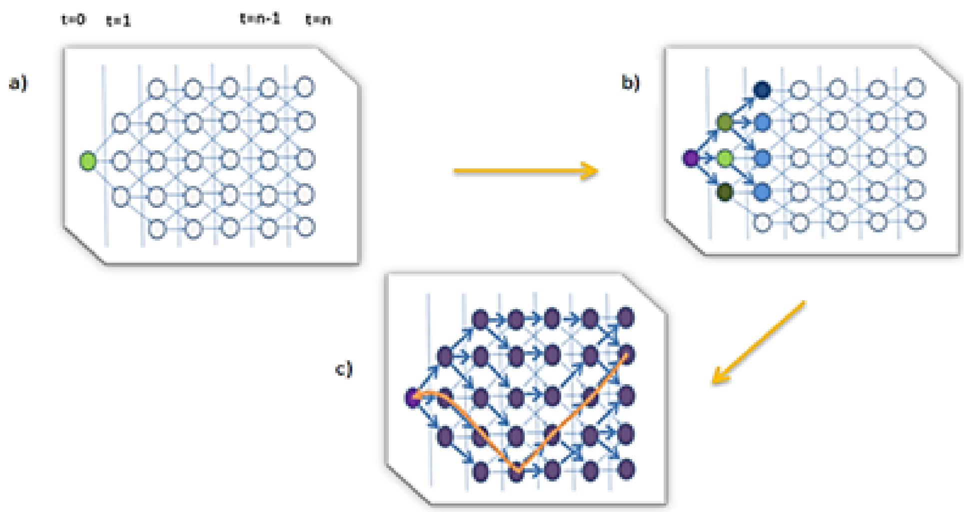


Figure 30: The Planning phase of Triana. a): The initial state of the device as defined by the Buffer Control Space. b): Determining the possible device states for all time intervals. c): Backtrace from the cheapest state of the last interval to its predecessors

Once, we have calculated all possible States of a Buffer device for all time intervals, the energy profile of the device is defined by the States that led to the minimum total cost in the last time interval.

4.1.5 Price-revising technique in Triana

The electricity price is calculated during every iteration of the algorithm for all devices. The price revising in Triana is based on the aggregated energy profile, the global objective (in our case a flat profile equal to the average consumption) and a probability that differs for every device. Since it is not realistic to achieve a totally flat aggregated energy profile, we use an upper and a lower bound used as criteria for revising the price of a device. Those boundaries have a distance of ± 10% with respect from the average consumption. If those boundaries are not included to the revision of the price vector, then the price tends to appear a strong oscillation. Namely to avoid numerical issues that occur when having very low price values during intervals of high consumption and very small values during intervals of low consumption. More specifically, after an aggregation of all profiles has been made by the global planner, each controller receives that profile and calculates the distance of the energy consumption value of every interval from the upper and lower bound. Then a random number for every interval is generated by the controller and if the distance mentioned is larger than the probability, then the price changes according to a price difference defined in the controller. The price difference is positive if the distance from the upper bound is positive or negative if the distance from the lower bound is negative. That procedure is followed for every time interval.

4.1.6 Local planning: a pseudocode representation

The next pseudocode snippet presents how Dynamic Programming is applied in the current implementation. This code presents the main task of a local Triana Controller bound to a Buffer device i.e. to determine the energy consumption of a device based on the global objective and the total energy cost during the operation of a device. The outcome of the process presented is an energy profile which contains the energy consumed during every interval. The first state of the Buffer device is explicitly defined by the Buffer Control Space that is provided by the RAI. Based on that Control Space the states of the next intervals are calculated. A different state is determined for every charge speed and it is considered valid only if the SoC is between 0 and 1. When the states of every time interval are calculated, we determine the state of the last interval that resulted in the minimum total cost. The final step includes a backtrace to the previous intervals to find all predecessor states that led to the state with the minimum total cost. The returned energy profile contains a value that shows the energy consumption during every interval. Note that this information is stored in the states.

Algorithm: *Dynamic Programming in local Triana Planning*
Input: *Heat Demand, Buffer Control Space (BCS), number of time intervals N*
Objective: *Find and return the energy profile that leads to minimum energy cost*

```
Local Planning
create energyProfile, Time Interval array //length N=number of time intervals
for i=0 to N-1 do
    if i=0
        first State = initialState(BCS)
        interval[i] ← first State
    else
        for State in interval[i].States
            State.nextStates //every State of an interval creates a number of successor States
            for all successors
                if successors is valid
                    interval[i].addState
                end if
            end for
        end for
    end if
end for
find endState //the State with the minimum total cost in the last interval
energyProfile←backTrace(endState)//find the end States predecessors and determine the energyProfile
return energyProfile //the energyProfile that leads to minimum total cost
end Local Planning
```

4.1.7 Time Shifters

The algorithm developed for planning the operation of Time Shifter devices differs in some points to the algorithm developed for the Buffers as the Control Space of these types differs. These differences are explained in this subsection.

As described in the previous chapter, the *Time Shifter Control Space* is determined by the attributes *startAfter*, *startBefore* and energyProfile. The first two variables are objects of the Java class *Date* which shows how many milliseconds have elapsed from the 1st January 1970 and define an exact date and time. The *energyProfile* consists of pairs of values one of them showing an amount of energy consumption and the other showing the duration in which that amount of energy gets consumed.

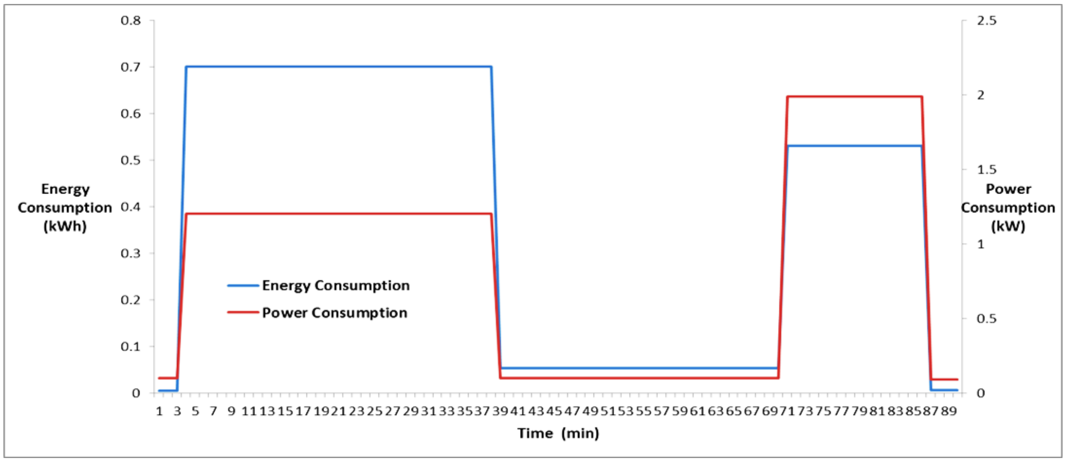
Therefore, those attributes have to be translated in variables that fit to the algorithm of Triana which uses the time interval notion to specify a timestamp during the planning horizon. Java provides a method that calculates the number of seconds (or hours or minutes) of a Date object that passed from the beginning of the current day. By dividing that number with the length of a time interval in seconds and by rounding that number, we can define how many intervals have passed from the beginning of the planning horizon. By doing that for the variables startAfter and startBefore, we define them as integer numbers that correspond to the number of the time interval they belong during the planning horizon.

Additionally, the energy profile as defined in *RAI* had to be translated in an energy profile which has an energy value for every interval. *RAI* contains a method that can split an energy profile to subprofiles. Each of them, is defined by an offset time (or the starting time of the sub profile) and the duration of the sub profile. In our case the duration is equal to a time interval. The offset is an integer multiple of the interval length. Another method of *RAI* is able to calculate the energy consumed during the sub profile, which is now the energy value of the interval.

In figure 31 we can see the form of a Time Shifter energy profile as defined in RAI and how it is split in sub profiles each one having a duration equal to the time interval. RAI defines a continuous (length-encoded) energy profile. This profile is defined by pairs of energy consumption and the duration in which this energy was consumed. For example during the minutes 4-38 (duration of 35 minutes) the Time Shifter consumed 0.7 kWh and this is defined by the command:

EnergyProfile.*create*().add(Measure.*valueOf*(35,NonSI.MINUTE),Measure.*valueOf*(0.7,NonSI.KWH))

During this conversion of a *RAI*-type energy profile to a Triana-type energy profile, a control is made that ensures that the operation of a Time Shifter does not end after the end of the planning horizon. That control is done using a variable representing the time interval that corresponds to the starting time of the operation. If the operation time (also calculated in intervals) added to the starting time interval exceeds the planning horizon, then a Triana energy profile cannot be determined.



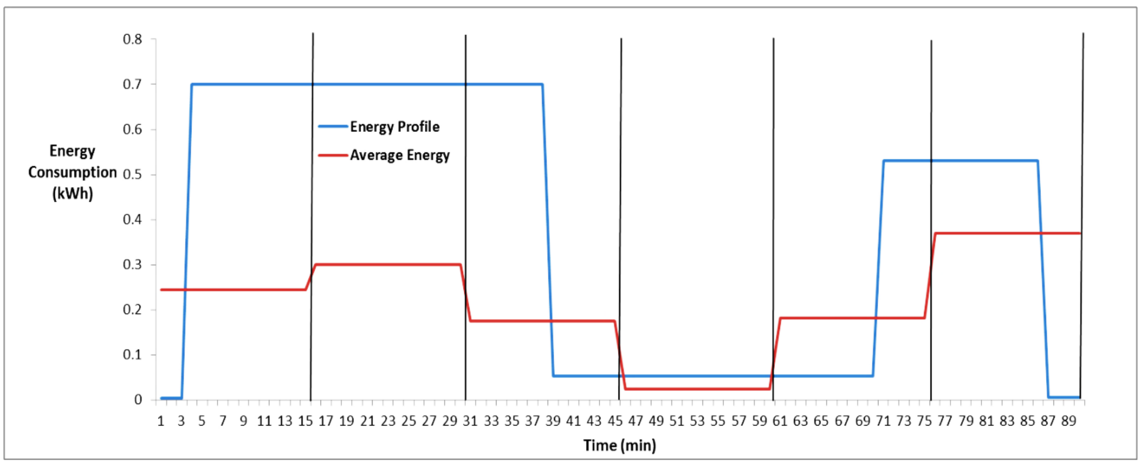


Figure 31: The energy profile of a time shifter as defined by RAI (above) and the energy profile split in discrete subprofiles with a duration equal to the time interval (15 min)

During Planning, for a given price profile and a time shifter profile we calculate where the energy profile should be placed during the planning horizon in order to get the minimum cost. In other words, given a price profile for the planning horizon we define the energy cost for all possible starting times. The starting time that reslts in the minimum cost determines when the Timeshifter will begin to operate.

In the figure 32, three possible positions of a Time Shifter profile (in our case it was a hypothetical profile of a washing machine) are shown. The profile has a duration of 90 minutes that corresponds to 6 intervals. Only case 2) shows an acceptable position. Case 1) has a starting time before startAfter. Cases 3) ending time slightly exceeds the planning horizon (begins at time interval 92). Case 2) represents a feasible starting time and also represents the optimum solution regarding the energy cost as the price vector has the minimum price over the operation intervals.

