UNIVERSITY OF LUGANO
ALaRI INSTITUTE
Faculty of Informatics
Master of science in Embedded Systems Design

Model driven system engineering methodology for VPS design

Master Thesis of:
Ing. Igor Kaitović
Student ID: **3005370**
email: *igor.kaitovic@usi.ch*

Tutor: **Ing. Slobodan Luković**
Supervisor: **Ing. Marcello Mura**

Academic years 2008 – 2010

Acknowledgments

This is to sincerely thank to the ALaRI committee, especially to Professor Mariagiovanna Sami and Ing. Umberto Bondi, for granting me a scholarship and giving me the opportunity to obtain a wide knowledge in the area of embedded systems design while working with world class professors and experts in this field. Furthermore I wish to express my gratitude to the AlpEnergy project partners, for all the consulting and provision of project materials. Most of the help and guidelines during, almost two years, work on this thesis and related publications I got from my advisors Slobodan Luković and Marcello Mura to whom I express a special gratitude. Moreover I wish to thank to all the ALaRI stuff, especially to Onur Derin and Alberto Ferrante for providing me a constant help and tools support. Last, but not the least acknowledgement for all the support, understanding and patient goes to my girlfriend, to my parents, my flatmates and all great friends and colleagues I met during my studies at ALaRI.

Author

Abstract

Economical and environmental concerns push towards solutions for sustainable energy supply. Introduction of numerous distributed resources (DER) and new types of loads, as expected to be in a near future, will cause power system instability due to uncontrollability, unreliability and low production predictability of DERs. A necessity for an advanced power system, SmartGrid, able to provide more enhanced algorithms for power stability and usage optimization emerges. Virtual Power System (VPS), tackles the challenge of unreliability, by aggregating various DERs and controllable loads and presenting them, to the rest of the power system, as one reliable and controllable entity in technical and commercial sense. In order to develop a mutual understanding platform for stakeholders coming from different areas of expertise and define requirements of embedded devices enabling VPS implementation, a model-driven system design methodology has to be adopted and applied on the VPS case.

Keywords

Virtual Power System, Smart Grid, Model-driven system engineering methodology, System Modeling Language, embedded systems requirements, behavioral and structural modeling.

To my loved ones.

Contents

List of Figures

List of Tables

“*We are whirling through endless space, with and inconceivable speed, all around everything is spinning, everything is moving, everywhere there is energy. There must be some way of availing ourselves of this energy more directly. Then, with the light obtained from the medium, with the power derived from it, with every form of energy obtained without effort, from the store forever inexhaustible, humanity will advance with giant strides. The mere contemplation of these magnificent possibilities expand our minds, strengthens our hopes and and fills our hearts with supreme delight*“

Nikola Tesla

Chapter 1

Introduction

Awareness of global climate changes caused by different kinds of pollution on the one side and increasing concern about sustainable energy supply on the other side have created a global sensitivity to energy efficiency as well as an interest in finding pollution-free and sustainable solutions. Smart Grid [2], has emerged as a promising response to these needs. It embraces wide range of technical issues starting from electricity generation paradigm shift (i.e. from central towards distributed) to wide deployment of intelligence in all segments of the power system boosting interaction among all actors in electricity distribution (e.g Substation automation, Demand Side Management etc.). Nevertheless, these trends would result in tremendous data transmission, storing and processing requirements which would urge for a kind of distributed system solutions.

On the other side, a novel concept of Virtual Power System (VPS) has been introduced in scope of AlpEnergy research project [1] aiming at efficient integration of Distributed Energy Resources (DERs) and controllable loads in a single entity in technical and commercial sense. A cluster created in such manner would be able to autonomously participate as an actor on energy market and trade energy still providing reliable supply and optimal energy prices for customers and producers inside it. VPS covers wide range of commercial, power system and ICT (information and communication technology) related issues. Nevertheless, due to the fact that many heterogeneous and uncontrollable generators must be coupled together with indeterministic open market conditions, such systems require complex and robust ICT solutions to support needed technical and commercial advances. The ICT structure must be optimally tailored to satisfy stakeholders' requirements considering at a same time the urge for constant system evolution and upgrade. These demands lead to a request for modular, scalable and flexible system design. VPS represent a cross disciplinary topic in which many fields as power engineering, automation, control, finance, legislation and many more, meet. Tackling the challenge posed by system complexity and inter-disciplinary cooperation, requires a holistic and standardized approach which systems engineering provides.

Bearing in mind all facts mentioned above, this thesis, as being to some extend, related to AlpEnergy project [1] introduces a specification, modeling, optimization and analysis methodology for VPS design flow developed. The methodology is based on model-driven system engineering design methodology and System Modeling Language (SysML), [69], [16]. The aim is to propose an instrument to cope with the complexity of the VPS providing a communication platform for experts and practitioners coming from different fields and at the same time facilitates analysis, simulation and design of the system while baring in mind role of embedded systems' role in both VPS and Smart Grid.

Proposed methodology is based on top-down approach starting with stakeholders identification and determination of their requirements. Using SysML sequence and use case diagrams further refinement of these requirements is done while their are being mapped into system ones. Defined system requirements are fatherly used as a starting point of high system model development. Besides provision of mutual understanding platform for different stakeholders, developed model is used for identification, functionalities extraction and structure proposal for VPS embedded devices. Moreover, the proposed modeling methodology shows system representations from different aspects and in different levels of granularity so that system architects can have better insight on whole design as well as in all components and their interactions.

The final result of the thesis and related part of the project would be an instrument that formally describes the system context, efficiently gathers user requirements, transfers them into system requirements and extracts concrete functional requirements for system components. The methodology and developed VPS model should serve as a medium to facilitate efficient system design bridging a gap among different fields experts and stakeholders.

The thesis is organized as follows: Chapter 2 gives a brief overview on power system evolution and introduces concepts of Smart Grid and VPS. Moreover, an overview of model-driven system engineering methodology is placed here. Chapter 3 discusses, in a general manner, the set of problems emerging with the VPS concept modeling and development and defines a set of particular problems to be tackled in scope of the thesis. Chapter 4 introduces the proposed VPS model-driven methodology and gives a brief overview on SysML and tools to be used. Application of proposed methodology on VPS concept, resulting structural and behavioral models and description of developed simulation framework are presented in Chapter 5. Chapter 6 gives an example on how obtained VPS models and simulation framework could be fatherly used for embedded devices requirements extraction, structural modeling and algorithm optimization. Finally, some conclusions being drawn out of the work behind this thesis, together with possible directions for future work are placed in chapter 7.

Chapter 2

State of the art

"We shall not satisfy ourselves simply with improving steam and explosive engines or inventing new batteries; we have something much better to work for, a greater task to fulfill. We have to evolve means for obtaining energy from stores which are forever inexhaustible, to perfect methods which do not imply consumption and waste of any material whatever"

Nikola Tesla

The purpose of this chapter is to give a brief overview on power system evolution and modeling concept, so to provide a basic motivation for this thesis. Without going into technical details, a basic structure of the existing power system, it's capabilities and limitations when compared with future power system trends are given first. A vision of a future power system is presented trough overview of Smart Grid and Virtual Power System concepts. Moreover, a model-driven methodology as a quite well-know and widely used concept in complex systems development is introduced. Finally, an overview on ongoing projects relevant from the point of view this thesis is given.

2.1 An overview on traditional power system

Evolution of the electrical power system has started by the end of 19th century when, thanks to the inventions of Nikola Tesla and support of George Westinghouse, Niagara Falls Power Company managed to develop the first AC (Alternating Current) power grid that included a hydro power plant of 75MW and network of about 50 kilometers long [7]. Development of the powers system as known nowadays has began in the 1950s, when the primary objective was to keep the lights on using centralized power plants that feed power over an electro-mechanical grid [67]. During the years, as a product of rapid urbanization and developments, driven by economical, political and geographical factors the electrical power system drastically grew. Nowadays, worldwide production is estimated to be nearly 17 billion KWh per year [15]. Still, conceptually, not much has changed since the first power grids being developed by the beginning of 20th century and the basic structure of powers system grid has remained unchanged. Next paragraph gives a short overview of that structure.

General structure of the powers system consists of: production, transmission, distribution and finally, consumption. Since of that, an interconnected power system may be subdivided into the following major subsystems: Generation subsystem, Transmission and Subtransmission subsystem, Distribution subsystem and Utilization subsystem. Generations subsystems essential element is the three-phase AC generator known as synchronous generator or alternator whose prime movers may be hydraulic turbine, steam turbines whose energy comes from the burning of coal, gas and nuclear fuel, gas turbines, occasionally internal, combustion engines burning oil. Typical power plant contains several of such generators and produces hundreds of MWs. An overhead transmission network role is to transfer electric power from generating units to the distribution system which ultimately supplies the load, typically operating at standard voltages in range from 69 kV to 765 kV. Large industrial users may be served directly from the subtransmission system with typical voltage level ranges of 69 kV to 138 kV. The distribution system distributes energy from the subtransmission station to the distribution substations and further to the consumers service-entrance equipment. It is consisted of primary and secondary distribution lines operating on 4 kV to 34,5 kV and 240/120V respectively [17], [27] . An overview of traditional power system structure is presented on figure 2.1, [67].

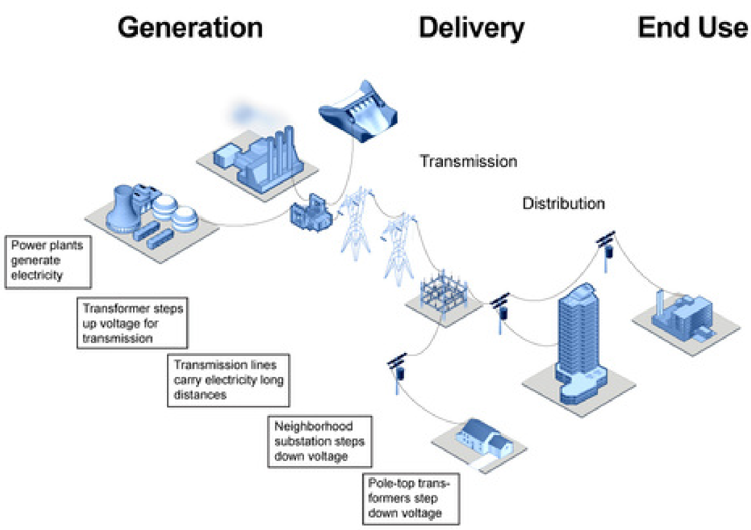


Figure 2.1: Structure of traditional power system

Based on this structure and previous discussion, one can easily conclude that existing power system is unidirectional system, meaning that energy flows only in one way - from production to consumption. Moreover, the existing electrical power systems are real-time systems, that do not store energy, but produce it when loads call for it. This is what is called on-demand production or load-driven production [17]. Such a system requires a highly reliable, stabile, high capacity and highly controllable generators. Still, the systems lacks in advanced communication, supervisory and control infrastructure that could provide its better efficiency, reliability and controllability. The core of existing power systems control that is widely deployed is known as SCADA (Supervisory Control And Data Acquisition). Such systems give utility companies limited control over their upstream functions, while the distribution network remains outside their real-time control. In order to enhance SCADA functionality, advanced software packages to enhance power system efficiency has been developed (EMS/DMS software). Still, worldwide, a low percentage of power systems is equipped with advanced monitoring and control structure that could provide better efficiency and better customer service [30].

To summaries, existing power system is characterize with: traditional structure, centralized production with one-way power flow, reliable generators of huge capacity, load-driven production, lack of advanced ICT (Information and Communication Technologies) structure, one-way communication, centralized limited control and monitoring, a few customers choice and low efficiency in energy distribution.

