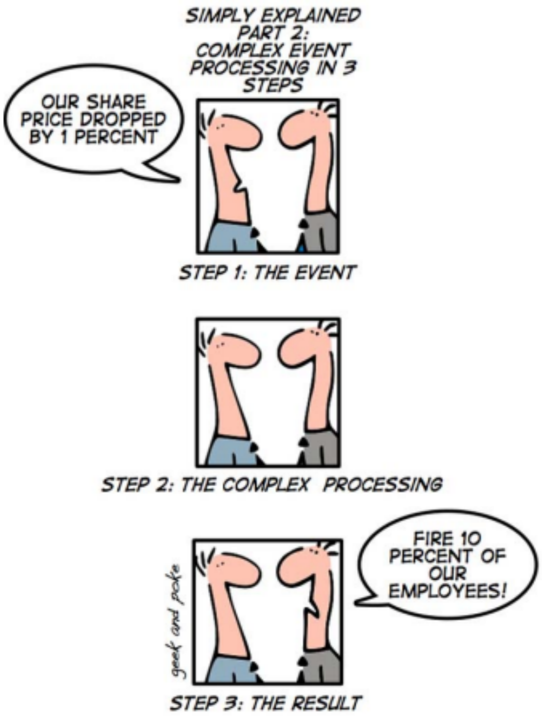
Smart Grid management using Java-based Complex Event Processing

Master's Thesis



Author	Ralph Matroos
Author Email	r.e.matroos@student.tudelft.nl
Student ID	1193791
Faculty	EWI, Delft University of Technology
Study program	Master Information Architecture
Supervisor	Dr. ir. A.J.H. Hidders
Date	12-03-2013

"The definition of insanity is doing the same things and expecting different results."

- ***Albert Einstein***

Cover picture: http://geekandpoke.typepad.com/geekandpoke/2008/02/minigeek---ed5.html

Smart Grid management using Java-based Complex Event Processing

Thesis

submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
in
COMPUTER SCIENCE
TRACK INFORMATION ARCHITECTURE

by

Ralph Emile Matroos

born in Willemstad, Curaçao

Web Information Systems Group
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Mekelweg 4
2628CD Delft, the Netherlands
www.ewi.tudelft.nl

Logica is now part of CGI.

Logica
George Hintzenweg 89
3068AX Rotterdam, the Netherlands
www.logica.com

(This page is intentionally left blank)

Smart Grid management using Java-based Complex Event Processing

Author: Ralph Emile Matroos
Student id: 1193791
Alternate e-mail: ralphmatroos@gmail.com

Abstract

In the past years the electricity demand has risen a great deal, and it is predicted that this trend will continue in the years to come. Two of the main reasons this rise in electricity demand is taking place is because of the growing world population and the introduction of Electric Vehicles (EV). There are also several types of small energy sources being introduced (Solar panels for example) in different parts of the electricity network causing electricity not to be generated at one point but rather several different sources (Distributed Generation). Not only will this increase the load on the Dutch electricity grid, but it will also make the electricity in such a network more volatile. The Dutch electricity grid is ageing and studies have shown that it cannot support these latest developments and future predictions. This causes a problem since the electricity grid will soon reach its maximum potential if demand keeps rising at this rate and the grid isn't updated or replaced. Simply adding more electricity cables and bigger transformers has proven undesirable due to cost, and this means that some other way is needed to deal with this problem.

To solve the previously described problem, a lot of research as been done on ways to improve the current grid, and the most promising development is that of the smart-grid. Using this approach, the grid will switch from a passive to a more active nature, one in which components in the grid will be able to communicate with one another. This communication can for example be in the form of status updates by one component to one or more of the other components in the grid, or listening to requests by one or more of the other components in the grid and performing actions based on those requests. When managed in a correct manner, the communication between components in the grid can ultimately result in a more efficient, reliable and secure grid, which in turn directly translates into an extended maximum potential of the grid, financial benefits for the electricity suppliers, consumers, and delivery companies and environmental benefits for the entire society. As stated in the previous sentence, these benefits can only be achieved when using proper management tools to make optimal use of the grid components functionalities. Due to the vast size of the Dutch electricity grid, and the "smart" components producing more events, a lot of data will become available for processing. Any sort of smart-grid management tool therefore must have the capability to work with large amounts of data. Though there are several running projects with each there own form of management method available, none of them are based on Complex Event Processing.

Previous research has shown that the problems in smart grid could in fact be solved by using complex event processing and the problem this research aims to answer is how such a system should be built and what other aspects should be addressed in order to implement a CEP system in the Dutch electricity grid. We started by researching the relevant areas of the field and gathered experts (Logica, Alliander, Enexis, Stedin) to conduct interviews. After collecting the appropriate knowledge we proceeded by creating a digital representation (simulation) of a (fictional) electricity grid to mimic all the events the smart-grid would pass on to the CEP-engine. The second step was to conclude with the experts in the field which rules were the most important to keep an electricity grid running in practice and these rules were then implemented in the CEP-engine. The result of the project was a demo application, which used the simulation model and defined rules to show how such a system would perform in real life situations.

Though this research was an exploratory one, a very clear overview is presented about using CEP in the utilities sector and several results where achieved during the course of this research. There was a lot of information available on smart grids and on Complex Event Processing separately, but this research combined the two and provides insight on what steps need to be taken to actually start using CEP in the Dutch electricity grid. Next to combining CEP and Smart Grid, several available CEP suites were compared and recommendations of which tool to use were made. Furthermore, an architecture of such a CEP system was created using the principles of Event-Driven SOA. The central question during this research was how CEP could be employed in the smart grid networks and we have accomplished just that. Using the input of experts in the field, three main smart grid challenges were defined according to priority, and implemented using CEP technology. To mimic a smart grid network for the CEP engine, a smart simulation grid was also designed and implemented.

It must be mentioned however that CEP relies on an event cloud being generated (event source) and the components in the grid being able to receive messages and remotely controlled (subscribers should be able to handle incoming actions and messages). Without the transition to smart-grid, the event-processing engine would be of little value since the engine would then have no way to know the state of the grid at a certain point in time and thus no analysis can be done. At the other end of the equation it is also extremely important that the components in the grid not only provide status updates, but can also be controlled to some extent. CEP can still be of value by just giving warnings to employees (SMS, e-mail for example), but it's true power lies in the fact that the analysis can fully automate the process which in turn results in quicker overall reaction times. Due to the lack of similar systems (smart grid management), time constraints, and aim of this research however, it has not been possible to test the performance of the designed system.

Keywords: Complex Event Processing, EDA, SOA, EDSOA, Smart grid

Graduation Committee:

Prof. dr. ir. G.J.P.M. Houben,	Full prof., EEMCS, TU Delft
Dr.ir. A.J.H. Hidders,	Assist. prof., EEMCS, TU Delft
Dr.ir. M.F.W.H.A. Janssen,	Assoc. prof., TPM, TU Delft
Ir. E. Essenius,	Logica, George Hintzenweg 89, Rotterdam

Preface

This document is the result of a master thesis project conducted at business and technology service company Logica in Rotterdam. It is the final part of the master program Information Architecture at the University of Technology Delft (TU Delft), and covers all the knowledge and results of the project in detail. This research gave me many new insights the energy sector and related business domains, a field I had very little knowledge about when embarking upon.

Though this research was conducted on my own, I couldn't have done it without the help of Edwin Essenius, my direct supervisor at Logica, and Bert de Neef, project manager at Working Tomorrow (Student research department of Logica), and my direct supervisor at the University, Jan Hidders, whom all helped in keeping me on the right track during the research. A special thanks also goes out to Willem Altena, consultant at Logica, for providing me with the much-needed contacts in the utilities field. I would also like to thank Martijn Bongaerts (Alliander), Danny Geldtmeijer and Joris Knigge (Enexis), and Inge Wijgerse (Stedin), for providing me with information and answering the questions I had for them during the project.

The end of this graduation project also marks the end of my study in Delft. As one of the last but most certainly not least I would also like to thank my parents and brothers for their unconditional support and encouragement throughout the years. Finally, thanks to my friends for the great times we've shared both on and off campus (We made Microsoft's XBOX 360 look good).

Ralph Matroos
Delft, the Netherlands
March 2013

(This page is intentionally left blank)

Contents

CHAPTER 1: INTRODUCTION 2

1.1 PROBLEM DESCRIPTION 2
1.2 RESEARCH QUESTIONS 4
1.3 GENERAL INFORMATION 6
1.4 SCOPE 6
1.5 RESEARCH METHOD 6
1.6 DELIVERABLES 8
1.7 REPORT OUTLINE 10

***PART 1: BACKGROUND & CORE KNOWLEDGE* 11**

CHAPTER 2: DUTCH ELECTRICITY SECTOR 12

2.1 DUTCH ELECTRICITY VALUE CHAIN 12
2.2 TRANSMISSION SYSTEM OPERATOR (TSO) 12
2.3 DISTRIBUTION SYSTEM OPERATOR (DSO) 13
2.4 ENERGY TRADING 14

CHAPTER 3: SMART GRID 15

3.1 INTRODUCTION 15
3.2 DEFINITION 15
3.3 RELATED WORK 17
3.3.1 PowerMatching city 17
3.3.2 Vacation park Bronsbergen (Zutphen) 18
3.3.3 The Los Angeles Power Grid 18
3.3.4 Boulder city Colorado (XCEL Energy) 19
3.3.5 Related work conclusion 20

CHAPTER 4: EVENT PROCESSING 22

4.1 EVENTS 22
4.2 EVENT PROCESSING APPROACHES 23
4.2.1 Simple Event Processing 23
4.2.2 Stream Event Processing 23
4.2.3 Complex Event Processing 23
4.3 SITUATION AWARENESS 24
4.4 ALTERNATIVE EVENT DRIVEN TECHNOLOGIES 25
4.5.1 Available CEP Suites 27
4.5.2 Esper for Java 30

CHAPTER 5: SYSTEM ARCHITECTURE 33

5.1 ARCHITECTURE 33
5.2 ENTERPRISE ARCHITECTURE 33
5.3 ARCHITECTURE TYPES 35
5.3.1 Business architecture 35
5.3.2 Information systems/Application architecture 35
5.3.3 Security architecture 36

5.4 SERVICE ORIENTED ARCHITECTURE 36
5.5 ENTERPRISE SERVICE BUS 38
5.6 EVENT DRIVEN ARCHITECTURE 39
5.6.1 Event Driven Architecture definition and principles 39
5.6.2 Event Driven SOA 40
5.7 ARCHITECTURE FOR DEVELOPED SYSTEM 41

PART 2: EVENT GENERATION 43

CHAPTER 6: EVENT GENERATOR 44

6.1 SMART SIMULATION GRID 44
6.2 SMART GRID COMPONENTS 45
6.2.1 Smart meters 45
6.2.2 Distribution transformers 47
6.2.3 Combined Heat and Power generators (CHP) 47
6.2.4 Solar panels 48
6.2.5 Energy storage 49
6.2.6 Wind turbines 50
6.3 SMART GRID COMPONENT DESIGN 51
6.3.1 Electricity cables 51
6.3.2 Smart meters (Smart houses) 53
6.3.3 Distribution transformers 54
6.3.4 Combined Heat and Power generators (CHP) 55
6.3.5 Solar panels 55
6.3.6 Wind turbines 55
6.3.7 Electricity storage batteries 55
6.3.8 Component identifiers 56

PART 3: CEP ENGINE DESIGN AND DEMO EVALUATION 57

CHAPTER 7: MAIN CEP-ENGINE CHALLENGES 58

7.1 CURRENT GRID VERSUS FUTURE GRID 59
7.2 LOAD BALANCING 59
7.3 OUTAGE DETECTION 60
7.4 CONGESTION MANAGEMENT 62

CHAPTER 8: CEP ENGINE AND DEMO 64

8.1 DEMO SETUP AND GOAL 64
8.1.1 Scenario #1 (Sun: None, Wind: None, Temperature: Cold) 64
8.1.2 Scenario #2 (Sun: Average, Wind: High, Temperature: Mild) 65
8.1.3 Scenario #3 (Sun: High, Wind : High, Temperature: Mild/Hot) 65
8.1.4 Test setup 65
8.2 TEST RESULTS 65
8.3 SYSTEM IMPLEMENTATION 66
8.4 GRAPHICAL USER INTERFACE 68

CHAPTER 9: ARCHITECTURE AND DEMO EVALUATION 71

9.1 ARCHITECTURE EVALUATION 71
9.2 DEMO EVALUATION 74
9.2.1 Event generator evaluation 74
9.2.2 CEP engine evaluation 74

9.2.3 CEP engine evaluation conclusion *77*
9.3 WORK OVERVIEW 78

***PART 4: SUMMARY, CONTRIBUTIONS, CONCLUSIONS AND FUTURE WORK* 79**

CHAPTER 10: SUMMARY AND CONCLUSIONS 80

10.1 SUMMARY 80
10.2 CONCLUSIONS 81

CHAPTER 11: CONTRIBUTIONS AND FUTURE WORK 83

11.1 CONTRIBUTIONS 83
11.2 FUTURE WORK 84
11.2.1 Future work in national grid *84*
11.2.2 Future work within designed system *84*
11.2.3 General *85*

LIST OF FIGURES 86

LIST OF TABLES 87

BIBLIOGRAPHY 88

***APPENDICES* 92**

APPENDIX A: LIST OF ABBREVIATIONS 93

APPENDIX B: INTERVIEW SETUP 94

(This page is intentionally left blank)

Chapter 1: Introduction

This master thesis researches the possibilities of complex event processing (CEP) in (smart) electricity grids. As the prices of sensor and communication technology are dropping, it is becoming more and more interesting and economically feasible to employ these techniques in different settings. One particular setting where there is a lot to be gained is that of electricity grid management. It is believed that implementation of sensors, accompanied by a control system, would result in an overall quicker reaction time to events generated by the grid. This would lead to increased efficiency in the grids since for example, safety thresholds can be minimized without posing an actual threat to the system.

This chapter starts with an introduction in which the relevance of the subject is stated. Several developments in the electricity sector are discussed and concluded upon. The rest of this chapter covers the research questions defined, the scope of the research assignment, the research method and the deliverables. The end of this chapter gives a short outline of the remainder of this document.

1.1 Problem description

Electrical power systems of industrialized economies, such as the Netherlands, have become one of the most complex systems ever created by mankind over the course of the 20th century. What started out as a novelty, has turned out to be an absolute necessity, and has caused worldwide electricity use to grow tremendously (J.Kok, 2010). The Energy Information Administration (EIA) predicts that worldwide electric power generation will grow 2.4% a year between 2004 and 2030. To give a better view of these facts I present you with the table below (table 1).

Category	1950	2000	2050(Estimates)
World population	2.56 Billion	6.22 Billion	8.29 Billion
Electricity usage	2.06TW	3.80TW	6.99TW
Electricity as % of total energy	10,4%	25,3%	33,7%
Televisions	0.6 Billion	1.4 Billion	1.9 Billion
Personal computers	0	500 Million to 1 Billion	6 Billion to 8 Billion
Cell phones	0	0.8 Billion	5 Billion
Electric hybrid vehicles	0	55,852	3,151,439

Table 1: Electricity growth estimates (Jesse Berst, 2008)

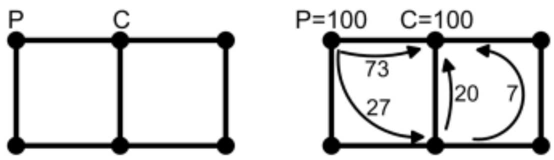
Research has shown that the biggest driving forces behind this growth of electricity demand are the following (Jesse Berst, 2008):

1. The rapid expansion of world population – Growth in the number of people needing electricity
2. Electrification of everything – Growth in the number of devices that require electricity.
3. Expectation inflation – Growth in the sense of entitlement that turns electrical conveniences into essentials demanded by all.

In addition to the growing energy demand there are also a number of ongoing changes in the electricity system that are forcing a change in infrastructure management. For example the shift to renewable energy sources means that weather will have a higher impact on generation and thus cause fluctuations. This shift to renewable energy sources, along with the introduction of CHP generators, causes a phenomenon otherwise known as Distributed Generation (DG). Distributed Generation causes the generation capacity in the medium and low voltage networks to rise (since energy is no longer just generated at power plants) (J.Kok, 2010).

Another development that has been gaining a lot of attention recently is that of electric cars which also plug into the grid to recharge their batteries. The prediction is that 20% of all cars in the Netherlands will be electric ones by the year 2020. This also forms a problem in electricity grid management because most of these cars will be recharged during peak hours (before and after office hours).

The increase of electricity demand along with distributed generation and the introduction of electric cars pose a serious threat on the aging power grid still widely used today. On top of everything previously mentioned, there is no way of directing the flow to follow a specified path. What happens instead is that the current follows the path of least resistance; possibly using a number of parallel trajectories an example is given below in figure 1. This behavior also has great implications for network planning and operation and furthermore means that if we would try to reinforce the grid, one specific section would not suffice (J.Kok, 2010).



Power loop flows in an electrical network. *Left*: simple example electricity network with six nodes and seven lines. At node *P* electricity is produced, at node *C* it is consumed, while the other nodes are passive. *Right*: the physical flows resulting from producing 100 units at node *P* while consuming the same amount at node *C*. The (resistance) characteristics of all seven lines have been chosen equal, while, for the sake of the example, transport losses have been neglected.

Figure 1: Power loop flow example (J.Kok, 2010)

The problems mentioned above (Growth of electricity demand, Distributed generation, Rise of the electric cars), along with the fact that the grid it operates on is ageing, make it necessary to look at new ways of getting the maximum out of the electricity grid and keeping it all in balance. The first thing that comes to mind when thinking of solutions for the problem of growing energy demand and generation is probably the following. Why can't we just strengthen the infrastructure by adding more electricity cables and bigger transformers? This modification would not be financially feasible since the entire grid would need changing as current chooses the path of least resistance.

Next to the fact that we need to upgrade the current electricity grid due to capacity problems, as stated above, the transition to smart grid brings along several other benefits like better reliability, reducing cost to produce, deliver and consume electricity, creating new jobs, reducing emissions and increasing safety by reducing the chance of injuries or even loss of life (K.Dodrill, 2010). All of these benefits can only be achieved when making optimal use of the smart grid's capabilities. Yogesh Simmhan et al. state the following in their research (Y.Simmhan, 2011):

"Power utilities are increasingly rolling out "smart" grids with the ability to track consumer power usage in near real-time using smart meters that enable bi-directional communication. However the true value of smart grids is unlocked only when the veritable explosion of data that will become available is ingested, processed, analyzed and translated into meaningful decisions."

What can be drawn from their research is that some form of management tool (for the explosion of data) is necessary to make sense of the available data, and that this data needs to be translated into meaningful decisions as soon as possible. Complex Event Processing can address these requirements perfectly, as the key characteristics of Complex Event Processing are analysis of large amounts of events by using sophisticated event interpreters, event pattern definition and matching along with correlation techniques, and being able to do it in real-time.

When looking at the facts and figures stated above, it is clear that something has to be done with the way we manage our electricity grid if we want to preserve our current lifestyle (in terms of electricity usage) and embrace future developments. Literature also points out that CEP should be a good way to tackle the large amounts of data made available in these smart grids. Previous research by a student at the university of technology Delft has shown that Complex Event Processing can be of great value in tackling these problems without reinforcing the grid, but he only covered this problem on a high level. This research actually goes further by identifying the available CEP suites, developing a basic architecture and actually building a CEP engine to show how CEP can be applied in this field.

1.2 Research questions

In the previous section the relevance of this research topic was discussed in detail, and in this section the main question will be formulated. The main question that this graduation project will attempt to answer is the following:

"How can Complex Event Processing be used for the management of electricity (control, purchase, sale) in a medium/low voltage (Dutch) smart electricity grid?"

Due to time constraints, only the part of electricity control will be addressed in detail. The parts of purchase and sale are included in the main question to stress their importance, and will be covered lightly in theory.

In order to answer the main question, next to the development of the engine itself, the following sub-questions (SQ) are defined:

SQ1. Why are the current ageing electricity grids not able to support the future electricity demand, technological advancements and the way in which the grids will be used in the future?

SQ2. What is a suitable grid technology that can support (if managed correctly) these forecasted demands, technological advancements and usage styles if applied to Dutch electricity grids?

SQ3. Is Complex Event Processing a suitable technology for the management of a smart grid in the Dutch electricity sector?

SQ4. What is an appropriate architecture that can be used to implement smart-grid using CEP in the Dutch electricity sector?

SQ5. How can the behavior of a smart-grid be simulated using Java, and what components should this simulation grid consist of in order to provide the demo application with realistic usage and generation data?

SQ6. What is an appropriate application design to test if CEP can support the transition to smart-grid in the Dutch electricity sector on the implementation level?

As mentioned, the relevance of the research topic was already addressed in the introduction of this document (Chapter 1.1) and tackles sub question 1. Chapters 2 and 3 are used to answer the second sub question. Chapter 4 is used to elaborate on CEP in general and also state the relevance of CEP in smart grid (sub question 3). Chapter 5 is dedicated to getting the necessary knowledge about architectures and creating one (sub question 4) and chapters 6 through 8 are dedicated to answering sub question 5 and 6. The result of these questions will be used in the final three chapters (Chapter 9, 10, and 11) to answer and conclude on the main research question.

1.3 General information

This project was conducted at the graduate department of Logica Rotterdam called Working Tomorrow. Logica is one of Europe's largest IT and management consultancy firms with over 5100 employees and 11 offices in the Netherlands.

My graduation project at Logica consisted of two parts:

- A literature study in the direction of smart grids and Complex Event Processing.

- The actual thesis project of implementing (a part of) the proposed solution to check whether CEP can actually be used for (smart) electricity grid management.

Next to the contributions this research gives science, Logica can also benefit from this research, as change is needed in the way we currently manage our electricity grid. Should Logica decide to do further research on this subject, they will have a good market position as there is little attention spent on the combination of CEP in the utilities market elsewhere.

1.4 Scope

Since the possibilities and fields of research on this subject are endless and the time available for this project is limited, a couple of boundaries are stated to ensure the timely completion of the project.

- The main topic of study for this research is CEP in the utilities sector. Though the main research question states control, purchase and sale, only control will be addressed in this research and will be elaborated upon by developing a small demo CEP application.

- The focus of the research will be on the distribution section of the Dutch electricity grid; this includes the Medium Voltage (MV) and the Low Voltage (LV) part of the grid.

- The demo that will be designed in this project will be based on the fictional grid defined in chapter 6. The events and rules used in the project will also result from this (fictional) grid.

1.5 Research method

At the point of starting this research there was little to no information available on the subject at hand. There was a lot of literature available about smart grids and separately also about Complex Event Processing, but nothing about the two subjects combined. It is for this reason that the main approach taken for this research is an exploratory one.

Exploratory research is often conducted because the research problem has not been clearly defined yet, or its real scope is as yet unclear, which is exactly the case here as this research is also used to familiarize with the subject. The results of exploratory research are not usually useful for decision-making by themselves, but they can provide significant insight into a given situation, as this research aims to do in the field of Complex Event Processing in smart grid networks.

As can be seen in section 1.2, research questions have been defined and sharpened throughout the duration of this research. The gathering of information will happen on both primary (gathering new information by interviews) and secondary levels (literature research). The research also includes a part of the *qualitative research* method since as already mentioned in the previous sentence; one of the qualitative research techniques used in this thesis is the interview. Interviews can also be part of the quantitative research method, but the main difference here is the depth of the interview. As the interviews conducted during this research project were relatively unstructured and in depth, they take more of a qualitative form in this research. Last, but most certainly not least, this research also delivers a *Proof Of Concept (POC)* in the form of a CEP engine demo to prove its feasibility.

Using the gathered information and the research questions defined an initial setup of a suitable simulation model will be made. This model will be verified by 5 experts in the field and adjusted accordingly. The next step will be that of identifying the most important challenges of electricity grid management and the generation of rules for the CEP-engine. When the development process is completed, the results will be used to conclude on whether CEP is a suitable solution to employ in the smart grid network. The course that the research will follow can also be seen in the project-roadmap (figure 2).

1.6 Deliverables

At the end of the project a conclusion can be made about the actual usability of the CEP system for use in the electricity sector. As mentioned in section 1.3, this information could give Logica an idea on whether to invest in this technology or not.

To sum it up the following should be the result of the thesis project:

- Thesis report (this document) for the Delft University of Technology and Logica (Working Tomorrow).
- Developed Complex Event Processing application for the management of an electricity grid.

Figure 2 on the next page gives a graphical representation of the project's stages. Part 0 refers to the research assignment, which was already completed before starting the main project (hence the number 0).

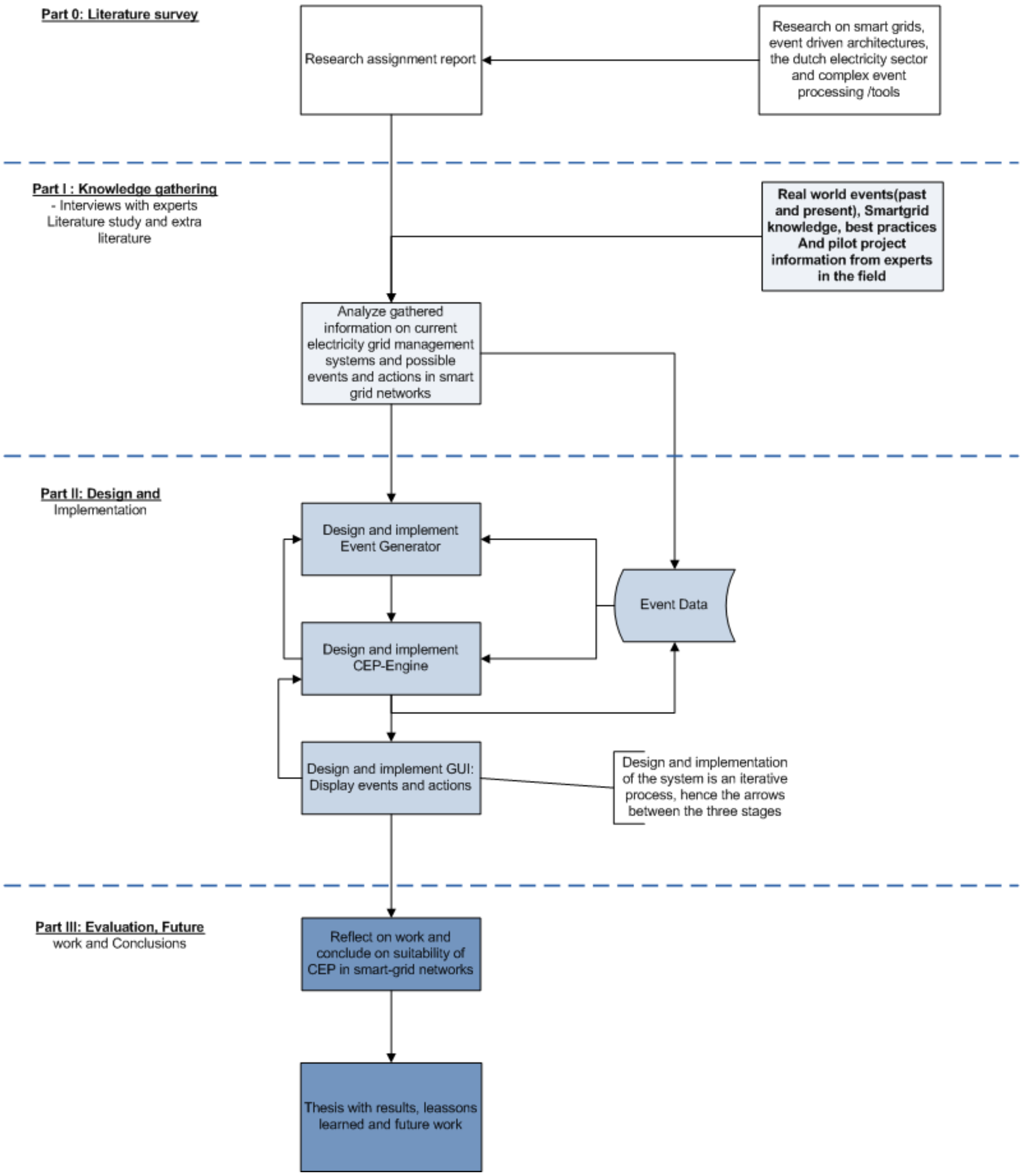


Figure 2: Project roadmap

1.7 Report outline

This section gives a short description of each chapter. This report will roughly follow the structure depicted in figure 2.