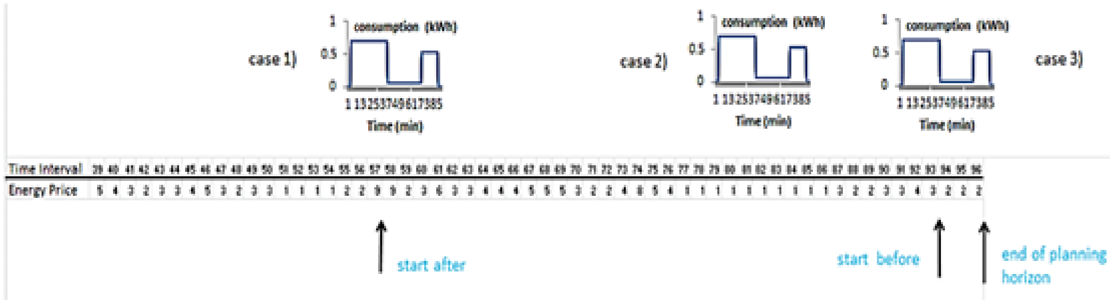


Figure 32: Placing an energy profile in the optimum position of the planning horizon

4.1.8 Uncontrolled devices

The implementation of Planning phase for Uncontrolled devices is simpler compared to Buffer and Time Shifter devices.

The energy profile that belongs to an uncontrolled device is filled with energy values that correspond to the production or consumption of such a device. No alterations can be done to the profile of an uncontrolled device. Its planning process returns to the Global Planner the energy profile that is shaped as described. In simulations executed in this assignment, an installation of PVs is represented as an uncontrolled device. The production values during a winter day were used to define its profile.

Finally it should be mentioned that the price-revising scheme used for Time Shifters is the same as for the Buffer devices, whereas the price-revising is not applied for Uncontrolled devices as there is no optimum cost solution that can be found for them. However they influence indirectly the price for the other devices with their contribution to the aggregated profile.

4.2 Incompatibilities between Triana and FPAI

To implement Triana using the FPAI there are several incompatibilities that have to be overcome.

Firstly we need to explain why Triana cannot directly be run on top of FPAI a fact that makes the bridging between Triana and FPAI more complex. Trianas planning phase uses parameters that are calculated during the first step of forecasting. FPAI does not yet contain a forecasting step that can provide such functions and thus certain data were passed to Triana by using data files for e.g. the heat demand, necessary to implement the planning step for the buffer devices. Another example is the startAfter and startBefore notions that were described in the previous section that were also passed manually and correspond to the information that would be given by the Resource Manager to a local Triana Controller.

Another incompatibility is the form the Time Shifter Control Space has, as provided by FPAI. The Time Shifter Control Space sends an energy profile that is continuous in time. Triana needs to receive an energy profile consisting of values that show the energy consumption of each interval. This problem was solved using a method provided by FPAI which can split an energy profile to sub profiles as explained in subsection 4.1.7.. Furthermore, FPAI does not take into account the end of the planning horizon and thus the operation of a device could take place outside the planning horizon which is unacceptable for Triana. The control explained in figure 32, was the solution to that problem.

One more difference is the form of an Allocation and the form of an energy profile that is provided by Triana as input. The difference is that Triana uses discrete time intervals (of the same duration) for each of those intervals an energy value is defined that shows the energy consumed during the interval. In contrary, the Allocation contains an *energyProfile*, defined by a list of energy/duration tuples where the duration is not equal for all elements of the list. Thus, in our case the Allocation consists of an array with size equal to the time intervals of the planning horizon and a value showing the energy consumption in every interval.

Additionally, an incompatibility between FPAI and Triana is that FPAI uses in its energy calculations only electricity properties. Thus all Triana attributes that expressed energy in the form of heat had to be converted in electricity. For example although Triana uses the Coefficient of Performance (COP) of a heat pump to calculate the SoC of a heat buffer based on the heat stored in it and the heat demand, in the current implementation the COP was not used and instead we took into account the COP to convert the heat demand of a house to electricity demand. Auxiliary heat devices were also not taken into account. Auxiliary devices are used in Triana to cover the heat demands in the extreme case where the heat stored in the buffer and the energy consumed by the heat pump are not enough to the heat demand during one interval. If auxiliary devices are used, an extra penalty is posed to the total energy cost.

Another incompatibility that should be taken into account when thinking for a full implementation of Triana (all phases) on top of FPAI is when a new Planning process should take place. Currently Triana uses data from the Forecasting phase to conduct the Planning for the next 24 hours with a rolling horizon of six hours. Nevertheless, the input which is now the Control Space can be changed in an arbitrary moment as a new Control Space is generated as soon as a new measurement is done regarding one of the parameters that determine the flexibility of a device, e.g. the state of charge of a heat buffer or when a new program for a Time Shifter is entered.

Finally it should be mentioned that Triana is written in the software language C++ using the Qt platform, whereas FPAI is written in Java programming language. Thus, any core code of Triana that had to be used has to be rewritten in Java as well as all other code developed to added extra functionality to the Triana code so that Triana can use as input data or methods provided by the FPAI.

4.3 Software tools provided by the FPAI

In this section some of the software tools that FPAI provides are described and their usage during the implementation of Trianas Planning phase.

FPAI uses a tool that makes calculations that include measurable properties more convenient - Measurable interface. With Measurable we can express any property in a SI or a non SI unit and using any multiplication factor. For example, in order to define that the capacity of a heat buffer is 10 kWh we use the command:

$$\mathit{ConstraintList}<\text{Power}>\ \text{cl} = \mathit{ConstraintList.create}(\text{SI.WATT}).\mathit{addSingle}(1000)\ .\mathit{addRange}(2000, 3000).\text{build};$$

For specifying all possible charge speeds of a heat buffer a class ConstraintList has been developed. It can define a series of single values or a range between which a heat pump could operate. For example the following constraint list represents a device which can consume 1000 watts and any amount between 2000 and 3000 watts.

$$\mathit{ConstraintList}<\text{Power}>\ cl = \mathit{ConstraintList.create}(\text{SI.WATT}).\mathit{addSingle}(1000)\ .\mathit{addRange}(2000, 3000).\text{build};$$

The sub profiles of the Time Shifters use the method:

subprofile (Measurable<Duration> offset, Measurable<Duration> duration)

where *offset* is in our case an integer multiple of a time interval and *duration* has the value of the length of a time interval.

4.4 The Class diagram of the code developed

In the appendix A of this document the Class diagram of the code developed is given in the form of a Unified Model Language (UML) diagram. A few remarks are put in this section to justify this specific structure of the Class diagram.

4.4.1 The Timeseries super class and its sub-classes

The Timeseries super class uses the planning horizon and the time interval length to calculate the number of time intervals of the planning horizon. Then, an array with size equal to the number of intervals is defined. The sub classes *EnergyProfile*, *PriceProfile* and *HeatDemand*, need such an array to store a value for each time interval. For example an EnergyProfile object stores a value in every element of the array and these values represent the consumption (or production) of a device. The aggregated energy profile is also an object of the same type holding in each array element the aggregated energy consumption of an interval. Similarly a PriceProfile object needs such an array to store the energy price values of each interval and a HeatDemand object the heat demand during each interval. In the Timeseries class two methods are written that are used to read from and write to CSV files in order to read necessary data for our algorithm (e.g. the heat demand profiles of the houses) or to extract data from the simulations run.

4.4.2 The State and TimeInterval classes

These two classes are only used in the planning implementation for Buffer devices. As mentioned in the previous section a State object contains all information that specifies the characteristics of the heat buffer (SoC, buffer capacity, charge speeds) but also methods and variables used to calculate the energy consumption during a transition from a State to another and the corresponding cost of this transition. Each TimeInterval object is composed of a number of States. A method makes sure that when two States have the same SoC, only the one with the minimum cost is stored. A TimeInterval is also responsible for finding the State of a time interval with the minimum total cost. This is used during the planning to find the cheapest State of the last time interval. That is the state we use, to find its predecessors that finally determine the optimal energy profile (this energy profile results in the minimum total cost). The usage of these two classes makes it easier to split different tasks, which are necessary to determine the desired energy profile, to more than one components-objects thus improving the structure and readability of the code and making it also easier to add extensions to it.

4.4.3 The TrianaController Interface, its implementations and the GlobalPlanner

The TrianaController interface contains three methods. One of them is the execution of planning for a single device and all classes that implement the interface have a different version of this method as described in the previous sections of this chapter. The two other methods are only implemented for the Buffer and the Time Shifter Controller. These two methods are responsible for creating an initial price profile and to updating the price profile for the next iteration.

The TrianaController implementations receive the Control Space of a device typically by the Resource Manager and execute the Triana algorithm using that information in order to return an Allocation the energy profile of the device. The Resource Manager is the connecting link between the device and an energy application. Similarly to the BufferTrianaController, in Powermatchers implementation a Buffer Agent is responsible for exploiting the Control Space of a buffer device.

The GlobalPlanner has a list of TrianaControllers. The first action of the GlobalPlanner is to define what kind and how many devices it will have under control defining the same time the data that need to be read by the local controllers. For example the Triana Controllers of Buffer devices need to read the heat demand data of a house. PV Triana Controllers need to read the data that give the PV production during the planning horizon. When the process of Planning for all devices begins, the Buffer and Time Shifter devices receive an initial price and then they execute their planning algorithms. Finally the Global Planner calculates the total mismatch and gives a command to the Triana Controllers to update their price profiles given the aggregated energy profile.

Conclusions

During the implementation of Trianas Planning phase, a part of the core functionality of the current Triana implementation was re-implemented in a way that the communication with FPAI could be made possible. Furthermore, several changes and additions were also necessary especially for the implementation of the Planning phase of Triana regarding the Time Shifter devices (section 4.1.5). Other incompatibilities also included the data type that normally Triana uses and the data type as provided by FPAI. The flexibility of a device is now revealed by its Control Space instead of the energy parameters that are normally provided by the Forecasting step of Triana. Triana is now able to handle a Control Space and execute the Planning phase for a fleet of devices.

The simulations of the next chapter prove that the implementation developed, produces the results that are normally expected when the Planning of Triana is executed.

5 Simulations Results

In subsections 2.1.4 a series of simulations conducted within [4] were presented. Those simulations give an idea of what could be achieved through the control of a fleet of devices using Triana. A typical goal that is set when using Triana to control a number of devices is to achieve an aggregated energy consumption profile that is flatter compared to the case when no control is used. In this chapter we present a series of simulation that were executed to test the software developed within this thesis which implements the Planning phase of Triana using the FPAI.

5.1 General information about the simulations

The target during the simulations conducted in this thesis was to reach a flat profile regarding the energy consumption of the devices controlled over a planning horizon. The flat profile means that aggregated energy consumption of all devices is steady over the planning horizon and has a value of E_{av}, where E_{av} is the average energy consumption of all devices.