2.2 Limitations of the existing power system

As previously mentioned, existing power system relies on the structure developed in the middle of 20th century. During the years, due to the economical and customer needs it has been in constant grow and under improvement that includes new ICT solutions providing higher efficiency, reliability, security and stability of the system. Still, it has been estimated, that less than a quarter of world's most advanced electrical power systems are equipped with information and communication structure, while distribution system automation is only around 15% [30]. Mostly due to the insufficient monitoring and communication infrastructure but also the structure of the system itself, existing power system suffers great losses (general estimation of 2% on transmission level and around 8% on distribution level)(See figure 2.2),[11]).

Moreover, existing power system needs to grow faster in order to follow constant energy consumption and population grow (figure 2.3(a), figure 2.3(b),[11]).

On top of this, there are still many parts of the world that do not have any power grid. It has been estimated that there are 1.4 billion people in the world with no access to the electricity and another 1.2 billion people who have inadequate

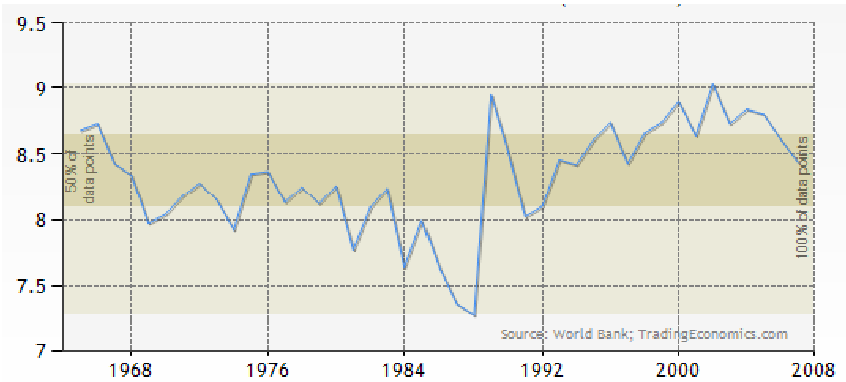
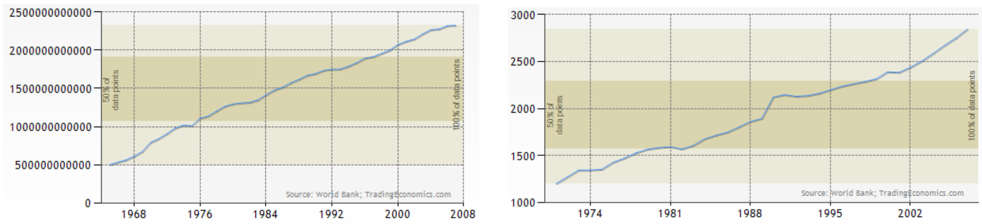


Figure 2.2: Electric power transmission and distribution losses (% of output) in world



(a) Electricity production (kWh) (b) Electricity consumption (kWh per capita)

Figure 2.3: World's electrical energy production/consumption growth

access to electricity (meaning that they experience outages of four hours or longer per day), [15].

Besides all the problems mentioned above, that can be summarized as how to providing a reliable and stabile service (as known today) to a constantly growing an demanding market, with an insufficient ICT structure, existing power system is not able to provide any new services to the market. An overview of the new power system services that are seen to be developed is given in next section.

2.3 Future trends in power systems

Constant energy market growth, economical and technological evolution and finally awareness to global climate changes urges for evolution of the existing power systems in both commercial and technical terms. This section gives an overview of power system future trends, main drivers, challenges and opportunities, while taking into account technical and commercial aspects of future grid as well as global and specific challenges and opportunities. The general challenges and opportunities concern everybody, while the specific ones concern only single stakeholder groups, for instance electric grid operators, established and new electricity producers, producers of components and systems for VPS (Virtual Power System), VPP (Virtual Power Plant), SmartGrids, and Smart Metering, potential suppliers of new services, electricity customers or public authorities [57]. Since the purpose of this section is to give an global overview of future trends rather than going into specifics problems discussion, it will mostly focus on global challenges and opportunities while specific ones will be briefly mentioned.

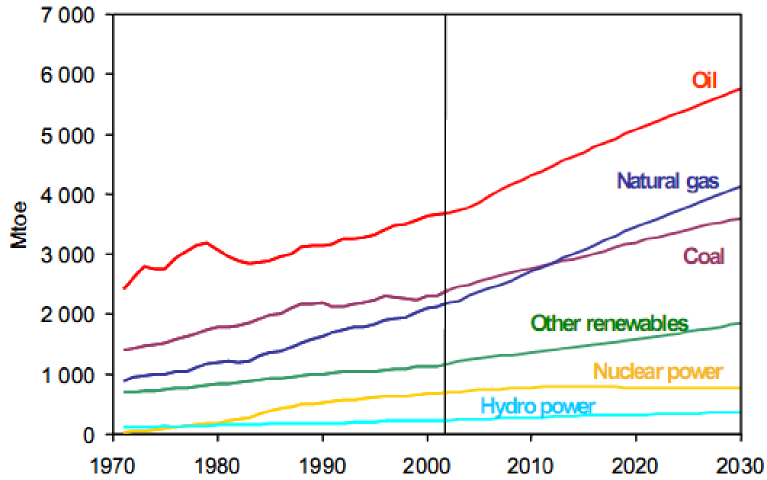


Figure 2.4: World's energy demand forecast till 2030

2.3.1 Distributed Energy Resources (DERs)

According to the International Energy Agency's latest energy projections till 2030 (See figure 2.4), the global CO2 emissions will rise up to 70% more when compared with toady's one. With fossil fuels continuing to dominate the energy mix global energy-related (See figure 2.5), CO2 emissions will rise by 1.8% per year reaching 38 billion tonnes per year in 2030, [14], [42].

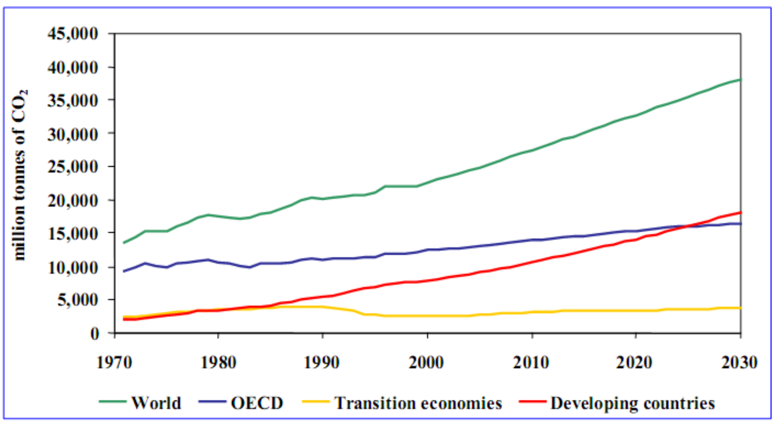


Figure 2.5: Forecast of energy related CO2 emissions by region till 2030

The need to limit climate change implies that global green house gas emissions must be quickly and strongly reduced. This mostly includes reduction of CO2-emissions from fossil electric power plants by introducing a higher use of either more diluted uranium sources or renewable energies resources. Due to high fuel efficiency, short construction lead time, modular installation and lower distribution losses since production is getting closer to consumption, renewable energy resources seams to be more promising future solution [57].

Descriptive definition of renewable resources describes them as: the small scale generating technology, such as microhydro, photovoltaic, windmill, wind turbine and microturbine, able to produce renewable energy (Electrical energy that comes from inexhaustible resources such as sunlight, wind, rain, tides and geothermal heat, which are naturally replenished. Besides term renewable generator, they are also refereed as distributed generators (DGs). The DG technology is often lumped with distributed storage and their combination is referred to as distributed energy resource (DER) [28]. Still, quite often term DER is used to describe DG. An overview on DERs production capabilities, that are mostly used today, is given in *table* 2.1, [60].

Technology	**Typical capability ranges**
Solar / PV	A few to several hundred kW
Windmill	A few to several hundred kW
Wind turbine	A few hundred W to a few MW
Geothermal	A few hundred kW to a few MW
Internal combustion engine	A few tens of kW to a few tens of MW
Fuel cells	A few tens of kW to a few tens of MW

Table 2.1: Overview of the characteristic of selected decentralized energy technologies.

Still, besides direct benefit of decreasing pollution and distribution losses, introduction of DERs comes with a several drawbacks that lies in the nature od DERs themselves. The power grid must operate so that there is a real-time balance of generation and load. Without this balance, the grid frequency will drift up or down the nominal value (50Hz in Europe). Only 30MHz frequency drop can reduce power delivery by 1GW, while large drops of 2Hz could seriously harm the system. Existing power system is mostly load-driven, meaning that it operates mostly by controlling generation to mach load at any particular time. But, load-driven strategy becomes more difficult with DERs introduction due to their low reliability, predictability and schedulability. One solution for this problem is introduction of more conventual generation sources to be used in peak hours, while other lies in development of dispatchable loads. Those loads can be controlled and scheduled from a distance depending on the production capabilities. That way, instead of shaping production on consumption needs, energy peaks could be shaped by scheduling consumption. Loads that potentiality could be controlled remotely fall into two categories:, [15],

[18].

- loads that, when remotely controlled, could result in some inconvenience or discomfort for the customer (e.g., lights, air-contitioning)
- loads that, when remotely controlled, would go largely unnoticed by the customer and so are good candidates for demand dispatch (dishwasher, washers and dryers, water heaters, HVAC system, battery chargers, plug-in vehicles and data centers with cloud computing).

Among all controllable loads, plug-in vehicles could provide the most benefit to a future power grid. Electric-drive vehicles, whether powered by batteries, fuel cells, or gasoline hybrids, have within them the energy source and power electronics capable of producing the 50 Hz AC electricity. One typical electric-drive vehicle can put out over 10kW, with the average draw of 10 houses. With adding connections that could allow energy flow from vehicle to the grid, not only that charging of the vehicle can be scheduled and remotely controlled, but vehicle itself could serve as an power source. [51], [35]

In summary, a quick large-scale implementation of renewable energy technologies increasing energy efficiency and saving measures is necessary to respond to the global climate change, the strongly increasing global energy demand, and the related price increase of conventional fossil energy carriers, [57].

2.3.2 Novel ICT structure for future power system

As presented above, distributed energy resources and a demand dispatch with controllable loads could be a solution for the current power system limitations and problems. Still, in order to successfully monitor, control and communicate with all the new devices, a robust ICT structure is necessary, [45]. That novel ICT structure of the power system should include [65]:

- Distributed modular device network for wide-area monitoring and control
- Demand area secure communications network
- Ad-hoc and plug-and-play sensor network for maintenance

Distributed embedded devices that could serve for monitoring, local data processing and control of field devices (both generators and loads) are seen to be essential for a future power system ICT structure. Those devices include AMIs (Advanced Measuring Infrastructure), whose main purpose is sensing and measuring and IEDs (Intelligent Electronic Device) whose main purpose is control and actuation, [45]. AMI is much more that simple automated meter reading (AMR). It is the deployment of a metering solution with two-way communications to the electric meter to enable key features including: [36], [20], [61]

- Two way communication to the electric meter to enable:
    - time stamping of meter data outage reporting
    - communication into the customer premise
    - service connect/disconnect
    - on-request reads
    - alarms and faults reporting, etc.
- Ability of the AMI network to self register meter points
- Ability of the AMI network to reconfigure due to a failure in communications
- AMI system interconnection to:
    - utility billing
    - outage management systems, ETC.