Chapter 1 introduces the subject by stating its importance to our society and also states the main question and sub questions, which this research will attempt to answer.

Chapter 2 is used to elaborate on the Dutch electricity sector in general, the energy trading market, Transmission System Operator, Distribution System Operator and the definition of smart grid. In the final part of this chapter, some pilot projects in the smart grid sector are given.

In **chapter 3** the focus is put on smart grid networks. The chapter is used to elaborate on the definition of a smart grid and its potential benefits and main drawbacks.

Chapter 4 is all about event processing. Now that we know all we have to know about the Dutch electricity grid, the fourth chapter introduces the concept of Complex Event Processing. This is the main technique the research is based upon. Different CEP platforms are also analyzed in this section and a conclusion is made on which platform to use.

Chapter 5 Is used to cover the architectural aspect of this research. The chapter starts with the definition and elaboration of several key architecture components followed by the main architecture defined for the designed application. Furthermore, the link to the Information Architecture track is also made in this chapter.

Chapter 6 will cover the case study in which the event generator (EG), which will be used as input to the CEP engine, is elaborated upon. This chapter also describes all of the components, which will be used for the simulation of events and elaborates on the manner in which the grid components will be implemented for the event generator, which in turn will be used as input for the designed CEP engine (defined in chapter 9)

Chapter 7 describes the main smart grid challenges we will attempt to tackle in this research by using CEP and furthermore, the main requirements of the system are specified, analyzed and transformed in to rules. These rules are used in the CEP engine to make sense out of the event input stream generated by the EG.

Chapter 8 is used to describe the implementation of the rules in the CEP engine. Next to the implementation details, this chapter also covers the graphical user interface of the designed Proof Of Concept (POC).

Chapter 9 covers the evaluation of the derived architecture and designed application.

Chapter 10 gives a summary of the entire project and thesis, followed by the conclusions based on the results acquired.

In **chapter 11** the results of the project as well as the contributions made to the research field are elaborated upon.

A list of the abbreviations used throughout this report can be found in appendix A. In appendix B the interview documents can be found.

Part 1: Background & Core knowledge

Chapter 2: Dutch electricity sector

As the title probably gave away this chapter will elaborate on the electricity distribution grids in the Netherlands. Since the main question also contains the management of purchase and sale next to control, the electricity (Fens, 2005)purchase/sale aspect is also covered in this section. The first sections will cover Dutch electricity sector and the energy trading followed by smart meters.

2.1 Dutch Electricity Value Chain

The Dutch Electricity sector consists of various parties and in order to get a view of the complete picture we take a look at the energy value chain as defined by (Fens, 2005). A value chain, also known as value chain analysis, is a concept from business management first described by Michael Porter (1985) and can be used for understanding activities in a specific industry. In this case the chain starts at the generation of electricity and at each stage gains more value to the end customer.

Figure 3: Energy value chain

As depicted in figure 3 the stages of the Dutch electricity sector can roughly be categorized in the following subsections (J.Makansi, 2007):

- Generation
- Transmission
- Distribution
- Delivery

In this chain electricity is generated and flows from left to right with the end consumer in the last step. The end consumer pays for the delivered energy, which initiates a monetary flow back to the first stage. The first step in the chain is generation. The production of energy in power plants is usually accomplished by burning fuel to generate steam. The steam is then pressurized and pushed through a turbine, which causes the generator to start working and thus generate electricity. The next step is transmission. In this step the electricity is transported from the power plants to the substations. In order to minimize the loss of electricity during transport the voltage, which can be seen as what drives the electricity down the wire, is increased to anything over 110KV. The responsible party for this part of the chain is TenneT (Chapter 2.1.2.). This is necessary because power plants are usually located far from population centers and increasing the voltage will get the electricity at the substations faster and thus decreasing loss of electricity (J.Makansi, 2007). Once at the substations the electricity is lowered in voltage and distributed to several points close to the end user. Lastly the electricity is transformed again and delivered directly to the end user.

2.2 Transmission System Operator (TSO)

The backbone of the Dutch electricity grid is the part that connects the electricity generating stations to the distribution companies (see figure 4). In figure 4 the part in the middle starting at generating step up transformer up to substation step down transformer is the responsibility of a TSO. The TSO is responsible for the management of this grid and is constantly working at minimizing network failures and delivering the electricity to the distribution companies. Next to the previously mentioned tasks, one of the biggest tasks at hand is that of balancing supply and demand of electricity in order to avoid the grid of getting overloaded.

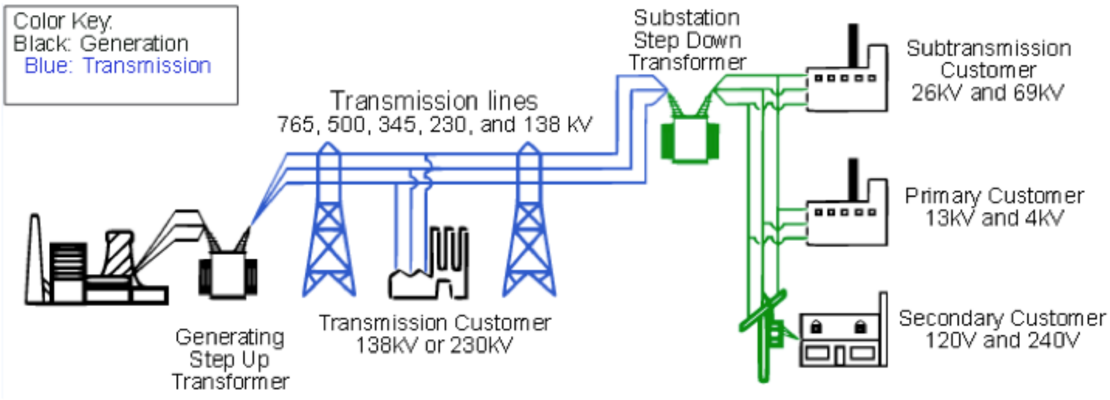


Figure 4: Electricity grid from generation to delivery

The Dutch Transmission System Operator (TSO), TenneT, is responsible for the proper handling of all the electricity above 110KV in the Netherlands and has the Dutch ministry of Finance as the sole shareholder. They currently have about 2685km of 220kv and 380kv high voltage transmission lines and 5716 km of 110kv and 150kv transmission lines in their management (TenneT-group, 2010). As can be seen in figure 4, the TSO (TenneT) is only responsible for the transmission of the electricity and does not generate any electricity itself nor does it deliver electricity directly to business or consumers.

2.3 Distribution System Operator (DSO)

The last two steps in the electricity value chain (recall figure 3) are that of distribution and delivery. As mentioned in the previous section the TSO Handles the high voltage network (HV) whereas the DSO handles the low and medium voltage network (LV and MV). We spend extra attention to the DSO since the part of the network that they control also happens to be the part of the network this research covers.

The main goal of the DSO is to connect the clients to the low voltage network and thus connect any customer to the power grid, and also to enable electricity consumption or production. A DSO has a monopoly in its region and for that reason they remain regulated by the government. Regulatory compliance and performance of the DSO's (in terms of reliability and power quality), are monitored by the Energiekamer. If a DSO fails to perform adequately, they will not be allowed to raise their tariffs for electricity transport (adjusting for inflation). This system forces a competition between DSO's even though they have a regional monopoly. Should one perform less than the others, they will not be allowed to raise tariffs and as a result have less money to spend.

The three following tasks are fundamental to the Dutch DSO's (Fens, 2005, p. 116):

1. Asset lifecycle management
2. Network design optimization
3. Network operations

In the figure below (Figure 5) you can find the Dutch regional DSO's with their market share (B.Baarsma, 2008) based on the residential and small-scale consumers.

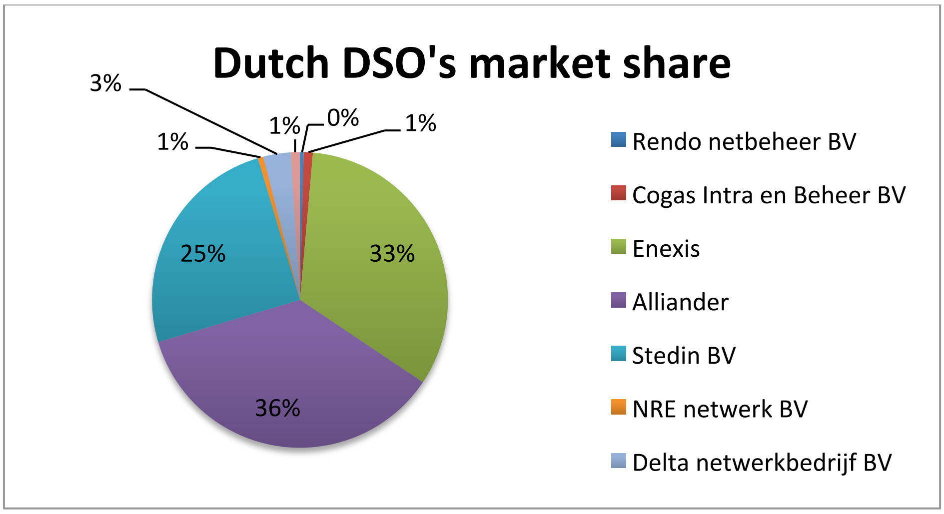


Figure 5: Regional DSO's with market share

When looking at figure 5 (above), we can see that the three biggest DSO's are Alliander, Enexis and Stedin. Together these 3 DSO's have a 94% market share, and therefore manage nearly the whole MV/LV network in the Netherlands. It is for this reason, that the experts consulted during this research (Next to the ones at Logica) are from these companies.

Next to the regional DSO's there are also the delivery companies like Essent, Nuon, Eneco, Oxxio and GreenChoice, which in fact are no more than administrative and commercial contact points for the clients (B.Baarsma, 2008). These companies are the main contact points for every customer that wants to be connected to the grid. As stated in section 2.1, the monetary flows follow the same path backwards; starting at the delivery companies (client pays for electricity) and traveling all the way back to the generation plants.

2.4 Energy Trading

In previous years demand and supply of electricity was balanced by adjusting energy generation to energy demand. In recent years however markets where electricity can be sold and purchased have emerged. These market are quite similar to stock markets and can also determine the price of electricity. The precise working of these markets differ per country but in the most common form there is a company in between the companies (APX-ENDEX) on the demand side who maintain the contracts with the customer and the companies on the supply side who generate the electricity. This means that companies like the APX-ENDEX (discussed in the next subsection) don't have to generate electricity. All they do is buy contracts on the market and sell them and this can be done day-ahead or real-time. If there are any differences between contractual obligations (futures) and actual usage they will be settled afterwards (J.Makansi, 2007).

Amsterdam Power Exchange

Up to a couple of months ago, there were two exchanges for gas and electricity in the Netherlands: The Amsterdam Power Exchange (APX) and the Dutch European Energy Derivatives (ENDEX). APX had its main focus on day-ahead and in day trading while ENDEX was a futures exchange. As of January 5th this year however the two companies have successfully merged into one company[1] and they currently operate under the name APX-ENDEX. The APX-ENDEX is one of Europe's most experienced energy exchanges and, since the merger, operate day-ahead trading, in day trading along with futures trading for electricity and natural gas in the Netherlands, the United Kingdom and Belgium. At the end of 2009 the combined power and gas volume of APX-ENDEX 412 TWh and the exchange had over 300 memberships from more than 15 countries (APX-ENDEX, 2010).

Chapter 3: Smart grid

This chapter will be used to elaborate on the smart grid in general and the problems it is expected to solve. There are 2 main reasons for the consideration of applying a smart grid on national level (K.Dodrill, 2010). Firstly the current electricity grid is aging and out-dated in many respects. Investments are needed to improve its condition and to ensure that the grid has enough capacity to handle the electricity demand. Secondly, and more important for this research, there are several substantial benefits of using CEP when implementing a smart grid.

3.1 Introduction

One of the biggest challenges of current power systems is running a supply on demand system that is truly reliable. In the past this has led to a power system based on a controllable supply to match the highly uncontrolled demand. With the concerns of climate change however, the application of renewable energy sources have risen a great deal and are now reaching significant levels of penetration. The problem this brings is that, since the renewable sources are largely dependent on climate and are therefore not controllable, the supply cannot easily be adjusted to meet the energy demands (Cameron W. Potter, 2009). These developments are making it impossible to continue electricity grid management the way it currently done, and therefor a change is needed. Smart-grid addresses that change by approaching the management of the grid in a different manner. The use of smart-grids allows a more flexible management system, which can be achieved in many ways from Demand Side Management (DSM) to electricity storage.

3.2 Definition

The US National Energy Technology Laboratory (NETL) defines 7 principle characteristics of the smart grid (K.Dodrill, 2010):

- Enables Active participation by customers
- Accommodates all generation and storage options
- Enables new products, services and markets
- Provides power quality for the digital economy
- Optimizes asset utilization and operates efficiently
- Anticipates and responds to system disturbances
- Operates resiliently against attack and natural disaster

Next to these characteristics they also describe the following 6 key values from which the benefits will result after improvements:

- Reliability - Improvements would lead to a reduced frequency and duration of outages, a reduction in the number of disturbances due to poor power quality, and virtual elimination of widespread blackouts.
- Economic - Improvements would be realized from improved meters resulting in more accurate information on the usage of equipment and thus a lower bill.
- Efficiency - Improvements would reduce the cost of producing, delivering and consuming electricity.
- Environmental - Improvements would also be an effect of implementing a smart grid since improved efficiency would lead to a reduction in emissions and discharges when compared to the current situation.
- Security - Improvements would lead to an increase in the robustness and resiliency of the grid from a physical and cyber perspective.

- Safety - Improvements would lead to reduced safety hazards as well as time of exposure to hazards.

From this list we find that the expected improvements resulting from the implementation of a smart grid will be noticeable throughout the whole society. The delivery companies will have the opportunity to significantly reduce their capital costs and earn a return for their shareholders, electricity suppliers have the opportunity to enter new markets created by the smart grid (renewable generation and storage for example), Consumers have the opportunity to reduce their energy bills and reduce the losses caused by outages and power quality events and Society will benefit from a stimulated economy, improved environmental conditions and improved national security.

Since the smart grid and its benefits discussed above are a bit abstract we present you with a couple of smart grid definitions and the one we will adhere to throughout this project.

Epic (Johan Boekema, 2009, p. 7)
A Smart Grid generates and distributes electricity more effectively, economically, securely and sustainably. It integrates innovative tools and technologies, products and services, from generation, transmission and distribution all the way to the customer appliances and equipment using advanced sensing, communication and control technologies. It enables two-way exchange with customers providing greater information and choice, power export capability, demand participation and enhanced energy efficiency.

Xcel Energy Smart Grid (Johan Boekema, 2009, p. 7)
While details vary greatly, the general definition of a smart grid is an intelligent, auto-balancing, self-monitoring power grid that accepts any source of fuel (coal, sun, wind) and transforms it into a consumer's end use (heat, light, warm water) with minimal human intervention.

Wikipedia (Wikipedia group, 2009)
A ***smart grid*** *is a digitally enabled electrical grid that gathers, distributes, and acts on information about the behavior of all participants (suppliers and consumers) in order to improve the efficiency, importance, reliability, economics, and sustainability of electricity services*

Martijn Bongaerts, Netbeheer Nederland (Johan Boekema, 2009, p. 7)
The term "smart grid" is a container-term for the development of intelligent and innovative electricity/gas-networks, capable of balancing peaks and dips in the electricity supply.

The definition found, best matching the line of thought in this research comes from the one defined at Epic

"*A Smart Grid generates and distributes electricity more effectively, economically, securely and sustainably. It integrates innovative tools and technologies, products and services, from generation, transmission and distribution all the way to the customer appliances and equipment using advanced sensing, communication and control technologies. It enables two-way exchange with customers providing greater information and choice, power export capability, demand participation and enhanced energy efficiency.*"

We will look at the smart grid as an interrelated set of consumers (houses), generators (wind turbines, solar panels) and those that do both (houses with CHP and/or solar panels).

3.3 Related work

At the time of this research, there were already a couple of smart grid projects, which just started or were just starting up. In this section, the most important of these pilot projects are described and compared to this research. To conclude this section, the sort of management used for each pilot project will be analysed to see in what aspect these projects differ from this research.

3.3.1 PowerMatching city

The first example we will describe here is a demonstration project, which displays the energy infrastructure of the future. PowerMatching City is an experiment in the Dutch city of Hoogkerk and includes 25 houses connected to each other by smart grid. The main idea is that energy will no longer be generated at central points, but will be decentralized and divided into small sustainable generation plants within the Hoogkerk network. Examples of decentralized generation include solar panels, wind turbines and Combined Heat and Power generators (CHP). There is however a problem with this approach, solar and wind power are heavily dependent on the weather and the use of CHP generators will lead to an excess of electricity during the winter period when a lot of heat is needed. All these fluctuations lead to the fact that usage and generation of energy are not a synchronous process. Solar panels for example generate a lot of electricity during the day, but this is also the time when most people are at work and the electricity demand is lower than in the morning or evening/night. In order to deal with this imbalance of supply and demand the PowerMatcher (figure 9) is introduced.

<u>PowerMatcher</u>

Within a PowerMatcher cluster the agents are organized into a logical tree. At the root you can find the auctioneer agent, which performs the price forming process. The auctioneer collects all the bids from the agents connected directly to it into one single bid, searches for the equilibrium price and communicates a price update back in case of a significant price change. The concentrator agents have the same tasks but do this for a subgroup of devices. Towards the Auctioneer it mimics the device agents, and towards the device agents it mimics the auctioneer. Next to these agents you also have the device agents which act as managers for each separate device. The device agents will bid on electricity units at the concentrator agents and will back out when electricity prices get really high (Supply lower than demand). This will cause the devices with a lower priority to cease their work right away and wait for the prices to drop (demand lower than supply).

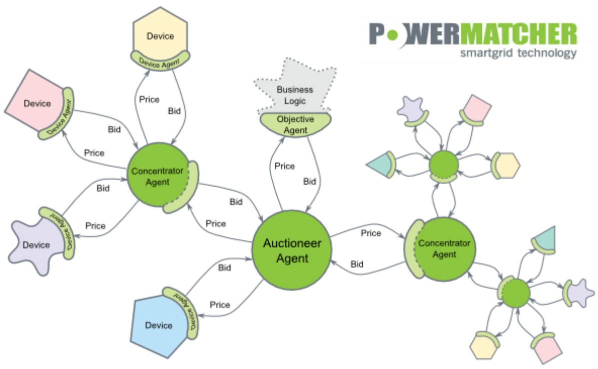


Figure 6: Power Matcher concept

3.3.2 Vacation park Bronsbergen (Zutphen)

In this second example we have a vacation park in the Netherlands, which started out as an experiment in 2005. The project was started, concerning to the CO2 emission reduction and less dependence on conventional fuels, hence the step to renewable energy resources. The park consists of 210 holiday houses, of which 108 have been installed with over 3000 m2 of solar panels on the roof. A MV/LV 10kV/400V, 400kV transformer is used to connect the Bronsbergen network to the national grid (D.T. Ho, 2010). Furthermore the grid consists of 2 battery banks, which are used as electrical storage. In this setting, the transformer can also be used to deliver electricity back to the grid. Figure 7 illustrates an image of the Bronsbergen grid on the left and its simplified single-line diagram on the right.

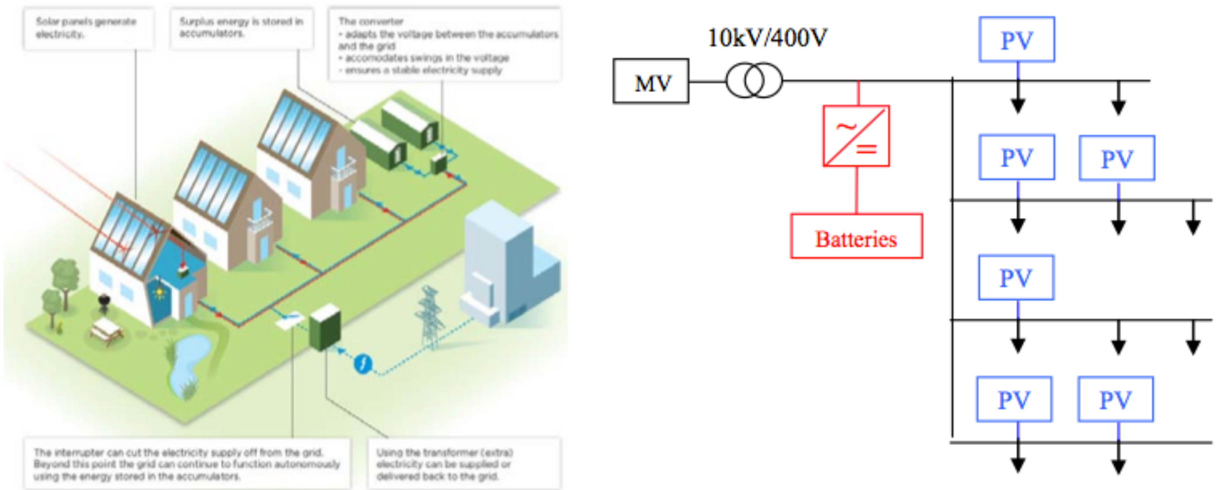


Figure 7: Vakantiepark Bronsbergen (D.T. Ho, 2010)

The main purpose of this grid is to monitor the effects of distributed generation on its network components (harmonic distortion) but its setup is similar to the simulation grid introduced in chapter 6.

3.3.3 The Los Angeles Power Grid

The city of Los Angeles is served by the Department of Water and Power (LA DWP) for it's electric utility needs. DWP is the among the largest public utilities in the united states, serving 4 million residents across about 748 square kilometres. As it turns out, this translates to about 1.4 million electricity and 0.7 million water consumers in that area. Compared to the Dutch utilities market, where there is a separate TSO and several DSO's, this public utility is vertically integrated and controls and operates it's own power generation, transmission and distribution systems. Due to the latest goals of reducing it's carbon emissions and increasing its share of renewable energy sources, the DWP has to find efficient ways to use available capacity on the grid. The increasing global concern on the environment and efficient use of energy is causing countries to invest in improved power infrastructure and research into the optimal use of energy. The Department of Energy (DoE) has started a funding exercise, which would stimulate the utility market to for example identify new energy sources and make the electric grid smarter and therefore the LA DWP has been awarded 60 million USD to research the possibilities (Y.Simmhan, 2011).

For this reason the LA DWP has formed two goals: one, is to reduce per capita power usage overall (lower carbon emissions), and the second, is to reduce the peak power usage overall by shifting power usage to off peak hours (reduces the extent of unused spare capacity). Yogesh Simmhan et al. give an example where the base load on a typical day in LA DWP varies between 2000MW at approximately 4 A.M., and 4800MW at 4 P.M.. They forecast that DWP will not by able to meet its peak load 50% of the time by 2020 without further expansion or load curtailment. DWP is planning to meet 500MW of peak load reduction through Demand Response (DR) programs to control and shit load during peak hours (Y.Simmhan, 2011).

They have started a demonstration project, which entails several activity areas all of which are supported by the large-scale installation of Smart Meters at DWP consumers used as test beds to research *demand response optimization*. The main idea behind their demand response optimization is by offering incentives to consumers to reduce energy consumption when a peak load or loss of reliability situation is encountered and this can be achieved by having a prior commitment by consumers to reduce load during power shortage, or using a variable rate market model that increases prices during shortage. Prior commitment by consumers would allow the control to turn off the consumers' equipment or send a signal to the consumer to reduce consumption in return for consumer incentives (Y.Simmhan, 2011).

3.3.4 Boulder city Colorado (XCEL Energy)

This smart grid pilot project started in 2008 and was aimed at the entire city of Boulder Colorado. The utility estimated its share of the cost to sum up to 15 million dollars and that investors would fund the rest. It is stated that the grid should be able to allow customers and utilities to collaboratively manage power generation, delivery and energy consumption. The key components of Smart Grid City include (Xcel-Energy):

- A dynamic system rich in information technology
- High-speed, real-time, two-way communications
- Sensors throughout the grid enabling rapid diagnosis and corrections
- Decision-making data and support for peak efficiency
- Distributed generation technologies (Such as wind turbines, solar panels, and plug in hybrid electric vehicles)
- Automated smart substations
- In- home energy control devices
- Automated home energy use

By 2009 Xcel Energy had installed sensing and communications equipment to monitor the Boulder distribution grid covering over 45,000 homes. Currently they have installed more than 20 disparate software applications and 95 new data integration interfaces to support SmartGridCity, along with approximately 4,500 grid condition sensors and 24,000 smart meters (MetaVu Xcel Boulder city evaluation, 2011). Information about the actual management system used in this project has proven very hard to find and can therefore not be included in this subchapter. The reason this project is still discussed here comes from the lessons learned in this pilot project.

Due to lack of a proper cost benefit analysis, the project exceeded its budget by nearly triple the initial amount adding up to 44,8 Million dollars (Jeffe, 2010). According to Xcel, the cost drivers were the added expense in laying fiber-optic cables to each of the 23000 smart meters and the necessary software to run the system.

3.3.5 Related work conclusion

Though several smart grid pilot projects are available throughout the world, it has proven rather difficult to acquire the level of information needed in order to make sense out of these different projects and compare them to the approach presented in this thesis. The previously mentioned projects are all focused on demand response optimization in which the utilities try to match the energy supply to demand. The main focus across most of these projects lies within the houses and making the components inside each house "smart". These smart components can then be switched on or off remotely, so that the management system can decide when is the best time to switch on each device. Using this technique power usage can be aligned with generation resulting in a more stable and efficient electricity grid.

Xi Fang et al. describe the smart grid as a whole by defining three so-called systems of which the smart grid consists (X. Fang, 2011):

- Smart infrastructure system – *The energy, information, and communication infrastructure underlying of the smart grid that supports: 1) advanced electricity generation, delivery, and consumption; 2) advanced information metering, monitoring, and management; and 3) advanced communication technologies.*
- Smart management system - *The smart management system is the subsystem in SG that provides advanced management and control services.*
- Smart protection system - *The smart protection system is the subsystem in SG that provides advanced grid reliability analysis, failure protection, and security and privacy protection services.*

The main focus of this research is on the part they defined as the "Smart management system", and the reason their work is presented here is because they also analysed which methods are currently being used for each of the three categories. In the figure below (figure 8) you can find the methods found for smart management (X. Fang, 2011).

Figure 8: Smart management methods

Most of the previously defined projects fall under the Action category as demand side management forms the central part of the management system. Complex Event Processing does not really fall in to one of these defined categories as it can be used for multiple purposes (Optimization, Auction) and can be combined with machine learning for example, which gives it an advantage over these methods.

X Fang et al. also state that there are currently approximately 90 smart grid related projects running all over the world as can be seen in the following figure.

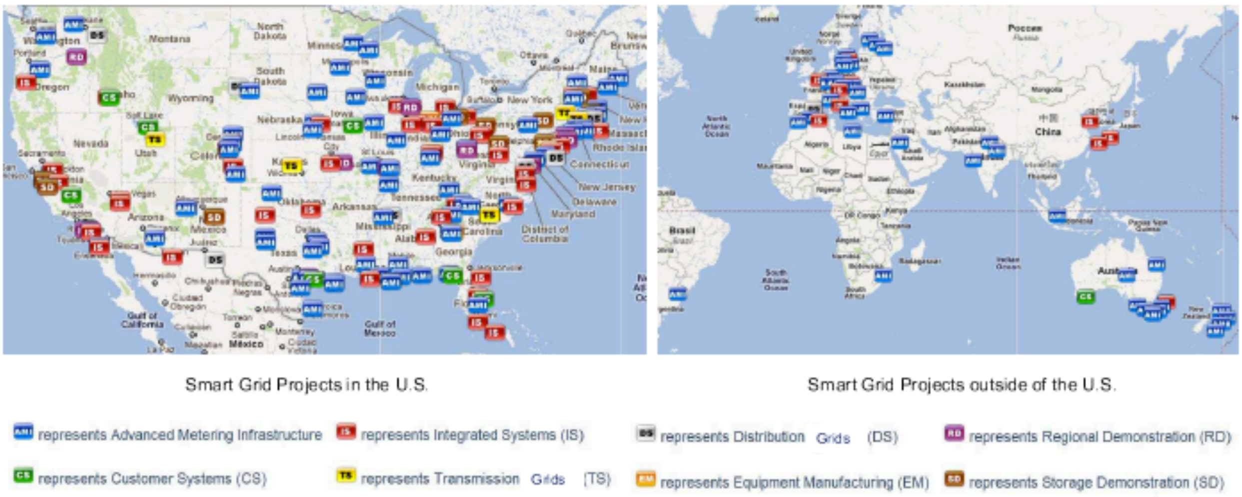


Figure 9: Smart grid pilot projects

As you can see there are several smart grid projects running, but in different directions. The important ones for this research are the blue (AMI) and red (IS) projects. The AMI projects focus more one smart metering and smart houses (house components) whereas the IS focuses on the entire grid (This project). Though there is some information available on most of these projects, the information found doesn't go in detail, which is why they are not mentioned in this chapter.