As we normally will not be able to reach this profile, we aim at minimizing the mismatch from this profile. The mismatch can be calculated using the next equation of squared mismatch that measures how much the actual profile deviates from the average consumption.

$$SquaredMismatch = \sum_{i=1}^{N}(E(i) - E_{av})$$

where

- N is the number of time intervals of the planning horizon,
- $E(i)$ is the aggregated energy consumption of all devices at interval i,
- E_{av} is the average energy consumption of the aggregated profile (a targeted-flat profile)

An alternative measure for the mismatch could be the primary mismatch calculated by:

$$SquaredMismatch = \sum_{i=1}^{N} M(i), \quad M(i) = \begin{cases} 0.9E_{av} - E(i) if E(i) < 0.9E_{av} \\ E(i) - 1.1E_{av} if E(i) > 1.1E_{av} \\ 0 \end{cases}$$

In that case we calculate how much the actual profile deviates from and upper bound which is $1.1E_{av}$and a lower bound which is $0.9E_{av}$. In this assignment the mismatch was calculated using the formula of the squared mismatch.

In figure 33 we show a real consumption profile (blue line) and the targeted profile which is the average consumption (red line). We also show the upper and lower boundaries which are the references for measuring the primary mismatch.

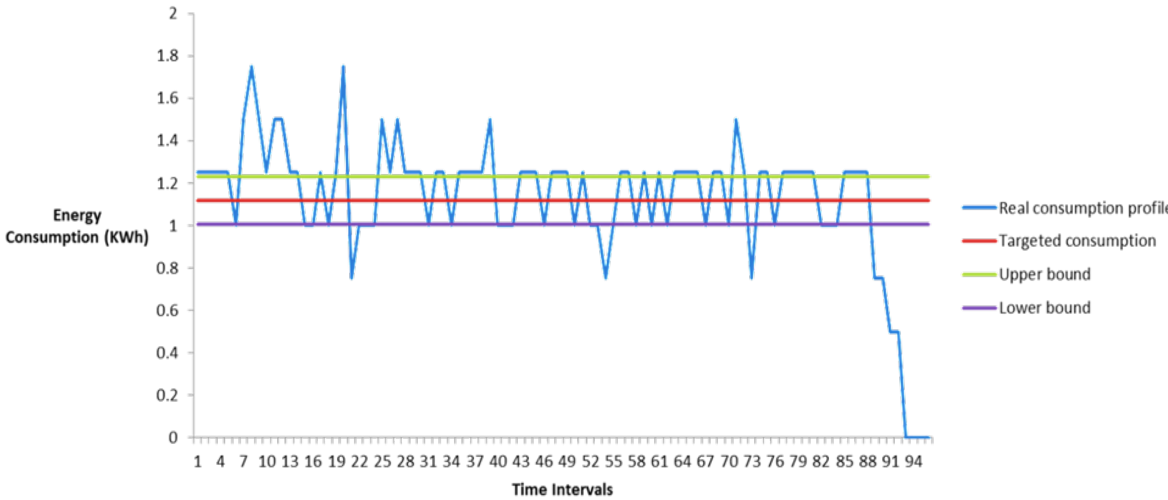


Figure 33: The real energy profile, the targeted-average consumption and the upper and lower bounds

In the following pages the graphs from a series of simulations are presented and explained. First we present simulations from the planning of a number of heat pumps accompanied by a heat buffer (Buffer devices). Simulations are run for different buffer sizes and initial state of charge. Then, simulations are done for houses that are equipped with heat pumps and a washing machine (Time Shifter devices). Finally we see the behavior of Triana when in the whole system we introduce the energy production from PVs. All simulations included 30 iterations of the algorithm. The planning horizon is 24 hours and the time interval length is 15 minutes, resulting in 96 intervals during the planning horizon.

5.2 Simulations

5.2.1 Planning for Buffer devices, Simulation 1

In this first simulation, we control the operation of twenty heat pumps using Triana. Each of the heat pumps is responsible for providing the energy required to cover the heat demand of one house. The heat pumps are connected to a heat buffer whose size is 10 kWh_el (30 kWh_th, COP=3). The heat pumps can consume one to five kW with a step of one kW. The initial SoC of the buffers is zero.

Real data that represent the heat demand of a Dutch household for a winter day (collected in 2007) were used. The heat demand of each house has a profile with high peaks at certain hours of the day and zero demand during other periods (figure 34). That would lead to a corresponding electricity profile if heat pumps consumed energy according to the heat demand. In figure 34 the heat demand of one house is shown. The abscissa of every point is the time interval and the ordinate the energy consumed during that interval.

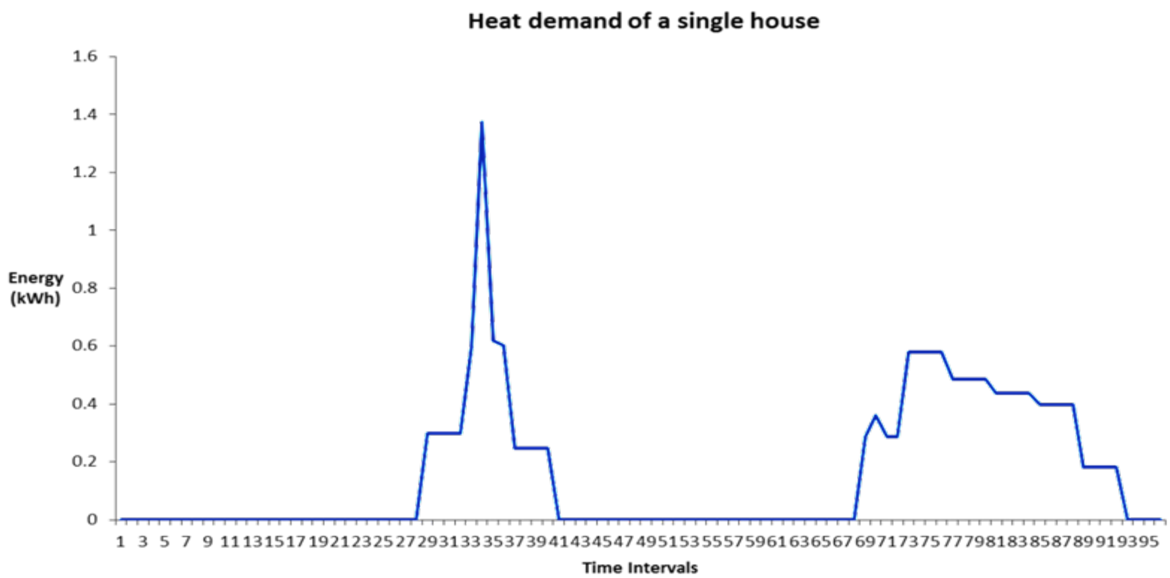


Figure 34: The heat demand of a single house

Figure 35 shows the resulting energy consumption profile of the 20 heat pumps when the algorithm developed is executed for two iterations. The energy profile presents very important oscillations regarding the energy consumption through the planning horizon. The aggregated profile is improved after the 5th algorithm iteration but the energy profile has still a wave shape with significant periods of peak consumption (figure 36). The energy profile is much closer to the targeted profile after the 30th iteration as can be seen in figure 37. The red line in every graph represents the average consumption which is our target profile.