Beside that, AMI systems should provide: conviviality, integrity, reliability, availability, data security and accountability. Following components are seen to be in future AMI systems, [21]:

- Smart Meter - the source of metrological and energy-related measurable information information form consumer loads and DERs
- Customer Gateway - an interface between the AMI network and customer systems and appliances within the customer facilities (Home Area Network (HAN) or Building Management System (BMS))
- AMI Communications Network - provides a path for information to flow from the meter to the AMI
- AMI Headend - manages the information exchanges between external systems (Meter Data Management (MDM), VPS control center, etc.) and the AMI network

On the other side direct controlling, or better to say imposing of control parameters to electro-mechanical devices in the power system is performed by actuators. This is performed trough IEDs that should also enable, [45]:

- Remote connect (remote administration)
- Demand Side Management
- Configurability and programmability
- Local history log data and elementary statistics

- Diagnostics and facilitated maintenance
- Interpretability
- Security from cyber attacks

Communication infrastructure of future grid should provide reliable, (near)real-time and secure communication among control center and all field devices. It can be realized with RF, Wi-Fi, cellular, Ethernet and/or low bit rate PLC. Ongoing standards IEC 61850, IEC 61400-25 and IEC 61950 give a full description of information model and information exchange for electric power system, [64].

2.3.3 New electrical energy market trends

Besides technical issues and trends in power system, future trends in power system include electrical energy market trends as well. Market transformation refers to the process of increasing incentives or reducing market barriers to support the adoption of cost-effective, energy-efficient and clean energy products in a sustainable manner. It simultaneously stimulate the development of new products and promote their market introduction by effectively changing consumer-purchasing practices so that targeted products become commonplace, [31]. New energy market trends include, [38], [44]:

- Expansion and modification of available tariffs to include: sales to grid, ancillary services and choice of back-up service.
- Access to markets (real time pricing signals, distribution only tariffs)
- Deregulation of energy market to introduce new producers
- Introduction of customer as active participant in energy market

2.4 SmartGrid

In order to fulfil all the trends, challenges and opportunities that are mentioned in the previous section and to overcame all drawbacks of the existing power system, a new solution, mostly for the power system ICT structure is necessary. That solution has to successful tackle the problem of DERs introduction into the power system, to handle their unreliability; to successfully accept new types of consumers/prosumers and fulfil their consumption needs; to introduce and implement new market offers for the customers. On top of that, new solution for the power grid has to keep and moreover to improve overall system's stability, reliability and security. A concept of a *Smart Grid* (also known as "future grid", "intelligent grid", "modern grid", etc.) aims at proposing solutions for all that has been previously mentioned. Still, one should keep in mind that Smart Grid concept is not revolution of power grid, but rather evolution that first of all aims at improving overall system efficiency. Since Smart Grid is (and will keep to be for some time) still a concept that is under development, there is no clear definition and agreement on what the Smart Grid actually is. Mostly, this term is used to refer to the ICT structure of the new power grid ("The smartness of the Smart Grid lies in the decision intelligence layer, all the computer programs that run in relays, IEDs, substation automation systems, control centers, and enterprize back offices", [63]). Purpose of this section is to give a brief overview on Smart Grid concepts by combining different points of view and solution proposals coming from various relevant projects, organizations and articles.

The European Technology Platform for the electricity networks of the future (ETP, [2]), defines Smart Grids as 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, economical and securable electricity supplies. A smart grid employs innovative products and services together with intelligent monitoring, control, communication, and self-healing technologies in order to: [2], [43]

- Better facilitate the connection and operation of generators of all sizes and technologies;
- Allow consumers to play a part in optimizing the operation of the system;
- Provide consumers with greater information and options for choice of supply;
- Significantly reduce the environmental impact of the whole electricity supply system;
- Maintain or even improve the existing high levels of system reliability, quality and security of supply;
- Maintain and improve the existing services efficiently;
- Foster market integration towards European integrated market.

According to the same source, the "smartness" of the SmartGrid is in making better use of technologies and solutions to better plan and run existing electricity grids, to intelligently control generation and to enable new energy services and energy efficiency improvements. While clearly making a difference between smart metering and Smart Grid, a summary of Smart Grid elements is given on figure 2.6, [2].

Another important view of Smart Grid concept that has to be mentioned here is presented by The U.S. National Energy Technology Laboratory (NETL) [3] that is owned and operated by the U.S. Department of Energy (DOE) [4]. The latest joint report of NETL/DOE on Smart Grid benefits, gives seven principal characteristics of the Smart Grid [71]:

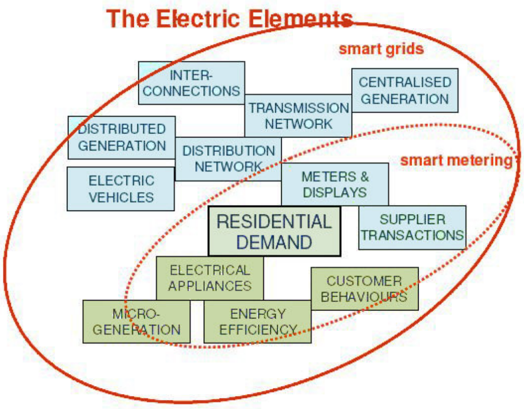


Figure 2.6: Smart Grid elements as seen by ETP

- It enables active participation by consumers - Consumer choices and increased interaction with the grid bring tangible benefits to both the grid and the environment, while reducing the cost of delivered electricity.

- Accommodates all generation and storage options - Diverse resources with ”plug-and-play” connections multiply the options for electrical generation and storage, including new opportunities for more efficient, cleaner power production.

- Enables new products, services, and markets - The grid’s open-access market reveals waste and inefficiency and helps drive them out of the system while offering new consumer choices such as green power products and a new generation of electric vehicles. Reduced transmission congestion also leads to more efficient electricity markets.

- Provides power quality for the digital economy - Digital-grade power quality for those who need it avoids production and productivity losses, especially in digital-device environments.

- Optimizes asset utilization and operates efficiently - Desired functionality at minimum cost guides operations and allows fuller utilization of assets. More targeted and efficient grid maintenance programs result in fewer equipment failures and safer operations.

- Anticipates and responds to system disturbances (self-heals) - The smart grid will perform continuous self-assessments to detect, analyze, respond to, and as needed, restore grid components or network sections.

- Operates resiliently against attack and natural disaster - The grid deters or withstands physical or cyber attack and improves public safety.

Moreover, in the same report Smart Grid benefits are seen as improvements trough six key value areas:

- Reliability - by reducing the cost of interruptions and power quality disturbances and reducing the probability and consequences of widespread blackouts
- Economics - by keeping downward prices on electricity prices, reducing the amount paid by consumers as compared to the "business as usual" (BAU) grid, creating new jobs, and stimulating the U.S. gross domestic product (GDP).
- Efficiency - by reducing the cost to produce, deliver, and consume electricity
- Environmental - by reducing emissions when compared to BAU by enabling a larger penetration of renewables and improving efficiency of generation, delivery, and consumption
- Security - by reducing the probability and consequences of manmade attacks and natural disasters
- Safety - by reducing injuries and loss of life from grid-related events

Finally, a list of technology solutions for smart grid is given while each of it is related to the value area in witch it is expected to create improvements, *table* 2.2.

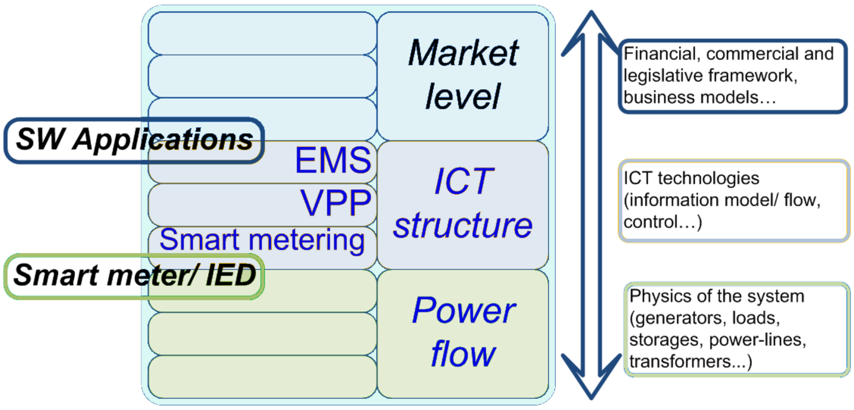


Figure 2.7: Smart Grid functionalities as a layered structure

Combining previously mentioned solution proposals and future trends given in previous section, one can concluded that future Smart Grid is a complex system that need to be hieratically decomposed in order to get a better overview of it's services and functionalities (see figure 2.7), [44]. From that approach, an representation of a Smart Grid as a layered structure rises. Proposed structure should not be considered just as on overview of Smart Grid services, but also as a starting point for further Smart Grid modeling, simulation an development.

Finally, figure 2.8 is given as a quick overview of new techonologies and capabilities of power system on it's evolution toward the Smart Grid [30].

Technology solution	**Improved value areas**
Advanced Metering Infrastructure (AMI)	Reliability, Economics, Efficiency, Environmental, Security, Safety
Customer Side Systems (CS)	Economics, Efficiency, Environmental
Demand Response (DR)	Reliability, Economics, Efficiency, Environmental, Security
Distribution Management System/Distribution Automation (DMS)	Reliability, Efficiency, Environmental, Security, Safety
Transmission Enhancement Applications (TA)	Reliability, Efficiency, Environmental, Security
Asset/System Optimization (AO)	Efficiency, Environmental, Security, Safety
Distributed Energy Resources (DER)	Reliability, Economics, Efficiency, Environmental, Security
Information and Communications Integration (ICT)	Reliability, Economics, Efficiency, Environmental, Security, Safety

Table 2.2: Smart Grid technology solutions and their impact on grid's value areas as seen by NETL/DOE

2.5 Virtual Power Plant and Virtual Power System

Smart Grid concept, as presented in previous section, appears to be highly complex system, or better jet system of systems, that includes power grid, power generation, complex ICT structure, etc. One of the greatest challenges of Smart Grid is efficient integration of DER's in both technical and commercial sense. Using a paradigm of encapsulation, as well known in other engineering disciplines, rises a concepts of Virtual Power Plant (VPP) and Virtual Power System (VPS).