Chapter 4: Event Processing

An owl hunts mice at night and will pounce on a mouse from flight when it detects it moving on the ground. The mouse might sense this and respond by jumping out of the way but will die if it responds inappropriately, this is referred to as a sense and respond system (K.Chandy M. A., 2007). What we learn from this is that a living being that does not detect threats gets killed. Likewise, one that frequently takes unnecessary action wastes energy and perishes. It is this same basis that has led event processing to where it is today and in a sense businesses work in the same way. In 1999 S.Haeckel has studied sense and response systems from an organizational perspective (S.Haeckel, 1999), so event processing has a long history, of which a great deal in computer science. If we apply this concept to the business we discussed earlier in this document (Dutch electricity sector Chapter 2.) we see that it fits perfectly. A central sense and respond system could monitor the entire network based on real-time incoming events. Upon receiving these events, actions and notifications can be generated. This sort of sense and respond system based on event processing could sense changes in the electricity grid (connector failures, broken cables) or in energy prices (buy electricity from the net, use own CHP or sell energy using the APX) and respond accordingly.

This chapter will cover all we need to know about event processing. We start with elaborating on the basic notion of an event, followed by the different types of event processing approaches. The next section dives back into the beginning of event processing in the form of situation awareness. In the section on situation awareness, the principles are linked to the current approaches. We end this chapter with the different kinds of Complex Event Processing-suites available for use and select the one best suited for this project.

4.1 Events

Before we can begin to understand this notion, it is necessary to define the word "event" as it will be used throughout this document and create a general idea about what is meant by it. An event can be defined as a significant *change* in state (K.Chandy, 2006), notice how the emphasis is on change rather than the state itself. A significant change (notable event) is one in from which some kind of response is required and an insignificant change (ordinary event) is one where no direct action is needed.

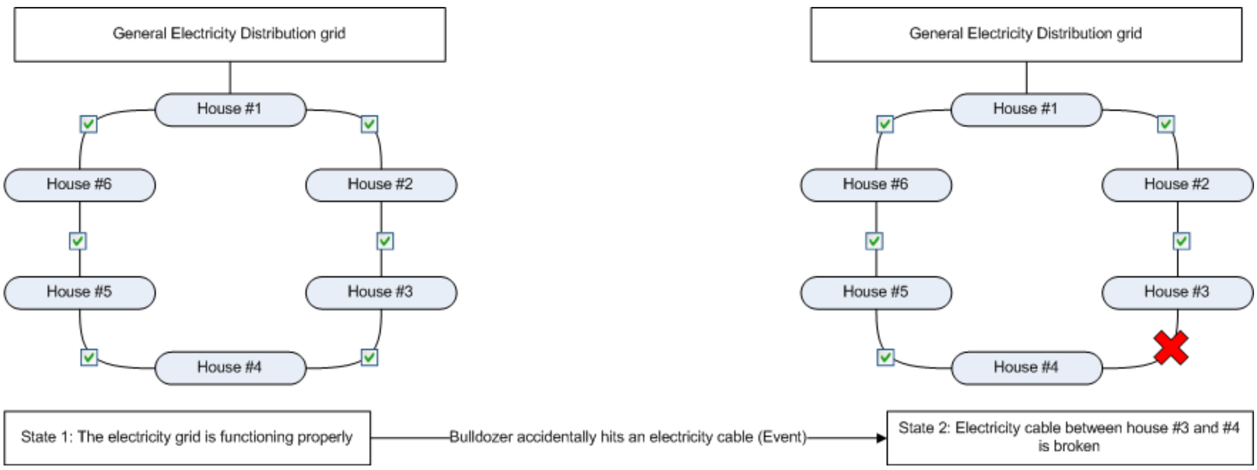


Figure 10: An event changing the state of an electricity grid

In the figure above you can see an electricity grid (fictional) consisting of 6 houses in two states. It depicts the event of a bulldozer breaking an electricity cable, causing a state change from 1 To 2, where the cables between house #3 and #4 are broken.

4.2 Event Processing approaches

The Patricia Seybold Group defines three kinds of event processing methods, the first one being simple event processing, the second one stream event processing and the third complex event processing (Michelson, 2006). In order to get a better grasp of the different types of event processing we first take a look at event flows to aid in describing the types and divide an event flow into 4 sections (Michelson, 2006). The first part is that of the event generators which, as the name implies, is the source of the generated events in the system. Secondly we have the event channel which can be seen as the messaging backbone of the system transporting the messages between event generators, processing engines and downstream subscribers (Michelson, 2006). The third part is the event-processing engine, which evaluates the received events against the event processing rules causing actions to be initiated. The last part of the event flow is the downstream event driven activity, which initiates numerous downstream activities. In the next subsections we will take a look at each of the event processing approaches mentioned above.

4.2.1 Simple Event Processing

In simple event processing, a notable thing happens, initiating downstream action(s) (Michelson, 2006). An example of this could be you standing at a car dealership looking for a new car. A form of simple event processing initiating downstream of actions would be the store manager requesting a new model for display, because of the event that an older model was sold and left an empty spot at the dealership. The request would be done at the dealer who then in turn requests shipping for the car to the location of the dealership. Upon delivery the store manager is notified and he arranges for an employee to pick up the car and drive it to the dealership where he notifies the store manager that the car arrived and the manager picks a spot for the car to be displayed. This form of simple event processing is commonly used to drive the real time flow of work, taking the lag and cost out of a business.

4.2.2 Stream Event Processing

In stream event processing several sources generate events, which are then processed directly (Michelson, 2006). Both notable and ordinary events can occur. Ordinary events can be screened for notability and streamed to information subscribers (Michelson, 2006).

4.2.3 Complex Event Processing

The difference between Complex Event Processing and Stream Event Processing is that CEP also allows you to access events that happened in the past and use them in any order whereas SEP only allows you to use and analyse the events in the order they arrive. The event can come from various sources and may occur over a long period of time. CEP requires sophisticated event interpreters, event pattern definition and matching along with correlation techniques.

4.3 Situation Awareness

Even before sense and respond systems as defined by S.Haeckel in 1999 (S.Haeckel, 1999), Mica R. Endsley researched the theory of situation awareness (SA) in dynamic systems (Endsley, 1995, pp. 32-34). Though initially the concept of situation awareness describes a model based on its role in dynamic human decision-making in a variety of domains, it is this same concept, which led to the event processing concepts we know today. Some of the theories used in Endsley's research date back to 1989. It is for this reason that the main steps of SA are discussed here to present you with a clear picture of Event Processing and its history.

We start by presenting the main definition of situation awareness (Endsley, 1995, p. 36):

"Situation Awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future".

From this definition it is already clear that there is a strong resemblance with Event Processing, as we know it today. There is a perception of the environment within a volume of time and space (event processing also reacts on a perception (or state) of the environment), the comprehension of their meaning (this step can be seen as the rules of CEP which add meaning to certain events), and the projection of their status in the near future (This part concerns the decisions that will be made based on the comprehension). The model defined for SA (Figure 9) shows three levels in Situation Awareness. We discuss these levels below:

Perception (Level 1 SA): The first step in achieving SA is to perceive the status, attributes, and dynamics of relevant elements in the environment. Thus, Level 1 SA, the most basic level of SA, involves the processes of monitoring, cue detection, and simple recognition, which lead to an awareness of multiple situational elements (objects, events, people, systems, environmental factors) and their current states (locations, conditions, modes, actions).

- ***CEP comparison****: this is about monitoring events and the states of entities - basic CEP functionality.*

Comprehension (Level 2 SA): The next step in SA formation involves a synthesis of disjointed Level 1 SA elements through the processes of pattern recognition, interpretation, and evaluation. Level 2 SA requires integrating this information to understand how it will impact upon the individual's goals and objectives. This includes developing a comprehensive picture of the world, or of that portion of the world of concern to the individual.

- ***CEP comparison****: this is about recognizing patterns of events and using reasoning tools (such as inference rules) to interpret and evaluate these patterns. Goals and objectives are usually represented as end states in a CEP system.*

Projection (Level 3 SA): The third and highest level of SA involves the ability to project the future actions of the elements in the environment. Level 3 SA is achieved through knowledge of the status and dynamics of the elements and comprehension of the situation (Levels 1 and 2 SA), and then extrapolating this information forward in time to determine how it will affect future states of the operational environment.

- ***CEP comparison****: this is about associating certain complex events with predictions about likely future states and reporting or making decisions based on them.*

SA also involves both a temporal and a spatial component. Time is an important concept in SA, as SA is a dynamic construct, changing at a tempo dictated by the actions of individuals, task characteristics, and the surrounding environment. As new inputs enter the system, the individual incorporates them into this mental representation, making changes as necessary in plans and actions in order to achieve the desired goals. SA also involves spatial knowledge about the activities and events occurring in a specific location of interest to the individual. Thus, the concept of SA includes perception, comprehension, and projection of situational information, as well as temporal and spatial components.

- ***CEP comparison**: this is about the importance of time and location. Indeed, you could substitute CEP for SA and still have a meaningful paragraph above.*

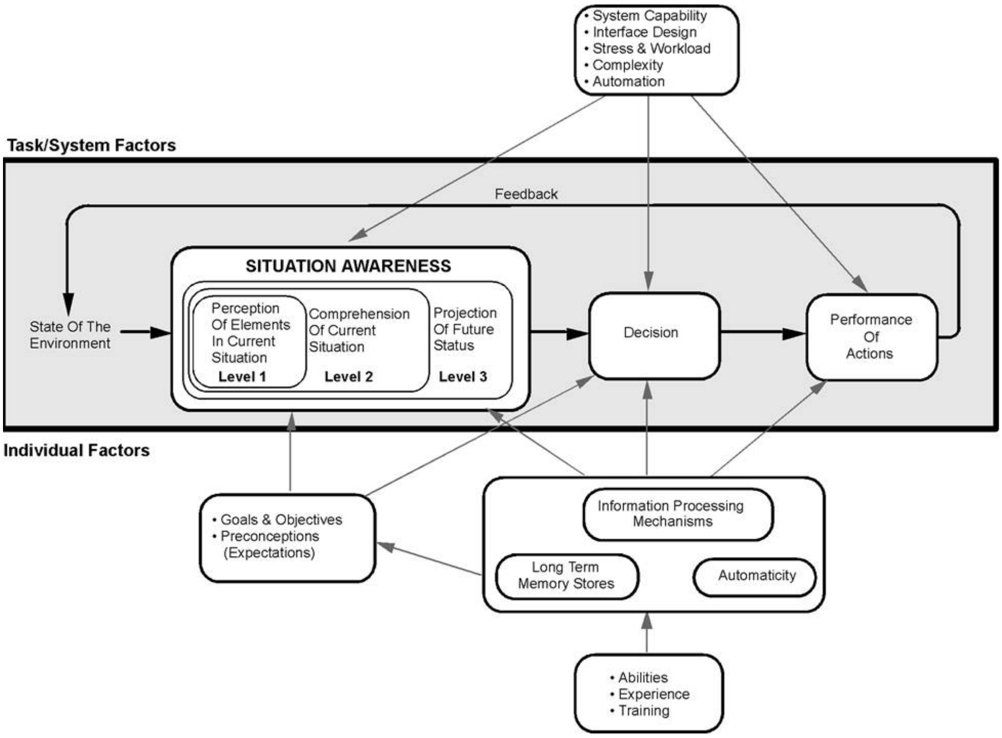


Figure 11: Model of situation awareness in dynamic decision-making (Endsley, 1995)

4.4 Alternative Event Driven technologies

Next to Complex Event Processing there are several other technologies, which can be used in event driven situations. The most important ones are briefly described in this section and will be compared to CEP. The main purpose of this section is to elaborate on why CEP was the technology chosen for this specific setting. We start by stating the most important Event Driven technologies (next to CEP) in the table below and also include a brief explanation of each technology.

Technology	Explanation
Business Activity Monitoring (BAM)	Used to provide real-time access to critical business performance indicators to improve the speed and effectiveness of business operations. Unlike traditional real-time monitoring, BAM draws its information from multiple application systems and other internal and external (inter-enterprise) sources, enabling a broader and richer view of business activities.
Business Process Management (BPM)	An activity undertaken by businesses to identify, evaluate, and improve business processes. With the advancement of technology, BPM can now be effectively managed with software that is customized based on the metrics and policies specified by a company. This type of action is essential to businesses seeking to improve process performance related issues so that they can better serve their clients.

Active databases	A database that includes an event driven architecture, which can respond to conditions both inside and outside the database. Possible uses include security monitoring, alerting, statistics gathering and authorization. Most modern relational databases include active database features in the form of SQL Triggers.
Operational Business Intelligence (BI)	Computer-based techniques used in spotting, digging-out, and analyzing 'hard' business data, such as sales revenue by products or departments or associated costs and incomes. Objectives of a BI exercise include (1) understanding of a firm's internal and external strengths and weaknesses, (2) understanding of the relationship between different data for better decision making, (3) detection of opportunities for innovation, and (4) cost reduction and optimal deployment of resources. See also competitive intelligence.

Table 2: Event Driven Technologies

Though all of the technologies described above (table 2) are based on dealing with events within a business or system, each of them is more limited than CEP (Mike Gualtieri, 2009, p. 2). Complex Event Processing excels in handling live data, diverse data, pattern detection, diverse business processes, and large volumes of data. Research has shown that out of the technologies discussed up to this point, CEP is best when used in dynamic, near real-time, rapidly changing situations (Mike Gualtieri, 2009, p. 4). Forrester defines the following 3 key advantages of using CEP (Mike Gualtieri, 2009):

1. *Detect and inform applications → CEP applications can act as intelligent information filters in real-time applications such as fraud detection applications, and other applications that must detect relationships within continuous streams of data.*

2. *Detect and act applications → CEP applications can recognize patterns of events and respond to those patterns by kicking off a process in an external application or pushing a critical event to a downstream system. CEP applications can also detect the absence of an expected pattern within a period of time.*

3. *Event processing architectures → Organizations can also use CEP to coordinate business processes that involve a jumble of heterogeneous information systems. Business processes are often triggered by one simple event, or sometimes by a complex pattern of events, and a CEP application can easily detect these events and patterns.*

When looking at these advantages compared to the other techniques available we see that CEP is indeed the best technology for this kind of problem. Due to the vast amount of information (events) this network is going to generate, some sort of information filtering (1) will be necessary to select the most important events and patterns. Also in every case some act should be performed upon detection of a pattern (or pattern absence) (2). Furthermore the system can easily be altered to add more functionality in the form of business processes (the addition of a new security policy for example)(3).

4.5 CEP suite selection

There are several CEP platforms available for use nowadays and in this chapter we give an overview of these platforms and make a selection on which to use for this project. Before we present you with the available suites and start analysing them, it is important we now exactly what a CEP suite is. The definition found, best matching the ideology in this research comes from Forrester and is the following (Mike Gualtieri, 2009, p. 2):

"A *software infrastructure that can detect patters of events (and expected events that didn't occur) by filtering, correlating, contextualizing, and analysing data captured from disparate live data sources to respond as defined using the platform's development tools.*"

Based on this definition we will continue our research in the next section by selecting the available suites. For every found suite we will give some basic information and state their strong points and weaknesses in general. Furthermore it has also been stated that that typical CEP systems cost on average, between 100,000 and 250,000 dollars (Leavitt, 2009, p. 17), a fact that is important when considering the budget for such a system. In the second section we will check which of these points are important for this research and select one CEP suite, which we will used to implement the demo.

4.5.1 Available CEP Suites

There are a several CEP suites available at the moment and in order to make a selection on which ones to analyse we first define a couple of selection criteria by which the suites must comply. All evaluated platforms have the following features (Mike Gualtieri, 2009):

- **Employ CEP as the primary programming model for custom application development** – CEP platforms that are designed to develop a wide range of custom applications.
- **Support the full application life-cycle** – CEP platforms that incorporate application development tools, runtime environments, and operational management facilities
- **Are being used by customers** – only CEP platforms that could provide at least two customer references, which have successfully implemented the suite in a production environment for an on-going business concern, are included.
- **Are sold and supported as independent CEP platforms** – Products that embed CEP in applications solutions or other development platforms are excluded
- **Are available** – To be included, the suite had to be available for customers by April 2010 (Start of research assignment)

The following table gives a list of the suites found based on the selection criteria, the results of the paper by Forrester was updated to include the recent changes.

Vendor	Product evaluated	Product version evaluated	Version release date
TIBCO Software	TIBCO business events	3.0	January 2012
Espertech	Esper enterprise edition, Esper for Java	4.5.0	January 2012
IBM	Websphere Business Events	7.0.1.1	November 2010
Oracle	Oracle CEP	11gR1	May 2011
Progress Software	Progress Apama	4.3	December 2011
StreamBase Systems	StreamBase Event Processing Platform	7.0.5	

Sybase	Sybase CEP	7.6	

Table 3: Evaluated CEP platforms

In the table below we present you with a brief summary of the Advantages and weaknesses of each platform.

Vendor	**Advantages**	**Weaknesses**
TIBCO Software	▪ Large userbase ▪ Strong product features ▪ Easy to use eclipse like interface	
Espertech	▪ The only open source Event Stream Processing engine for Event-Driven Architectures ▪ No risk at trying for simple things and moving on to more critical applications ▪ Fully embeddable in existing java platforms	▪ Lacks strong tools for business end users and administrators ▪ Small userbase
IBM	▪ Empowers business users to define and proactively manage business events ▪ Improves line of business insight and awareness of event driven business situations	▪ Installation and customization are less simple than main competitors
Oracle	▪ Capable of delivering extreme throughput and latency performance	▪ Needs to improve tools for developers and business end users
Progress Software	▪ Large userbase ▪ Product features ▪ Strength in runtime architecture, platform administration, CEP features and tools for developers	
StreamBase Systems	▪ Event processing features, business end tools	
Sybase	▪ Rapid application development through modeling toolkit ▪ Strength in runtime architecture, platform administration, CEP features and tools for developers	

Table 4: Platform advantages and weaknesses

The leaders in the CEP section are currently Progress Software with their Apama suite and TIBCO Software with their business events suite. These platforms provide a good development environment, dashboard and analysis tools next to a large user base (handy when troubleshooting). It is for this reason that I would recommend these platforms when actually implementing such a system (costs should still be considered). For this project however Esper was chosen because of its open source nature. Even though like many open source projects, Esper lacks the tools for business end users and administrators, which put it at a disadvantage against other platforms (Mike Gualtieri, 2009), Esper was still selected for use in this project. It's highly embeddable Java architecture, strong CEP feature set, high throughput (hundreds of thousands of events per second), and open source status make it the top candidate for this research. In the next section we elaborate on the Esper platform and present you with its basic features.

4.5.2 Esper for Java

Esper is an Event Stream Processing (ESP) and event correlation engine (CEP, Complex Event Processing). Targeted to real-time Event Driven Architectures (EDA), Esper is capable of triggering custom actions written as Plain Old Java Objects (POJO) when event conditions occur among event streams. Like the other applications mentioned, the Esper engine has been developed to address the requirements of applications that analyze and react to events.

Esper provides the user with two principal methods to process events. The first one is that of event pattern matching and the second one is by using stream queries. Underlying the pattern-matching engine is a state machine implementation. This method of event processing matches expected sequences of presence or absence of events or combinations of events. It includes time-based correlation of events.

In standard databases the query language (SQL) is used to retrieve data from information stored on hard disks. This approach tends to be too slow in a setting where hundreds or more events have to be analyzed. It is for this reason that the Esper database works "upside down". Instead of storing events and running queries on them, queries are stored and are performed on the events passing through the system. This approach resulted in the Event Processing Language explained below.

Esper also offers event stream queries that address the event stream analysis requirements of CEP applications. Event stream queries provide the windows, aggregation, joining and analysis functions for use with streams of events. These queries are following the EPL syntax. EPL has been designed for similarity with the SQL query language but differs from SQL in its use of views rather than tables. Views represent the different operations needed to structure data in an event stream and to derive data from an event stream.

Event Processing Language (EPL)

As mentioned previously the Esper techniques are based on the Event Processing Language. EPL statements derive and aggregate information from one or more streams of events, to join or merge event streams, and to feed results from one event stream to subsequent statements. EPL is similar to SQL in it's use of the *select* clause and the *where* clause. However EPL statements instead of tables use event streams and a concept called *views*. Similar to tables in an SQL statement, views define the data available for querying and filtering. Views can represent windows over a stream of events. Views can also sort events, derive statistics from event properties, group events or handle unique event property values.

In order to clarify these approaches I restate some examples found in the Esper reference manual (Espertech, 2012, p. 304).

This is a sample EPL statement that computes the average price for the last 30 seconds of stock tick events:

```
select avg(price) from StockTickEvent.win:time(30 sec)
```

A sample EPL that returns the average price per symbol for the last 100 stock ticks.

```
select symbol, avg(price) as averagePrice
   from StockTickEvent.win:length(100)
group by symbol
```

This example joins 2 event streams. The first event stream consists of fraud warning events for which we keep the last 30 minutes (1800 seconds). The second stream is withdrawal events for which we consider the last 30 seconds. The streams are joined on account number.

```
select fraud.accountNumber as accntNum, fraud.warning as warn, withdraw.amount as amount,
```

```
    MAX(fraud.timestamp, withdraw.timestamp) as timestamp, 'withdrawlFraud' as desc
from FraudWarningEvent.win:time(30 min) as fraud,
    WithdrawalEvent.win:time(30 sec) as withdraw
where fraud.accountNumber = withdraw.accountNumber
```

Event Pattern Matching

Event patterns match when an event or multiple events occur that match the pattern's definition. Patterns can also be temporal (time-based). As mentioned earlier, pattern matching is implemented via state machines.. In order to match events with predefined patterns, operators are used to connect these statements. Expressions can contain further nested pattern expressions by including the nested expression(s) in round brackets. There are 5 types of operators that can be used within the Esper environment:

- Operators that control pattern finder creation and termination: *every*
- Logical operators: *and, or, not*
- Temporal operators that operate on event order: *-> (followed-by)*
- Guards are where-conditions that filter out events and cause termination of the pattern finder, such as *timer:within*
- Observers observe time events as well as other events, such as *timer:interval, timer:at*

In the following section a couple of pattern matching examples are given using some of the operators mentioned above. These samples are restated from the Esper Reference Manual (Espertech, 2012):

A sample pattern that alerts on each IBM stock tick with a price greater than 80 and within the next 60 seconds:

```
every StockTickEvent(symbol="IBM", price>80) where timer:within(60 seconds)
```

A sample pattern that alerts when event A occurs, followed by either event B or event C:

```
A -> ( B or C )
```

An event pattern where a property of a following event must match a property from the first event:

```
every a=EventX -> every b=EventY(objectID=a.objectID)
```

Combining Patterns Matching with Event Stream Analysis

Last but not least I also provide an example using pattern matching and stream analysis. Patterns match when a sequence (or absence) of events is detected. Pattern match results are available for further analysis and processing.

The pattern below detects a situation where a Status event is not followed by another Status event with the same id within 10 seconds. The statement further counts all such occurrences grouped per id.

```
select a.id, count(*) from pattern [
    every a=Status -> (timer:interval(10 sec) and not Status(id=a.id)
] group by id
```

Whether using Event Pattern Matching, Stream analysis or a combination of both, these techniques will form the basis of the CEP systems. The rules which in the end initiate all the actions will be created using statements of the form given in the examples.

Chapter 5: System architecture

Now that the main subjects regarding the proposed system (Dutch electricity sector and Complex Event processing) have been covered, and the tool that will be used to develop the Complex Event Processing engine, it is time to take a look at the general architecture aspects of the complete system. The architecture will define the way in which the CEP engine (will be developed using chosen toolkit Chapter 4.5) will fit in the whole system. We start by giving some general knowledge about architectures and there function (5.1, 5.2), followed by the definition of the Service Oriented Architecture (SOA), which in turn gives us the basic knowledge to get a better grasp of the Event-Driven Service Oriented Architecture (EDSOA). The EDSOA will form the basis of the architecture presented at the end of this chapter. In this architecture we define the use of an Enterprise Service Bus (ESB) to connect all the components to each other, which is way the ESB will also be elaborated on in this section.

5.1 Architecture

The need for architecture and structure has always been a part of business but is becoming increasingly more obvious within the Information System and information technology area (A.T.M.Aerts, 2003). The step to smart grid networks brings with it a large part of Information Technology. Take a look at the definition of enterprise architecture below:

*"**Enterprise Architecture** – the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution."* (Sessions, 2007).

What we see here is that an architecture describes the complete system and the principles guiding its design and evolution. So in order to make sense of this complex grid, it is key that we also plan and structure the system using an architecture. Another very important benefit of architecture except bringing structure is increased flexibility. Flexibility will support adaptation of information systems to different situations. Architecture will provide top-end flexibility since changes can be made to the architecture resulting in a different system as well as bottom-end changes in which existing components can be replaced by new ones or new ones can be added. Before diving into the system architecture in this chapter, we take a look some other relevant terms of architecture in general.

*"**Architect** – one whose responsibility is the design of an architecture and the creation of an architectural description."* (Sessions, 2007).

*"**Architectural artifact** – a specific document, report, analysis, model, or other tangible that contributes to an architectural description."* (Sessions, 2007).

*"**Architecture framework** – a skeletal structure that defines suggested architectural artifacts, describes how those artifacts are related to each other, and provides generic definitions for what those artifacts might look like."* (Sessions, 2007).

In the first section the notion of enterprise architecture is elaborated upon, followed by an explanation of the main architecture types and the application architecture best suited for the proposed CEP application.

5.2 Enterprise Architecture

For centuries, architecture has been used to plan houses and even cities. Enterprise architecture can be seen as its counterpart within a business and is used for planning and structuring information technology aligned with business needs and business strategy. As previously stated, (Sessions, 2007) gives the following definition for enterprise architecture:

“***Enterprise Architecture*** *– the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.”*

Whittle and Myrick state that many enterprises don’t have a formal business structure and emphasize that enterprise is not about chaos (R. Whittle, 2005). Every business, including the ones in the utilities market, has the need to achieve business goals, and enterprise architecture will provide the necessary support to achieve this goal. The key is to create a link between the strategy, its supporting architectures, and its planned activities (R. Whittle, 2005). The figure on the next page (figure 10) gives a basic overview of the links defined by Whittle and Myrick.

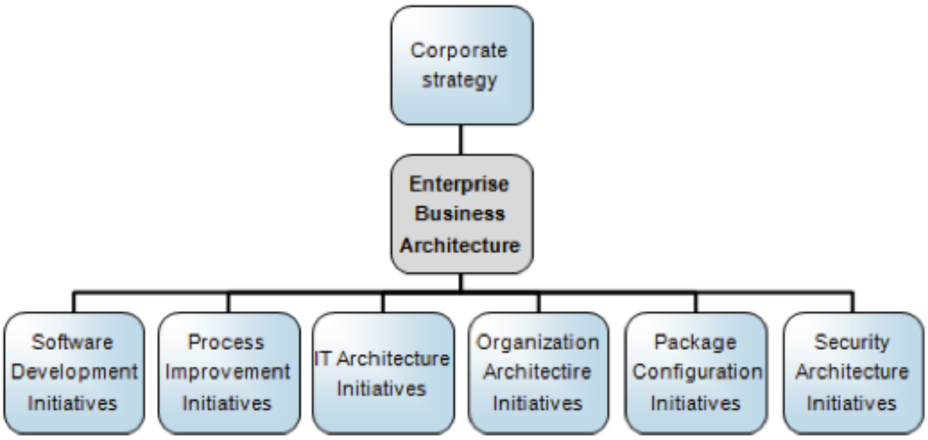


Figure 12: Formal links Whittle and Myrick (2005)

As the figure above depicts, enterprise architecture can be seen as the connecting node between the different aspects within a business. Over time, the scope of enterprise architecture has expanded and now includes business, information, application and technology domains (A.T.M.Aerts). In the next section (5.2) we elaborate the domains relevant to our research and conclude this chapter with a section on application architecture (5.3).