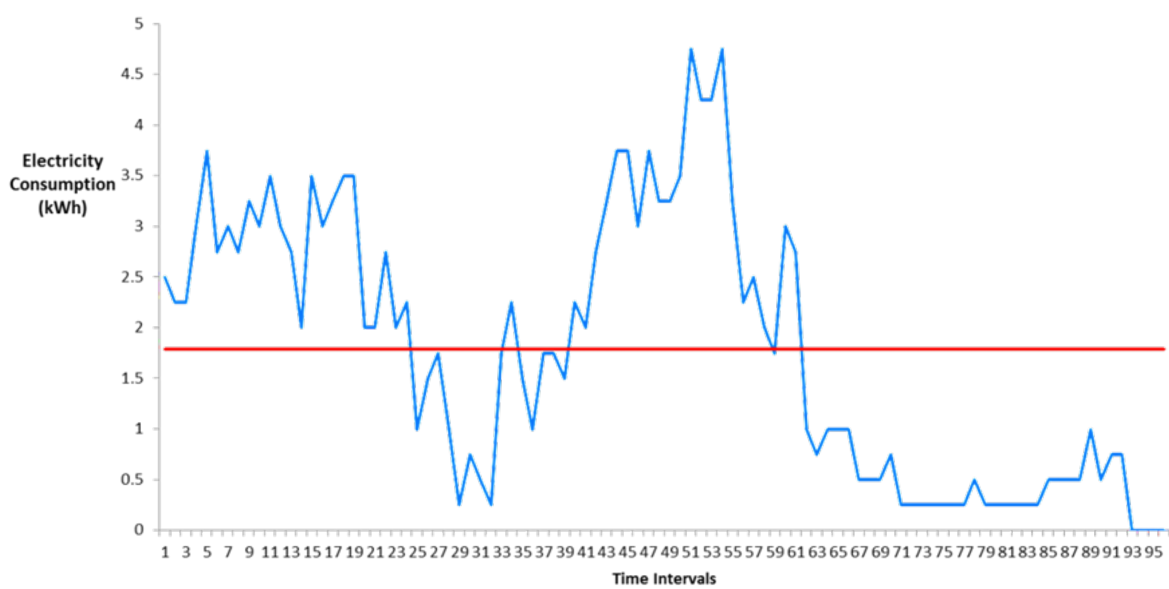


Figure 35: Initial SoC=0, Buffer Size=10 kWh_el, 2 iterations

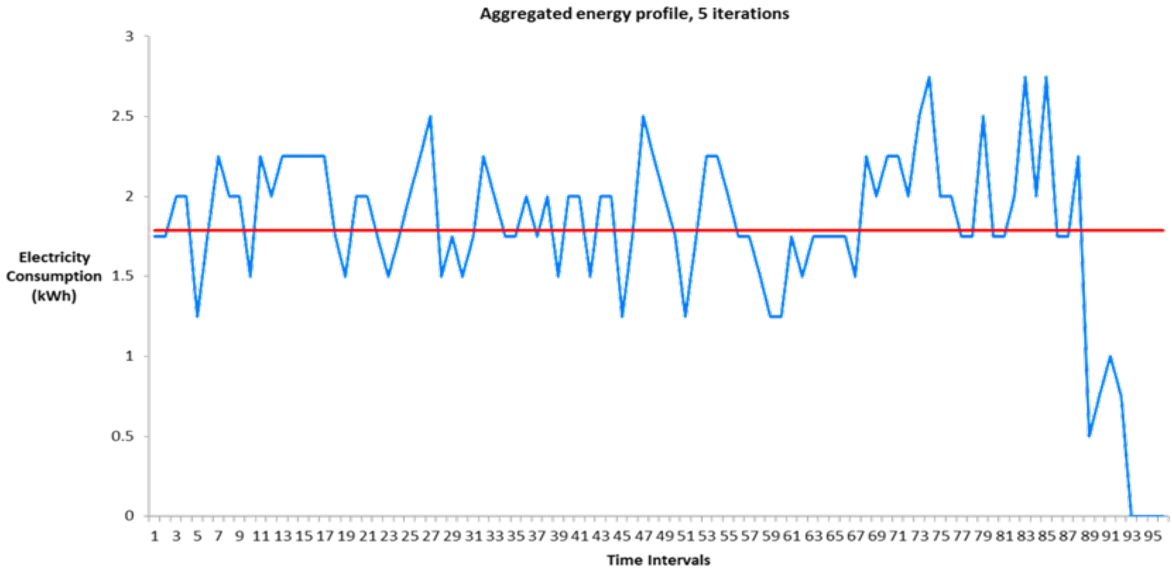


Figure 36: Initial SoC=0, Buffer Size=10 kWh_el, 5 iterations

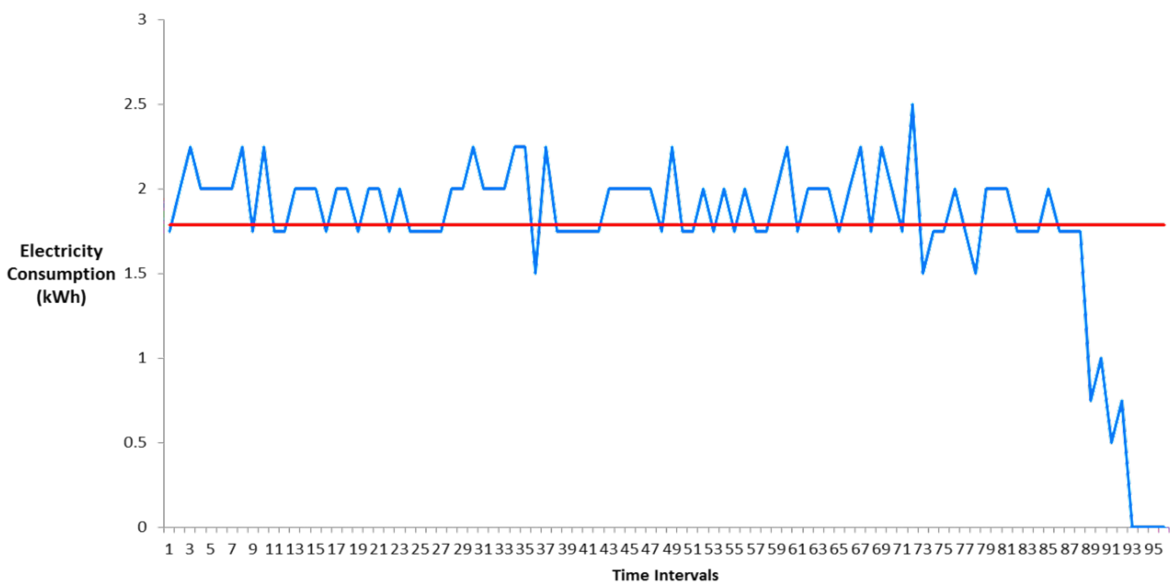


Figure 37: Initial SoC=0, Buffer Size=10 kWh_el, 30 iterations

The three profiles have a decreasing squared mismatch as the number of iterations increase; an expected result as it shows that the algorithm tries to achieve the global objective which is reaching an aggregated energy profile as close as possible to the average energy consumption. The graphs of the 2nd and 5th iteration have significantly higher peak consumption periods compared to the graph which shows the aggregated energy profile after the 30th iteration. The squared mismatch converges to a value close to 22 already after the 10th iteration of the algorithm as can be seen from the next graph (figure 38).

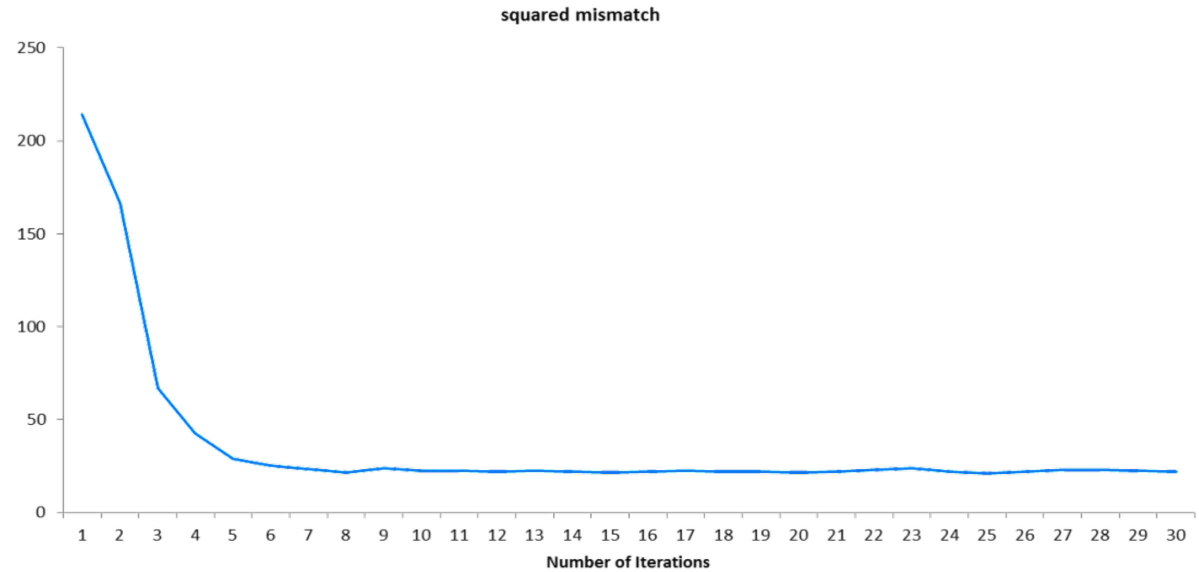


Figure 38: Initial SoC=0, Buffer Size=10 kWh_el, squared mismatch

Table 3 summarizes the results of the first simulation regarding the mismatch from the targeted profile.

Iteration number	Squared Mismatch M_2	Average Energy E_{av} (kWh)
2	166.416	1.79
5	28.79	1.79
30	22.04	1.79

Table 3: The squared mismatch in different iterations

The same conclusion can also be derived from the next graph which shows the aggregated energy profile after 2, 5, 10, 20, 30 iterations. The high peaks after the 2^{nd} (dark blue) and 5^{th} (red) iteration are eliminated after the 10^{th} iteration (figure 39).

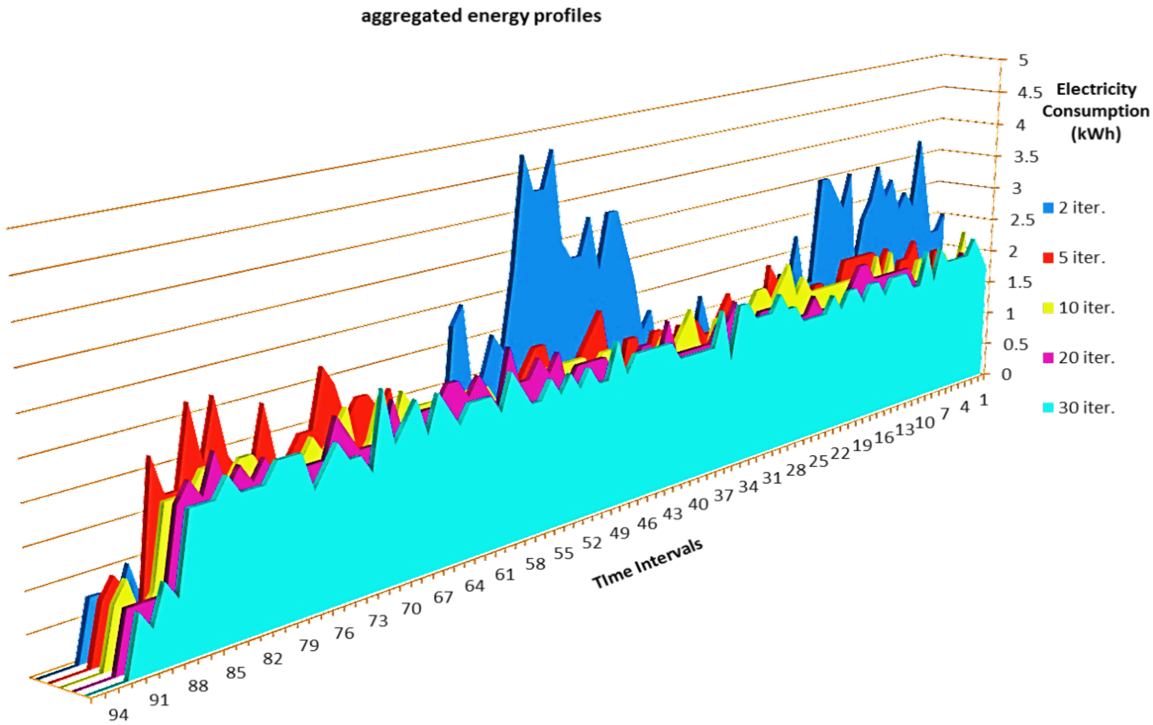


Figure 39: Initial SoC=0, Buffer Size=10 kWh_el, aggregated energy profiles

A common characteristic for all graphs is that the energy consumption has a steep reduction during the last 5-10 time intervals (except for the case of the 2nd iteration when the algorithm has not influenced significantly the whole consumption). A simple explanation would be that the energy price signal is very high during those intervals and that this cost increase reflects to the energy consumption. Nevertheless that is not true. On the contrary, the energy cost during those intervals is respectively much lower compared to the energy price of the previous intervals. However the stored energy of the heat buffer is enough to cover the heat demand and thus the heat pumps do not need to consume further energy for the last intervals.

As an example the next graph (figure 40) shows the aggregated electricity consumption (primary axis) and the aggregated electricity price which is the sum of al price signals over all devices (secondary axis). The reduction in the price during the last intervals does not result to an increasing energy consumption (as would be expected) because the heat pumps are not turned on during these intervals as the heat demand can be covered by the heat stored in the heat buffers. Thus in the last intervals the electricity price and consumption appear a parallel reduction. Figure 41 shows the average price compared to the price of one controller and the aggregated price of all controllers.