A Virtual Power Plant (VPP) aggregates capacity of various DER's, encapsulates them and presents, to the rest of the system, as a single controllable and reliable entity that is similar to the transmission-connected generator. As already mentioned in this chapter, individual DER's, due to the nature of it's energy sources, are usually highly unpredictable and uncontrollable. Moreover when operating alone, DERs mostly do not have sufficient capacity and flexibility to participate in market-based activities, while with the creation of a VPP from a group of DER, these issue can

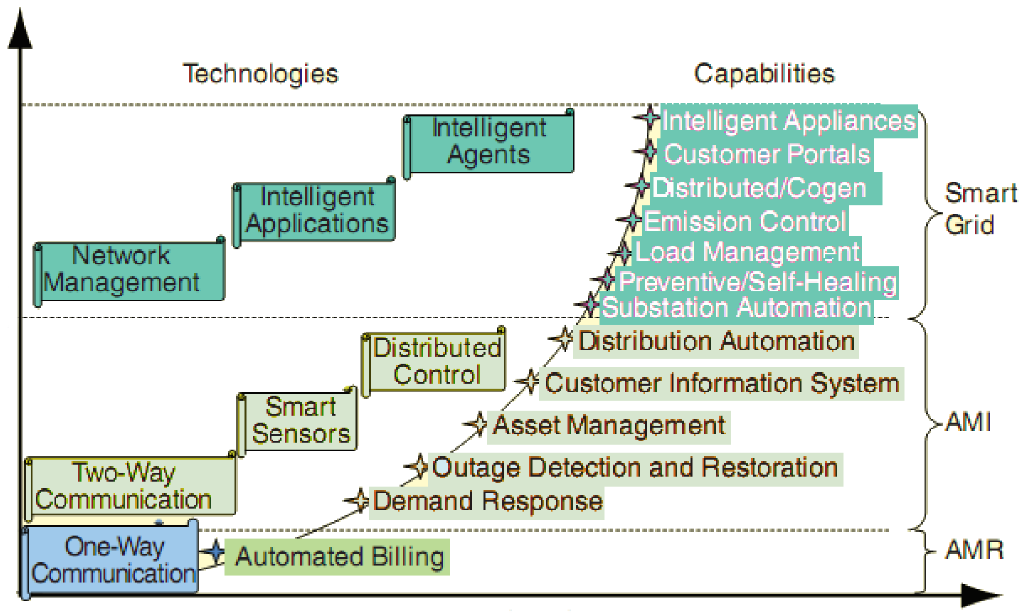


Figure 2.8: Power system evolution

be overcome [59], [46], [25].In summary, trough the VPP concept,

- Individual DER will be able to gain access and visibility across all the energy markets, and will benefit from VPP market intelligence in order to optimize their position and maximize revenue opportunities;
- System operation will benefit from optimal use of all available capacity and increased efficiency of operation. [59], [55]

In order be better distinguish, describe, model and design VPP it can be considered trough two separate entities encapsulating two main groups of VPP's functionalities that are Commercial Virtual Power Plant (CVPP) and Technical Virtual Power Plant (TVPP). Basic functionality of CVPP is energy selling, optimization and scheduling of production based on prediction needs of consumers and price negotiation. As a part of SCADA/DMS control system, TVPP has to guaranty power quality and provide production feasibility depending on grid topology. A full list of TVPP and CVPP functionalities, as seen by Fenix project, [55] are given in *table* 2.3.

Virtual Power System (VPS) is a more general concept that has been introduced in scope of AlpEnergy project, [1]. It relies on VPP while introducing active consumer into DER's integration system. The concept and project of Virtual Power System (VPS) has been developed in order to promote Transnational Cooperation and sustainable energy supply in the Alpine space, [1].

A short and simple definition of VPS as seen by AlpEnergy is: A Virtual Power System integrates, manages and controls distributed energy generators and storage ca-

CVPP functionalities	**TVPP functionalities**
• Maintenance and submission of static physical characteristics of DERs. • Administration of planned and non-planned outages on a long, medium and a short-term basis. • Forecasting of generation based on weather data. • Forecasting of demand based on demand profiles. • Outages Demand Management. • Building DER bids. • Submission of bids to the market and balancing offers to the TSO. • Day-to-day optimization and re-balancing of real-time generation.	• Feasibility check of the base schedules within the distribution network. • In the event of problem detection, recommend remedial actions to provide a feasible combination of schedules and network configuration. • Feasibility, calculation & evaluation of scheduled generation values of the DER units. • In the event of problems detected in the distribution network caused due to basic schedules, planning of corrective measures. • Validation of day-ahead schedules. • Same day verification of the DSO schedule. • Switching procedures for safe reconfiguration of the network management. • Volt VAr Control (VVC).

Table 2.3: Virtual Power Plant functionalities

pacities and links their technical operation to the demand of consumers and the energy market, [57].

Main motivation for development of a new concept such as VPS lies in following, [1], [57]:

- to effectively manage the assessment of electricity consumption data (metering) to prevent electricity theft, fraud and non-payment of electricity bills;

- to ensure the power balance between the generation and the load profile even in presence of embedded dispersed generation, including fluctuating renewable energy sources and mobilizing the potential electricity savings;
- to reduce energy losses in the electric grid;
- to enhance the electric grid performance with regard to power quality and coordination of relaying/ protections in presence of dispersed generation.

VPS is more general concept of VPP, even that the second one is more know in industry and scientistic community. VPS gathers and manages all DER's from one area (that can be in ranges of a small settlement to entire county) and summes them up in a single production profile. Moreover, VPS combines a number of consumption units, that can be considered as controllable loads. That way, the whole system is internally controlled and presented to the rest of the power system (usually connected to the distribution network) as one single entity that is, depending on current internal production / consumption ratio considered to be producer or consumer. VPP presents a boundary case of VPS when it contains only production units (DER's). When VPS contains only consumption units it is referred as VLP (Virtual Load Plant), that is another boundary case opposite to VPP. Moreover, VPS can include various types of energy storage elements (batteries, electrical vehicles, heat, biomass, water in barrages, etc.). Figure 2.9 gives an summary on VPS definition and it's differences with other similar concept like VPP and VLP, [1].

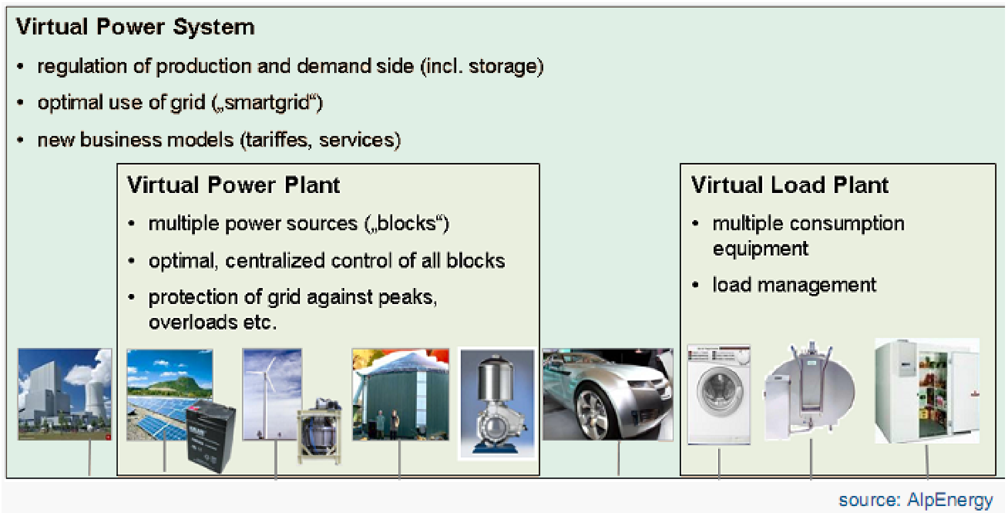


Figure 2.9: Virtual Power System definition

On top of all that has been mentioned in a previous paragraph, VPS has to include a robust ICT structure that includes, [45], [46]:

- **Embedded devices (AMI, IED)**, attached to all the VPS elements (DERs, loads and storages) that are able to measure/impose all the necessary parameters (production, consumption, weather parameters, device status parameters, power quality parameters, etc.)

- **Control center(s)**, centralized and local depending on the system size, able to process all the data from the field device, apply advanced optimization algorithms to achieve the best system efficiency, and control field devices.

- **Communication infrastructure**, able to provide secured, and reliably communication between control center(s) and embedded device but also local communication between embedded devices and field devices.

- **Interface** to the rest of the power system.

As a system that gathers all the elements mentioned above, and that can cover smaller or larger areas and even include traditional generators (operation on distribution network) in extreme cases, VPS includes most of the important characteristics of Smart Grid itself. Still, VPS is not a Smart Grid that is much larger and more complex system of national or international size including also industrial users and large power plants. It is rather a cluster of Smart Grid, that can even operate independently till some extend. As such it is ideal candidate for a case study of a power system of the future. Figure 2.10 gives a quick overview on VPS elements and it's relation to the rest of the system as seen in the future.

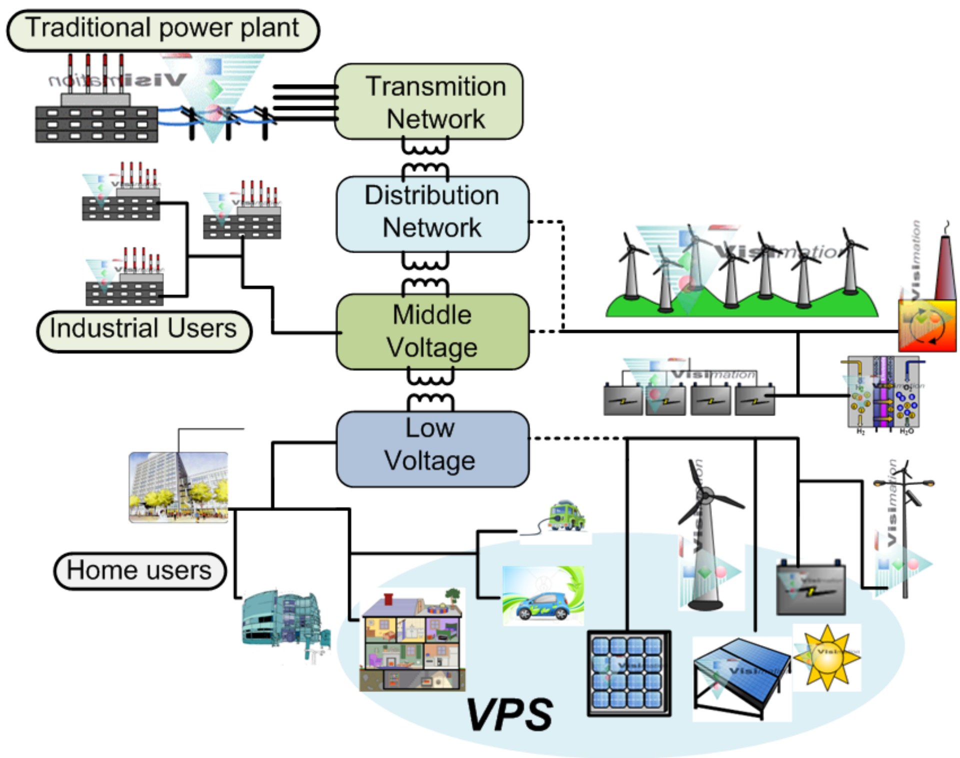


Figure 2.10: Virtual Power System as a part of Smart Grid

2.6 Model-driven system engineering methodology

Complex systems or better to say "systems of systems", like Smart Grid, VPP and VPS are, always involve stakeholders coming from different areas of expertise and having different views to a system as a whole. Still, most of the tools that are proposed to be used for modeling and simulation of Smart Grid, regard it only from the power flow aspect while forgetting the fact that future power system is more about it's ICT structure and that a large part of it's complexity is related to commercial and technical characteristics. In scope of this thesis VPS is considered from a more general point of view, as a system of a systems, while emphasizing it's ICT structure and most of all role of embedded systems in future power grid. For that reason analysis of modeling tools and methodologies that are considering VPS only from a view of a power system engineer is not of the interest for thesis. Since of that, the purpose of this section is not to give an overview on existing power system modeling and simulation tools (for more details regarding existing tools reader is referred to [40], [37]), but rather to introduce a general concept of complex system modeling and design through a proven concept of Model-Driven System Design, MDSD (also Model-Driven Methodology, MDM; Model Driven System Development; Model-Based System Engineering, etc.).

"Model-driven systems development is the progressive, iterative refinement of a set of models to drive development of the system", [50]

Before going in-deep about MDSD itself, a few basic terms and concepts, that will be in the rest of this text, has to be clarified, [50].