5.3 Architecture types

As could be seen in the previous section, Whittle and Myrick define several domains in their formal links model and Aerts expanded the selection over time. In this section we will elaborate (high level) on the domains relevant to this research.

5.3.1 Business architecture

The business architecture describes the business objects, functions, processes, actors, and how the business should function. What a business architecture aims to do is clarify the complexity within an organization and form a useful starting point, which can then be used to develop information, process and application architectures (G.Versteeg, 2006). A business architecture should be defined prior to any further individual aspects since it arranges the responsibilities around the most important business activities (G.Versteeg, 2006). According to G.Versteeg, a business architecture should contain the following:

1. A lay-out of business domains (including their occurrences on various levels) and their assigned business activities and added value ("business case")
2. Business functions and business concepts (high-level data descriptions), that these business domains need (and are responsible for) to perform their assigned business activity.
3. High-level business processes, which show how these domains work together to achieve the organizational goals and strategies.

The figure on the next page (figure 11) shows the relation between business strategy, business architecture and subsequent architectures.

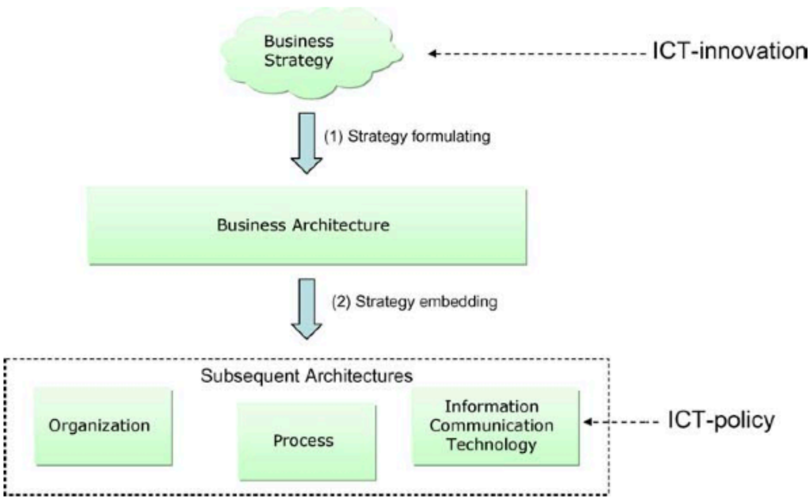


Figure 13: Business relation (G.Versteegh)

What we can see in this model is that a business strategy is transformed to the corresponding architectures. In this research the business strategy could be to stay ahead of the competition by innovation, which in turn could be done by converting their part of the aging grid to a smart grid. The resulting business architecture and subsequent architectures would that be completely focused on achieving this goal.

5.3.2 Information systems/Application architecture

Information systems (IS) architecture is amongst the most common architecture areas within organizations. Also called Application architecture it entails the software application components and their interaction (A.T.M.Aerts, 2003). IS architecture has become highly distributed, consisting of relating components that hold functionality (A.T.M.Aerts, 2003). The mapping and integration of components and systems forms the main part of IS architecture and is often referred to as the integration architecture. Integration of components brings a couple of challenges as described by Magoulas and Pessi. They state that the number of systems within enterprises is growing through mergers and acquisitions, which need to be maintained, updated or removed. Existing systems need to function and interact with new systems and businesses (Thanos Magoulas, 1998). In the case of the proposed system, further research in this direction will have to be done to find out what effect the system will have on the rest of the business.

5.3.3 Security architecture

According to the Electric Power Research Institute (EPRI) ((EPRI), 2009), One of the biggest challenges in the adoption of the Smart Grid is related to the cyber security of systems. The authors of the EPRI report state that cyber security forms a critical issue due to the increasing potential of cyber attacks and incidents against this critical sector as it becomes more and more interconnected. In the figure below (figure 12) the blue lines depict the communication between the components in a smart grid network.

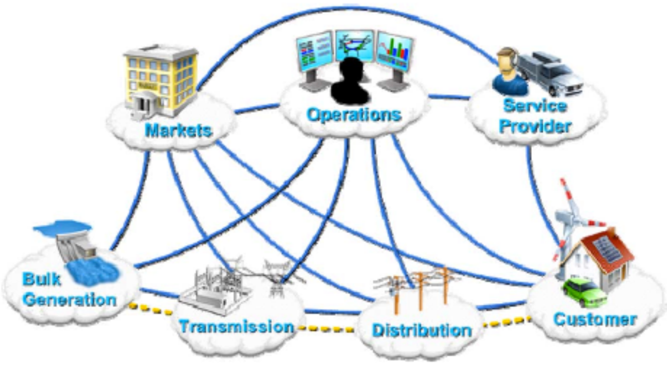


Figure 14: Communication between smartgrid components (EPRI)

If we take a look at figure 12 the CEP system developed during this research would be placed in the operations component with communication lines between the distribution and costumer components. Though there are many organizations working on the development of smart grid security requirements, the National institute of Standards and Technology (NIST) and the Smart Grid Interoperability Panel (SGiP) are the most prominent source of security requirements for the smart grid and should therefore be taken in to consideration when creating the security architecture for a DSO in the utilities market (Anthony R. Metke, 2010). In order to succeed in implementing such a system in reality, it is key that the security architecture is built from the beginning and the first steps to building a good security architecture layer is selecting a cohesive set of requirements and standards for smart grid security.

5.4 Service Oriented Architecture

Since the Event-Driven architecture (was used in the developed system) is often mixed with the Service Oriented Architecture we discuss the latter in this section. There are several definitions for SOA and below we look at three of them and extract the main idea of what a Service-Oriented Architecture actually is.

H. Haas, A. Brown – Web services glossary. W3C Working Group Note (Hugo Haas, 2009)
"A set of components which van be invoked, and whose interface descriptions can be published and discovered."

Thomas Earl - Service-Oriented Architecture: Concepts, Technology and Design (utschig-utschig, 2008)
"A term that represents a model in which automation logic is decomposed into smaller, distinct units of logic. Collectively, these units comprise a larger piece of business automation logic. Individually, these units can be distributed."

Boris Lublinsky – Defining SOA as an architectural style (Lublinsky, 2007)

"SOA can be defined as an architectural style promoting the concept of business-aligned enterprise service as the fundamental unit of designing, building, and composing enterprise business solutions. Multiple patterns, defining design, implementations, and deployment of the SOA solutions, complete this style."

What we see here is that a SOA describes a system to be broken down in smaller distinct units of logic (components) that maintain their own state (Hugo Haas, 2009). Furthermore these components must have interface descriptions, which can be published and discovered. This will result in a loosely coupled system in which it is possible to look for services and invoke these in your own services without having to build the functionality yourself. The interface contract to the service should also be platform independent (Hugo Haas, 2009), this implies that a client from anywhere on any operating system in any language can consume the service. The main idea is presented in figure 13.

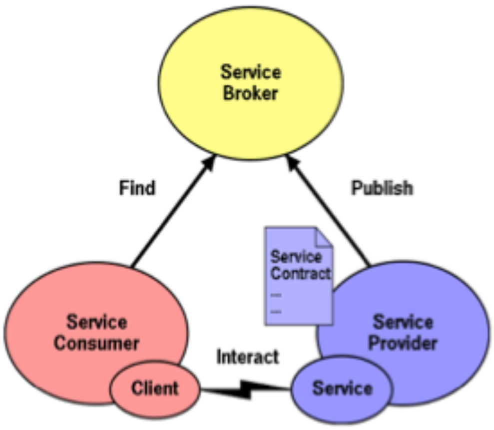


Figure 15: Service oriented architecture model

Every defined component (service) has a service provider and a service contract (interface description), which explains how to use the service and what is needed/provided by it. The service contract is published to the service broker, a library where service consumers (clients) can find services.

The most likely connection technology of service-oriented architectures is that of web services. A service provider creates a Web Service Description Language (WSDL) file, which is an XML language to define the interface of the provided service. These WSDL files will mostly be accessed and read by other services and applications using the Simple Object Access Protocol (SOAP), which is used for communication (XML messages) between de service provider and de requester (usually over HTTP). The service registry can be implemented using the Universal Description, Discovery and Integration (UDDI) language and can be queried using SOAP requests to find the WSDL files of necessary services (Michelson, 2006) (See figure 14 below).

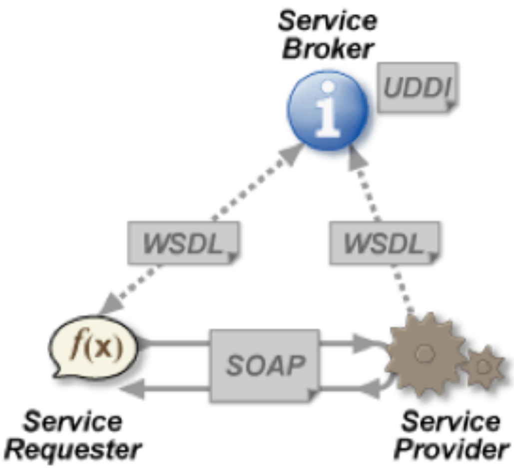


Figure 16: SOA model with webservices

5.5 Enterprise Service Bus

In order to get a better understanding of the architecture described in the next section, the Enterprise Service Bus is discussed beforehand since the proposed system will also rely on an ESB.

An Enterprise Service Bus provides an abstraction layer on top of an enterprise messaging system, and can be seen as the transport medium between several applications/services. The main purpose of an ESB is to easily connect applications/services to each other without actually having to implement detailed connection methods for each new application/service. This connection can be established with applications running on different platforms written in different programming languages and even using different programming models (Jr., 2007). ESB's offer real value when there are 3 or more applications to integrate. In figure 15 you can see an example where an ESB forms the main transport bus between a frontend application and 3 backend support systems.

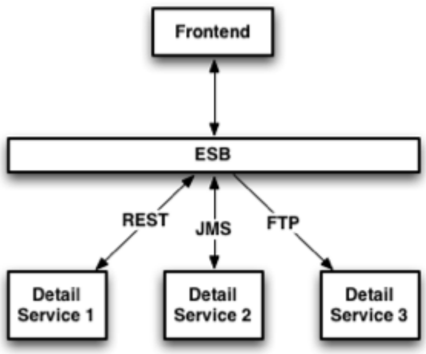


Figure 17: ESB scenario with 4 applications

An ESB lets one application send data to another by first accessing it via a locator and then sending it a message. In the communication process between these 2 applications the ESB handles the transformation of data formats, message acceptance, processing, routing and the sending of multiple messages at the same time (Jr., 2007).

5.6 Event Driven Architecture

In this section the architecture and principles used for the development of the CEP system itself is described. Notice how the first two sections covered the entire business, in this case the utilities market, and this section goes within the actual application. We start by elaborating on Event Driven Architecture, followed by its comparison to SOA. In the end of this section (and chapter) the reference architecture for the developed system is presented.

5.6.1 Event Driven Architecture definition and principles

Event Driven software Architecture (EDA) and design patterns have been used and implemented for a long time. The main reasons for using EDA has been in applications with high performance requirements handling thousands of transactions (events). This is also the reason CEP and EDA are closely related and why it's the best suitable architecture for the proposed CEP system. To get a clear understanding of what EDA stands for, take a look at the definition by the Patricia Seybould group (Michelson, 2006):

"In an event-driven architecture, a notable thing happens inside or outside your business, which disseminates immediately to all interested parties (human or automated). The interested parties evaluate the event, and optionally take action. The event-driven action may include the invocation of a service, the triggering of a business process, and/or further information publication/syndication."

Although this definition clearly focuses on an actual business instead of a software system it still holds for software systems as well. By nature an event-driven architecture is extremely loosely coupled, and highly distributed. The architecture is highly distributed because an event can be almost anything and can exist almost anywhere. The architecture is extremely loose coupled since an event itself does not know about the consequences of its cause but only knows the event occurred. If we have software system for example which is used to access personal banking information and a user enters the wrong password three times trying to log in, the password checker just reports the "wrong password entered" event three times but is not aware of the fact that some other part of the system uses this information to block the account until reactivated by the user.

Since there is no clear view on the principles of EDA, I have selected the most important ones, since they form an important part in the development of the CEP engine. The principles will be elaborated on in the following list:

- **Reusability** – A crucial factor in designing the events is that it shall be designed in such a way that it not only fits the current requirement, but also future requirements and out-of-scope usage. Events must be designed in a generic way, which is able to reuse (J.V.Hoof, 2007).
- **Real-time notification** – The technical infrastructure must support real-time creation and delivery of events. The human infrastructure must support real-time transformation of information first into knowledge and then into intelligent and following action (Ranadive, 1999).
- **Publish/Subscribe –** Notifications are pushed by the event source, not pulled by the event consumer. The event producer determines when the message that contains the event is sent (Schulte, 2008).
- **Immediate response –** The arrival of a notification causes the event consumer to act immediately. However in some cases, the action is simply to save the event for subsequent processing (Schulte, 2008).
- **Freedom to act –** A notification does not specify what action the event consumer will perform. It is a report not a request. The consumer contains the logic that determines how it will respond (Schulte, 2008).

5.6.2 Event Driven SOA

Over the course of the last years EDA and SOA have been in the spotlight but in a lot of cases both of these have come up short in certain areas (utschig-utschig, 2008). The major aim of enterprise architecture is realized in the ED-SOA concept by SOA combining business functions and IT, and EDA focusing on data as well as business relevant event orientation. ED-SOA can be seen as a hybrid application of EDA and SOA, and is even more suitable for this project since they complement each other (utschig-utschig, 2008).

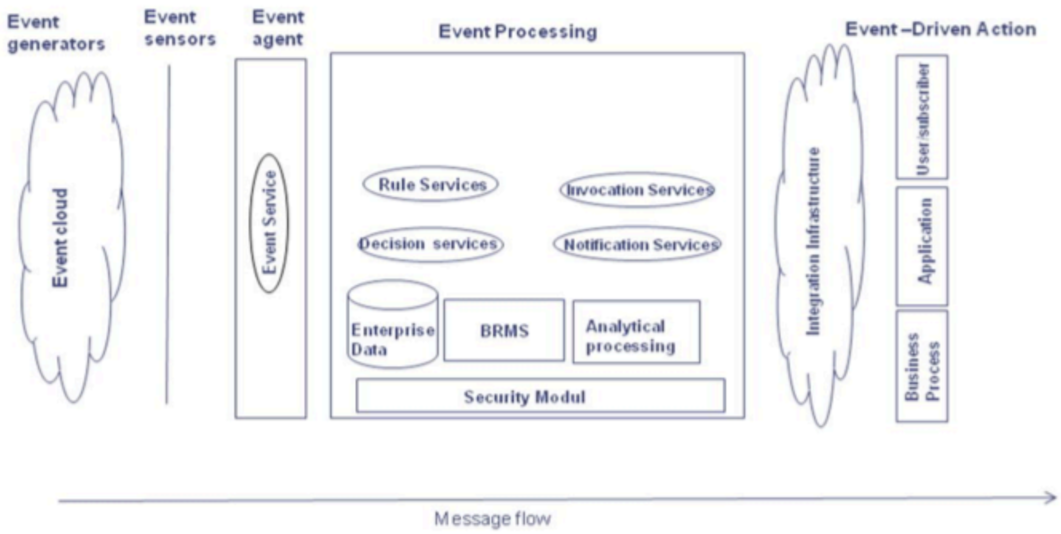


Figure 18: Event Driven SOA (Olga Levina, 2009)

In the figure above (figure 16) you can find a reference architecture for an EDSOA environment. On the left we have the event cloud, which through the event sensors uses an event service to register events (EDA). Within the Event processing unit Services are defined and stored as usually done in a SOA.

There are 2 main EDSOA composition theories (Orchestration and Choreography) of which the first one would best fit the situation of the CEP in the utilities market. In orchestration, a central process takes control over the involved web services and coordinates the execution of different operations on the web services involved in the operation. The EDSOA services do not know that they are part of a higher service. In this case only the CEP engine of the orchestration knows this, so the orchestration is centralized with explicit definitions of operations and the order of invocation EDSOA services.

Choreography on the other hand does not rely on a central coordinator like the CEP engine. In the case of choreography each EDSOA service involved in the choreography knows exactly when to execute its operations (based on defined trigger criteria) and whom to interact with. Choreography is a collaborative effort focused on exchange of messages (Events). All parts of the choreography need to be aware of the business process, operations to execute, messages to exchange, and the timing of message exchanges.

5.7 Architecture for developed system

In this section the main architecture of the developed system is presented. As input we use the reference architecture discussed in the previous section and the Event Processing reference architecture as described by Paschke and Vincent (Adrian Paschke, 2009). They define an event processing architecture to have the following stages (Adrian Paschke, 2009):

- Event originator – *Possibly distributed set of event sources.*
- Event Modeler – *Specifies the interface between the sources and the medium.*
- Event processing medium – *The platform that processes the events.*
- Event consumer – *Receives events which are communicated by the processing medium.*

They also define a lifecycle for the different phases supported by the entities as can be seen in the figure below.

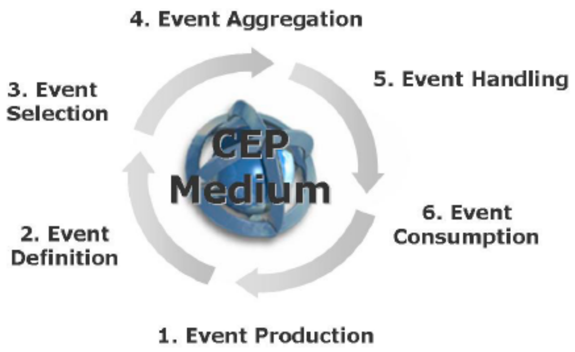


Figure 19: EP lifecycle (Adrian Paschke, 2009)

It all starts at the production of events where the different sensors, weather information and generated complex events publish to the ESB. At this point the CEP medium, which is subscribed to these events, picks the events up from the ESB where they pass through definition, selection, aggregation and handling before being published to the ESB so the subscribed consumers can pick them up. As depicted by the lifecycle this is a continuous process since the stream of events are endless.

In the figure below you find the architecture for this project. As you can see it can be roughly divided in two sections. The first being the event generator, which is the part discussed in the next chapter (Chapter 6), and the second is the CEP system itself, which will be presented in chapter 9.

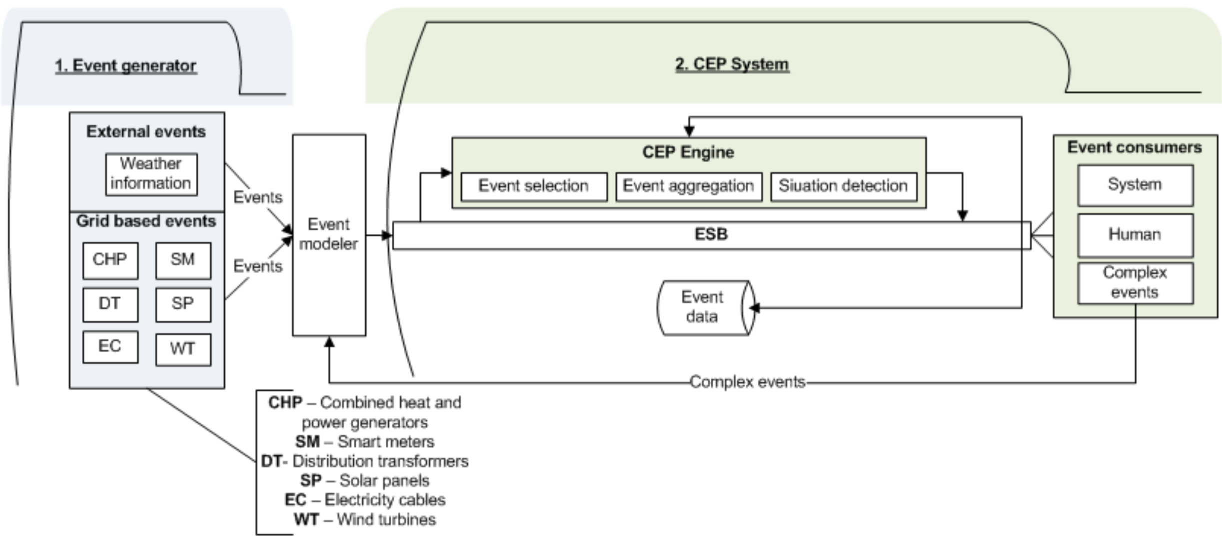


Figure 20: Complete system architecture

Though the architecture in figure 18 shows a complete system architecture, The parts actually covered by this research are the ones marked in blue (1. Event generator) and green (2. CEP system). Due to time constraints the CEP system will be directly attached to the event generator and event consumers without the use of the enterprise service bus. Also the event modeler is not a separate instance but is part of the CEP system.

Part 2: Event generation

Chapter 6: Event generator

In this chapter we will cover every aspect about the event generator. The event generator is the part of the system that will simulate an electricity grid and provide the Complex Event Processing engine with input. The figure below (figure 21) depicts the event generator in context of the full system. The designed application can be controlled by changing the three mentioned weather factors and will result in the system generating a different sequence of events. Changing the weather factors will directly influence the output of the simulated solar panels, wind turbines and CHP generators in the amount of electricity that they produce and deliver back to the grid.

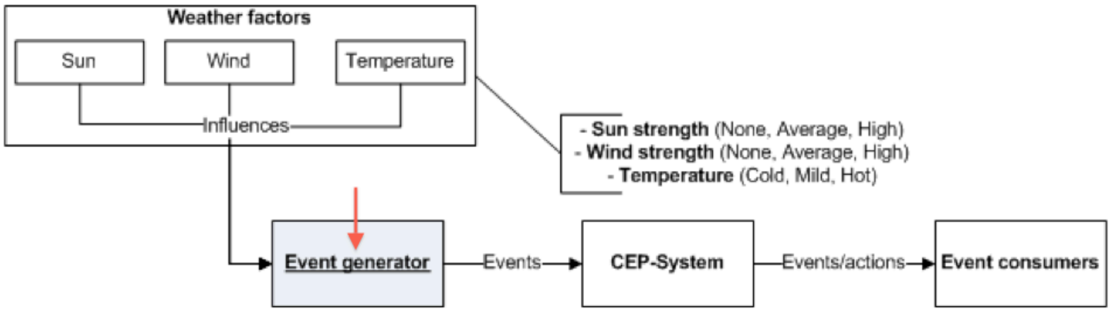


Figure 21: The event generator in context of the whole application

First we will discuss the model of the simulation grid, which will be used to mimic the events we use as input to the CEP-engine (section 6.1). This model was constructed by using the input of several experts in the field working at Logica, Alliander, Enexis and Stedin (more information about this at the beginning of section 7). In section 6.2 we elaborate on the different components that form the simulation grid followed by the way they are modelled into the event generator in chapter 6.3.

6.1 Smart simulation grid

As discussed in chapter 2, the Dutch electricity grid can be roughly divided in three sections, the high voltage (HV), medium voltage (MV) and low voltage (LV) grids. Below (figure 19) you find the (fictional) model of the grid, which will be used in this project. The model is set up as follows, first we have the high voltage (HV) network operating between 110Kv and 380Kv (marked with red lines), which uses transformers to connect to the medium voltage (MV) and low voltage (LV) network operating below 110Kv (marked with yellow and green lines). The two rings in between the lines are depictions of the transformers, which are used to convert voltage. Connected to the MV (yellow lines) you can find a couple of transformers (to and from HV/LV), a small wind farm and a small solar farm. The LV network connects a total of 200 houses to the MV grid where the larger part only consumes electricity and some of them also generate electricity using CHP's and/or solar panels.

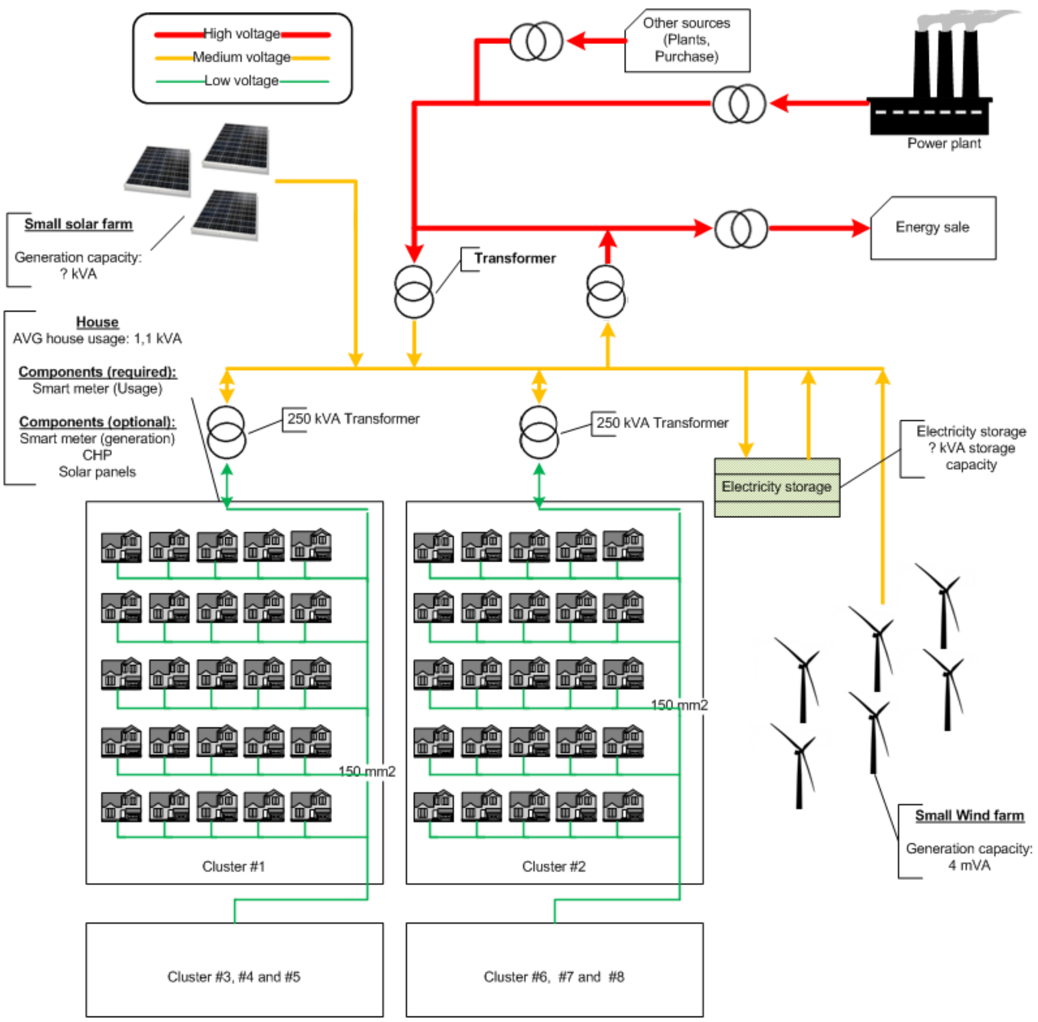


Figure 22: The smart simulation grid

6.2 Smart grid components

In this section, the components used in the network are explained (briefly). This section will help when trying to understand the designed simulation model and the way certain factors influence them when using CEP to manage the grid.

6.2.1 Smart meters

In this section we will elaborate on the concept of smart metering. We will start by covering the traditional methods followed by the recent developments and the pro's and con's of smart metering. The traditional electromechanical meters used in electricity grids are pretty straightforward and only allow for manual reading of electricity usage. Though these traditional meters are still widely used, among others utility providers have been looking at better ways to manage their electricity grids for years.

Next to the traditional way we can currently distinguish three kinds of metering techniques (Deconinck, 2008) starting with Automatic Meter Reading. With the main advantage of saving the utility provider periodic trips to each physical location to read a meter, Automatic Meter Reading (AMR) was introduced. Using this technology the collection of consumption, diagnostic and status data from metering devices can be completed remotely without physical access to the meter. The data is transferred to a central database where it can then be used for billing, troubleshooting and analyzing. The next step in automatic metering lead to Automatic Meter Management (AMM), which is basically the AMR system extended with the ability to manage meters remotely. It could for instance be used to disconnect non-paying customers or dimming usage. The last of the three techniques is smart metering which enable 2-way communication between the meter and an IT system, which reads, analyzes and stores the data. The combination of these smart meters, communication channel and IT system are referred to as the Advanced Metering Infrastructure (AMI). Smart meters have been defined as digital meters, which have the following capabilities (Rob van Gerwen, 2006):

- Real-time or near-time registration of electricity use
- Offering the possibility to read the meter both locally and remotely
- Remote limitation of the throughput through the meter (in the extreme case cutting of the electricity to the customer)
- Interconnection to premise based networks and devices
- Ability to read other, on-premise or nearby commodity meters (e.g. gas, water)

In order to be able to provide these services a smart meter must at least have the following components (figure 8). In this example the smart meter is used for electricity and gas metering. At the core of the meter lies a processing unit, which takes care of the logging, processing and control of the sensor data. The processing unit is also connected to a display unit for in home display of usage and a modem which will provide the connection to the utility and energy company. At the left and right side we see that the processing unit is connected to 2 sensors, one for electricity and one for gas. What you can also see is that next to the sensor they both have control units, which allows the electricity companies to regulate the electricity/ gas without actually going to the physical location. At the left side you can also find a switch, which allows the energy companies to switch on or off the electricity entirely, for safety reasons this is not implemented in the gas sensor.