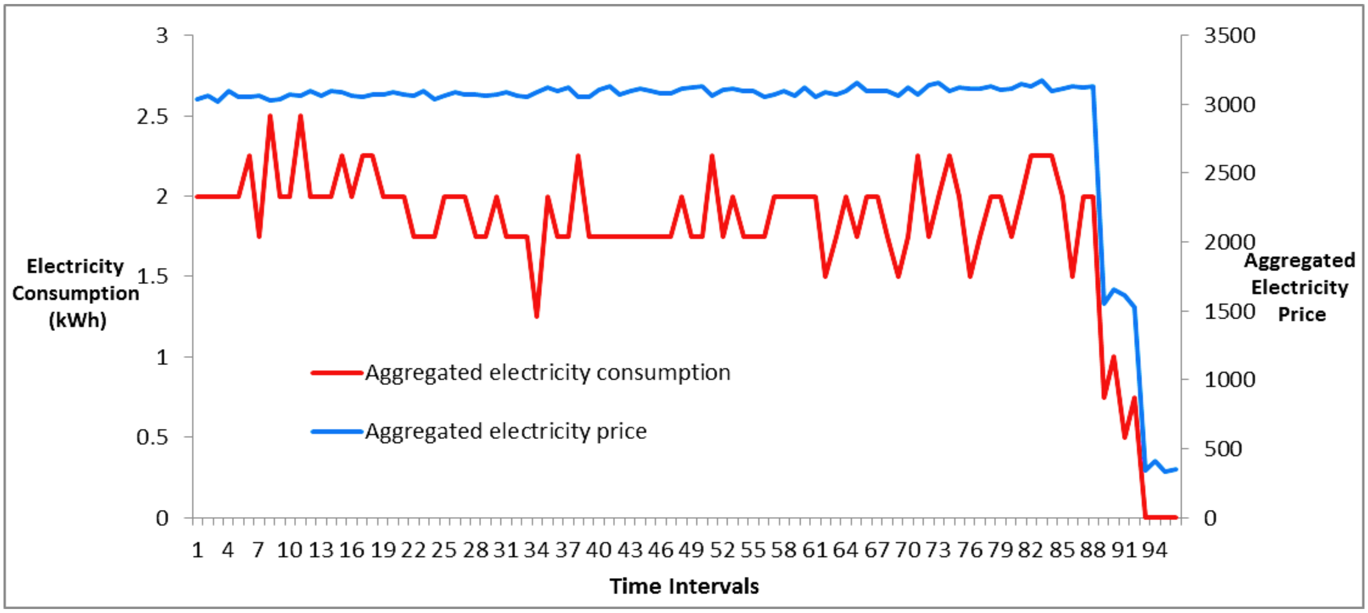


Figure 40: Aggregated energy consumption and the aggregated electricity price

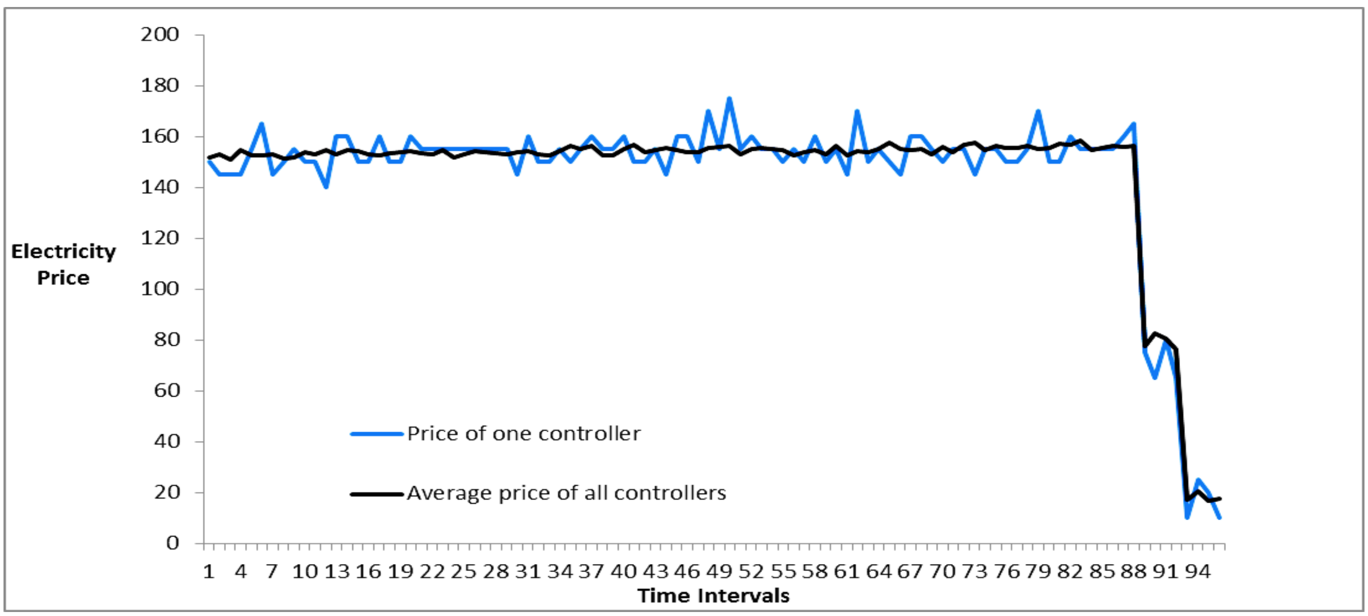


Figure 41: Price of one controller and average price of all controllers

5.2.2 Planning for Buffer devices, Simulation 2: influence of the initial State of Charge

The input of simulation 1 was only changed in the initial state of charge of the buffers which was set to 0.5. The influence of the initial state of charge of the buffers is examined regarding the average energy consumption and the squared mismatch during the execution of the Planning phase of Triana applied to 20 heat pumps.

Figure 42 and figure 43 show the aggregated energy profile after the 2^{nd} and 30^{th} iterations respectively.

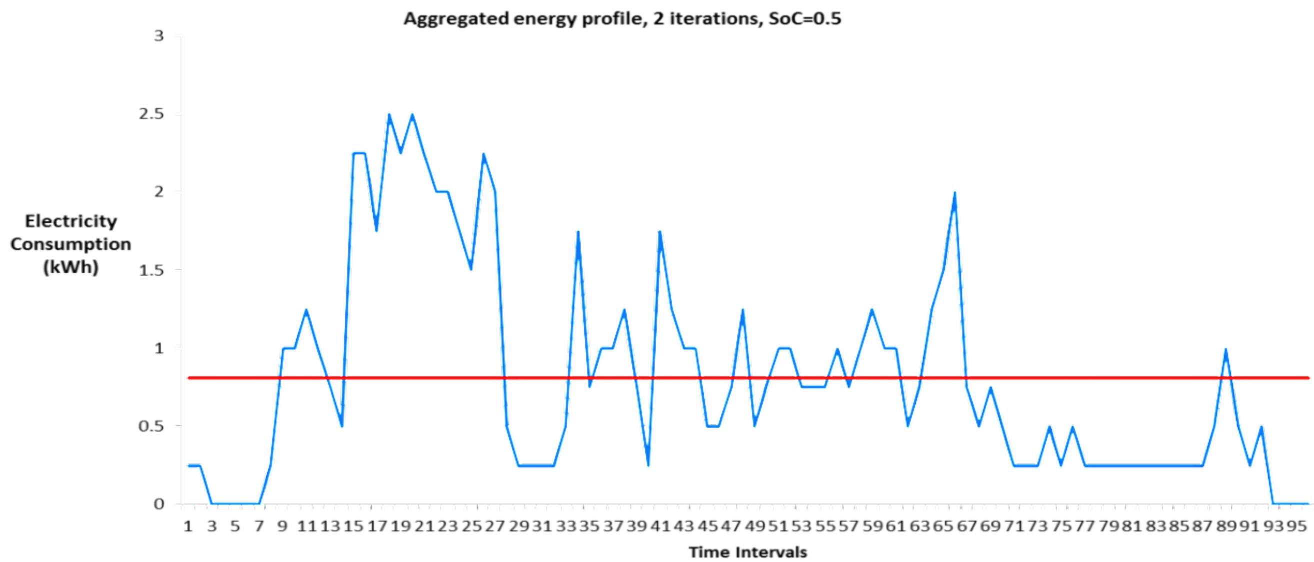


Figure 42: Initial SoC=0.5, Buffer Size=10 kWh_el, 2 iterations

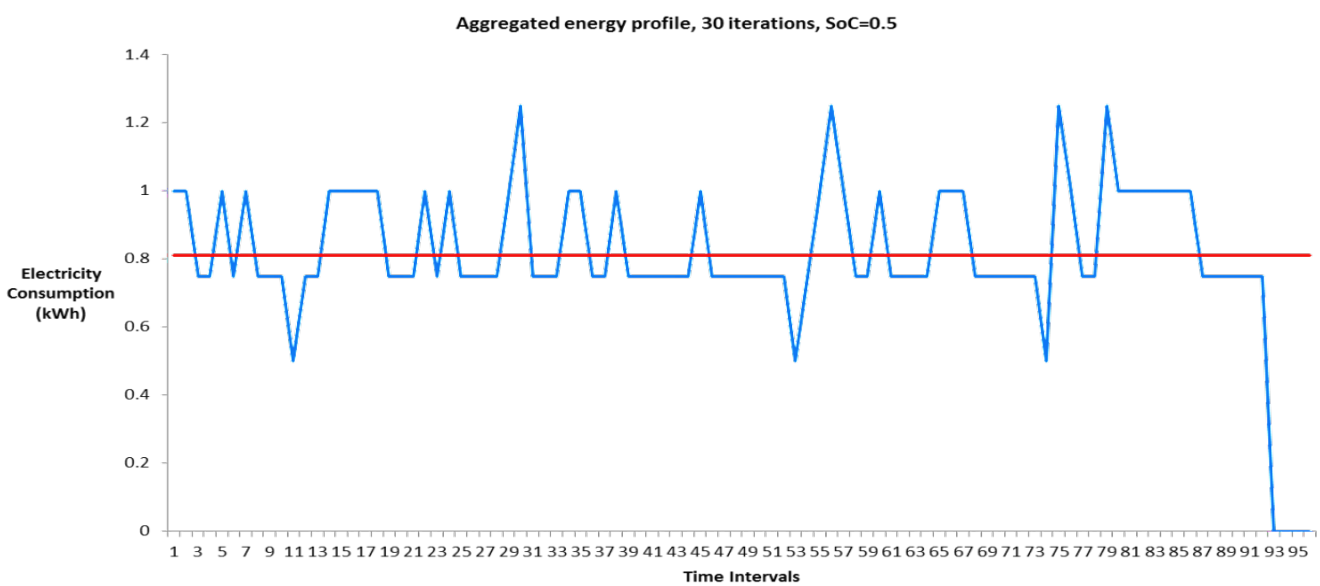


Figure 43: Initial SoC=0.5, Buffer Size=10 kWh_el, 30 iterations

Figure 44 shows the decrease of the squared mismatch as the number of algorithm iterations increases.

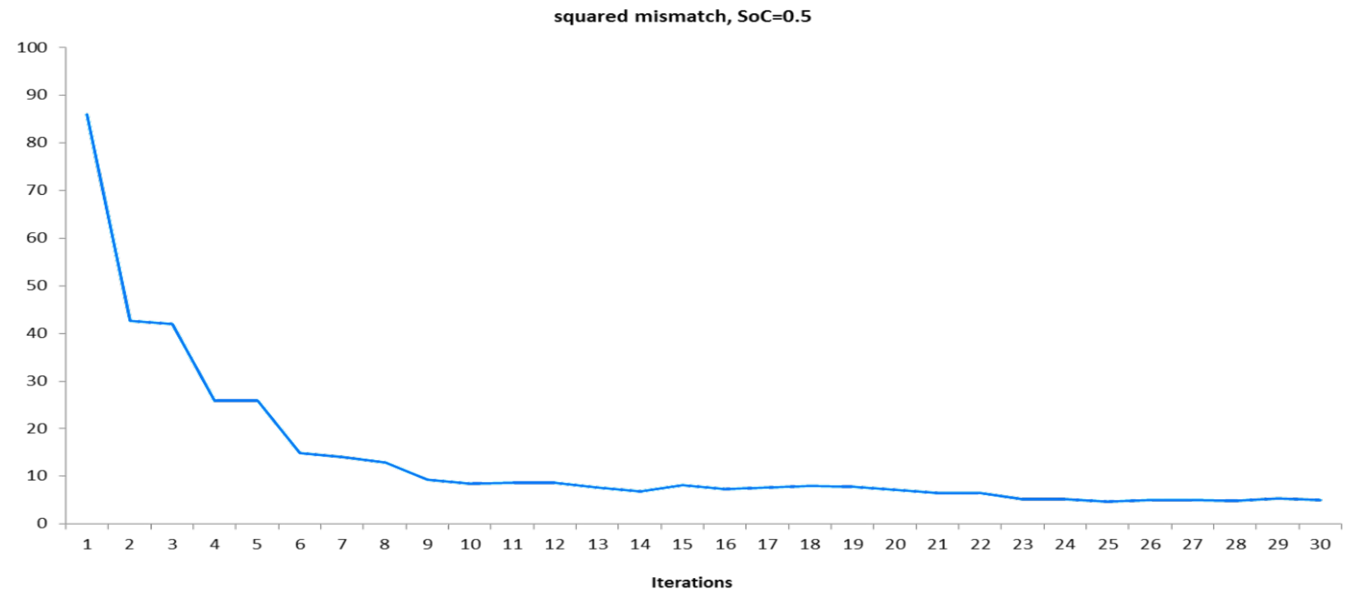


Figure 44: Squared mismatch, Initial SoC=0.5

Table 4 sums up the results from the two simulations performed. It can be concluded that when the initial SoC of the buffers was set to 0.5, the squared mismatch was reduced as a result of the larger flexibility offered by a buffer which has an initial SoC of 0.5 compared to a buffer with initial SoC of 0 which is forced to consume energy in the initial time intervals as it holds the minimum SoC. Furthermore, the average energy consumption is smaller as a result of the stored energy in the buffer. The peak consumption is also significantly reduced, probably because the high heat demand during certain intervals is partially covered from the energy that is already stored from the beginning and thus less energy needs to be consumed during that intervals.