- A **system** is a set of resources that is organized to provide services. The services enable the system to fulfill its role in collaboration with other systems to meet some useful purpose. Systems can consist of combinations of hardware, software, other objects and flows, workers, and data.
- At a high level, a **service** is a mechanism by which the requirements of the customer are satisfied.
- A **requirement** is a condition or capability to which the system must conform.
- A **model** is defined as a collection of all the artifacts that describe the system.
- An **artifact** is defined as any item that describes the system, including a diagram, matrix, text document, or the like.

Logical and physical complexity of systems that include hardware, software, other objects and flows (for example energy flow or water flow), user interaction, etc. is constantly rising. System complexity is increasing in response to the capability of languages, technology, and global information flow. Moreover, those subsystems that system is consisted of are complex by themselves till such extends that requires evolvement of many experts from different fields. There is a common understanding that building complex systems requires careful planning, good architectural design and well-controlled development, testing and deployment processes. Using traditional methods and engineering techniques, developed for one particular engineering discipline, is mostly inadequate and usually fails in successful system development and leads to a fatal business mistakes. Model-driven development is one approach to raising the level of abstraction that has been successfully exploited for more than a decade mostly in software engineering. It's basic premise is to use models as nonlinear forms. For that reason, models are usually rendered using visual notations, such as diagrams, instead of pure text, [49], [50]. Model-driven system design is a concept that is already highly accepted in areas of like aerospace and avionics industry and defence; areas that require serious planing and evolvement of many stakeholders. [69], [44]

Basic elements of model-driven system design are, [16]:

- **Modeling language** - The visual and semantic standard used.
- **Modeling tool** - The application used to create and manage the model.
- **Modeling process** - Defines the set of work-flows, artifacts, and other elements to be followed when modeling.

An overview on MDSD structure is given on figure 2.11, [16].

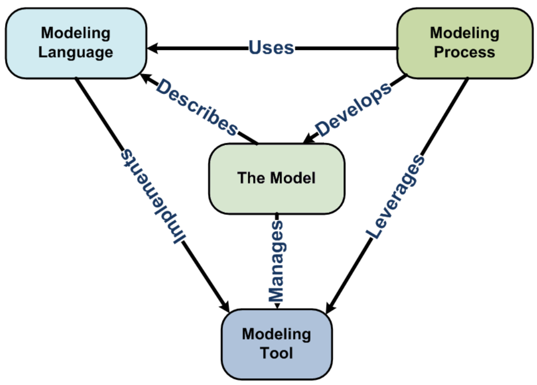


Figure 2.11: Elements of Model-driven system design environment

Main problems that MDSD tackles are, [50]:

- Managing system complexity by managing levels of abstraction and levels of detail
- Considering multiple views to address multiple concerns
- Integrating of form and function of the system while making sure, from the early phase of design, that system has to meet requirements

- Defining system as isomorphic composite recursive structures and defining methods to unsure scalability

As a conclusion, MDSD is an iterative, modular, scalable system development method, that deals with system complexity through abstraction. It combines various engineering technologies, and from the very begins of system design unsures: reduction of risk, enhanced team communication, explicit processes for reasoning about system issues and performing trade studies, early detection of errors, better architecture, traceability, project documentation.

What is considered to be a major enabler of model-driven system development is System Modeling Language (SysML). Based on Unified Modeling Language (UML), SysML has been developed as a general purpose modeling language for systems engineering applications that supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems.

More details regarding MDSD for the VPS design as well as SysML particularities description are given in the chapter 4.

2.7 An overview on relevant projects

Purpose of this section is to give a brief overview and description on projects that are tackling subjects of SmartGrid, VPP, VPS and Smart metering. A list of project that follows is not exhaustive. Projects to be presented here are chosen by the criteria of project's relevance to this thesis, ALaRI involvement to the project and project's overall relevance.

AlpEnergy, [1] is a European Territorial Cooperation Project bringing together power suppliers, development agencies, research institutes and public administrations from five different countries of the Alpine Space - France, Germany, Italy, Slovenia and Switzerland - to address the central issue of renewable energy supply. It explores concept of Virtual Power Systems (VPS) focusing on both technical as well as economical aspects to introduce an efficient operational model that aims at a standardization of both technologies and procedures. It intends to provide new knowledge-based incomes and business opportunities to farmers, traditional and innovative enterprises, thus supporting the competitiveness of the Alpine ventures and making the Alpine space a showcase for other mountain areas in the world. It has been recognized as an Official Partner of the Sustainable Energy Europe Campaign, [19] and contributes in the area of Demonstration and dissemination projects.

FENIX (Flexible Electricity Networks to Integrate the eXpected energy evolution) project, [55] is an European collaborative project that includes 20 partners (among which Siemens, Avera, Ecro, etc.) , partly funded by the European Commission within the 6th Framework Programme for Research lunched in 2005 and concluded in 2009. Main results of project includes: Development of VPP concept that fits the European power system; Development of ICT architecture that is scalable and hierarchically flexible; Development of a commercial and regulatory framework that allows the beneficial integration of the VPP concept in the future EU power system; Cost-Benefit-Analysis that quantify the economic benefits of the VPP concept; Demonstration of two field projects in real networks in Spain and UK.

ADDRESS, [52] is a large-scale Integrated Project co-founded by the European Commission under the 7th Framework Programme, in the Energy area for the "Development of Interactive Distribution Energy Networks". It involves 25 partners and has started in June of 2008 and will last for 4 years. ADDRESS stands for Active Distribution network with full integration of Demand and distributed energy resources and its target is to enable the Active Demand in the context of the smart grids of the future, or in other words, the active participation of small and commercial consumers in power system markets and provision of services to the different power system participants. ADDRESS is framed in the Smart Grids European Technology Platform, whose vision for the electricity networks of the future may be expressed in just 4 words: flexibility, accessibility, reliability, economy.

INTEGRAL, [56] project is a 3 years European project under the 6th Framework programme, that involves 9 partners (from The Netherlands, France, Greece, Spain and Sweden) and has started at November of 2007. Project goal is to develop and demonstrate an industry-quality reference ICT-platform for Distributed Control in Electricity Grids at aggregated levels in the electricity grid, based on commonly available ICT components, standards, and platforms.

E-Energy: ICT-based energy system of the future, [54] is a new major funding project initiated by the Federal Ministry of Economics and Technology (BMWi) in line with the German Federal Government's technology policy. The primary goal of E-Energy is to create E-Energy model regions that demonstrate how the immense potential for optimization presented by information and communication technologies (ICT) can best be harnessed to enhance the efficiency and environmental compatibility of the power supply and to ensure supply security.

DISPOVER, [53] is a project of 38 partners coming from different expertise areas (utilities, power industry, service companies, research centers and universities) and 11 European countries, coordinated by ISET e.V., Kassel/Germany. The project goal is to investigate field of integrating small and distributed generators into the electricity distribution grid. The central question was as follows: how does the technology has to be developed so that the growing number of decentralized energy resources and it's further integration into the European electricity grids in the future, without losing reliability, safety and quality.

Chapter 3

Problem Description

"A problem well stated is a problem half solved"
Charles Kettering

As presented in a previous chapter, driven by a rising environmental concerns and requests for energy market liberalization, existing power system has to evolve toward more reliable, "green", secured, decentralized, advanced, etc. or simply to present more complex system. A future vision of power system is described in the concept of Smart Grid. Virtual Power System, as it has already been presented, includes most of the main characteristics of Smart Grid itself. Moreover, VPS can be considered as a cluster of Smart Grid that has a certain amount of autonomy. As such, it ia ideal candidate for Smart Grid concept implementation.

Developing a complex system, of "system of systems" like VPS is requires at least fours steps: system specification; modeling; analysis (simulation) and verification; and deployment. Still, most of a tools meant for Smart Grid, and so VPS modeling and simulation are strictly power system oriented and rarely take into account other system aspects. In order to successfully specify, model and simulate a VPS as a complex system, in a way that will be understandable for various stakeholders and experts that VPS involves, a methodology that uses a higher level of abstraction has to be applied.

As already presented in section 2.5 of this thesis, VPS is still a concept that is still not fully specified. The first step for a full specification includes identification of all the possible stakeholders and shareholders and identification of their roles particular views on the VPS as a system. For example, local authorities are mostly interested in legal aspects of the system, it's geographical deployment and commercial functionalities, etc.; distribution companies are interested on system connection interface, controllability, power and voltage quality and reliability, etc.; software engineer wants to know what to know what are data inputs and outputs of the system and what algorithms should be programmed; etc. A clear **identification of VPS's stakeholders** and development of the model that clearly identifies their roles has to be the first task of this thesis.

The exact system specification has to start with stakeholders requirements collection and analysis. From a collected set of requirements, system requirements has to be extracted. The second task of this thesis is to **define and apply a methodology for requirements engineering**.

Using extracted system requirements, that can be considered as informal system description, more formal VPS specification has to be given. This specification includes VPS structure and behavior modeling while baring in mind multidisciplinary nature of the system. Model developed is this way, and able to provide different views, will serv as a mutual understanding planform for various stakeholders. Moreover, since of the possibility that VPS specification will be improved and changed, and possibility to use model parts for SmartGrid specification, developed model has to be scalable.**Development of a high level system model** is the next step of this thesis.

Importance of embedded devices in a future power system and so in VPS as well has already been identified (See sections 2.3.2, 2.4 and 2.5). Moreover, since this thesis and related research are being done at ALaRI as embedded systems institute, the role of embedded devices in VPS is of particular interest. For both reasons, one more task of this thesis is to clearly identify role and **functional requirements for embedded devices** and present them in a form of model describing devices interface and structure.

Developing a high level VPS model and specifying functional requirements for embedded devices concludes the original problem set for this thesis. Still, in order to provide a more complete solution, two more tasks have been added. This includes: providing of a framework for **algorithms development and optimization** (this mostly refer to the algorithms of prduction/consumtion scheduling) and exploring possibilities for **system level validation**.

A methodology that conceptually includes most of the problems presented above is a **model-driven methodology** (already presented in section 2.6). Still, model-driven methodology itself does not provide solution on those problem, nor a straight-forward mechanism for finding a solution. For that reason, model-driven system development methodology has to be properly **adopted and extended** for the purpose of VPS modeling. Moreover, exploring and suggesting a proper **languages and tools** for implementation of adopted model-driven methodology has to be provided.

A general description of task to be fulfilled is: *Extending, adopting and applying a model-driven methodology using a chosen set of languages and tools in order to specify, model, simulate, optimize and verify Virtual Power System concept.* Particular set of tasks includes:

- Full stakeholders identification
- User requirements gathering and analyzing
- System requirements extraction and analyzing
- Development of high level abstract system model (structural and behavioral)
- Development of mutual understanding platform for stakeholders
- Identification of functional requirements for embedded devices

Moreover, this thesis aims at providing a framework and proposing solution for:

- Algorithms optimization and verification
- System level simulation and verification

Due to a complexity of VPS as a concept, development of a VPS model that fully describes VPS behavior and structure from all possible views and till each detail requires a whole team of researches and engineers. For that reason, contribution of this thesis is seen to be more in demonstrating benefits of a concrete methodology (MDM) while focusing on importance of early stage model development, rather than proposing a fine-grained verified model than can be directly used for system development.

Major result of this thesis should be in proposing a **model-driven methodology for VPS design**, development of **full high level system model for VPS** with **demonstration** of usage of developed **framework for algorithms optimization** and methods for **embedded systems functional requirements identification** and structure proposition.

Chapter 4

Methodology

"No problem can be solved from the same level of consciousness that created it"
Albert Einstein

This chapter gives an description of model-driven methodology extensions and adaptation for the purpose of VPS. Moreover, a short overview on System modeling Language (SyML) and it's benefits as well as detailed description of particulars diagrams to be used are presented. Finally an overview on IBM Rational Rhapsody and OMNeT++ as tools chosen for solution implementation is given.