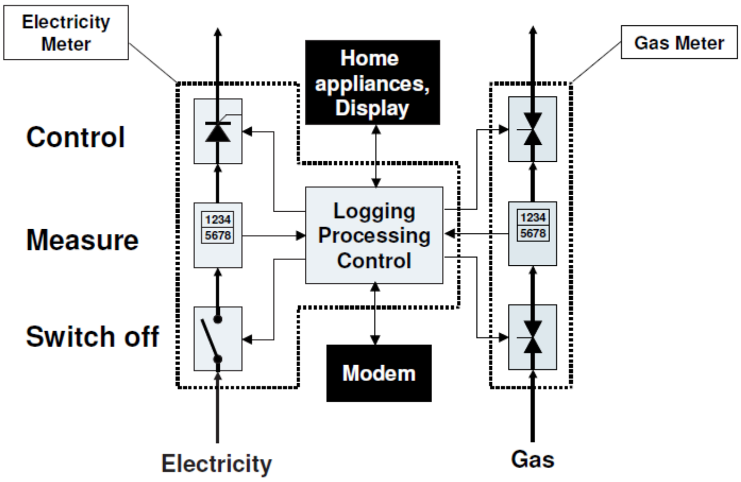


Figure 23: Schematic overview of a typical smart meter configuration

In order to control and manage these smart meters a form of communication from the meters to the electricity companies is needed. Research has been done in this area and three categories in general have been studied in detail. The first of these is the **Power Line Carrier (PLC)** which uses the power grid for data communication, the second **Communication over Telephone and Cable Infrastructure** which could use either a small band or broadband communication over phone/TV line, and third, Wireless **communication (Mobile phone, RF , PMR)** using licensed or unlicensed RF (Deconinck, 2008).

6.2.2 Distribution transformers

Distribution transformers are fundamental components of the power distribution system and their purpose is to either convert a higher voltage to a lower voltage (Step-down transformer), or to convert a lower voltage to a higher voltage (Step-up transformer). A distribution transformer (see figure below) is a static device constructed with 2 or more windings which are used to transfer alternating current from one circuit (primary) to another (secondary) using the same frequency but different values of voltage and current.

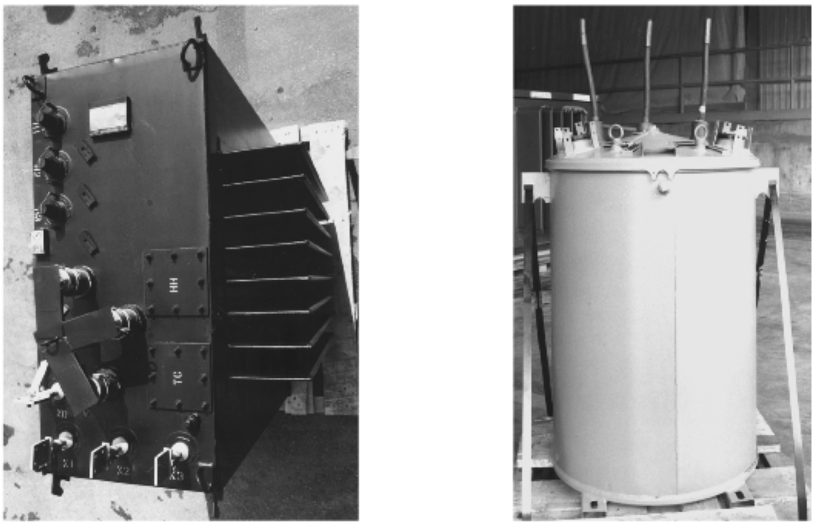


Figure 24: A three-phase vault-type distribution transformer (left) and a submersible distribution transformer (right)

In electrical engineering you have single phase and three-phase electric power. The single phase refers to a system in which all the voltages of the supply vary in unison and are mostly used in lighting and heating. In the three-phase system, the currents in each conductor reach their peak instantaneous values sequentially and not simultaneously. This difference in phase can therefore also be found in distribution transformers. Distribution transformers are classified into different categories based on certain factors such as:

- Type of insulation - *liquid-immersed distribution transformers or dry-type distribution transformers*
- Number of Phases - *single-phase distribution transformers or three-phase distribution transformers*
- Voltage class (for dry-type) – *Low voltage distribution transformers or medium voltage distribution transformers*
- Basic impulse insulation level (BIL), for medium-voltage, dry-type.

In the simulation grid these distribution transformers are used to transform the electricity between the LV, MV and HV (out of scope) grid. The ones defined here are "smart" in the sense that they produce several events as input for the CEP engine.

6.2.3 Combined Heat and Power generators (CHP)

The (micro) Combined Heat and Power generators (CHP) have been steadily gaining popularity in all sectors of the energy economy (M.Hommelberg, 2007), due to for example the increased cost of fuels, and are therefore an important part of our simulation grid. It is for this reason that they will be explained in this section. CHP Generators are more efficient in their production of usable energy from fossil fuels when compared to traditional power stations. A lot of energy is wasted in traditional power plants in the form of heat produced by the turbines for example. A technique called Cogeneration is applied which allows the released heat to be captured in the form of high-pressure water vapor or hot water, this increases the efficiency of the generators drastically (see figure 25).

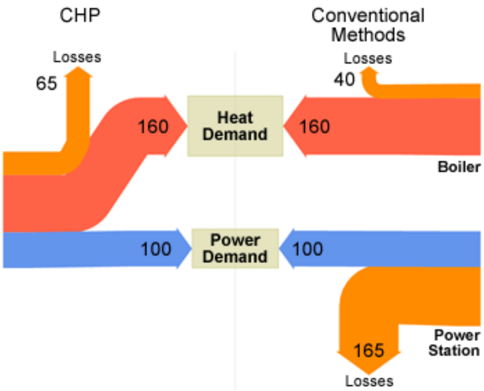


Figure 25: Efficiency CHP compared to Traditional Energy sources (N. van der Velden, 2008)

What we can see here (Figure 25) is that for the same heat and power demand (160 units heat, 100 units power) the Conventional methods require 465 unit of fuel whereas the CHP requires only 325 units of fuel. This leads to an efficiency of 80% for the CHP and 55% for the conventional methods. Since the simulation grid is based on the Dutch electricity sector in the Netherlands, which is know for its potentially cold winters and hot summers, micro CHP's form an excellent addition to our simulation grid.

6.2.4 Solar panels

The sun can potentially provide the equivalent of about 25,000 times the total amount of energy presently used from all other sources in the world. However, up until now only a very small fraction of this freely available energy is exploited through direct means for human use (Aftim Acra, 1984, pp. 13-14). A solar panel (photovoltaic module or photovoltaic panel) is a panel consisting of several solar cells (Figure 26), which generate a direct current (DC) when exposed to sunlight. This electricity, when left unaltered, can be used to charge small devices (calculators, telephones and streetlights for example) and batteries. When used as a component of a larger photovoltaic system however, these can be used to generate and supply electricity in residential applications or when used in an even larger setting, even power an entire district. The appliances in our homes use a different type of electricity called alternating current (AC). DC electricity flows in one direction only, while AC electricity changes direction rapidly, offering certain advantages in transmission (greater distances through smaller wires). This means that the generated energy needs to be converted from DC to AC (using an inverter) before it can be used in this setting.

Figure 26: Solar cells connected in an array (Solar panels)

A solar cell

There are 5 main technologies used to make solar cells, the building blocks of panels. Below you find a list of these types and their properties (Wiles, 2011):

- **Monocrystalline** solar panels are often the most expensive due to the manufacturing process, which uses large amounts of highly purified silicon and a great deal of energy. Monocrystalline solar cells are about 13-16% efficient at converting sunlight to electricity.
- **Polycrystalline** cell efficiencies range between 11-14% so polycrystaline solar panels are slightly less expensive than monocrystalline ones on a price-per-Watt basis.
- **String ribbon** solar panels use less silicon in the cell manufacturing process than the other crystalline types and therefore achieve efficiencies in the 12-14% range.
- **Amorphous** solar panels, or thin-film amorphous silicon (A-si). While these solar panels have lower efficiencies, (usually 7-10%), they offer certain advantages. They can often be used in hotter climates since they suffer less power loss than other types under hot conditions. Additionally, the amorphous technology does not use the typical "glass sandwich" construction, allowing for the creation of flexible solar panels, which are also very durable.
- The **CIGS technology**, or Copper Indium Gallium di-Selenide, uses no silicon at all, and can be made into panels with or without discrete cells.

The main reason that solar panels are not widely adopted is because of its cost effectiveness. Researchers are still working on lower-cost, higher-efficiency alternatives, but for the foreseeable future, these five types represent what is commercially available. The simulation grid in this project will contain solar panels on houses and a small solar farm.

6.2.5 Energy storage

Energy storage is achieved when some sort of device or physical media can store energy for use at a later time; these sorts of devices are also called accumulators. Energy storage in our setting will play an important role in conserving energy for later use since most energy sources are intermittent in nature. Solar energy for example is only available during the day, and hence, the application of solar energy will require efficient energy storage so that the excess heat collected during daytime can be stored for use during the night. The CHP generators have similar problems as the waste heat available and the period during it can best be utilized are usually different. Electricity consumption in electricity grids varies significantly during day and night, especially in extremely cold and hot climate countries where the major part of the variation is due to domestic space heating and air conditioning (discussed in previous chapters). There are several types of storage methods:

- *Chemical*
- *Biological*
- *Electrochemical*
- *Electrical (The storage of energy such as chemical or kinetic that can later be converted to electricity)*
- *Thermal (The temporary storage or removal of heat for later use)*
- *Mechanical*

The simulation grid in this project will contain energy storage banks (electrical and thermal) in order to deal with the distributed generation and electricity consumption peaks.

6.2.6 Wind turbines

Wind turbines are used to convert kinetic energy into mechanical energy, which is used to produce electricity. Wind turbines can rotate about the horizontal axis, Horizontal Axis Wind Turbines (HAWT), or the vertical axis, Vertical Axis Wind Turbines (VAWT). The VAWTs have a couple of advantages over the HAWTS, they are for instance independent of wind direction, produce less sound then the HAWTs and are easier to maintain (closer to the ground). They have one big drawback in that they generate less electricity then the HAWTs, which is why the HAWTs are used more frequently then the VAWTs (H.Riegler, 2003). In this research the focus is on the HAWTs since these are more common in practice.

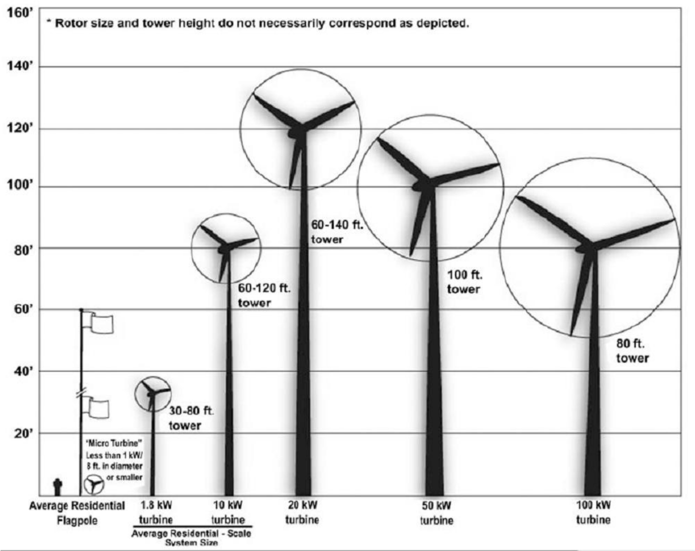


Figure 27: Different sizes of HAWTs

The horizontal wind turbine technology is one of the most emerging renewable technologies and started in the 1980's ranging from a few kW of production power to multi Megawatt installations (figure 27). The technology used in wind turbines is not new, but the price per KWh produced has dropped to levels that make them a great renewable energy source (F.Blaabjerg, 2006).

6.3 Smart grid component design

In the previous section general information of each of the simulation grid's components were given. This section covers the way we use them in the simulation grid and will follow along the same path as section 6.2.

6.3.1 Electricity cables

In order to create an event generator, which throws events in, the same way an ordinary electricity grid would, extra information will be needed about each line segment in the grid. This subchapter gives an impression of how the line segments of the electricity grid will be modelled in java. Below in figure 28 you can see the base situation (smaller subset) in which all the nodes are connected to each other. A node in this example can represent a house or a connection point between two different lines.

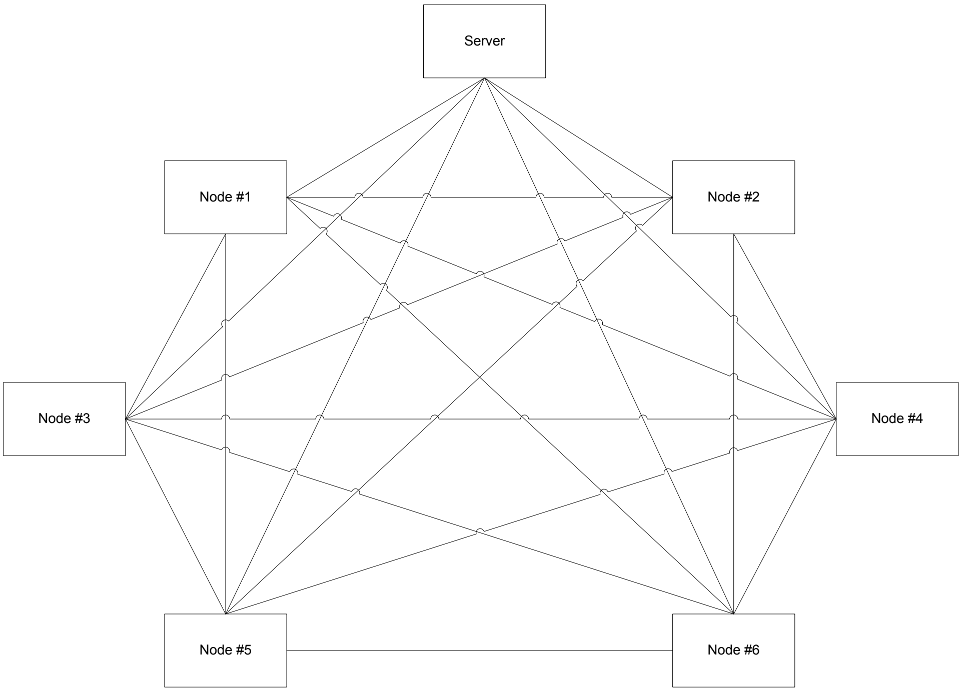


Figure 28: Base situation with all the lines connected

The electricity cable segments will be programmed in a double array, which is represented by table 5. It is evident that no information is needed about one node to itself; this is why all of these options are marked with an "x". The remaining entries could either be "null" if no connection exists or contain an object of the type "line" which contains the relevant information regarding that specific line segment. In order to save space and calculation time only half of the grid is used. This can be achieved by storing all necessary information in the line objects. (e.g. line thickness, line temperature, max line load)

	Server	***Node #1***	***Node #2***	***Node #3***	***Node #4***	***Node #5***	***Node #6***
Server	X						
Node #1	LineObj.	x					
Node #2	LineObj.	LineObj.	x				
Node #3	LineObj.	LineObj.	LineObj.	x			
Node #4	LineObj.	LineObj.	LineObj.	LineObj.	x		
Node #5	LineObj.	LineObj.	LineObj.	LineObj.	LineObj.	x	
Node #6	LineObj.	LineObj.	LineObj.	LineObj.	LineObj.	LineObj.	x

Table 5: Table corresponding to figure 28

Since the situation in figure one will hardly ever be the case in real life, another example is given which represents a more plausible setup. If we consider node #3 and node #6 to be houses, then the node #5 could be a connection point between the lines to the houses and the server.

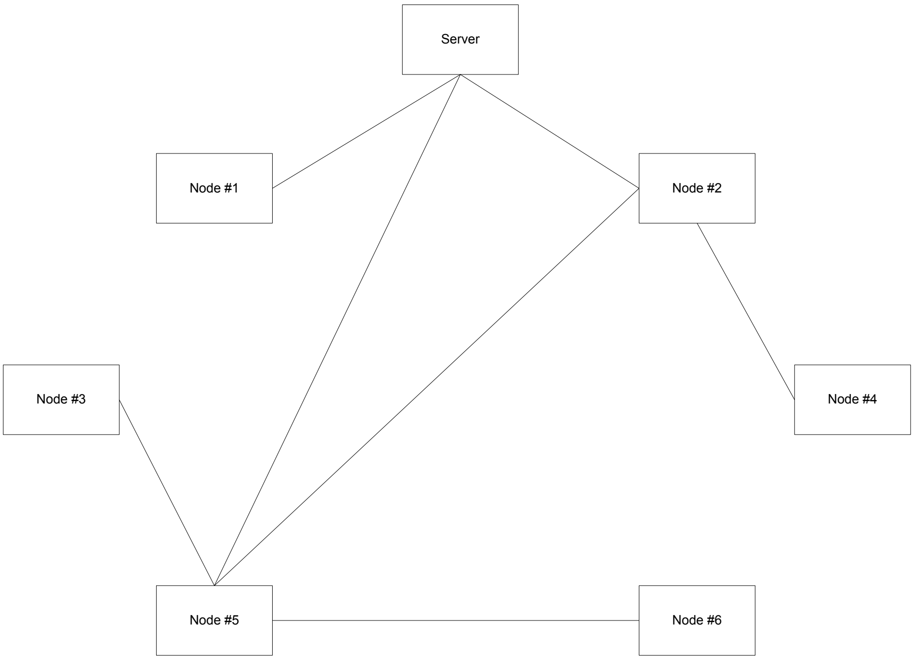


Figure 29: Example setup

In the table below you can find the line information about the example setup in figure 29. As you can see, the nodes, which do not have a connecting line, are given the null value.

	Server	***Node #1***	***Node #2***	***Node #3***	***Node #4***	***Node #5***	***Node #6***
Server	X						
Node #1	LineObj.	x					
Node #2	LineObj.	null	x				
Node #3	null	null	null	x			
Node #4	null	null	LineObj.	null	x		
Node #5	LineObj.	null	LineObj.	LineObj.	null	x	
Node #6	null	null	null	null	null	LineObj.	x

Table 6: Table corresponding to figure 28

The nodes used in this setup can represent different entities in the network. Those entities also generate events of their own and will be discussed in the next subchapters. To keep track of all the different cables in the grid, a unique identifier is defined for each cable in the form of **ECXXXX (EC0001 for example).**

6.3.2 Smart meters (Smart houses)

Each house in our setup is equipped with a smart meter. The houses with solar panels and/or CHP generators have meters, which provide us with production and consumption events whereas the houses without generation capability just provide us with consumption events. In the first subsection we will focus on the houses that just consume energy and in the second subsection we focus on the houses that also generate energy.

6.3.2.1 Energy consumption in houses

For this project we define 5 types of houses as drawn in the figure below from separate houses on the left to apartments on the right.

Figure 30: Different types of houses

Research has shown that for these 5 types of houses the electricity demand roughly follows the same curve as shown in figure 31 below (M. van Lumig, 2009).

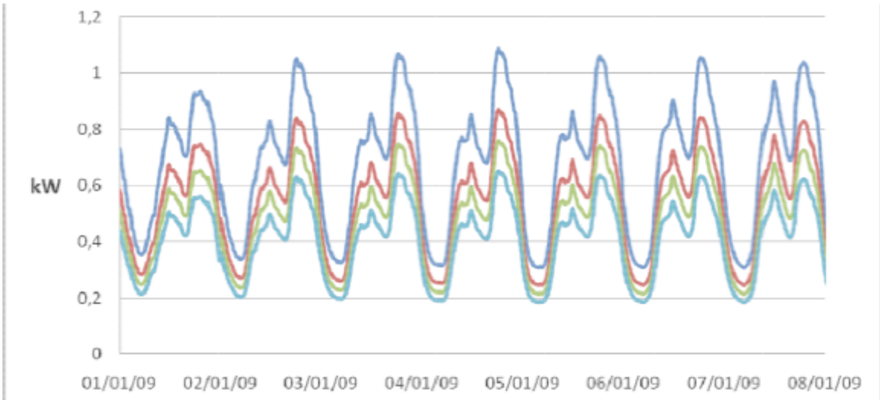


Figure 31: Electricity demand for the different types of houses (7 days)

If we take the average of these houses and make a graph for one day, we get the following figure. The 200 fictional houses in the designed grid will follow this curve (0 – 1.2kW) when generating events.

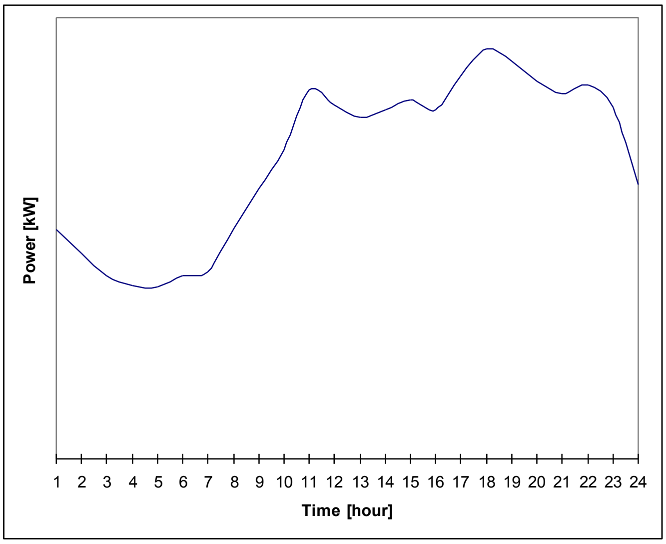


Figure 32: Average daily electricity demand.

To keep track of each individual smart meter, a unique identifier is defined of the following form **'H001SMUS'.** This code contains the house number "H001" and "SMUS" defines that the event generating smart meter in that house is monitoring consumption.

6.3.2.2 Energy generation in houses

The houses in the grid that are equipped with solar panels or CHP generators will also deliver energy back to the grid (small scale distributed generation). The settings in the designed application for sun strength and temperature will change the simulated load accordingly to show how the CEP-engine handles these situations. The simulated loads for small-scale CHP and solar panel generation will be discussed in section 6.2.4.and 6.2.5 respectively.

The unique identifier for the energy generation smart meter only differs from the one defined in section 6.3.2.1 in that US is replaced by GEN in the following manner **"H001SMGEN".**

6.3.3 Distribution transformers

Since distribution transformers do not generate energy, the distribution transformers in the defined grid will only generate events in the form of status updates. For the distribution transformers the following events are defined:

- **Heartbeat event** – Every 5 minutes a "heartbeat event" is generated to ensure the system that the specific distribution transformer is still operational.
- **Temperature event** – Every minute a "temperature event" is generated to inform the system of the current load on that specific distribution transformer. This information can also be logged to generate the "maintenance needed" event.
- **Maintenance needed event** – As described in the "Temperature event", the maintenance needed event will be generated upon assessment of past temperature data. Due to the temperature (can be translated to workload) of a distribution transformer the maintenance dates can be roughly predicted. This prediction is handy since it helps prevent outages due to distribution transformer failures.

To keep track of each individual distribution transformer, a unique key is also defined for the distribution transformers in the following manner **"DT01"**, where DT stands for Distribution Transformer and the number 01 will be used to mark each individual DT.

6.3.4 Combined Heat and Power generators (CHP)

The Combined Heat and Power generator (elaborated upon in chapter 6.2.3) will generate electricity during the cold periods when a lot of heat is needed for the house it is installed in and thus causes fluctuations in the electricity grid balance. The temperature setting in the designed application will directly affect the output of each CHP and the event this component will generate is of the amount of electricity being sent back to the grid.

To keep track of each individual CHP generator, the unique key for CHP's is defined in the following manner "**H001CHP01"**, where H001 stands for a specific house and CHP01 will be used to mark each individual CHP.

6.3.5 Solar panels

In our demo grid we have small scale and large scale solar panels (elaborated upon in chapter 6.2.4). The events these solar panels will send to the CEP engines are also in the form of the amount of energy generated. The sun setting in the designed application will have a direct effect on the output of these panels.

To keep track of the solar panels in our grid we define the following keys "**H001SP01 or LSSP01"**. The first one "H001SP01" will be used to indicate a solar panel mounted on a rooftop and the "LSSP01" will be used to indicate a panel in the solar farm.

6.3.6 Wind turbines

The grid in this demo will only contain wind turbines in wind parks as opposed to also having them in the houses. The events the turbines will generate also come in the form of amount of energy generated. The wind setting on the designed application will directly impact the amount of energy generated by the turbines.

To keep track of the wind turbines in our grid we define the following key **"LSWT001"** for the wind turbines in our grid. "LSWT" stands for Large Scale Wind Turbine, and the number will be incremented for each turbine.

6.3.7 Electricity storage batteries

In order to help keep the balance in the electricity grid, we also defined a bank of electricity storage batteries. As discussed in section 6.2.5 we will narrow the forms of storage down to electrical and thermal storage. The battery banks will be connected to the grid at MV level and will be used in such a way that when the demand is too high or the supply is too high the batteries will deliver or charge. The battery banks will generate events for electricity usage (charging) and delivery in the form of the amount of energy generated.

To identify the battery banks we defined the following key, **ESB001,** to identify the events in the event stream. ESB (not to be confused with Enterprise Service Bus) stands for Electricity Storage Bank and like all the previous components, the number will be incremented for each battery bank.

6.3.8 Component identifiers

For each component in the previous sections we defined an identifier to keep track of the component in the grid. In the table below you can find the complete overview of the event generator's identifiers.

Component name	Component identifier	Description
Electricity cable	EC-**XXXX**	XXXX-Electricity cable number
Smart house (Smart meter)	H-**XXX**SM-**XX**US, H-**XXX**SM-**XX**GEN	XXX-housenumber, XX smart meter number, GEN for generation, US for usage
Distribution transformer	DT-**XX**	XX – Distribution transformer number
CHP generator	H-**XXX**CHP-**XX**	XX – CHP Generator number
Solar panel	H-**XXX**SP-**XX** (small scale) SP-**XX** (large scale)	XX – Solar panel number
Wind turbine	LSWT-**XX**	XX – Wind turbine number
Electricity storage battery	ESB-**XX**	XX – Battery number

Table 7:Component identifiers

These components will be the source of all events we will use in the CEP-engine. In the next chapter (chapter 8) we take a look at the functional requirements defined for our designed system.

Part 3: CEP engine design and demo evaluation

Chapter 7: Main CEP-engine challenges

In this section, the main problems (Load balancing, Outage detection, Congestion management) are introduced and elaborated upon. In each of the main problem sections we will also state the requirements defined for the specific problem and their corresponding rules, which will be used to provide the CEP engine with the necessary knowledge. As was the case with the construction of the simulation grid, the input of several experts was used to establish the three most important problems (described in this section) and their corresponding requirements and rules for the CEP engine by conducting an interview (Original interview in appendix B). In the table below you can find a list of the functions of the people used as input for these interviews (names are omitted for privacy reasons).

Company	Function	Relevance
Logica	Principle business consultant utilities	General knowledge about the utilities and running projects in the Netherlands
Logica	EUT Business consultant	Knowledge about the utilities market and involvement in smart metering projects
Stedin	Senior specialist innovation	Thorough knowledge about smart grids and solutions, possibilities within the current grid. Knowledge of all smart grid related projects and research within Stedin
Alliander	Innovation manager	Thorough knowledge about smart grids and solutions, possibilities within the current grid. Knowledge of all smart grid related projects and research within Alliander
Enexis	Innovator	Thorough knowledge about smart grids and solutions, possibilities within the current grid. Knowledge of all smart grid related projects and research within Enexis

Table 8: Conducted interviews

The interview setup (Appendix B) was used as a base situation to get the interview going and does not cover all the interview topics. It should be stated that among the interviewee's, there was little to now knowledge about CEP, which makes this research all the more relevant to Logica and the Dutch utilities, since CEP shows a lot of promise for smart grid management.