Variable	Simulation 1	Simulation 2
Initial SoC	0	0.5
E_{av} (kWh)	1.79	0.81
M_2 (30 iterations)	22.04	4.96
M_2 (2 iterations)	166.416	42.71
Peak consumption (kWh)	2.5	1.25

Table 4: Comparison, Simulations 1 and 2 (Buffer size = 10 kWh_el)

5.2.3 Planning for Buffer devices, Simulation 3: influence of the heat buffer size

The input was the same as simulation 2 except the capacity of the buffers which was changed to 5 kWh_el (15 kWh_th ,COP=3), which is half of the buffer size of simulation 2, allowing us to derive conclusions regarding the influence of the buffer size to the aggregated energy profile.

Figure 45 shows the aggregated energy profile after 30 algorithm iterations and figure 46 the squared mismatch over the planning horizon.

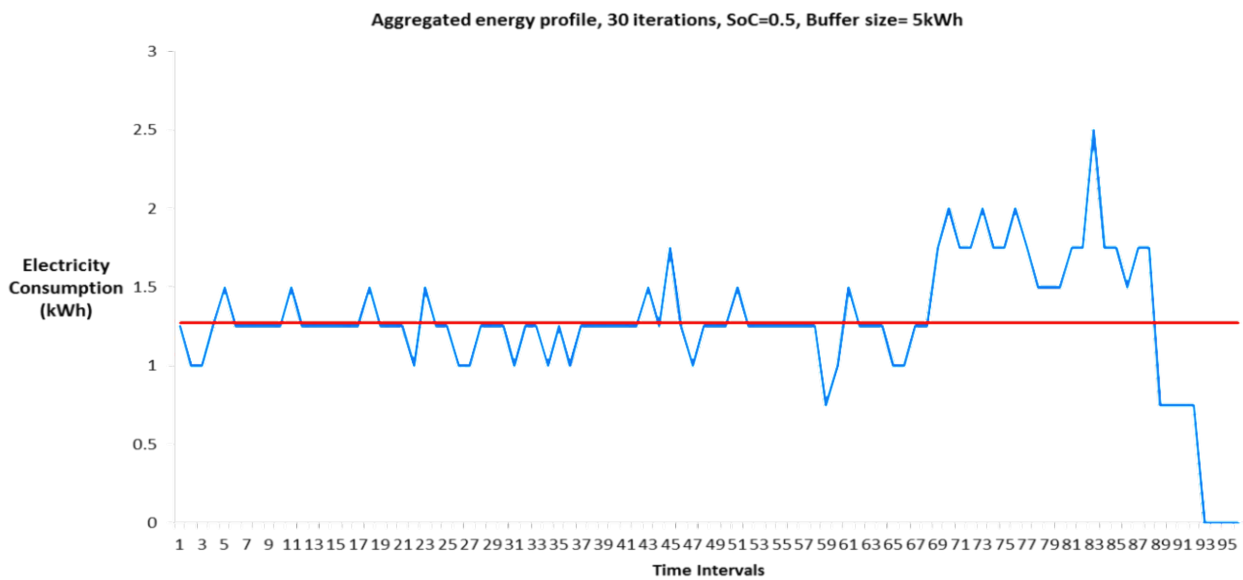


Figure 45: Initial SoC=0.5, Buffer Size=5 kWh_el, 30 iterations

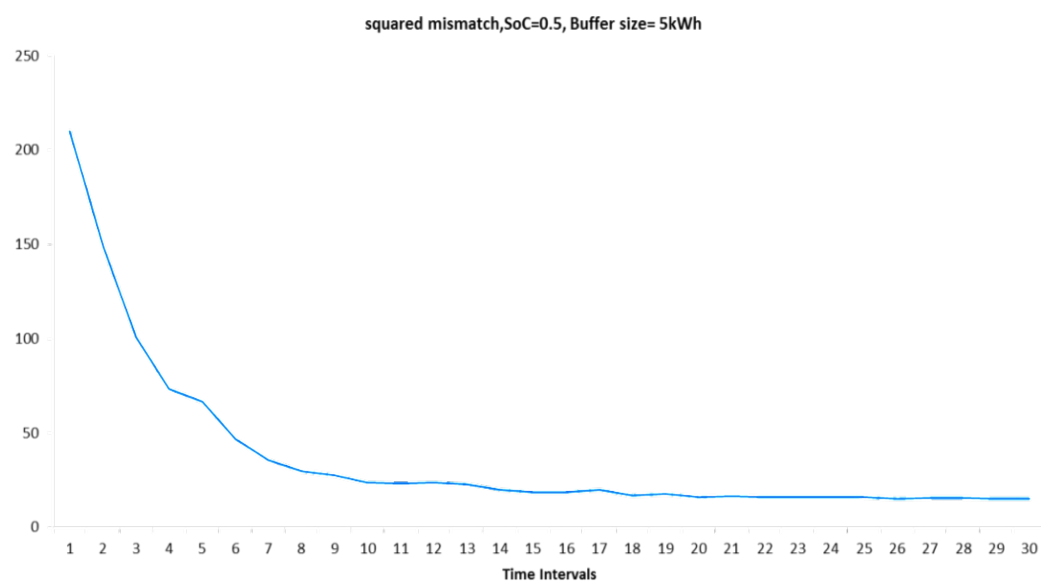


Figure 46: Squared mismatch: SoC=0.5, Buffer Size=5 kWh_el

Table 5 compares the simulations 2 and 3. In these simulations the initial SoC was the same (0.5) but the buffer size in simulation 3 was half of that of simulation 2.

Variable	Simulation 2	Simulation 3
Buffer Size (kWh)	10	5
E_{av} (kWh)	0.81	1.27
M_2 (30 iterations)	4.96	15.38
M_2 (2 iterations)	42.71	148.7598
Peak Consumption (kWh)	1.25	2.5

Table 5: Comparison, Simulations 2 and 3 (initial SoC=0.5)

It can be derived that when the buffer size was reduced, the squared mismatch and the peak and average consumption increased. That is an expected result as the flexibility of a buffer increases with an increasing capacity. The average consumption is also larger in the case of the smaller buffer as less energy is stored initially.

5.2.4 Planning for Buffer and Uncontrolled devices, Simulation 4

In this simulation we added a PV system to the twenty houses equipped with heat pumps and buffers (buffer size = 10kWh_el, initial SoC=0.5).

The data related to the PV production were received from a real system of 1.4 kWp during a winter day (of a house located in the Netherlands) and were multiplied by 50 and 100 to represent systems of 70 kWp and 140 kWp respectively. We need to mention that because the data were received during a winter day, the PV system did not reach the maximum power it could produce during the measurements. Data from a winter day were chosen as the data for the heat demand were also collected during winter day. The PV production profile can be seen in figure 50 (yellow color).

The goal is again to reach a flat aggregated profile exploiting the same time the energy produced by the PV system. The next figures (figure 47 and figure 48) represent the aggregated profiles in the two cases of 70 and 140 kWp after 30 iterations.

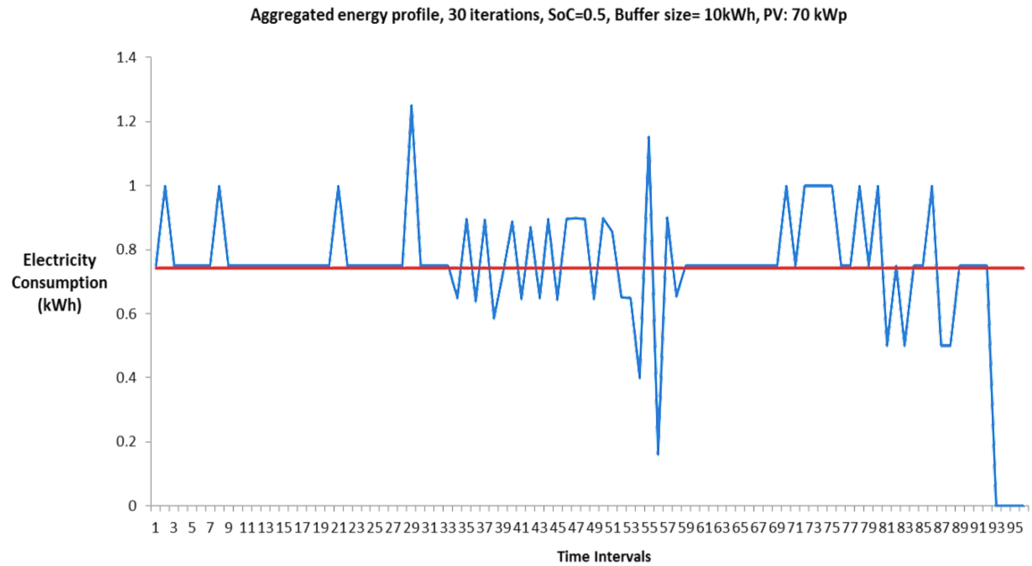


Figure 47: PV capacity of 70 kWp

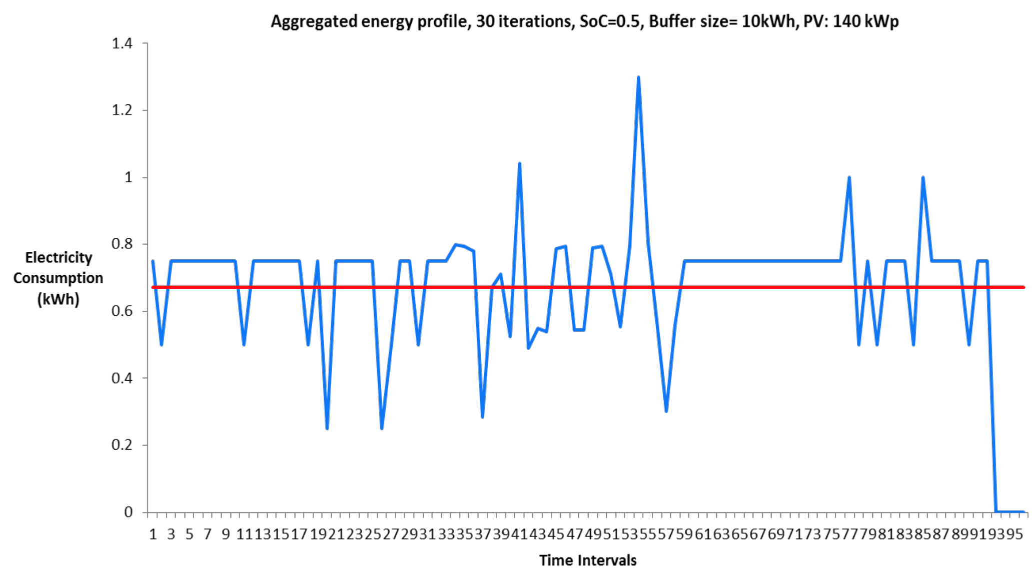


Figure 48: PV capacity of 140 kWp

The results regarding the squared mismatch and the peak and average consumption of the two cases are summed up in table 6. The squared mismatch presents a very small difference between the two cases of 70 and 140 KWp cases. The average consumption is smaller in the 140 kWp case as the energy offered from the PV system to the buffer is larger. In both cases the average consumption is smaller compared to the system where the PV system did not exist since the PVs cover part of the electricity demand. The squared mismatch is also decreased.