4.1 Adopting and extending model-driven system engineering methodology for VPS design

A general concept of model-driven system engineering methodology (also refereed as model-driven methodology, model-based design, etc.) has been already briefly introduced in section 2.6. As already noted, model-driven methodology is more concept than a set of rules to be applied, and can be adopted for the purpose of particular needs.

Adopting and applying a model-driven system engineering methodology for VPS modeling purpose is one of the original contributions of this thesis. When choosing a concert steps, languages and tools for the methodology and it's application author constantly had in mind characteristics of VPS as they are presented in section 2.5 as well as set of problems as described in previous chapter. Proposed methodology and models presented in next chapter are mostly focusing on the ICT structure of VPS, but due to methodology and model scalability future extensions and modifications are possible.

A full methodology flow is presented on figure 4.1. As a presentation mean, a SysML activity diagrams are used. For more details regarding activity diagrams and other SysML diagrams used in this section for the purpose of presentation or description, reader is referred to section 4.2.

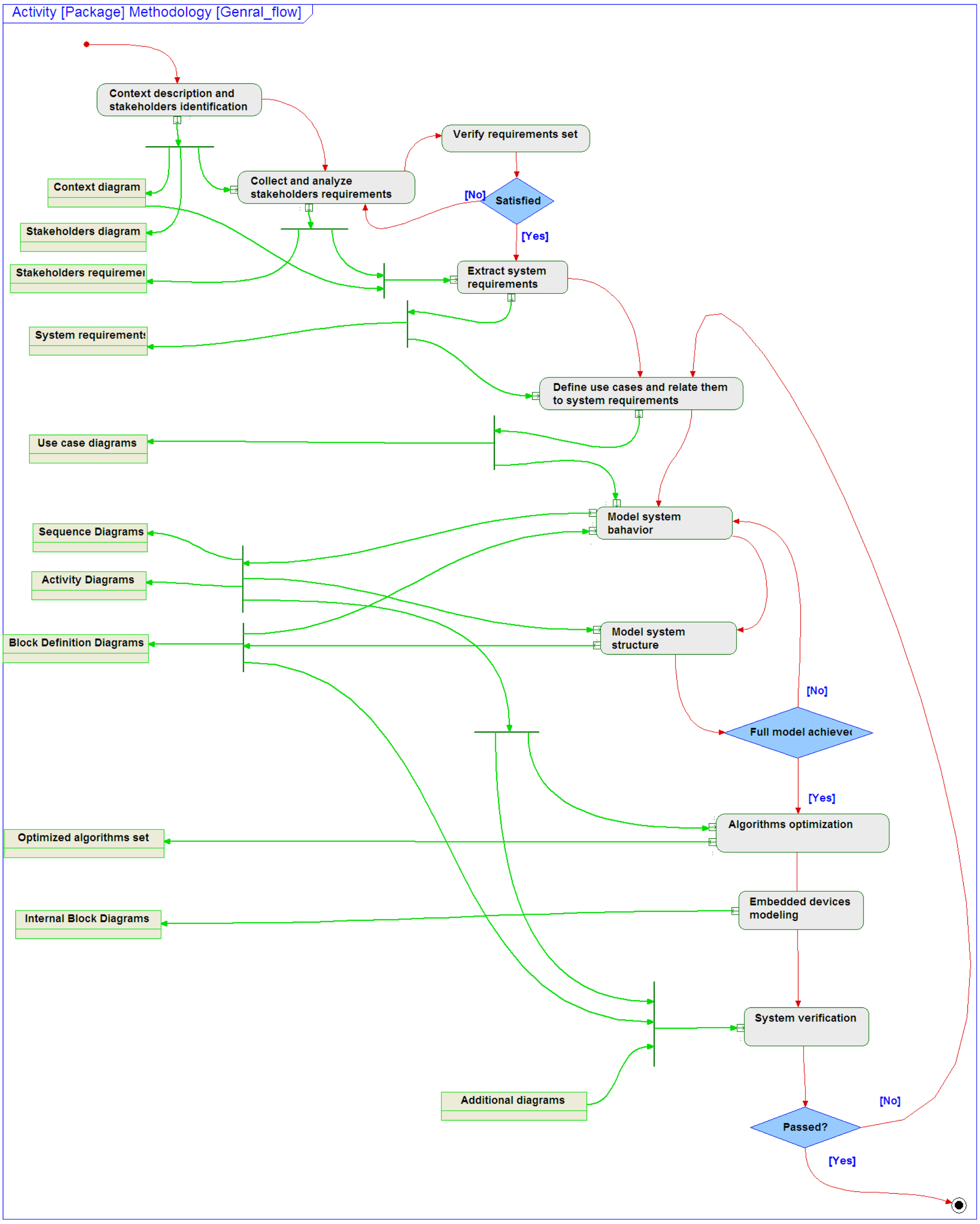


Figure 4.1: Adopted model-driven methodology flow for VPS modeling

In order to keep diagram more consistent and easier to follow, shown activities are mostly not atomic ones. A detailed description of each activity that includes description of all sub-activities as well is given in next paragraphs.

Context description and stakeholders identification action consist of two sub-actions that are, to some extend, done in parallel. Context description includes identification of external elements and systems that are not a part of the system under consideration and systems positioning in terms of relation to external elements. As a mean of presentation use-case diagrams are suggested for stakeholders identification while but block definition diagrams can be used to identify system context. Still other UML/SysML diagrams can be used as well. Main propose of this diagram is to get a high level overview of system and help stakeholders identification. Stakeholders identification includes targeting and listing all stakeholders (both, functionalities requestors and providers) involved in systems functionalities. Besides additional system context description stakeholders diagrams as a result of this activity provide a basis for a next methodology activity.

Next step is to collect and analyze stakeholders requirements (also referred as users requirements). Collecting of requirements also includes requirements representation in a textual form and it's grouping based on stakeholder or type of requirements. A user defined requirement's stereotapes or packages can be used for this purpose as shown in a Solution chapter. Requirements analysis includes analysis for ambiguities and conflicts. This methodology does not provide the exact mechanism for this purpose, but applies methods presented in [26] and [62]. Users requirements can letter be presented in a tabular or diagram form and used for black-box system verification.

Analyzing of requirements can oslo include requirements reorganization and adaptation in order to avoid conflicts. Since of that, requirements set should be verified with stakeholders before proceeding with a next step of methodology.

User requirements set can not be direly used for system modeling and simulation since they are defined by a stakeholder who are not necessarily familiar with system structure and development. For that reason a system requirements has to be extracted from users requirements. For system requirements extraction, this methodology uses solution similar to the one presented in [39]. Each user requirement is considered as realized trough interaction of system under construction (in particular case VPS) and external elements as presented in context diagram. For interaction representation SysML sequence are suggested, but other UML interaction diagrams can be used as well. Each interaction activity initiated by a system under consideration is considered as a new system requirements. After considering all user requirements system requirements has to be organized (using packages or stereotypes) and analyzed for ambiguities and conflicts. Final system requirements set can be presented in table or diagram.

Basic system functionalities are presented with a few use case diagrams that can later serve for system functionality overview. Those, general use case diagrams, are decomposed to atomic use-cases that are related to previously different system requirements. Each use-case can be related (refine) one or more system requirements. Still, it is possible that one system requirement is refined by more than one use-case. Use cases diagrams as system functionalities that are later used for system behavioral and structural modeling.

Behavioral and structural system modeling is an iterative process in which a straight line between modeling of behavior and structure can not be defined. If system modeling starts with requirements gathering, that system behavioral should be modeled first. On the other hand it is not possible to model behavioral without knowing the structure of the system since interactions and system activities presented in behavioral models are performed by elements of structure. For that reason, despite the fact that deferment authors are suggesting that structural modeling should proceed behavioral modeling (or vice versa), this methodology suggest that behavioral and structural modeling should be done in parallel trough a few iterative processes.

Once when all use-cases are described trough behavioral diagrams, modeling process can be concluded. After this step, system can be considered as fully modeled. This model can be used for further fine-grained modeling, verification and development but also serves as a mutual understanding platform for stakeholders. After this, model can be further defined and optimized from a strictly software point of view and

Next step proposed by methodology is algorithms optimization as a part of further model development from the software (strictly functional but not structural) point of view. IBM Rational Rhapsody offers solution of using a connection with other tools for this purpose but does not provide any powerful language for algorithms optimization by itself (action language included in Rational Rhapsody that is a subset of C++, by the authors opinion, can not be used for any fine-grained algorithm definition). For tat reason using OMNeT++ for optimization framework development is suggested but still not mandatory (for OMNeT++ see section subsection 2.2 of this chapter).

Defining a functional requirements for embedded devices and structural modeling trough internal block diagrams can be done in parallel with a previous step. Still, this is not explicitly shown on the diagram in order to keep it more simple. With this step a fine-grained structural modeling is introduced. Besides showing internal structure, embedded devices model should also show all the communication and flows in the system.

Finally, system verification can be done. For a high level system verification it is not necessary to fulfil two previous steps, but some additional diagrams has to be developed. Since system verification will not be presented in scope of this thesis, development of additional diagrams is not done. In short, system verification can be done trough system simulation/animation. For that purpose state-chart diagrams have to be developed on various system levels. Moreover, system constrains has to be modeled and applied trough parametric diagrams while activity can be modeled using, already developed, activity diagrams and action language.

4.2 System Modeling Language (SysML)

System modeling Language (SysML) is a general-purpose graphical modeling language developed by Object Management Group (OMG) that supports the analysis, specification, design, verification, and validation of complex systems that may include: hardware, software, data, personnel, procedures, facilities and other elements of man-made and natural systems. SysML provides a means to capture the system modeling information as part of MBSE (Model-based system engineering) approach without imposing a specific method on how this is performed. The language is intended to help specify and architect systems and specify its components that can then be exported with XML and designed using other domain-specific languages such as UML for software design and VHDL for hardware design. SysML can represent systems, components, and other entities trough different levels of hierarchy and abstraction including operation level model, system level model and components level model. It provides semantics and notation that is fully independent on methodology or tools to be used. Still, SysML is considered to be a major enabler for model-driven system engineering.

SysML has been developed as an extension of UML (Unified Modeling Language used as a standardized modeling language in the field of software engineering, [12]). It reuses a modified subset of UML 2.0 and provides additional extensions needed to address requirements in the UML for Systems Engineering. A relationship between UML 2.0 and SysML 1.2 is summarized on a diagram shown on figure 4.2.

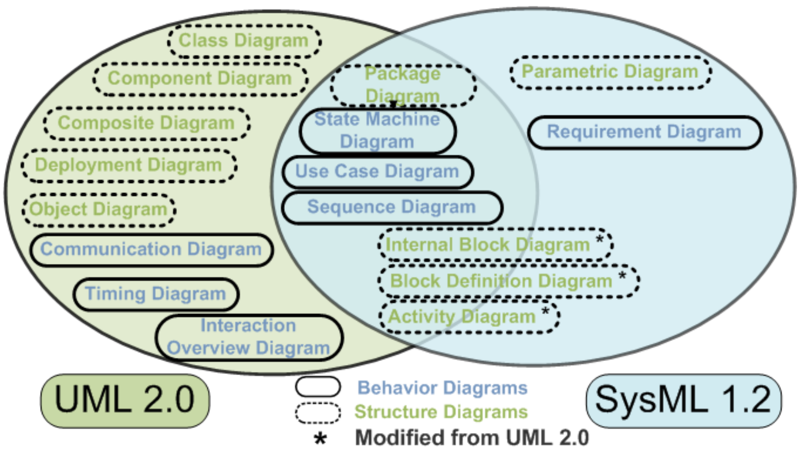


Figure 4.2: Overview of SysML/UML interrelationship

Since not all the SysML diagrams and parts will be used for the purpose of this thesis, only a subset of SysML diagrams and elements will be described in a following paragraph. For a full description of SysML and it's usage reader is referred to [13], [33], [32] and [69]. For more details regarding SysML diagrams and elements presented in this section, same set of references applies. For a quick overview on SysML diagrams, reader is referred to [70].