Before we dive into the three previously mentioned problems (Load balancing, Outage detection, Congestion management), we will compare the future grid to the current one in order to get an idea of the changes necessary for the CEP-engine to function correctly. As already mentioned, all of the previously mentioned subjects were discussed with experts in the field and are numbered in the order of priority (Consensus among interviewee's). We start with the grid comparison in subchapter 7.1, followed by the main problems "load balancing" in subchapter 7.2, "outage detection" in 7.3 and "congestion management" in 7.4. The sections 7.2-7.4 will also contain the relevant basic requirements set up with the experts as well as their transformation to rules for the CEP engine.

7.1 Current grid versus future grid

A large part of the current MV network cannot be monitored or controlled remotely as the equipment to make that possible is not in place. Most equipment, such as transformers, are protected by simple circuit breakers that break the circuit when power runs too high (Similar to fuses in houses and cars). When such a fault occurs however, the fault needs to be located by the repair crew and then restored manually. Once the power enters the MV and LV grid, the operators can no longer track where the power went or in what way it is being used. This means that with the electricity grid in its current state it is not possible to monitor changes in load, which up until now was fairly easy to predict. With the latest developments in the electricity sector, Distributed Generation (DG) is a rising phenomenon, and will cause the loads to be more unpredictable. This brings us to the future (smart) grid in which a lot of components will be automated. Substations for example can be equipped with remote control stations so they can be shut of remotely in case of a fault. Another example is that of the smart meters which make it possible to give insight in the electricity load. In the remaining sections of this chapter we elaborate on the subjects from which it is believed that the most can be gained when using a smart grid in combination with CEP. The way this will by done is by explain a set of possible scenario's for each problem, which are then translated to requirements and rules for the CEP system

7.2 Load balancing

One of the most important aspects, which can gain significantly from the implementation of a CEP-system, is that of load balancing. Load balancing is the act of matching electricity supply to electricity demand in order to keep a balanced network. As mentioned earlier, balancing demand and supply has become a more difficult task due to distributed generation for example. In traditional grids the electricity is generated at the electricity plant and based on analysis of (historical) usage statistics (supply matched to demand), but with solar panels, chp's and wind turbines installed this is no longer the case. Most of these new energy sources are dependant on the weather, which makes it difficult to predict generation capacity. Though there are several scenario's possible in this problem area in which CEP can excel, only the ones that will be covered in the demo will be explained below.

If we for example take a look at the houses equipped with CHP generators we see that during a cold winter the heating will be running at full capacity, which implies that a lot of electricity will be generated and sold back to the grid. This might cause the grid to get overloaded and the CEP engine will be able to detect this from an early point and generate predefined events for the subscribed parties. One of the events for example can be sent to the electricity producers to decrease production. Another possibility is to redirect the surplus of electricity towards charging the batteries, which can then be used at a later time. This scenario can be simulated in the demo by setting the temperature to cold, as this creates the maximum amount of generated electricity for the CHP's in the simulation grid.

In the case of solar panels we can consider a large group of clouds passing over the panels, which will cause production to decrease. Should this cause the electricity demand to be higher than production, another event can be generated asking the producers to increase energy production. The wind-turbines can also affect the balance of the grid depending on the speed of the wind at a given point in time. During a stormy period a lot of electricity will be generated and during a period with little to no wind the wind turbines wont generate a lot of energy.

What we can see if we look at the previous examples is that all of these generators are largely dependant on the weather at a given place and time. The CEP system can also use weather information at specific locations and times to predict the amount of energy that will be generated and consumed (consumption can be predicted based on historical data).

The examples given above have been transformed into the following requirements so that they can be implemented as rules in the CEP application. First you find the requirements, which are established by discussion with the involved parties. These requirements are translated into rules that can be interpreted by the CEP engine using the Event Processing Language (EPL).

LOAD BALANCING and CONTROL

RQ01. Monitor demand/supply and notify designated employees in case of a mismatch of 10% or more

RQ02. When supply exceeds demand (generated energy from solar panels/CHP's is too high), lower the energy flow into LV-grid from MV-grid (Allow LV to feed on itself) and charge battery banks

RQ03. When demand exceeds supply in the LV grid, increase production in solar farms or wind parks if possible or use the battery banks to supply energy to the grid

RQ01, RQ02 and RQ03

SELECT Load[Usage] as U, Load[Generation] as G FROM CurrentLoad.win:time(1 min) where (G/U) < 0.1 (*There is an imbalance on the monitored grid, usage is exceeding generation (old events are discarded). Execute actions:*

- *Send msg to lower energy flow from MV to LV. Allow battery banks to charge.*

SELECT Load[Usage] as U, Load[Generation] as G FROM CurrentLoad.win:time(1 min) where (G/U) > 1.1 (*There is an imbalance on the monitored grid, generation is exceeding usage.*

- *Send msg to increase the energy flow from the MV to the LV. Use electricity from battery banks.*

7.3 Outage detection

Another important aspect of the electricity grid in which CEP can significantly improve the process is that of outage management. The Dutch electricity grid is a very large network with numerous components and cables of which each one can fail for various reasons. What happens in the current grid is that the DSO's what for the customers to complain to register such a fault. The next step in the process is to find the faulty hardware, which can currently only be done by sending a repair-crew out to check all of the substations individually. This is a time consuming process, which can drastically be improved by outage detection using CEP.

The CEP engine can monitor the grid in such a way that events (or lack thereof) can result in a new action or event, which notifies the repair crew from the moment it happens instead of when the first complaint reaches the DSO's. Next to the repair crew being notified at a much earlier stage, the location of the fault might also be available by reasoning over the generated events. The CEP system would also be able to make use of the repair crew locations (Vehicles should be equipped with GPS modules) and just notify the nearest one. CEP can be used to process data from several sensors in the network. The data received from the sensor will be converted into event objects, which in turn can be used by the CEP engine. When an unexpected rise in the power current occurs it can usually be deducted that a fault has occurred in the grid. In order to find the location of the fault the events of the transformers in the network can be analysed since the fault should be located in between where one transformer notices a peak current ant the other one doesn't. An addition to this concept could be if the DSO's were to automate the transformers in the grid. This would make it possible to isolate the fault and restoring the electricity back to the unaffected surrounding areas, therefore keeping the outage area to a minimum.

Another purpose for the use of CEP in outage detection comes in the form of logging purposes. The Dutch DSO's are obliged to report any outages occurred in their network to the Energiekamer for monitoring purposes. This is a simple addition to the outage detection process and logs can automatically be made when an outage occurs.

The advantage of using CEP for this purpose is that the generated events can be consumed by several other relevant services (repair crew, logging purpose, asset management, automatic customer notification). Below you can find the basic requirements for the application of outage management using CEP.

OUTAGE DETECTION

RQ04. Notify designated employee when load drops suddenly, include the list of transformers where a zero load has been measured.(Use a 5 minute time window and select the first event as location)

RQ05. Notify designated employees when a smart meter sends a "no load" event (Probably an outage)

RQ06. Save "outage" event along with "back online" event to document (Reporting to Energiekamer)

RQ07. Notify a designated employee when a transformer doesn't update its status within a time window of 1 minute (Expected outage).

RQ08. Notify a designated employee when a smart meter doesn't update its status within a time window of 1 minute (Expected outage).

RQ04,RQ05 and RQ06

FOR EACH DT-**XX**

SELECT DT-**XX**[Load] AS Load, DT-**XX**[Location] AS Loc FROM DT-**XX**.win:time(5 min) WHERE DT-**XX**[Load]=0

- *If a zero load is measured msg to designated employees*
- *Log an outage event to the database for the energiekamer*

RQ07

FOR EACH DT-**XX**

SELECT DT-**XX**[Load] AS Load, DT-**XX**[Location] AS Loc FROM DT-**XX**.win:time(5 min) WHERE count(DT-**XX**[Load]) = 0

- *When no status updates are received within 5 mins, send msg to designated employees (expected outage)*

RQ08

FOR EACH H-**XXX**SM-**XX**US

SELECT H-**XXX**SM-**XX**US [Usage] AS Usage, H-**XXX**SM-**XX**US [Location] AS Loc FROM H-**XXX**SM-**XX**US.win:time(5 min) WHERE count(H-**XXX**SM-**XX**US [Usage])=0

- *When no status updates are received within 5 mins, send msg to designated employees (expected outage)*

7.4 Congestion management

The third most important problem for the Dutch DSO's is that of congestion management. In order to understand this phenomenon it can best be explained by using a well-known Dutch problem as a metaphor. The highways in the Netherlands are heavily crowded and during peak hours traffic jams are formed at the points where the available road is less than the amount of cars require. This metaphor exactly describes the problem of congestion in an electricity grid if we replace the roads by electricity cables and the cars by electricity current. Regions that generate or consume a lot of electricity require a higher capacity for transport and when the capacity is lower than the generated electricity, the grid becomes congested. This is not desirable as it can and eventually will cause equipment failure. For this reason the DSO's have a safety margin on the different parts of the grid where a percentage of the total capacity is used. When a CEP engine is monitoring the grid, the quicker response times will allow the DSO's to lower this margin and hence increase capacity using the same infrastructure.

The events generated by cable and transformer sensors can be monitored for load levels. When a cable or transformer reaches a peak load, a designated employee can be notified to lower production of the electricity generators. Next to notifying employee's, the events can also be used to automatically perform actions on the grid such as redirecting the power flow to charge battery banks.

Below you can find the requirements for the part of congestion management in the demo application. First the requirements are displayed followed by the translated requirements into the event processing language.

CONGESTION MANAGEMENT

- RQ09. Notify designated employee when load on a specific cable reaches upper threshold.
- RQ010. Notify designated employee when load on a transformer reaches critical (upper) threshold.
- RQ011. Save critical threshold message to maintenance log (A high amount of these events probably means that the transformer will need maintenance or replacement sooner).

MAINTENANCE

- RQ012. Upon receiving a "maintenance needed" event from a transformer, send an e-mail (with location and other transformer information) to the designated employee.
- RQ013. Keep track of "above average" transformer and cable temp for component wear. If temp higher than avg for a certain period of time. request extra check-up

<u>RQ09 (and RQ11, RQ13)</u>

FOR EACH EC-**XXXX**

SELECT EC-**XXXX** [Load] AS Load, EC-**XXXX** [Location] AS Loc FROM EC-**XXXX**.win:time(1 min) WHERE EC-**XXXX**[Load] => [Threshold]*0.90

- *When Load on a specific cable surpasses the predefined [Threshold], send a msg to designated employee to investigate.*
- *Log events to database for maintenance*
- *Monitor logged events and increase maintenance needed event counter*

<u>RQ10 (and RQ11, RQ13)</u>

FOR EACH DT-**XX**

SELECT DT-**XX** [Load] AS Load, DT-**XX** [Location] AS Loc FROM DT-**XX**.win:time(1 min) WHERE DT-**XX** [Load] => [Threshold]*0.90

- *When Load on a specific transformer surpasses the predefined [Threshold], send a msg to designated employee to investigate.*
- *Log events to database for maintenance*
- *Monitor logged events and increase maintenance needed event counter*

<u>RQ12</u>

FOR EACH DT-**XX**

SELECT DT-**XX** [Maintenance] AS Main, DT-**XX** [Location] AS Loc FROM DT-**XX**.win:time(1 min) WHERE DT-**XX** [Maintenance] = TRUE

- *Send e-mail with location information to designated employee for transformer maintenance.*
- *Log events to database*

Chapter 8: CEP engine and demo

Now that we have covered all the relevant information and gathered everything we need, we will dedicate this chapter to the actual CEP-engine and the user interface of the designed application. As mentioned before, the event generator described in chapter 6 will serve as the input to the CEP-System discussed in this chapter. The CEP-System forms the centre of this research as it contains the predefined logic to control the smart electricity grid and the ability to perform actions. The figure below gives an overview of the entire system with the focus on the CEP-System.

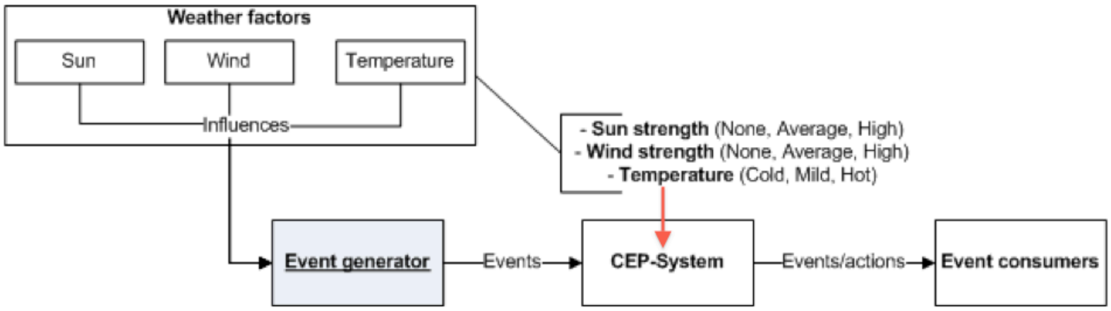


Figure 33: CEP application in context of the whole system

We first start by elaborating on the setup of the demo and present the table with the results of the tests ran on the system. In the last 2 sections of this chapter we briefly give an overview of the Java code and the graphical user interface.

8.1 Demo setup and goal

In this section we explain the setup of the demo and elaborate on the tests done with the system. First and foremost the event generator is switched on to create the input to the CEP engine. The amount of events generated by the components modelled in the event generator (chapter 5), are regulated with timers in Java. Based on this output, the CEP engine will start reasoning over the data and report which of the implemented requirements are met. Upon analysing the incoming events and completing the matching process with the predefined events in the CEP engine, the engine will execute these results by performing (fictional) actions in the network. The main goal of the demo is to show that the CEP based engine can independently manage the simulated grid by keeping the total amount of electricity passing through the grid below the specified threshold and balanced. This should also be the case in different weather conditions, which directly influences the amount of power generated and requested by the components in the grid. To test the demo against the main goal, a couple of tests will be executed on the system. The different scenarios are elaborated upon in the following sections.

8.1.1 Scenario #1 (Sun: None, Wind: None, Temperature: Cold)

As the title implies, in this scenario the sun and wind setting will be set to none, and the temperature will be set to cold. This mostly corresponds to the winter season and means that the solar panels and wind turbines wont generate a lot of energy. The heat however will be turned up high and this means that the CHP installations will be delivering electricity back to the grid. In this scenario it is likely that the MV/LV grid will be in need of extra power (extra electricity needed in winter). The expected behaviour of the CEP engine here is that it will automatically request electricity from the HV grid and battery banks when needed.

8.1.2 Scenario #2 (Sun: Average, Wind: High, Temperature: Mild)

This scenario reflects a windy autumn in which a lot of wind energy is generated and the amount of power generated by CHP generators or solar panels is below average. In this scenario the CEP engine has to prove it can maintain balance in the grid as at some points extra energy might be needed from HV network and battery bank (demand exceeds supply), and at some points there might be a surplus of energy in the grid (supply exceeds demand).

8.1.3 Scenario #3 (Sun: High, Wind : High, Temperature: Mild/Hot)

This scenario reflects a spring or summer day with the wind and sun settings to high and the temperature being mild/hot. The solar panels and wind turbines in the MV/LV grid will be generating electricity at maximum capacity and power usage will be lower than in winter. It is expected that the CEP engine will not request electricity from the HV grid but rather feed on its own distributed generation. Excess electricity should be redirected to and stored in the battery banks.

8.1.4 Test setup

The scenarios described above will each be executed in three runs (cycles) each of 1, 5 and 10 minutes. During this run (cycle) we will define a (fictional) threshold, which compares to the threshold used in the Dutch electricity grid at the moment, and monitor the result during each run since the CEP engine should keep the maximum electricity below the threshold to prevent equipment and/or cable damage. The amount of events processed per run will also be monitored along with the amount and types of complex events generated.

8.2 Test results

In this section we present the table with our findings based on the setup defined in the previous section along with a basic reasoning about the found data. A detailed view and conclusions about the CEP system and this data can be found in the next chapter (Chapter 9).

	Scenario #1 (1 min)	Scenario #1 (5 min)	Scenario #1 (10 min)	Scenario #2 (1 min)	Scenario #2 (5 min)	Scenario #2 (10 min)	Scenario #3 (1 min)	Scenario #3 (5 min)	Scenario #3 (10 min)
Average grid load	152	154	146	143	164	170	167	129	177
Difference (170)	-18	-16	-24	-27	-6	0	-3	-41	+7
Events processed (avg over runs)	5939	29547	52811	6233	24560	49520	5576	30349	62062
Complex events generated (Union over runs)	3	8	17	9	16	21	6	11	18

Table 9: demo test results

8.3 System implementation

In this section a general overview of the application code is provided. We start by showing the event generator initialization in the figure below. The event generator is the part of the simulation grid in which all the possible events are registered and generated using a timer.

```
public void initializeEngine(){
    Configuration config = new Configuration();
    config.addEventTypeAutoName("ElectricityCEP.events");
    ConfigurationEventTypeLegacy legacyDef = new ConfigurationEventTypeLegacy();

    // add all the fields of the class to esper
    legacyDef = new ConfigurationEventTypeLegacy();
    java.lang.reflect.Field[] fields =  ElectricityCableEvent.class.getFields();
    for (int i = 0; i < fields.length; i++) {
        legacyDef.addFieldProperty(fields[i].getName(),fields[i].getName());
    }
    config.addEventType("ElectricityCableEvent", ElectricityCableEvent.class.getName(), legacyDef);

    legacyDef = new ConfigurationEventTypeLegacy();
    fields =  SmartMeterEvent.class.getFields();
    for (int i = 0; i < fields.length; i++) {
        legacyDef.addFieldProperty(fields[i].getName(),fields[i].getName());
    }
    config.addEventType("SmartMeterEvent", SmartMeterEvent.class.getName(), legacyDef);

    epService = EPServiceProviderManager.getDefaultProvider(config);
}
```

Figure 34: Initialization of events in the event generator

What we can see in figure 34 is that the event is initialized with some (basic) configuration settings and that each event should be declared in this method as shown above. Next we create the statements for each requirement as described in chapter 7. The figure below gives a representation of the declaration of to basic statements in Java.

```
public void createSmartMeterStatement(){

    String expression = "select usage,id from SmartMeterEvent";
    EPStatement statement = epService.getEPAdministrator().createEPL(expression);

    SmartMeterEventListener listener = new SmartMeterEventListener();
    statement.addListener(listener);
}

public void createElectricityCableStatement(){

    String expression = "select avg(load) as AL,id from ElectricityCableEvent";
    EPStatement statement2 = epService.getEPAdministrator().createEPL(expression);

    ElectricityCableEventListener listener = new ElectricityCableEventListener();
    statement2.addListener(listener);
}
```

Figure 35: Declaration of statements (requirements) using EPL in Java.

The last step in the event generator is that of generating the events as declared in chapter 7. Below you can find the event processing service in a timer for execution.

```
public void GenerateEvents(){
    //This will create and start the electricityCableThread
    ( new Thread() {
        public void run() {
            logger.info("Started the electricityCableThread");

            // Cable events should be defined as ElectricityCableEvent (int serialNr,int cableThickess, double cableLength, String cableMaterial)
            while(started){

                epService.getEPRuntime().sendEvent(new ElectricityCableEvent(1,10,20,"copper"));
                //sleep for a while and continue
                try {
                    sleep(200);
                }
                catch (InterruptedException e) {
                    //Do nothing
                }
                epService.getEPRuntime().sendEvent(new ElectricityCableEvent(2,10,20,"aluminum"));
            }

            logger.info("Ended the electricityCableThread");
        }
    }).start();
```

Figure 36: Event generation

The events are modelled by creating event classes containing the necessary variables and methods as shown below.

```
public class ElectricityCableEvent extends Event  {

    private double load;
    private double temp;

    //define generation properties
    private int tempLowerbound = 80;
    private int tempUpperbound = 110;

    //maximum load in KW
    private int maximumLoad;
    //cable thickness in mm
    private int cableThickness;
    private double cableLength;
    private String cableMaterial;


    public ElectricityCableEvent (int serialNr,int cableThickness, double cableLength, String cableMaterial){
        //value EC is the type of this event, EC = Electricity Cable
        super(serialNr,"EC");

        //set all variables
        setCableThickness(cableThickness);
        setCableLength(cableLength);
        setCableMaterial(cableMaterial);

        //calculate and set remaining variables
        generateTemp();
        setLoad();
    }

    public void setLoad(){
        //Actual function still has to be defined
        this.load = getTemp() * (getCableThickness()/10);
    }
```

Figure 37: Event class

Last but not least, the events are connected to the listeners in which the actions needed can be modelled (Figure 38).

```java
public class ElectricityCableEventListener implements UpdateListener {

    private static Logger logger = Logger.getLogger("com.ElectricityCEP.EventListener");

    double roundTwoDecimals(double d) {
        DecimalFormat twoDForm = new DecimalFormat("#.##");
        return Double.valueOf(twoDForm.format(d));
    }

    public void update(EventBean[] newEvents, EventBean[] oldEvents) {
        EventBean electricityCableEvent = newEvents[0];

        EventsToGUI ETG = new EventsToGUI();
        ETG.addToEventsList("ID=" + electricityCableEvent.get("id")+" CableLoad=" +
                roundTwoDecimals(Double.parseDouble(electricityCableEvent.get("AL").toString())));
    }
}
```

Figure 38: Listeners for the events

8.4 Graphical user interface

As the name implies, this section will elaborate on the Graphical User Interface of the developed application. In the figure below you find the layout of the application as designed. Each of the elements in this main screen will be elaborated upon separately.

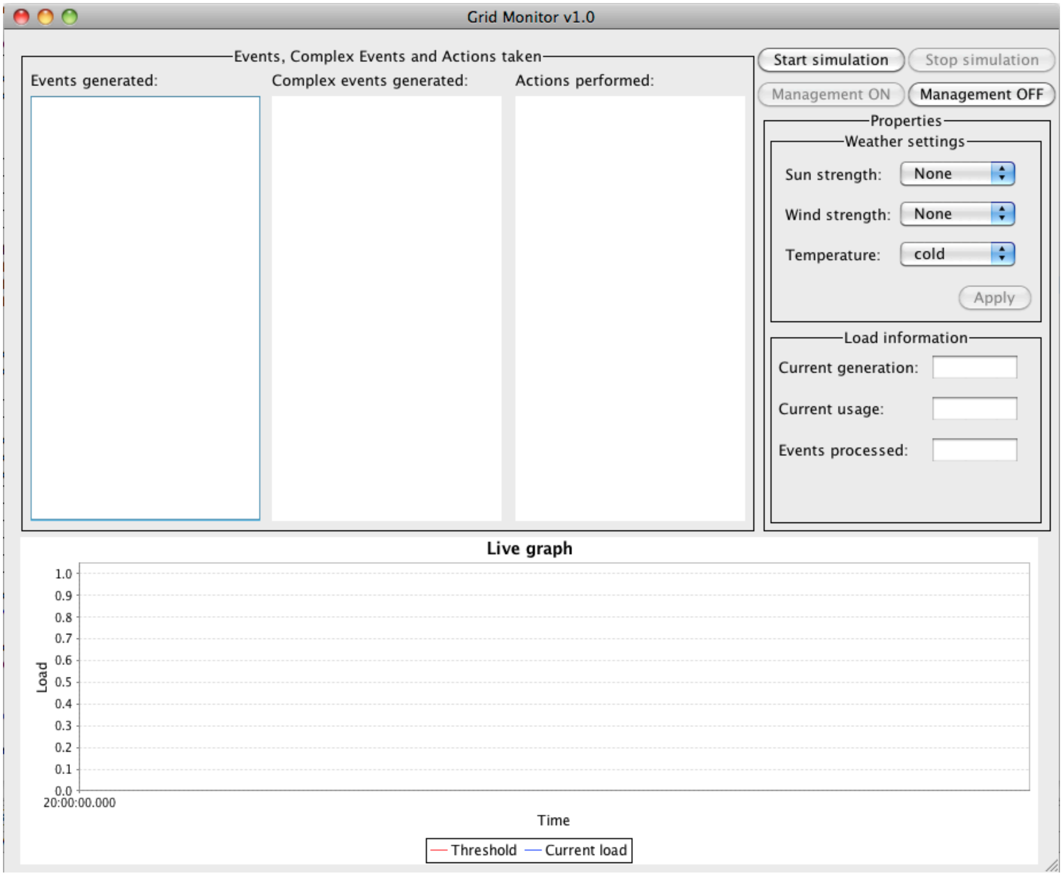


Figure 39: Developed application

To stress the vast amount of data available by these event generators, the first of the three event windows (events generated), displays all events. As you will be able to notice, no sense can be made of this information as it passes by to quickly to assess. The second window (Complex events generated) will show the complex events generated based on the defined rules (Information after filtering) and the third window (Actions performed) will show the defined actions. These windows are shown in the following figure.

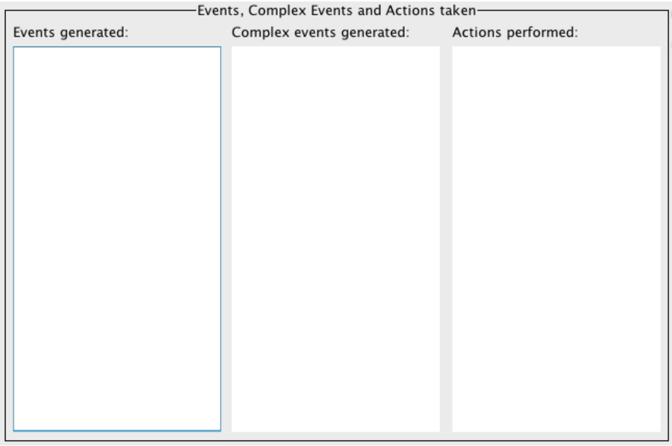


Figure 40: Event windows

Next we have the live graph, in which the total usage and demand are measured and plotted.

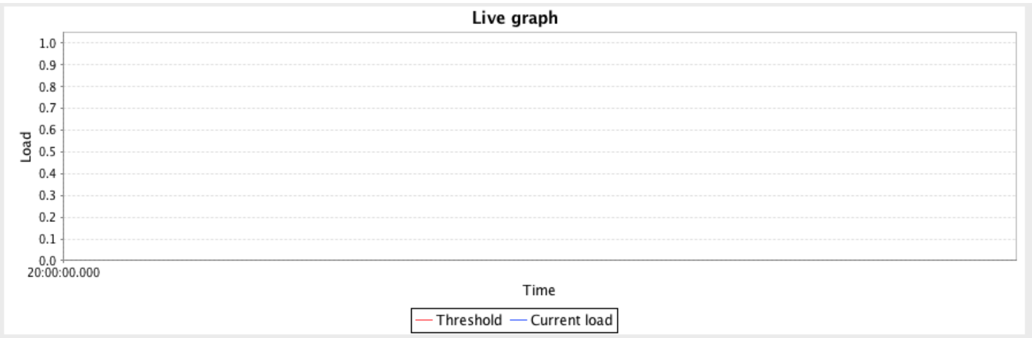


Figure 41: Live graph

The three defined weather settings can be changed in the properties section, which also contains some basic information such as the amount of events processed by the engine and the current electricity usage and generation.

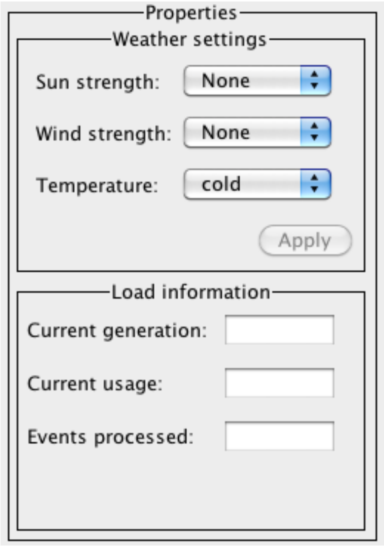


Figure 42: System properties

In the upper left corner there are 4 buttons found which can be used to control the application. The first two can be used to start and stop the engine whereas the last to can be used to compare the difference in loads with or without management.