Variable	Simulation 2	Simulation 4a	Simulation 4b
PV power (kWp)	0	70	140
E_{av} (kWh)	0.81	0.74	0.67
M_2 (30 iterations)	4.96	4.40	4.08
M_2 (2 iterations)	42.71	43.06	52.05822
Peak Consumption (kWh)	1.25	1.25	1.3

Table 6: Comparison: 0, 70, 140 kWp PV production)

5.2.5 Planning for Buffer, Uncontrolled and Time Shifter devices, Simulation 5

In this simulation we add twenty washing machines to the system of simulation four (one per house). The simulation also includes the PV production of 140kWp. The algorithm produces again satisfying results as the mismsatch converges again around the 10th iteration as can be seen in figure 50. Figure 49 shows the aggregated energy profile after the 2^{nd} and the 30^{th} iteration.

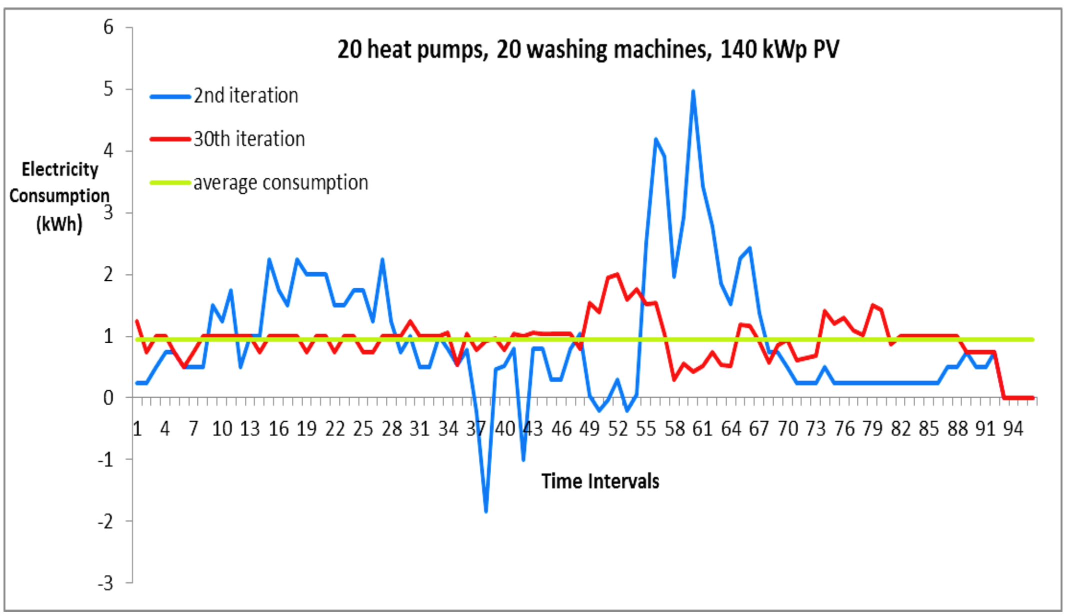


Figure 49: The aggregated energy profile after the 2nd and 30th iteration

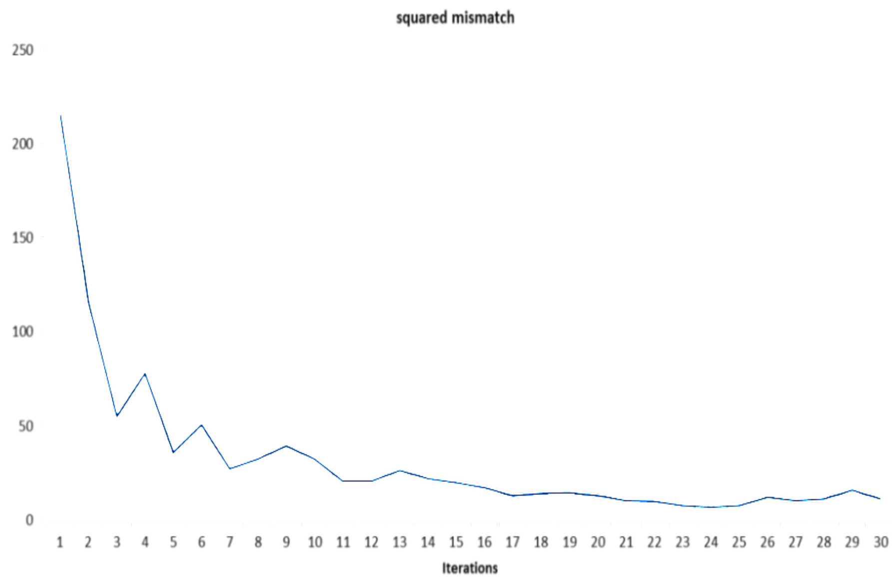


Figure 50: Squared mismatch, simulation 5

It would also be interesting to compare the energy consumed by the heat pumps and the washing machines, with the energy produced by the PV system. In figure 51 the energy consumed by the heat pumps is shown above the energy produced by the PV system. The peak consumption occurs to take advantage of the high PV production during the same period and by doing so the overall aggregated consumption does not result in any peak consumption period during those intervals.

In figure 52 the energy consumed by the heat pumps and the washing machines is shown. The restrictions posed by the permissible starting times, forces the washing machines to operate during a certain period and that explains why all washing machines operate between the 48th and 82nd intervals. This fact can also be seen in the aggregated consumption of the washing machines, shown in figure 53. By comparing the graphs of the heat pump consumption and washing machine consumption we can see that in the periods when the washing machines are turned on, the heat pumps have a very low overall consumption to counterbalance the large energy consumption of the washing machines.

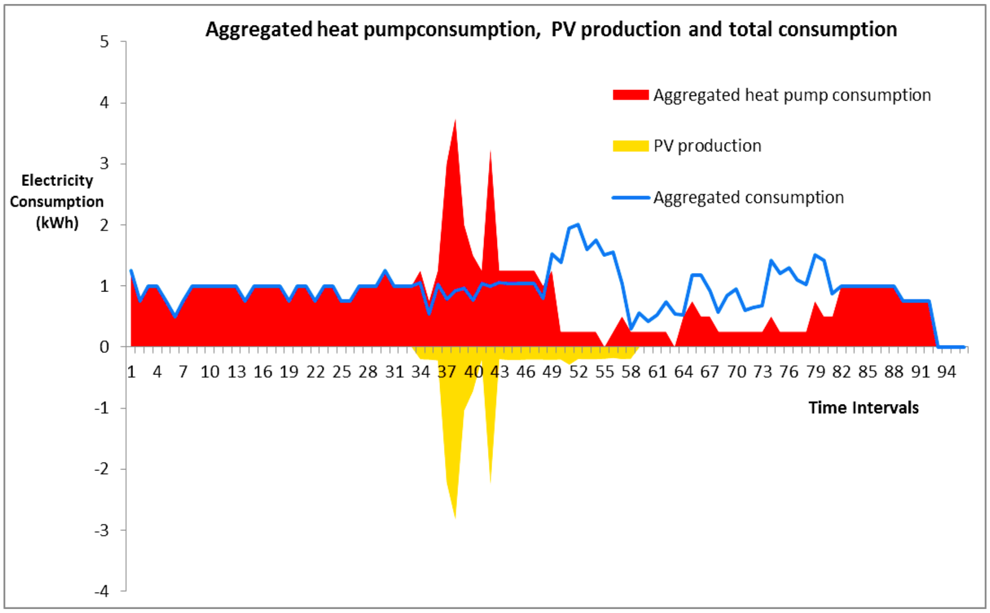


Figure 51: Aggregated heat pump consumption, PV production and total consumption

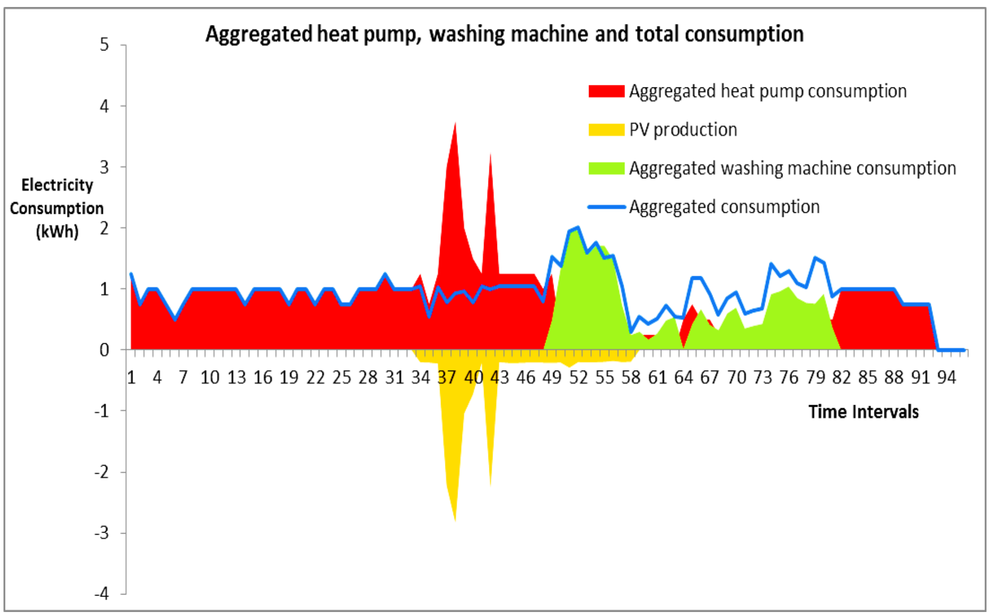


Figure 52: The aggregated energy consumed by the 20 heat pumps and the 20 washing machines