4.2.1 Requirements modeling

A requirement specifies a capability or condition that must (or should) be satisfied. A requirement may specify a function that a system must perform or a performance condition a system must achieve. Requirements describe a contract between the principal and all those who create the system design and implement the system. SysML provides modeling constructs to represent text-based requirements and relate them to other modeling elements. A requirement is a stereotyped class with two main properties: unique requirement ID and textual description of requirement. Stereotype "requirement" is mostly used for functional requirements that can be related to use-cases. For presenting non-functional and other type requirements user can derive own stereotype. Besides tabular representation, requirements can be presented trough diagrams. Besides requirements presented as lock, requirements diagrams usually include representations of relations between requirements as well as relations between requirements and other modeling elements. Possible relations include, [69]:

Derive relationship using stereotype "deriveReqt" describes that a requirement was derived from another requirement.

Namespace containment describes that a requirement is contained in another requirement.

Satisfy relationship using stereotype "satisfy" describes that a design element satisfies a requirement.

Copy relationship using stereotype "copy" describes that a requirement is a copy of another requirement.

Verify relationship using stereotype "copy" connects a test case with the requirement that is verified by that test case.

Test case relationship is a flow that checks whether or not the system satisfies a requirement.

Refine relationship using stereotype "refine" specifies that a model element describes the properties of a requirement in more detail.

Trace relationship A trace relationship is used as a general relation for traceability reasons only.

4.2.2 Structure modeling elements

SysML diagrams that are used to model system structure and static relationships include: package diagrams, block definition diagrams, internal block diagrams and parametric diagrams. Since parametric diagrams are not used in scope of this thesis, the will not be presented in this section. Other diagrams used for structural modeling as well as included elements are briefly discussed below.

Package Diagram (PD) can be used quite flexibly to organize the model in packages and views. As such, a package diagram can include a wide array of packageable elements. Using packages, a model can be organized in a multiple ways (by system hierarchy, diagram kind, views). Packages provide a mechanism for model organization, while the way of organizing is left to a user. Package diagram can be used to provide a personalized stakeholders perspective views to a model.

Block Definition Diagram (BDD) is used to define blocks in terms of their features, and their structural relationships with other blocks. Block can be used to represent: system, hardware, software, date, procedure, facility, person, etc. Relation with other blocks include: inheritance/generalization, association, composition, aggregation. Block properties can be described trough multiple standard block compartments that include: properties (part, reference, value, port), operations, constraints, allocations from/to other model elements (e.g. activities), requirements the block satisfies. Moreover, additional compartments can be defined by user. Block, just like class in UML, defines a general property and can be used in different context. When used in the context of a composing block, block is referred as part or role.

Internal Block Diagram (IBD) describes the internal structure of a block in terms of its properties and connections. Elements of IBD are parts, as instances of abstract blocks. Parts connection are modeled with ports, connectors and flows. There are two types of ports: standard ports and flow ports that besides differen application have a different notation. Standard ports are modeling provided/required interface and are mostly used for informaton/data flow. A more general flow (as energy flow, water flow, etc.) can be modeled with flow ports. Connection between two port is modeled with connectors while for modeling particular item that flows trough connector, flow item element is used.

4.2.3 Behavior modeling elements

Behavioral modeling of the system is realized using use case diagrams, sequence diagrams, activity diagram and state-chart diagrams. Diagram set used in scope of this thesis as well as included elements are presented below.

Use Case Diagram (UC) describes the usage of a system (subject) by its actors (environment) to achieve a goal, that is realized by the subject providing a set of services to selected actors. They are the mean to provide description of basic functionality and/or capabilities that are accomplished through the interaction between the subject and its actors. Use case diagrams include the use case and actors and the associated communications between them. Realization of functionality that is described with an use case is latter represented trough activity diagrams, sequence diagrams, and state machine diagrams. Relationship between use-cases can be modeled trough include, extend, generalize and communication relationship. The "include" relationship provides a mechanism for factoring out common functionality that is shared among multiple use cases, and is always performed as part of the base use case. The "extend" relationship provides optional functionality, which extends the base use case at defined extension points under specified conditions. The "generalization" relationship provides a mechanism to specify variants of the base use case. A concrete usage of use-cases is methodology dependent. In scope of this thesis use-cases are related to one or more system requirements and are used to describe system functionalities.

Sequence Diagrams (SD) are used to describe interactions between system entities. As mostly used type of interaction diagrams in UML, they are accepted in SysML without any changes. SDs describe the flow of control between actors and systems (blocks) or between parts of a system. This diagram represents the sending and receiving of messages between the interacting entities called lifelines, where time is represented along the vertical axis. The sequence diagrams can represent highly complex interactions with special constructs to represent various types of control logic, reference interactions on other sequence diagrams, and decomposition of lifelines into their constituent parts. In scope of this thesis, sequence diagrams are used to represent use-cases realization.

Activity Diagrams (Act) are used to specify transformation of inputs to outputs through a controlled sequence of actions while giving a fine-grain description of system functionalities previously described with use-cases. Activity diagram shows what will be done in which sequence, which data all of this requires, and which data will be output. For that reason, they are also referred as control flow and object flow diagrams. Activity diagram can contain swim-lines related to system blocks to show how particular activities and flows are delegated to system elements. Flow modeling is realized trough activity blocks, it's ports and connections and special nodes, that includes: initial node, activity final node, flow final node, fork node, join node, decision node and merge node. SysML activity diagrams are different that UML activity diagrams. The most important extension, from the point of view of this thesis is support for continuous flow modeling. Activity diagrams can be used to represent an activity to a different level of details. In particular case, the are used to represent a more fine-grained use-case realization.

4.3 An overview on tools to be used

Model-driven system development methodology and SysML as a language chosen for realization, as already stated, are fully tool independent. Choosing of tool for a solution implementation is based on tools comparing on various parameters and is not, in any case, suggested by methodology or language. Solution that will be proposed in next chapter can be realized different set of tools. This section aims at providing a brief overview on chosen tool with respect on capabilities and options that are relevant for this thesis. Since not all the details are, nor should be presented, additional references are provided as well.

4.3.1 IBM Rational Rhapsody

IBM® Rational® Rhapsody® is a visual development environment for systems engineers and software developers creating real-time or embedded systems and software. Rational Rhapsody helps diverse teams collaborate to understand and elaborate requirements, abstract complexity visually using industry standard languages (UML, SysML, AUTOSAR, DoDAF, MODAF, UPDM), validate functionality early in development, and automate delivery of innovative, high quality products, [23].

The main reason for choosing IBM Rational Rhapsody for the purpose VPS design is it's wide range of possibilities that includes possibility of different languages usage as well as as possibility of model exchange and further development using a wide range of tools from a same family. It represents a reliable solution with a good customer support and improvements available on both, Windows and Linux platform. Besides that it is highly available to students and researches trough IBM academic programs. Still, usage of Rational Rhapsody tool is not obligatory for proposed solution. Moreover, with XML export as a property of SysML, there is a possibility of further model usage with any other tool.

Since this thesis deals with VPS design on system level a IBM Rational Rhapsody Designer for System Engineers edition is used. The Rational Rhapsody Designer for System Engineers solution provides system engineers a collaborative development environment with simulation for early requirements, architecture and behavioral validation, helping communication of complex requirements and trade off analysis of complex systems. For more details regarding installation and additional tools required for a particular edition reader is referred to [22] and [41]. A chosen set of IBM Rational Rhapsody Designer for System Engineers edition that are considered relevant for the work presented in this document as well as for a future development is given below:

- Integrated requirements and modeling environment using industry standard SysML or UML diagrams
- Full life-cycle traceability and analysis from requirements to design
- Customizable automatic documentation capabilities
- Static model checking analysis that helps improve design consistency
- Automate model based testing with Rational Rhapsody Test Conductor Add On
- Dynamically analyze and execute SysML parametric diagrams (starting with Rational Rhapsody 7.5.2) to assist in trade study analysis

For more detail regarding IBM Rational Rhapsody and it's usage for Model-driven system design reader is referred to following references: [16], [50], [22], [23], [23], [8].

4.3.2 OMNeT++

OMNeT++ is a discrete event simulation environment, with primary application area in the simulation of communication networks. Still, due to it's flexibility, generality, extensibility, modularity and component-based property it is successfully used in other areas like the simulation of complex IT systems, queueing networks or hardware architectures as well. Architecture is based on C++ simulation library and framework, with an Eclipse-based IDE and a graphical runtime environment. OMNeT++ provides a component architecture for models, that are programmed in C++, then assembled into larger components and models using a high-level language (NED). It has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into user's applications.

OMNeT++ is not a network simulator itself, but rather network simulation platform. Mostly used for wireless sensor networks simulation, it comes with a wide range of add-ons and user defined model library that extend it's functionalities. Using a NED language as an extension to C++ for components development and fully visualized environment for network creation saves user's time. Among others, OMNeT++ supports following add-ons and extensions:

- support for simulation of communication networks, queuing networks, performance evaluation, etc.

- extensions for real-time simulation
- extensions for network emulation
- support for alternative programming languages (Java, C#)
- support for database integration
- possibility of SystemC integration, etc.

OMNeT++ is available on various platforms (Linux, Mac OS X, other Unix-like systems and on Windows) and free for academic and non-profit use. Since of that, and other previously mentioned benefits, it become a widely accepted in the scientific community worldwide.

In scope of this thesis, OMNeT++ is used for development simulation framework aimed at VPS algorithm optimization. In particular, developed framework can be used for scheduling algorithms exportation and optimization, whit a possibility of components reuse in other models. Since OMNeT++ is based C++, algorithm developments can be done in language that is already highly acceptable in academics and industrial community. Even that OMNeT++ does not come with add-on for Smart Grid simulation, components can de defined and edited fast and easy (by authors opinion, easier that in any other simulation environment). Moreover, Eclipse IDE provides good environment for network development and results analyzing.

For more details regarding OMNeT++ reader is referred to [9].

Chapter 5

Solution

"Make everything as simple as possible, but
not any simpler"

Albert Einstein

This chapter proposes the main part of the solutions for the problem set described in chapter 3. It includes all the steps of the model-driven methodology presented in chapter 4 (note that methodology it self presents a part of solution and original contribution of this thesis) applied to a VPS. As a final result a full structural and behavioral system model is given. Moreover, presentation of a developed framework for scheduling algorithms testing and optimization is presented. Due to a complexity of VPS, identification of functional requirements for all the embedded devices and optimization of all algorithms can not be achieved in scope of this thesis. For that reason, application of the methodology and framework are presented for a specific device and algorithm in next chapter as case studies. Moreover, one should keep in mind that VPS is a concept and that a full set of requirements is not completely defined. Since this thesis, is to some extend related to AlpEnergy project, [1] some of the stakeholders requirements are gathered with help of project partners, while the others are obtained trough various publications. Still, author strongly believes that even requirements that are not coming directly from interviews with stakeholder do reflect their real needs since coming from a deep and exhaustive explorations of VPS and Smart Grid.