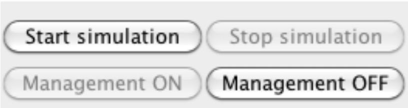


Figure 43: Application buttons

Chapter 9: Architecture and demo Evaluation

As the title already implies, this chapter will be used to further elaborate on the results of this thesis by evaluating the derived architecture and demo application. The first thing that should be mentioned is that the time frame used to answer such a complex research question is not nearly enough to cover all aspects in detail. It is for this reason that, as mentioned in chapter 1, this research became an exploratory one. A lot of time has gone into familiarizing in the field and acquiring the necessary contacts to be able to perform the research. A set of interview questions along with a small introduction into CEP was set up to serve as a guideline during the interviews with the experts. Adhering to these guidelines has proven harder than initially thought since the field is very large and each expert had his/her own vision about the smart grid and what it should be like in the future. Using the input gathered by conducting these interviews, an architecture was developed and an event generator was created to serve as input to CEP engine. In chapter 9.1 the evaluation of the architecture (figure 20) is done, followed by the evaluation of the developed event generator and demo application in chapter 9.2. The evaluation summary and conclusion can be found in chapter 9.3.

9.1 Architecture evaluation

The architecture in figure 20 was developed using proven principles in literature, and was adhered to when developing and implementing the demo application. In order to determine whether an architecture is suitable for a given system, Bachman (Bachman, 2009) defines the following model (figure 44). The first thing necessary for an architecture is the definition of the business goals. The business goals are the core of a company and consist of what is important to the stakeholders. Using these business goals it becomes possible to determine the quality attributes of a system, which in turn can be used to determine the architecture needed to support the business goals. The other way around we see that an architecture should satisfy the quality attributes, which in turn should support the business goals.

Figure 44: Architecture evaluation model

Bachman (Bachman, 2009) defined the following 10 principles to describe the model (where 8,9, and 10 mean that you shouldn't blindly trust anything that is provided and always ask for convincing evidence):

1. Quality attributes determine the architecture
2. Business goals determine quality attribute requirements
3. Business goals represent what is important to its stakeholders communities
4. Quality attributes need to be specified with good measures
5. To understand an architecture, you must understand its quality attribute properties.
6. The most important quality attribute requirements determine the parts of the system to focus the analysis on.
7. The distribution of functionality in the architecture contributes to the quality attribute properties.

8. Guilty until proven innocent.
9. Proper analysis disallows assumptions. Only facts count
10. Evaluated organizations must own the evaluation results.

The application architecture in this research is based on a subset of quality attributes determined by the business goals set by the experts in the field. The results are therefore partly biased, as these experts do not form the complete set of stakeholders. The set of quality attributes defined by the experts were functionality, performance, integrability and modifiability. Due to the time available for this research however it was not possible to do a complete stakeholder research and quality attribute definition and hence, a couple of important attributes remain. The first step that should be taken when continuing this research is that of completely defining the architecture in figure 20 by determining the business goals and quality attributes using stakeholder input.

Since this research only focussed on the application architecture, the next step would be to make a good enterprise architecture, with extra focus on the security architecture due to the fact that more and more components are remotely accessible and sharing information over wireless connections, the risk of hacking and hijacking these components will also increase. A good enterprise architecture is a key requirement for the transition to smart grid as it will give us an overview of the complete system (from stakeholders to components and governance), which will be used as a guideline when making the transition and also helps insure proper management of the system (entire grid). In order to keep an overview on the large grids and its organization and governance, it is recommended to use a proven architecture framework and its principles to create the overall architecture in a correct manner. The Open Group Architecture Framework has proven excellent with dealing with situations of this magnitude (for example imagine the different suppliers necessary for the creation of a smart grid). They give the following definition:

''An effective enterprise architecture is critical to business survival and success and is the indispensable means to achieving competitive advantage through IT.'' (The Open Group, 2009, p. 5)

They also present the TOGAF Architecture Development Method (figure 45), which the iterative process that should be used to define the enterprise architecture. For each iteration of the ADM, a couple of decisions must be made beforehand and these are summarized below.

1. The breadth of coverage of the enterprise to be designed.
2. The level of detail to be defined.
3. The extent of the time period aimed at, including the number and extent of any intermediate time periods.
4. The architectural assets to be leveraged (including assets created in previous iterations of the ADM cycle and assets available elsewhere in the industry)

If we take a look at the ADM cycle in figure 45, we can conclude that in this research the requirements management process was started (part in the the middle), along with an application architecture (part of Phase C). Due to time constraints however, it hasn't been possible to dive further into the TOGAF Architecture Development Method, but it is strongly recommended to implement the Dutch utilities market according to these guidelines. Since building the application is only half the work and a solid infrastructure and governance plan are key necessities for the success of CEP in the utilities sector.

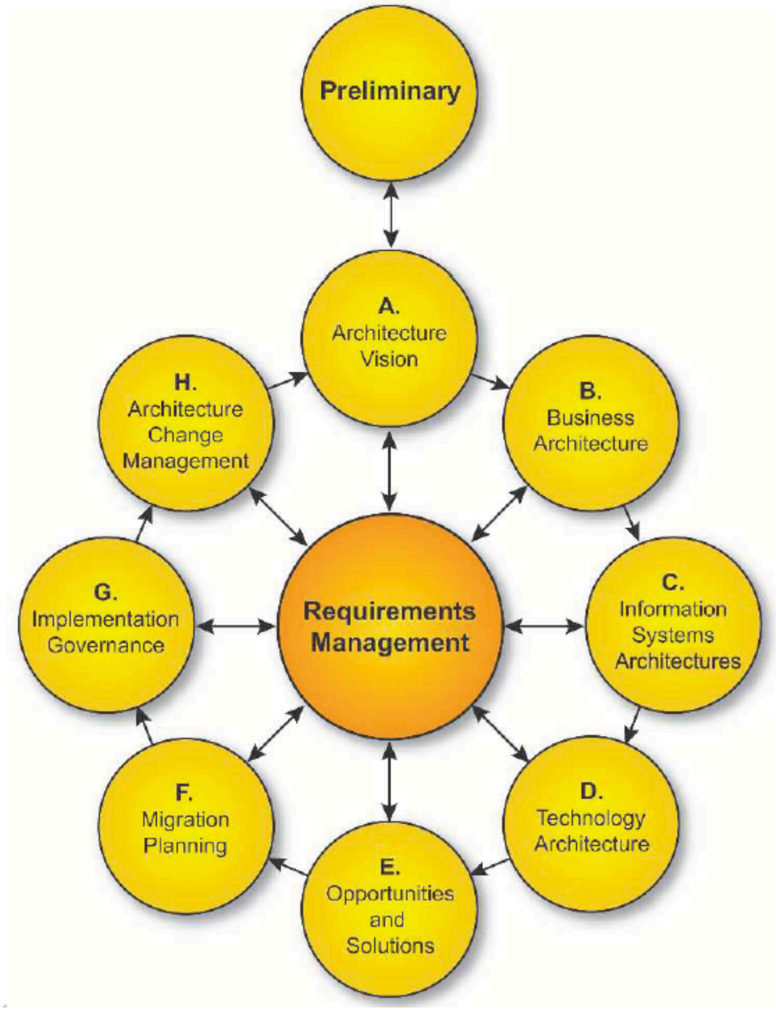


Figure 45: TOGAF architecture development cycle (The Open Group(2009))

The architecture derived in this research is based on the ADM method developed by TOGAF (The Open Group, 2009), but needs to get more in depth to give a better understanding of the impact on the Dutch electricity grid when making the transition to smart grid with event complex processing. Evaluation of the derived architecture has shown that more attention should be paid to establishing the stakeholders and extracting the business goals and expectations from these stakeholders for the transition to smart grid. The quality attributes used in this research follow from interviews with specific kinds of stakeholders (mostly innovation managers) and does not include the complete more realistic set of attributes. Using the ADM in more detail and completing the cycle (at least once) should take care of these gaps, but time restrictions did not allow for an in depth application of the ADM in this research. Therefore the application architecture derived during this research does not cover all the aspects needed to implement a CEP engine in the Dutch electricity grid. A complete system architecture including government of such a system is needed to gain more insight in this area.

9.2 Demo evaluation

Since no other project as attempted to combine CEP with smart grid, the goal of the designed system was to understand such a system rather than measuring its performance at this stage of the research. Even though the development of the application went as expected, the wish remained to go more in depth in the requirements of the CEP engine for the utilities market. Unfortunately time constraints did not allow for a more in depth requirements set. The developed application does give a clear example on how to build such a system by including all the steps necessary (from requirements to implementation). We first start with the evaluation of the event generator, followed by the evaluation of the CEP engine itself.

9.2.1 Event generator evaluation

A very important part of the demo is of course the event generator presented in chapter 6 since there is no way of properly testing the CEP engine without simulating the load and events in a smart grid network. The approach taken in this research is to model such a simulation grid's components in Java and find information in literature about the usage/ delivery statistics of each component. The usage statistics are then "hard coded" into the software to provide the CEP engine with input and these statistics are then looped and provide events for as long as needed by the CEP engine. This approach only allows for a set of static situations (Temperature mild for example), which can be simulated by changing the settings in the application.

In order to increase the reliability of the tests done, it is key that the simulation grid is as near to the actual grid as possible. This can be achieved by creating a more dynamic grid with several sets of statistics for the different seasons in a year. There are a couple of simulators for power grids available such as Mathworks' Simulink (Mathworks) and DOE's GridLAB-D (U.S. Department Of Energy (DOE)), however some research has shown that it would be difficult to connect these tools to the CEP engine within the given time-span.

The simulation model is limited in the fact that it is static and only has few predefined statistics hard-coded into it. The approach taken was feasible in the time available and allowed for quick connection to the CEP engine and easy alteration of the number of generated events. On the bad side this approach gave less depth in event types and less realistic simulations but allowed for a complete system implementation due to significant time gains. Before continuing the work on the CEP engine it is highly recommended to collect more statistics and create a more dynamic event generator to test with. Another key point is that no transport infrastructure was needed during the development and execution of the demo, since the event generator was in the same project as the CEP engine. In a more realistic setting this will not be the case and therefore more attention and research should be done in the area of transporting these generated events to the CEP engine.

9.2.2 CEP engine evaluation

The main part of the research was the CEP engine itself and up until this step we have derived an architecture explaining the working of the application, and the event generator to provide the engine with events. In chapter 5 we described an event processing language (EPL), which was used to model the requirements (set up with experts) into the Java application. The chosen requirements (rules) were relatively simple since the time available didn't allow for more requirements research and EPL modelling time, yet they form a solid base to prove how CEP can be used in a smart grid setting.

According to Espertech, a CEP engine created using the Esper toolkit is able to process about 500.000 events per second on a computer with a 2Ghz dual core processor (Espertech, 2012). With about 7.5 million homes in the Netherlands (Centraal Bureau voor de Statistiek), and each house producing up to a predicted 10 events per second (including all LV/MV events), the CEP engine would have to deal with about 75 million events per second. The designed system was able to handle around 350.000 events per second on average (mostly due to slightly slower laptop hardware), which is not nearly enough to process events in the entire grid. The main advantage of CEP however, is that several CEP engines can be connected together and be used to control different parts of the grid (about 150 engines needed for entire grid). During the research it became clear that the goals set for the demo application beforehand were to broad, and would have to be adjusted due to time constraints. Due to the limited time available for this project the requirements, which were implemented in the engine using EPL were of a basic nature. For more realistic performance results these requirements should be more extensive.

The evaluation of the CEP engine can best be analysed by answering a subset of the evaluation questions defined by the Sybase group (Sybase (Coral8), 2006). A quick literature study has shown this list to be the most comprehensive one regarding this projects specific needs and furthermore, every effort has been made to keep the evaluation questions objective and vendor neutral. The literature study regarding this topic also revealed some criticism on this paper by the Event Processing Technical Society (EPTS) Chairman Opher Etzion (Etzion, 2009). Further inspection of that criticism however has shown that it was not the sections and questions defined that caused the criticism but the rather narrow view of CEP in general. The paper generalised CEP into one category, which is certainly not the case, but it does happen to be the category this project is based on. Opher Etzion also states that most of the evaluation criteria are valid based on the fact that he had done similar research for IBM, which came up with similar criteria (Etzion, 2009). Since he is not allowed to publish those findings, we cannot compare them and hence, proceed with the questions and sections as set up in this paper.

The Sybase group arrange the evaluation questions in the following sections (Sybase (Coral8), 2006):

1. Programming model (Basic questions)
2. Programming model (Advanced questions)
3. Ease of integration and deployment questions
4. Enterprise features questions
5. Performance questions

The most important questions (relevant to the research) in each of these sections are stated below along with the findings done during the course of this research. The number in front of each question states in which of the sections defined above it belongs.

(1.) Does the programming model look familiar? – *The reason this question is asked is because if a programming model does not look familiar, the developers will resist using it.*

The EPL language is fairly similar to other types of programming software and should not be a problem for most programmers.

(1.) Is the primary programming model language-based or GUI-based?

The primary Esper programming model is language based and does not have an option to model in GUI.

(1.) Are continuous incremental queries supported? – *The queries reside in memory, and continuously compute and produce the requested outputs in response to the events.*

Continuous queries are an absolute necessity in this sector for the real-time handling of events. The queries developed during these research are continuous.

(1.) Are windows supported? – *(time) Windows allow you to define a subset of data from de data stream.*

Yes.

(1.) Are there rich features for joining and correlating multiple streams? – *This functionality allows for the creation of more complex queries for deductions.*

Yes.

(1.) Are event patterns supported? – *For example "If A and B happen and C does not happen within 5 minutes, then throw D"*

Yes.

(2.) What testing and debugging capabilities are available?

The Esper manual refers to J-Unit testing for all testing purposes. This can't be considered enough, as the modelled requirements should also be tested for correctness (is it really modelled according to the requirements specification?)

(3.) How does data get into the engine?

Since the events in this research are simulated, it is not possible to evaluate the used engine on this particular subsection. To evaluate the engine's data entry options, a more detailed demo should be created which makes use of actual data sources (sensors in the grid for example) instead of the generated ones in this research.

(3.) How are data anomalies handled? – *How does the engine handle delayed, out of order and/or lost messages?*

This is an important part of the engine, but was not included due to time constraints.

(3.) How does the engine produce output?

The fact that the engine is custom built in Java gives a lot of freedom when deciding on the output adapters. In this case the output results were transformed in Java methods being executed and simple print statements. However in subsequent more detailed versions of the engine, the output could be used to send text messages to mobile phones, send e-mails, used as input for other engines etc. These adapters should not be a problem when developing a tool from scratch.

(3.) Is there a publish/subscribe transport layer included with the engine?

The developed engine allows publishing to listeners, and printing on screen directly but does not include a transport layer.

(3.) Do I need to restart the engine to accomplish?

The developed engine does not allow the adding of a new query, data source, subscriber/publisher or a new action or alert without restarting the system. This is a much-needed feature in the area of smart grids; as the systems cannot be shut down to add new features (The electricity grid should be monitored at all times).

(3.) How easy is the engine to install?

The Esper API is easy to set up and use with java but does not come with a server platform. This should still be researched on best practices.

(3.) What platforms are supported?

Since this tool was created using Java, it is platform independent and can be used on any system with Java installed.

(4.) What security features are available?

Esper does not come with a set of features for security and these were not built during the research. This is a key factor and a necessity before actually implementing such a system in the Dutch electricity grid.

(4.) What monitoring and management capabilities are available?

Esper is a pretty basic tool and does not come with monitoring and management capabilities. For the purpose of this research however a small monitoring and management tool was created.

(5.) What is the throughput of the engine?

The throughput of the engine is about 350.000 events per second. It should be stated that the modelled requirements were of a rather simple nature and therefore allowed for faster event handling.

9.2.3 CEP engine evaluation conclusion

What we learn from this evaluation is that the designed engine does have a couple of weaknesses, some due to time constraints and some due to lack of support in Esper. The engine does show the capability of CEP and suitability of CEP for the Electricity grid at first. The modelling of requirements should however be taken to a much deeper level to make sure the EPL can handle the most complex problems as well.

9.3 Work overview

The figure below (figure 42) gives an overview of the work done during this research and the total amount of work necessary before the CEP engine can actually be used in the Dutch electricity grid. Two other really important factors that can decide the success or failure of CEP in smart grid, and can't be found in the figure below are that of arranging funding for the project and the technological advancement of the grid components. The items marked in blue have been completed in this research and those marked in red still remain open

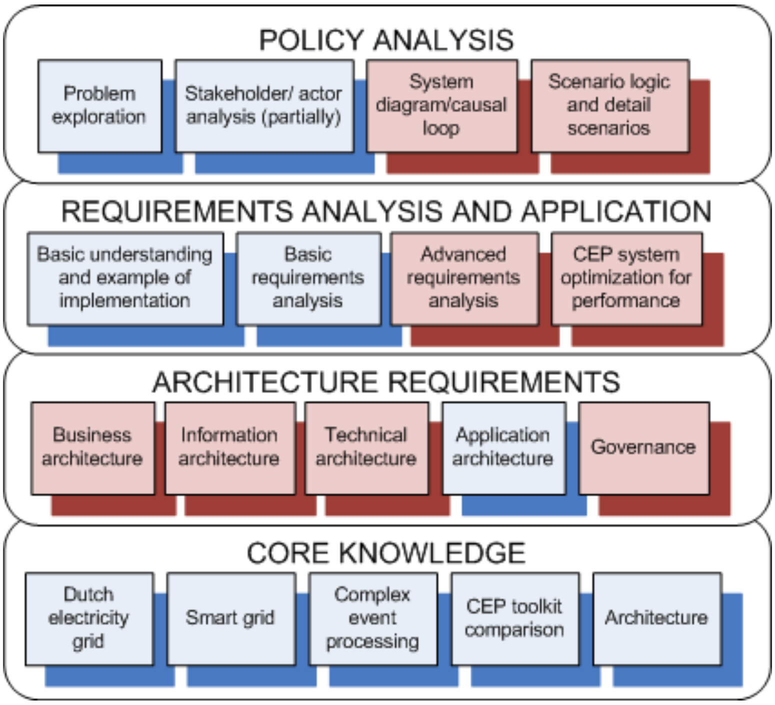


Figure 46: Complete picture of research

The tests done with the designed application showed very much promise in handling the smart grid requirements, however before any actual conclusions can be made, the next step should be taken. The next step includes creating a simulation by using actual components instead of the event generator, using a larger grid so that the connection of multiple CEP engines in a grid can also be tested and deriving more complex requirements to insure the suitability for the smart grid network.

Part 4: Summary, Contributions, Conclusions and Future work

Chapter 10: Summary and conclusions

This chapter gives a summary of the entire thesis followed by the conclusions of this research. We first start with the summary, in which all the key points of the previous chapters will be briefly repeated, followed by the conclusions we can draw from the performed research.

10.1 Summary

At the beginning of this thesis we introduced the subject by drawing a main picture of the current Dutch electricity grid and stating the predictions made about future power consumption and generation within this grid. Several articles in literature suggest that the increased electricity demand along with the increased electricity production from distributed sources (DG) will soon require more from the ageing electricity grid, than it is capable to deliver.

One of the most promising solutions to the current grid comes in the form of the smart grid. In a smart grid, the components and the way they communicate are switched from a passive to a more active nature. Smart meters and several sensors in the grid provide real time information about the grid and allows for more efficient usage of the existing grid. Due to the vast amount of generated information, it will be too much to handle and therefore rendered useless without a correct way of filtering the available information. Literature suggests that Complex Event Processing (CEP) might be the solution to this problem, but none of the articles actually test the suitability of CEP in a smart grid network.

This thesis attempts to fill that hole and in the process specifies all the relevant aspects that should be taken in to consideration when developing and implementing a CEP system for smart grid management. In order to get a better idea of the Dutch electricity sector and the smart grid, chapter 2 was used to elaborate on the Dutch electricity value chain followed by the managing parties of the electricity grid (TSO's and DSO's). To narrow down the scope of the research the part covered by the DSO's was chosen for the demo (Part of the network with most challenges, DG for example). From the several DSO's available in the Netherlands a selection was made of the 3 largest ones (Alliander, Enexis and Stedin, together form a 94% market share), since these also have had the most problems within the network (more components and cables than smaller DSO's). After getting to know all there was to know about the Dutch grid, the smart grid was introduced and elaborated upon in chapter 3. In the following chapter, chapter 4, the basic concept of event processing and the different types were elaborated upon, the alternative event driven technologies were stated, and the different CEP suites were discussed. Based on the assessment of these different suites, Espertech's Esper was chosen to implement the demo application.

With the knowledge of smart grids, the Dutch electricity sector, complex event processing and a chosen complex event-processing suite, chapter 5 focused more on the architectural aspects of the designed application. Besides elaborating on the relevant architecture subjects, chapter 5 was also used to briefly construct an architecture for the developed application. In chapter 6 the case study was introduced along with the event generator, which would function as the source of all the events for the event generator. The components of the electricity grid, which are represented by the event generator, are introduced and explained in the remainder of chapter 6. In chapter 7, the main challenges in smart grid the CEP engine will attempt to solve are introduced and elaborated upon. In this chapter the basic requirements derived during interviewing the experts, are show and translated to EPL.

Chapter 8 is used to describe the design of the developed application and show screenshots of the graphical user interface and also describes how the application works. The results of the above mentioned approach and developed application were evaluated in chapter 9.

10.2 Conclusions

In the years to come, the Dutch electricity sector will be confronted with a significant growth in electricity generation and demand and studies have shown that in the near future, the current grid will not be able to support these changes and developments. Using this input it has become clear that some form of change in the way we currently manage our electricity grid is necessary in the near future. Literature has shown that the most promising of these new electricity grid ideas comes in the form of the smart-grid. Though a lot of research is being done on the definition of a smart-grid and ways to transform the current ageing grids into a smart-grid, there is still little attention being paid to the actual management of these grids.

The chosen research subject has opened up many other questions and subjects that still need to be studied before CEP can actually be used in the smart grid network. Though the demo clearly shows the capability of CEP in the smart grid setting and shows in what way to achieve this, nothing can be concluded about its actual adoption and performance in the grid until extra research is done. This research also identifies those key factors where more research is needed and groups all the knowledge needed to start with such a project in one place. Furthermore this research can serve as a guideline for other researchers in this area as all the relevant fields are (briefly) covered. The fields that are briefly covered will be further elaborated upon in the future work section of this thesis. As for the main research question defined in chapter 1:

"How can Complex Event Processing be used for the management of electricity (control, purchase, sale) in a medium/low voltage (Dutch) smart electricity grid?"

Before we continue with the main question we take a look at the sub questions defined at the beginning of this research.

Why are the current ageing electricity grids not able to support the future electricity demand, technological advancements and the way in which the grids will be used in the future?
This question was answered in chapter one by stating the growth of the world population and the electrification of everything. We also presented some trending technologies as the electric cars which need to be charged and distributed generation. Traditional electricity grids will not be able to support all of these changes properly, which is why a new method is needed.

What is a suitable grid technology that can support (if managed correctly) these forecasted demands, technological advancements and usage styles if applied to Dutch electricity grids?
Smart grid is an upcoming technology that is already being experimented with throughout the world. Literature research has shown that this is indeed the most promising of technologies, which is why we present it in this research.

Is Complex Event Processing a suitable technology for the management of a smart grid in the Dutch electricity sector?
CEP has proven excellent for the processing of high volumes of events, which by just looking at them have no meaning. The possibilities provided by the EPL allows for filtering and reasoning over these events in order to provide useful information in (near) real-time.

What is an appropriate architecture that can be used to implement smart-grid using CEP in the Dutch electricity sector?
To answer this sub question a study was done on the several architecture types available, and the matching one was used to create a basic architecture, which in turn can be used for further research.

How can the behavior of a smart-grid be simulated using Java, and what components should this simulation grid consist of in order to provide the demo application with realistic usage and generation data?
This sub question resulted in the creation of the event generator, which was used as input to the developed CEP engine.

What is an appropriate application design to test if CEP can support the transition to smart-grid in the Dutch electricity sector on the implementation level?
The main goal of the research was to see complex event processing in action on a simulated grid. The answer of this question has caused the development of the CEP engine.

As you can see above, the question of how CEP can be used in smart grid was answered by separately researching and combining all the relevant subjects (Dutch electricity grid, Smart grid, Event processing, Architecture) needed to achieve this. Using this knowledge and that of interviewing experts in the field, an event generator and a CEP engine were developed to show how CEP should be used in such a setting. The answer to our main question is the following: The set up was proven successful in a demo and the engine responded to the generated events by performing predefined actions.

During this research, development and execution of the demo however, there were also a couple of issues discovered, which are summarized below:

- Better usage statistics should be used to create a more realistic event generator
- Performance of the CEP engine is largely dependent on EPL complexity
- A complete system architecture of the Dutch grid is necessary to gain more insight in this area
- More research should be done on methods to get the event data to the event processor

As already mentioned, at the moment a lot of effort is being done in the research of smart-grid technology without considering the actual management of these grids. This thesis makes a step in that direction by proposing and developing a system based on Complex Event Processing to achieve that goal and proves that it can be a key solution in managing these smart-grids, however there is still a lot of research to be done in this field. Because the vast area of the conducted research partly based on hypothetical situations and uncertainties, a follow up on the future work section of this thesis will be extremely important for the success of CEP in the utilities market.

With the proper background knowledge (this thesis), a proof of concept (this thesis) and further examination of the subjects covered in the future work, Complex Event Processing can play a major role in the latest developments of the utilities market, and therefore the world’s future.

Chapter 11: Contributions and Future work

In order to get a better idea of the importance of the future work in this research we start this chapter with the last sentence of the conclusion in section 10.2:

"With the proper background knowledge (this thesis), a proof of concept (this thesis) and further examination of the subjects covered in the future work, Complex Event Processing can play a major role in the latest developments of the utilities market, and therefore the world's future."

The first thing we notice is the fact that we refer to "this thesis" twice, and this of course cannot be done without substantiating what the contributions of this thesis to science are. Secondly we stress the importance of the future work to increase the chances of CEP succeeding in the utilities market. Section 11.1 will elaborate on the contributions of this research and chapter 11.2 will elaborate on the future work.

11.1 Contributions

Performing research on a new subject can be very difficult since there is no information of the problem and/or solution readily available. Therefore, the part that this research doesn't cover is that of comparing the results and performance to other similar systems. Instead what this research does contribute to science, are the points described in the following paragraph.

As literature points out we have discovered during this thesis that change is needed in the current Dutch electricity grid, and the "smart grid" is most likely to be the solution to the problems the current grid is facing. Switching over to smart grid however brings with it a lot of questions and challenges in terms of grid operation. One of the first things done in this research is identifying and gathering information about the relevant fields needed to tackle the described problem. The contributing factor here is that overall knowledge is now available in one document for software engineers not familiar with the utilities market. This also increases the potential group of people available to solve the problem as engineers can roll into the problem without prior knowledge of the utilities sector. It has also been shown that the developed CEP engine can handle the problem of data increase in smart grid by defining patterns that look for specific combinations of events (or lack thereof) in the event cloud and discarding the rest. This approach makes it possible for the system to deal with the hundreds of thousands of events per second. The application clearly shows the amount of events generated by the grid compared to the complex events and actions generated as in the first of the three columns (recall figure 33) it is nearly impossible to make sense of the posted information. Furthermore this research has compared several CEP suites and assessed there pro's and con's and giving a simple overview of all that is available. The architecture aspect of the system has also been covered in this thesis and gives the future researcher a basic set of requirements and guidelines that can be considered when implementing such a system. The described architecture (recall figure 18) forms an excellent starting point, but should be further developed in order to fully meet the requirements for the Dutch national grid. In the evaluation chapter the recommended direction is presented in the form of The Open Group Architecture Framework (TOGAF). The main contribution of this research comes from the results of the developed CEP-engine. The first step was defining the most important challenges facing the current Dutch electricity grid and future Dutch smart grid in collaboration with the experts of the 3 largest DSO's. After implementing the proposed system it has become clear that CEP dealt well with the test set of rules. The next step would be to make such a system based on actual smart grid components and testing on a real electricity grid.

11.2 Future work

Due to the limited time available for this research it has not been possible to cover all the parts in detail. This section is used to elaborate on what next steps should be taken in line of this research using 2 views. In the first one we look at the future work for the CEP-engine in the electricity grid, and in the second one we look at what future work has to be done within the developed system (figure 47).