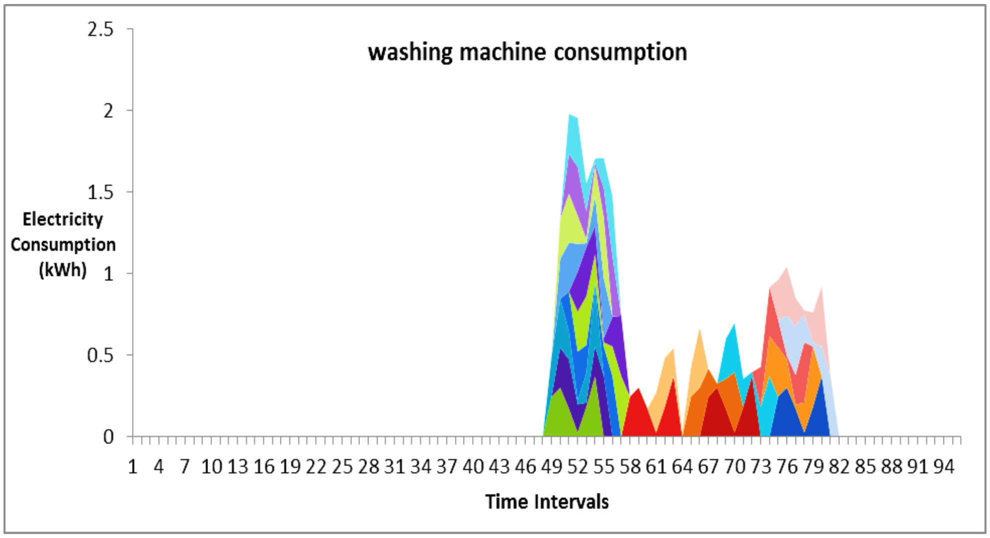


Figure 53: The energy consumed by the washing machines (stacked profiles)

5.3 Evaluation - Conclusions

The goal of the simulations executed and presented in this section, was to examine if the software developed within this assignment, implements successfully the Planning phase of Triana. The objective in every simulation was to control a number of devices, aiming at a flat aggregated consumption profile.

Simulations 1-3 included the control of Buffer devices. Simulations 2 and 3 examined the influence of the initial state of charge and heat buffer size to the overall operation of the Buffer devices. In simulation 4, the PV production added to the system of Buffer devices represented an Uncontrolled device. Finally, simulation 5 included twenty Time Shifter devices to the system of Buffer and Uncontrolled devices.

The results of all simulations showed that the execution of the algorithm developed produced the desired results; the mismatch between the actual and the goal profile, decreased with the number of running iterations. A convergence of the mismatch was also observed after a certain number of algorithm iterations showing that no further improvement can be achieved after this number of iterations.

6 Conclusions and future work

The goal of this thesis was to research whether it is possible to develop a software programme that implements the planning phase of Triana using the FPAI platform.

The first step to answer that question was a detailed study of Triana and FPAI. The study of Triana refers to both the theoretical background which includes the structure of Triana as well as the mathematical tools that it is built on but also to the programming implementation of the planning phase. Regarding FPAI, FAN provides a series of documents that describe all platform specifications. Understanding the RAI software implementation of FPAI was also essential as every energy application that is built on FPAI needs to understand the device flexibility as it is described by the Control Spaces but also because RAI provides all the other tools given in Chapter 4.

The next step was the development of the software that would execute the planning phase of Triana using the FPAI. Part of the software was developed during the internship conducted at TNO. During the implementation a few incompatibilities were observed between Triana and FPAI as discussed in Chapter 4. For the software development the core Triana functionality was used as a base. Nevertheless, new approaches were used as to overcome certain problems like in the Time Shifter planning algorithm. A number of simulations were also executed to test if this specific implementation produced the expected results that the planning phase of Triana should result in.

This thesis showed that FPAI provides all necessary tools to expose the flexibility of devices in a manner that can be handled by Triana in order to execute its planning phase.

This thesis addressed the issue of implementing the planning phase of Triana only and not the implementation of the other two phases of Triana, Forecasting and Real-time Control. FPAI does not include yet a component that would be able to forecast the energy parameters (e.g. the heat demand for the planning of Buffer devices or the permitted starting times for Time Shifter devices) that describe the energy flexibility of the devices in the form that the existing Triana implementation needs to execute the next phase, planning. To overcome this problem static heat demand profiles were used (Buffer device case) or the Control Space was used to provide essential information for the planning of other devices (e.g. the Time Shifter Control Space determines the permitted operation times). The execution of the Forecasting and the Real-time Control phases of Triana could be subjects for future master theses as the researchers working on the FPAI development have also considered the development of a platform that is capable of producing forecasts related to the operation of the devices.

Another possibility for future work could be a comparison of Triana and Powermatcher implementations when executed using the FPAI platform. Such a comparison would expose the advantages/disadvantages of each methodology over the other and would provide useful information for their improvement.

References

[1] Co2 in atmosphere per year. http://www.documentary-video.com/resources/documents/289.jpg.

[2] ABB. Efficiency - along the whole value chain. Website. Available online at http://new.abb.com/smartgrids/why-smart-grids/efficiency, Accessed: June 2014.

[3] R.N. Anderson, A Boulanger, W.B. Powell, and W. Scott. Adaptive stochastic control for the smart grid. *Proceedings of the IEEE*, 99(6):1098–1115, June 2011.

[4] Vincent Bakker. *TRIANA : a control strategy for Smart Grids : forecasting, planning and real-time control*. PhD thesis, Enschede, the Netherlands, January 2012.

[5] S. Bertolini, M. Giacomini, S. Grillo, S. Massucco, and F. Silvestro. Coordinated micro-generation and load management for energy saving policies. In *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*, pages 1–7, Oct 2010.

[6] C. Block, D. Neumann, and C. Weinhardt. A market mechanism for energy allocation in micro-chp grids. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pages 172–172, Jan 2008.

[7] M. Bosman. *Planning in Smart Grids*. PhD thesis, University of Twente, 2012.

[8] BP. World energy consumption. Website. Available online at http://www.bp.com/en/global/corporate/about-bp/energy-economics/statistical-review-of-world-energy.html, Accessed: June, 2014.

[9] R. Caldon, AR. Patria, and R. Turri. Optimisation algorithm for a virtual power plant operation. In *Universities Power Engineering Conference, 2004. UPEC 2004. 39th International*, volume 3, pages 1058–1062 vol. 2, Sept 2004.

[10] M. Chertkov. Optimization and control theory for smart grids. Talk availble online at author's site: http://cnls.lanl.gov/~chertkov/Talks/Grid/ColGrid.pdf. Accessed in: July 2014.

[11] L.M. Costa and G. Kariniotakis. A stochastic dynamic programming model for optimal use of local energy resources in a market environment. In *Power Tech, 2007 IEEE Lausanne*, pages 449–454, July 2007.

[12] S. Dagioglou. A literature review on energy control methodologies. Capita Selecta Assignment for partial fulifillment f the MSc of Sustainable Energy Technology, University of Twente, 2013.

[13] AL. Dimeas and N.D. Hatziargyriou. Agent based control of virtual power plants. In *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*, pages 1–6, Nov 2007.

[14] M. Erol-Kantarci and H. T. Mouftah. Wireless sensor networks in the smart power grid. Presentation in IEEE Ottawa Section Seminar, November 22 2010. Available online at: http://www.ieeeottawa.ca/comsoc/IEEESeminar-WirelessSensorNetworksWSNinPowerGrid-MErol-Kantarci.pdf.

[15] The Natural Resources Group. Average global temperature. http://www.thenrgroup.net/news/images/Average-Global-Temperature.gif. Accessed: June 2014.

[16] D. J. Hammerstrom. Pacific northwest gridwisetm testbed demonstration projects. Prepared for U.S. Department of Energy by the Pacific Northwest National Laboratory, Richland, WA, 2007. Available online at http://gridwise.pnl.gov/docs/op_project_final_report_pnnl17167.pdf.

[17] M. P F Hommelberg, C.J. Warmer, I G. Kamphuis, J. K. Kok, and G.J. Schaeffer. Distributed control concepts using multi-agent technology and automatic markets: An indispensable feature of smart power grids. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1–7, June 2007.

[18] J. K. Kok, C. J. Warmer, and I. G. Kamphuis. Powermatcher: Multiagent control in the electricity infrastructure. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 75–82, New York, NY, USA, 2005. ACM.

[19] K. Kok. *The Powermatcher: Smart Coordination for the Smart Electricity Grid.* PhD thesis, Vrije Universiteit, Amsterdam, 2013.

[20] A. Kooijman. Technology and sustainable development. Course lectures at SET progrmme, University of Twente, 2014.

[21] Fan H. Lowe M. and Gereffi G. U.s. smart grid, finding new ways to cut carbon and create jobs. Center on Globalization, Governance & Competitiveness, Duke University, April 2011. Available onlineat at `http://gridwise.pnl.gov/docs/op_project_final_report_pnnl17167.pdf`.

[22] Albert Molderink. *On the three-step control methodology for Smart Grids.* PhD thesis, Enschede, the Netherlands, May 2011.

[23] FlexiblePower Alliance Network. Flexiblepower application infrastructure developer tutorial, November 2013. Available online at `http://www.flexiblepower.org/downloads/`.

[24] FlexiblePower Alliance Network. High level functional specification of the flexiblepower application infrastructure, January 2013. Available online at `http://www.flexiblepower.org/downloads/`.

[25] Global Footprint Network. Footprint for nations. Website. Available online at: `http://www.footprintnetwork.org/en/index.php/GFN/page/footprint_for_nations/`, Accessed: June 2014.

[26] Global Footprint Network. Glossary. URL: `http://www.footprintnetwork.org/en/index.php/GFN/page/glossary/`. Accessed: June, 2014.

[27] Global Footprint Network. World footprint. Website. Available online at: `http://www.footprintnetwork.org/en/index.php/GFN/page/world_footprint/`, Accessed: June 2014.

[28] Climate Action of European Commission. The 2020 climate and energy package. Website. Available online at `http://ec.europa.eu/clima/policies/package/index_en.htm`.

[29] United Nations World Population Prospects. World population. Available online at: `http://esa.un.org/wpp/unpp/panel_population.htm`. Accessed: June, 2014.

[30] W. Saad, Zhu Han, H.V. Poor, and T. Basar. A noncooperative game for double auction-based energy trading between phevs and distribution grids. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 267–272, Oct 2011.

[31] W. Saad, Zhu Han, H.V. Poor, and T. Basar. Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications. *Signal Processing Magazine, IEEE*, 29(5):86–105, Sept 2012.

[32] M.A Sofla and R. King. Control method for multi-microgrid systems in smart grid environment–stability, optimization and smart demand participation. In *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, pages 1–5, Jan 2012.

[33] V. Thompson. Using neural networks to create a new control methodology for smart grids. In Website of National Science Foundation, http://www.nsf.gov/discoveries/disc_summ.jsp?cntn_id=127605, April 2013. Accessed in: July 2014.

[34] Wikipedia. Electric energy consumption. Accessed: June, 2014.

[35] Wikipedia. Stochastic control. Accessed: 25-03-2013.

7 Appendix – Class Diagram (UML)