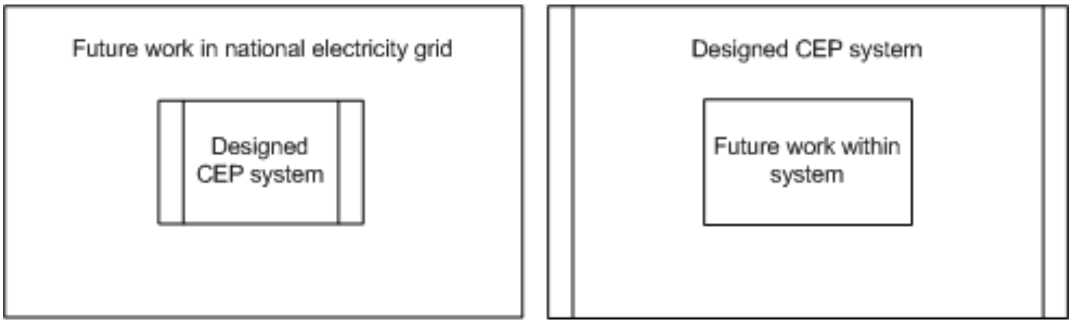


Figure 47: Future work areas

Let us start by restating the issues found in the developed demo below:

- Simulated load in generated events were static and hard coded into the system, more electricity usage and generation patterns give more realistic results. **(Within system)**
- Input events were generated using Java, therefore little attention was paid to how to get the events from the components to the engine (Wireless infrastructure, ESB, etc.). **(In national electricity grid)**
- The performance of the engine (processed events per second), are largely dependent on the complexity of the modeled EPL statements. **(Within system)**
- On architecture level there isn't a clear view on how the approach in this research could actually be implemented and governed in the Dutch electricity grid. **(In national electricity grid)**

For each of these issues it has been indicated in which category of the future work areas it falls and all of the issues will be discussed in the according sections. These 4 issues are the ones that need to be solved in order to make hard statements about the effectiveness of CEP in Smart grid. In the last section some general points are also stated and will need to be researched before implementation of such a system in the Dutch electricity grid.

11.2.1 Future work in national grid

One of the findings during the research was that input events were generated using Java, therefore little attention was paid to how to get the events from the components to the engine (Wireless infrastructure, ESB, etc.). In order to improve the reliability of the results in the demo it is key that research is done on the different possibilities to connect the components to the CEP engine.

The next issue found during this research was that on architecture level there isn't a clear view on how the approach in this research could actually be implemented and governed in the Dutch electricity grid. This complete system architecture is necessary to find out what (if any) parts were not discussed in this research.

11.2.2 Future work within designed system

Within the designed system, we found that the simulated load in generated events were static and hard coded into the system and more electricity usage and generation patterns give more realistic results. The usage and/or generation statistics of each component should be analyzed more carefully. As previously mentioned, there are several tools available that generate continuous streams of data and thus allow for a better event generator and hence better test cases for the CEP engine.

Another point is that the performance of the engine (processed events per second), are largely dependent on the complexity of the modeled EPL statements. The first thing that needs to be done in this area in order to get more accurate results from the demo, is to plan meetings with experts and get a thorough set of requirements which should then be translated to EPL. The performance of the engine should then be re-measured when running with the new set of rules.

11.2.3 General

One of the first things that should be researched next to the previously mentioned subjects, is what other possibilities are available to address the problem of the ageing electricity grid. This research focuses on smart-grid being "the solution" and therefore does not address the other options. It is however important to asses these options and determine the chance of smart-grid making its actual appearance in the Dutch electricity sector. As already mentioned CEP will be of little use without the active smart-grid and a different approach for the current grid problem might render further research useless.

Secondly one of the things that will have a big impact on the success or failure of CEP in smart-grid and should be further researched is that of the total cost of implementing such a system in the Dutch national grid. In chapter 4 it has already been mentioned that on average the implementation of a CEP system costs between 100.000,- and 250.000,- dollars but keep in mind that this is just for the CEP-engine and not the remaining cost.

In order to manage the entire Dutch grid using CEP, further research should also be done in the area of coherence. Several CEP-engines could be connected to each other and exchange high level event where each engine would be responsible for its own part of the grid.

And last but most certainly not least, one of the aspects that will become important instantaneously with the implementation of smart-grid is security. With all the components being "on-line" and some of those components even being controlled remotely, a lot of thought should be put in proper security for all of the communication within the grid as well as the CEP-engine itself.

List of Figures

Figure 1: Power loop flow example (J.Kok, 2010) 3
Figure 2: Project roadmap 9
Figure 3: Energy value chain 12
Figure 4: Electricity grid from generation to delivery 13
Figure 5: Regional DSO's with market share 14
Figure 6: Power Matcher concept 17
Figure 7: Vakantiepark Bronsbergen (D.T. Ho, 2010) 18
Figure 10: An event changing the state of an electricity grid 22
Figure 11: Model of situation awareness in dynamic decision-making (Endsley, 1995) 25
Figure 12: Formal links Whittle and Myrick (2005) 34
Figure 13: Business relation (G.Versteegh) 35
Figure 14: Communication between smartgrid components (EPRI) 36
Figure 15: Service oriented architecture model 37
Figure 16: SOA model with webservices 38
Figure 17: ESB scenario with 4 applications 38
Figure 18: Event Driven SOA (Olga Levina, 2009) 40
Figure 19: EP lifecycle (Adrian Paschke, 2009) 41
Figure 20: Complete system architecture 41
Figure 22: The smart simulation grid 45
Figure 23: Schematic overview of a typical smart meter configuration 46
Figure 24: A three-phase vault-type distribution transformer (left) and a submersible distribution transformer (right) 47
Figure 25: Efficiency CHP compared to Traditional Energy sources (N. van der Velden, 2008) 48
Figure 26: Solar cells connected in an array (Solar panels) A solar cell 48
Figure 27: Different sizes of HAWTs 50
Figure 28: Base situation with all the lines connected 51
Figure 29: Example setup 52
Figure 30: Different types of houses 53
Figure 31: Electricity demand for the different types of houses (7 days) 53
Figure 32: Average daily electricity demand. 54
Figure 36: Event generation 67
Figure 37: Event class 67
Figure 38: Listeners for the events 68
Figure 39: Developed application 68
Figure 40: Event windows 69
Figure 41: Live graph 69
Figure 42: System properties 70
Figure 43: Application buttons 70
Figure 45: TOGAF architecture development cycle (The Open Group(2009)) 73
Figure 46: Complete picture of research 78
Figure 47: Future work areas 84
Figure 48: Event producers, processors and consumers. Source: Bou-Ghannam & Faulkner(2008) 95
Figure 49: Smart grid (http://smartgrid.epri.com/Demo.aspx) 96
Figure 50: Project roadmap 97

List of tables

Table 1: Electricity growth estimates (Jesse Berst, 2008) 2
Table 2: Event Driven Technologies 26
Table 3: Evaluated CEP platforms 28
Table 4: Platform advantages and weaknesses 28
Table 5: Table corresponding to figure 27 51
Table 6: Table corresponding to figure 28 52
Table 7:Component identifiers 56

Bibliography

(EPRI), D. v. (2009). *Report to NIST on Smart Grid Interoperability Standards Roadmap.* Cimetrix inc., EPRI, Cox software Architects LLC, Enernex corporation, Hypertek inc., UCI, UTInnovation LLC, XCI.

A.T.M.Aerts, J. D. (2003). *Architectures in context: on the evolution of business, application software, and ICT platform architectures.* Elsevier B.V.

Adrian Paschke, P. V. (2009). A reference architecture for event processing (DEBS '09). *Proceedings of the third ACM international conference on Distributed Event-Based Systems* , 3-4.

Aftim Acra, Z. R. (1984). *Solar disinfection of drinking water and oral rehydration solutions.* Department of environmental health.

Anthony R. Metke, R. L. (2010, June). 2010 Innovative Smart Grid Technologies ISGT: Security technology for Smart Grid Networks. 1-7.

APX-ENDEX. (2010). *http://www.apxendex.com/index.php?id=15030-07-2010)* (Vol. website). APX-ENDEX group.

B.Baarsma, B. M. (2008). *Elektriciteit: een spannende markt p. 12-14.* Dies Natalis Faculteit Economie en Bedrijfskunde en Kring van Amsterdamse Economen.

Bachman, F. (2009, 01 01). *The principles of successful architecture evaluation.* Retrieved 08 16, 2012 from Software engineering institute: http://www.sei.cmu.edu/library/assets/20090910webinar.pdf

Cameron W. Potter, A. A. (2009). Building a smarter smart grid through beter renewable energy information. *Power systems conference and exposition, PSCE '09, IEEE/PES* (pp. 1-5). IEEE (10.1109/PSCE.2009.4840110).

Centraal Bureau voor de Statistiek. (n.d.). *CBS*. Retrieved 05 08, 2012 from http://statline.cbs.nl/StatWeb/publication/?VW=T&DM=SLNL&PA=37312&D1=a&D2=a,!1-4,!6-7&HD=120607-0605&HDR=G1&STB=T

D.T. Ho, J. C. (2010). *Network losses with photovoltaic and storage.* European Association for the Development of Renewable Energies, Environment and Power Quality (EA4EPQ).

Deconinck, G. (2008). An evaluation of two-way communication means for advanced metering in Flanders (Belgium). *IEEE Instrumentation and Measurement Technology Conference* (pp. 900-905). Canada: City.

Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *(A. M. Huberman & M. B. Miles, Eds.). 14*, pp. 532-550. The Academy of Management Review.

Endsley, M. R. (1995). *Towards a theory of situation awareness in dynamic systems.* Texas Tech University.

Espertech. (2011). *Esper for Java: Reference manual (http://esper.codehaus.org/esper/documentation/documentation.html).* Espertech.

Espertech. (2012). *Esper for Java: Reference manual (http://esper.codehaus.org/esper/documentation/documentation.html).* Espertech.

ETP. (2011). *(http://www.smartgrids.eu/?q=node/163).* European technology platform (Electricity networks of the future) .

F.Blaabjerg, Z. R. (2006). *Power electronics in wind turbine systems.* Aalborg university, Institute of energy technology . Denmark: IEEE.

Fens, T. (2005). *Trends in energy: Utility value chain optimalization 2005.* Capgemini utrecht. Utrecht: Capgemini.

G.Versteeg, H. (2006). *Business architecture: A new paradigm to relate business strategy to ICT.* Springer.

Giovanni Montana, F. P. (2008). *Data Mining for Algorithmic Asset Management.* Imperial College London, Department of Mathematics. ICL.

H.Riegler. (2003). HAWT versus VAWT. 44-46. Italy: REFOCUS.

Hugo Haas, A. B. (2009). *Web services glossary (Working group note)*. Retrieved 06 12, 2012 from W3C: http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/

J.Kok, M. a. (2010). *Intelligenve in electricity networks for embedding renewables and distributed generation.* Energy Research Centre of the Netherlands (ECN) p. 179 - 209.

J.Makansi. (2007). *Lights out: the electricity crisis, the global economy, and what it means to you.* Wiley.

J.V.Hoof. (2007). SOA and EDA: Using events to bridge decoupled service boundaries. *SOA Magazine issue 5* .

Jeffe, M. (2010, 8 24). *Xcel says costly SmartGridCity pilot project in Boulder won't be repeated or expanded Read more: Xcel says costly SmartGridCity pilot project in Boulder won't be repeated or expanded - The Denver Post http://www.denverpost.com/business/ci_15872574#ixzz21fbT51dQ Read The Denver Post's Terms of Use of its content: http://www.denverpost.com/termsofuse*. Retrieved 7 22, 2012 from The Denver post: http://www.denverpost.com/business/ci_15872574

Jesse Berst, P. M. (2008). *The electricity economy: white paper.* Global Environment Fund (GEF) p.12-20.

Johan Boekema, R. B. (2009). *Viesie op smart grids van netbeheerders, de ontwikkeling van een gezamenlijke toekomstvisie 2025.* Netbeheer Nederland, Vereneging van Energienetbeheerders in Nederland, Projectgroep smart grids. Netbeheer Nederland.

John Bates, M. P. (2007). *Next-Generation Algorithmic Trading (DOI: 10.3905/tr.2007.681982).* Apama products.

Jr., S. O. (2007). Getting on board the enterprise service bus. *Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4160214S (01-02-2012).* Computer, 40.

K.Chandy. (2006). *Event-driven applications: Cost, benefuts and design approaches.* Gartner application integration and web services.

K.Chandy, M. A. (2007). Towards a theory of events (DOI: 10.1145/1266894.1266929). *DEBS '07 Proceedings of the 2007 inaugural international conference on Distributed event-based systems* (pp. 180-187). ACM.

K.Dodrill. (2010). *Understanding the benefits of the Smart Grid.* DOE/NETL-2010/14/13.

Leavitt, N. (2009). Complex-Event Processing Poised for Growth (http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=5054866&contentType=Journals+%26+Magazines). *Computer , 42* (4).

Lublinsky, B. (2007). *Defining SOA as an architectural style: Align your business model with technology*. Retrieved 04 18, 2012 from IBM: http://www.ibm.com/developerworks/architecture/library/ar-soastyle/

M. van Lumig, M. L. (2009). *Impact DG en 'nieuwe belastingen' op het LS-net in bestaande woonwijken.* Labored, Hermes DG 3. Stijn Uytterhoeven.

M.Hommelberg, B. C. (2007). *Aggregratie van micro-wkk's in een virtuele centrale (First trial smart power system).* ECN, ESSENT. ECN.

Mathworks. (n.d.). *Simulink.* Retrieved 06 01, 2012 from Simulation and model-based design: http://www.mathworks.com/products/simulink/

MetaVu Xcel Boulder city evaluation. (2011, 12 15). Retrieved 07 22, 2012 from CSRWIRE: http://www.csrwire.com/press_releases/33515-MetaVu-Completes-Independent-Evaluation-of-Xcel-Energy-s-SmartGridCity-TM-

Michelson, B. (2006). *Event-driven architecture overview: Event-Drive SOA is just part of the EDA story.* From http://www.psgroup.com/detail.aspx?id=681

Mike Gualtieri, J. r. (2009). *The Forrester Wave: Complex Event Processing (CEP) Platforms, Q3 2009.* Forrester.

N. Museux, J. M. (2006). Complex Event Processing approach for Strategic Intelligence. *Information Fusion, 2006 9th International Conference* (pp. 1-8). Thales group.

N. van der Velden, P. S. (2008). *Energiemonitor van de Nederlandse glastuinbouw 2008.* Wageningen UR, LEI. LEIW .

Olga Levina, V. S. (2009). Realizing Event Driven SOA. *Fourth International Conference on Internet and Web Applications and Services* (pp. 37-42). IEEE Computer society.

R. Whittle, C. M. (2005). *Enterprise Business Architecture: The formal link between strategy and results.* Auerbach publications.

Ranadive, V. (1999). *The power of now (ISBN-13: 9780071590129).* Computing McGraw-Hill.

Rob van Gerwen, S. J. (2006). *Smart metering.* Retrieved 12 01, 2011 from IDC-ONLINE: http://www.idc-online.com/technical_references/pdfs/electrical_engineering/Smart_Metering.pdf

S.Haeckel. (1999). *Adaptive enterprise: Creating and leading sense-and-respond organizations.* Harvard business press.

Schulte, R. D. (2008). *Event Driven SOA is driving ahead (http://www.gartner.com/id=708313).* Gartner.

Sessions, R. (2007). *A Comparison of the Top Four Enterprise-Architecture Methodologies.* Object watch Inc.

Sybase (Coral8). (2006). *Comprehensive guide to evaluating event stream processing engines.* Sybase (www.sybase.com).

TenneT-group. (2010). *TenneT website (http://www.tennet.org/tennet/organisatie/kerngegevens.aspx20-07-2010).).*

Thanos Magoulas, K. P. (1998). *Strategisk IT-management: Volume 13 of Gothenburg studies in informatics.* Department of Informatics, Göteborg University.

The Open Group. (2009). *The Open Group Architecture Framework (Version 9 ISBN: 978-90-8753-230-7).* The Open Group.

U.S. Department Of Energy (DOE). (n.d.). *GridLAB-D, the smart grid simulator.* Retrieved 06 01, 2012 from GridLAB-D: http://www.gridlabd.org/

utschig-utschig, C. (2008). *Architecting Event-Driven SOA: A primer* . Retrieved 10 6, 2011 from ORACLE: http://www.oracle.com/technetwork/articles/oraclesoa-eventarch-097519.html

Wikipedia group. (2009, 04 01). *Smart_grid*. Retrieved 05 22, 2012 from Wikipedia: http://en.wikipedia.org/wiki/Smart_grid

Wiles, J. (2011). *Introduction to solar energy.* http://www.altestore.com/howto/Getting-Started-Renewable-Energy-Sustainable-Living/Introduction-to-Solar-Electricity/a89/.

X. Fang, S. M. (2011, 12 9). Smart Grid — The New and Improved Power Grid: A Survey (10.1109/SURV.2011.101911.00087). *Communications Surveys & Tutorials, IEEE* .

Xcel-Energy. (n.d.). The Smart Grid City. *http://www.toyota.com/esq/pdf/SmartGridCity.pdf* .

Y.Simmhan, S. B. (2011). *An informatics approach to demand response optimization in Smart Grids.* University of Southern California, Computer science. Los Angeles: Center for Energy Informatics.

Zachman, J. (1996). *Enterprise Architecture: The issue of the Century.* Zachman institute for Framework Advancement (ZIFA).

Appendices

Appendix A: List of abbreviations

CEP	:	Complex Event Processing
CHP	:	Combined Heat and Power
DSO	:	Distribution System Operator
EDA	:	Event Driven Architecture
EP	:	Event Processing
ESP	:	Event Stream Processing
TSO	:	Transmission System Operator

Appendix B: Interview setup

Below you can find the interview as conducted with experts in the field. The first part of the document was used as an introduction to the subject and research for the experts, and the questions were used to initiate a discussion, which would later on result in useful information.

1.1 Introduction

As the final part of my master Information Architecture at the university of technology Delft, I am currently working on creating a CEP –Engine (More on CEP in 1.2), which can semi-autonomously manage a smart grid network. This project builds on previous work by a student at TU Delft[1], which, in his work, analyzed the possibility of applying complex event processing in electricity grids.

This project will address the effect of Complex Event Processing on the electricity grid by creating a model, which will mimic the events processed in an actual electricity grid, and use it as input to the developed CEP-System. The strength of CEP lies in the fact that it can process a lot of real-time information reaching the engine in the form of events. This, along with the fact that components used in networks are becoming "smarter" by allowing two-way communication, makes CEP an excellent candidate for smart grid management.

The main focus will be on the changes that the transition to smart grid (for example Distributed Generation) will bring to load balancing, outage management and congestion management, the most important challenges of today's electricity grids (tested in a demo). Furthermore I will attempt to cover all the fields necessary for actual implementation in theory (Future work).

Due to the fact that this problem (challenge) will most likely take place in the medium and low voltage network, the main parties involved in this project will the Dutch Distribution System Operators, and several companies involved in the pilot projects like PowerMatcher in Hoogkerk (KEMA for example).

The following components will be included in the demo-electricity grid (Feel free to comment on list):

- Houses with/without smart meters
- (Micro) CHP's
- PV panels
- Small Wind turbine farm in MV network
- Electricity cables
- Transformers

One of the biggest challenges (next to extracting rules for the CEP engine) is programming the components in the list displayed above to mimic the events such a network would generate in real-life.

[1] Roland de Boo – Application of Complex Event Processing in Electricity Distribution Systems (2009)
[2] European smart grids technology platform (http://www.smartgrids.eu/?q=node/163) (2011)

1.2 Complex Event Processing

An owl hunts mice at night and will pounce on a mouse from flight when it detects it moving on the ground. The mouse might sense this and respond by jumping out of the way but will die if it responds inappropriately. This reaction (or lack thereof) is referred to as a sense and respond system. What we learn from this is that a living being that does not detect threats gets killed. Likewise, one that frequently takes unnecessary action wastes energy and perishes. It is this same idea that I will try to apply in the smart grid network in the form of Complex Event Processing. CEP is referred to as the process of timely detecting events and patterns that require action, and executing these actions in order to keep the whole system (smart grid) alive.

In complex Event Processing different events from different sensors in a network are fired at the CEP Engine in the form of a cloud (events are fired at the system in an unordered fashion). Complex in this technique does not refer to the word "difficult", but to the fact that events are generated in a cloud and the CEP engine has to make sense out of all the data received.

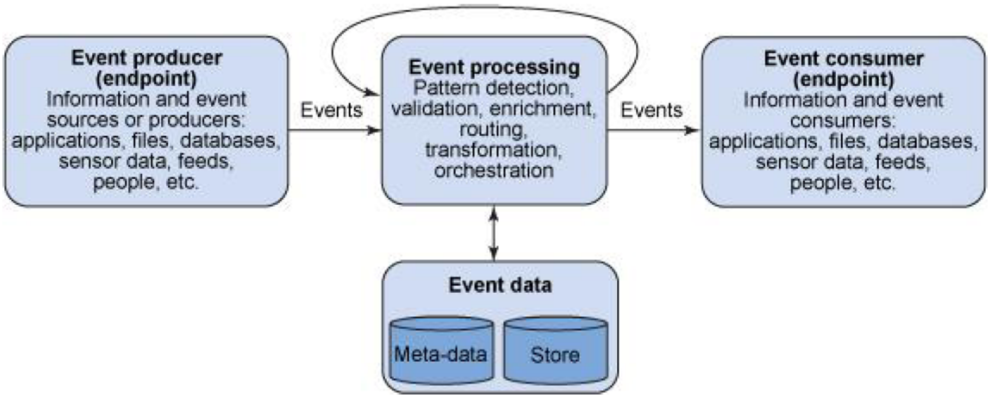


Figure 48: Event producers, processors and consumers. Source: Bou-Ghannam & Faulkner(2008)

In figure 1 you can see the basic working of the CEP-Engine. We have event producers (sensors in the network), which constantly fire events in turn caught and analysed by the engine based on predefined rules and patterns. When the events are processed and detected they are passed to the subscribers and appropriate action is taken, i.e. the CEP-Engine will attempt to fire commands back at the grid in order to balance it out. Should managing some part (or problem) of the network autonomously not be possible with technology available today (or in the near future), the system should fall back on traditional ways (e.g. notifying an employee with an sms alert)

1.3 Smart grid and casus

The European smart grids technology platform defines a smart grid as the following[2]:

"*Electricity networks that can intelligently integrate the behavior and actions of all users connected to it - generators, consumers and those that do both – in order to efficiently deliver sustainable, economic and secure electricity supplies*"

Though there is no general definition for the term "smart grid", the one described above, will in large part be what this project will build on. We will look at the smart grid as an interrelated network of consumers (houses), generators (wind turbines, solar panels) and those that do both (House with CHP and/or solar panels for example).

Figure 49: Smart grid (http://smartgrid.epri.com/Demo.aspx)

The case, which would be modeled in Java (Programming language), consists of a small network with a couple (amount still has to be specified) of houses connected in a network with smart meters and small CHP's. To complete the picture (Figure 1), we could also add solar panels and wind turbines to the network. Due to the fact that Distributed Generation (DG) occurs in this network (solar panels, CHP's, wind turbines), one of the most important aspects of operating this grid is **load balancing**. This, as mentioned before, along with **Outage management** and **Congestion management**, will be the main problems used to test the applicability and efficiency of CEP software in the "regular " electricity grid. If we once again take a look at figure 1, the CEP engine could be placed in the center of the picture (Where the control room is depicted).

In order to start the development of the CEP system, a model (simulation) of the smart grid network will have to be created first. This model has to be based on actual knowledge and facts since the events generated in this simulation will be used as input for the CEP-Engine. The main goal of this interview is to reveal the possible events within a smart grid network with distributed generation and the biggest problems encountered when running such a network.

[2] European smart grids technology platform (http://www.smartgrids.eu/?q=node/163) (2011)

1.4 Project roadmap

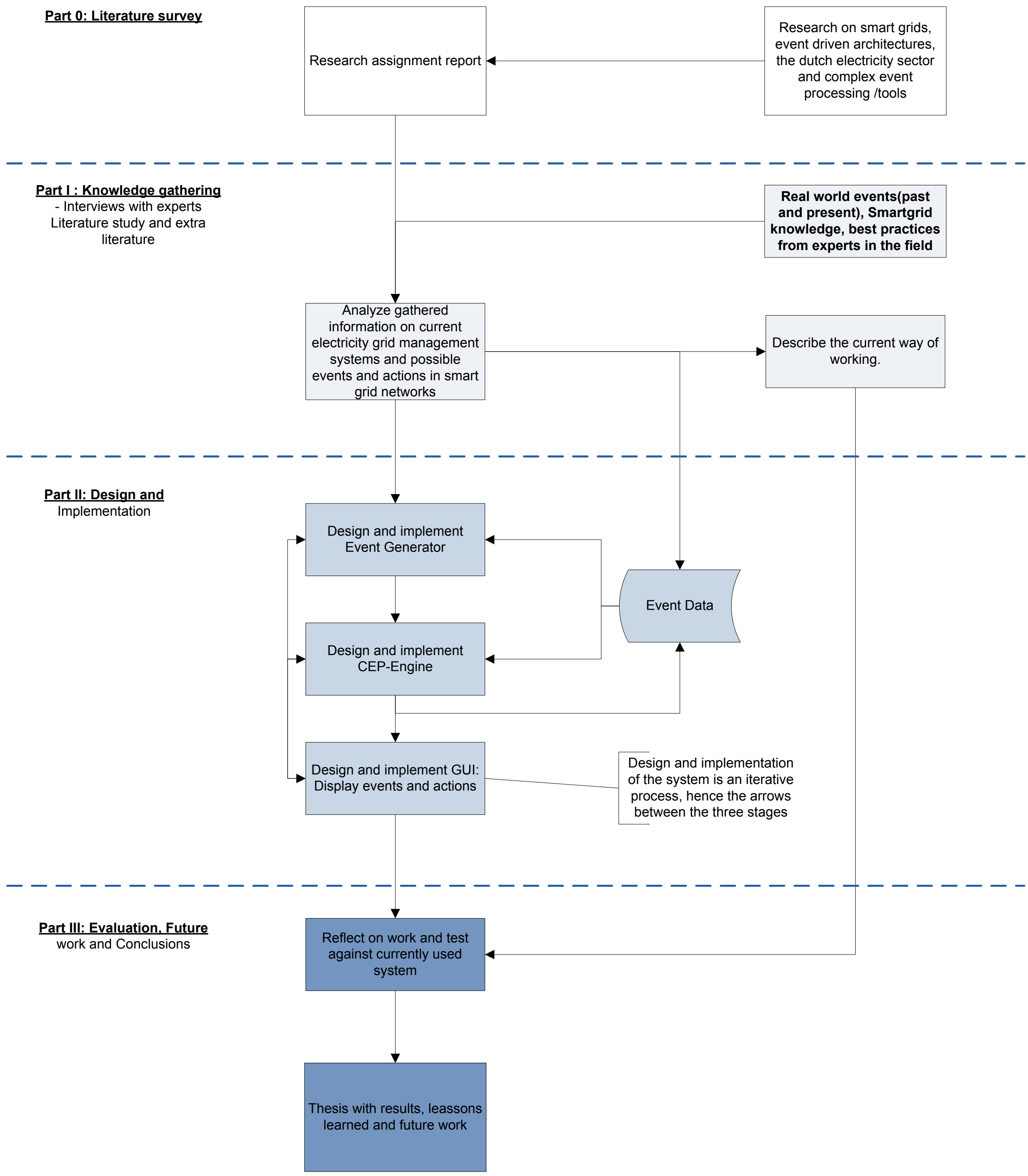


Figure 50: Project roadmap

1.5 Proposed solution

In figure 2 you can see the project roadmap consisting of four parts. In “Part 0: literature survey” research was done on all the relevant subjects and along with that report and the result of this interview, an event generator will be made (mimicking the events an ordinary electricity grid would throw). The events generated in this step will be used in the CEP-Engine and finally shown in the GUI. The CEP system will be evaluated against the current way of working (load balancing, outage management and congestion management) in order to assess the efficiency and effectiveness of CEP in the regular (and future) electricity grid.

1.6 Interview questions:

1. How should the simulation model be constructed? (To give a representative view of the real situation) What components should be in the simulation grid (amount of houses, CHP’s, smart meters, solar panels, wind turbines, kind of electricity cables)? → *The result of this question should be a schema, which will then be used in the thesis to elaborate on the simulation model.*

2. The current electricity grids (in terms of load balancing, outage management and congestion management) require various rules and measures to keep operational. What are these rules and measures? → *The result of this question should be an extraction of the “intelligence” which keeps the grids operational. This information will be used to create the CEP rules necessary to run the CEP system.*

3. How can we assess the performance of the designed CEP-system against running pilot projects? And how were the pilot projects assessed for performance? → *The result of this question should be several points on which the CEP system can be evaluated (Point used in the pilots for example?)*