5.1 VPS context description and stakeholders identification

Identification of all VPS stakeholders and context description represents the first step of project design that reflects to all steps that follows. The subsequent design heavily depends on this step and that is why it implicitly requires a deep understanding among stakeholders.

The major goal of VPS is to aggregate heterogeneous, uncontrollable renewable resources (windmills, photovoltaics etc.) and controllable loads into single reliable entity in commercial and technical terms. In order to achieve such an aim additional energy storages and controllable generators (traditional distribution level generators, e.g. gas turbines) have to be introduced in the system. VPS has to interact with the rest of power systems as well. Since, as it has been already described (section 2.5), VPS operates at distribution level, it has to interact with this part of power grid as well as with Distribution System Operators (DSOs). Interaction with environment is necessary in order to get current filed measurements (temperature, pleasure, humidity) and predictions that are used for production/consumption scheduling. Moreover, in order to provide optimal economical performance, interaction with Energy Exchange Stock is mandatory. VPS, as a system, has to interact with VPS operator that encapsulates all human factor in the system. Finally, in order to establish communication between VPS and all elements mentioned above, communication infrastructure is necessary. Figure 5.1 elaborates Virtual Power System context in form of SysML block definition diagram.

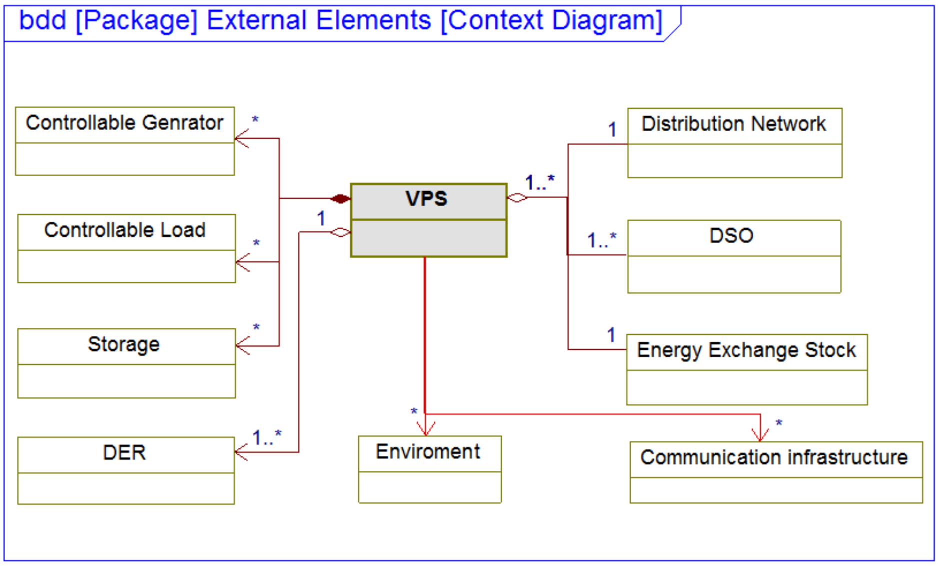


Figure 5.1: VPS context diagram

VPS stakeholders identification starts with analyzing system environment presented trough system context. Still, VPS is system under development, other stakeholders could appear later. This could affect some major changes in the system modeling that follows. Stakeholder identified so far are presented on following use-case diagram, 5.2.

5.2 Stakeholders requirements engineering

VPS stakeholder requirements engineering starts with collecting of requirements from all the stakeholders identified (see figure 5.2). Still, due to the novelty

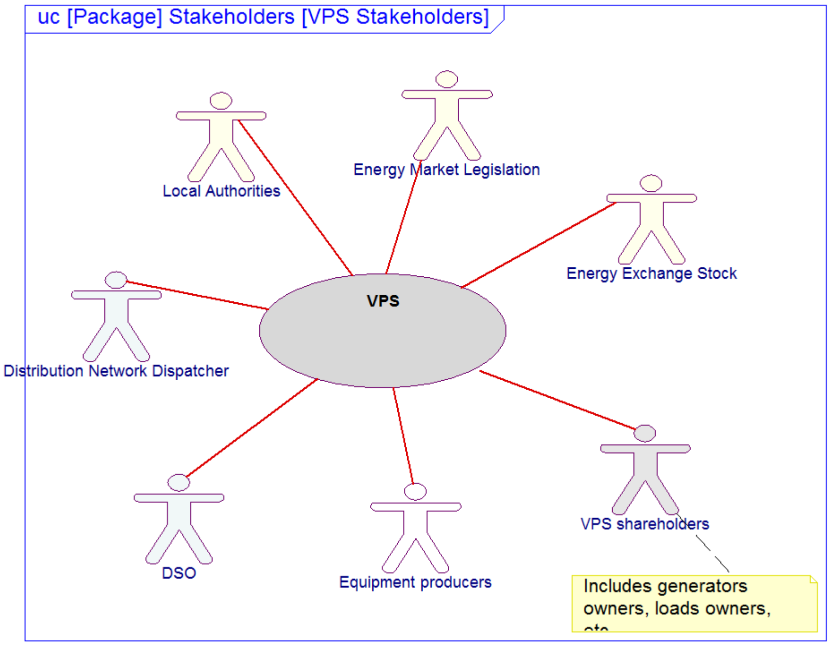


Figure 5.2: VPS context diagram

of VPS concept it is not possible to get in contact with all stakeholders and collect their requirements. For instance, market legislations for a future power system are still undefined; VPS legal regulation; standardization of communication is still on-going; etc. For that reason, not all stakeholders requirements are included here, while some of the requirements serve more for demonstration than practical purpose and will not be included in further model development. Depending on the group of stakeholder they are coming from, requirements are grouped in three packages: communication requirements, market requirements, technical requirements and legislation requirements. In order to defer between functional and non-functional requirements, requirement stereotypes are defined. Part of stakeholders requirements is presented in table 5.3 together with a short specification of each requirement. A full set of users requirements (so far identified) is presented in a more compact way on diagram 5.4.

ID	Name	Specification
M03	Price prediction	Based on production/consumption predictions, VPS has to be able to provide energy selling price
M04	Risk Management	VPs has to include local risk management
M01	Price negotiation	VPS has to be able to negotiates price for energy selling/baying
E02	Enviromental costs	VPS has to provide pollution information to the rest of the grid in form of environmental costs
C01	Reliability	VPS has to provide reliable communication interface for communication with Distribution Network and DSO
C02	Interoperability	VPS ICT structure must provide a full interoperability among field devices manufactured by different
C03	Plug-and-Play	All devices must be fully self-identifiable and self- describable
C04	Device standardization	All devices must fulfill communication standards requirements
C05	Security	VPS has to provide secured communication interface for communication with Distribution Network and
C06	Communication Protocol	Standardized communication protocols between VPS and rest of the power system have to be fulfilled.
L01	Data Privacy Protection	Privacy of generators/loads owner has to be guaranteed.
L02	Energy Market Operation Legislation	Market operation has to fulfill rules that will be defined for a future power grid.
L03	Local Legislation	VPS has to follow all local authorities legislation (like land usage rights, pollution, etc.)
L04	State Legislation	VPS has to follow all state legislations that will be defined for Smart Grid
L05	Low pollution	VPS has to operate while keeping CO2 (and other tocsins) emission in defined

Figure 5.3: Stakeholders requirements table (partial set)

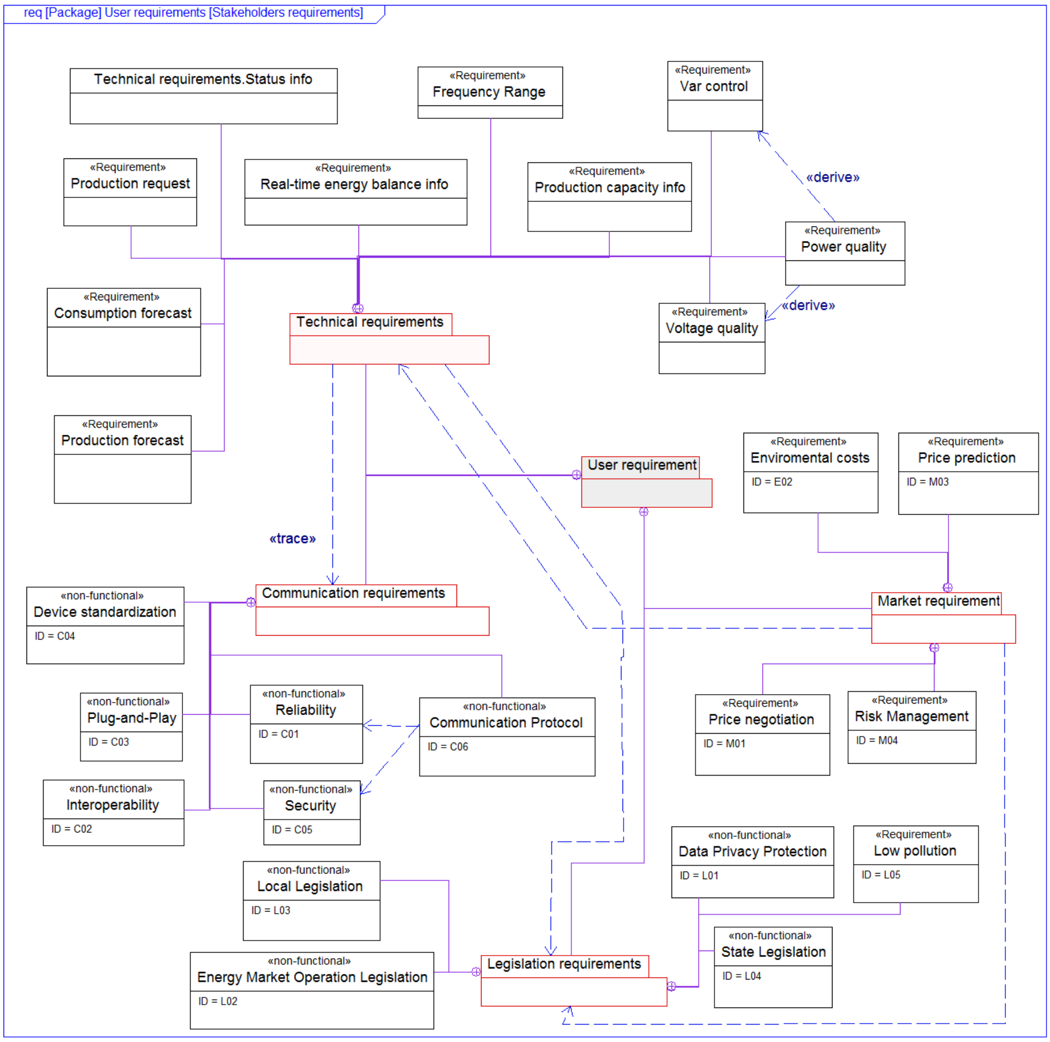


Figure 5.4: Stakeholders requirements diagram

5.3 System requirements extraction

As previously explained (see section 4.2) after stakeholders requirements have been collected and organized, system requirements have to be extracted trough set of sequence diagrams used to describe each of the requirements as interaction between VPS and other external elements presented in VPS context diagram (figure 5.1).

A full example on requirements extraction will be presented trough "Production request" technical requirement. Details on other system requirements extraction will not be presented in detail, while full list of system requirements and it's relations are given on figure 5.6.

Extraction of system requirements from user requirements "Production request" is presented on diagram 5.5. This requirements comes from DSO representing request for a specific amount of electrical energy trough some specific time. It is in interest of VPS to provide that energy in the most optimal way. For that reason following actions have to be done by VPS:

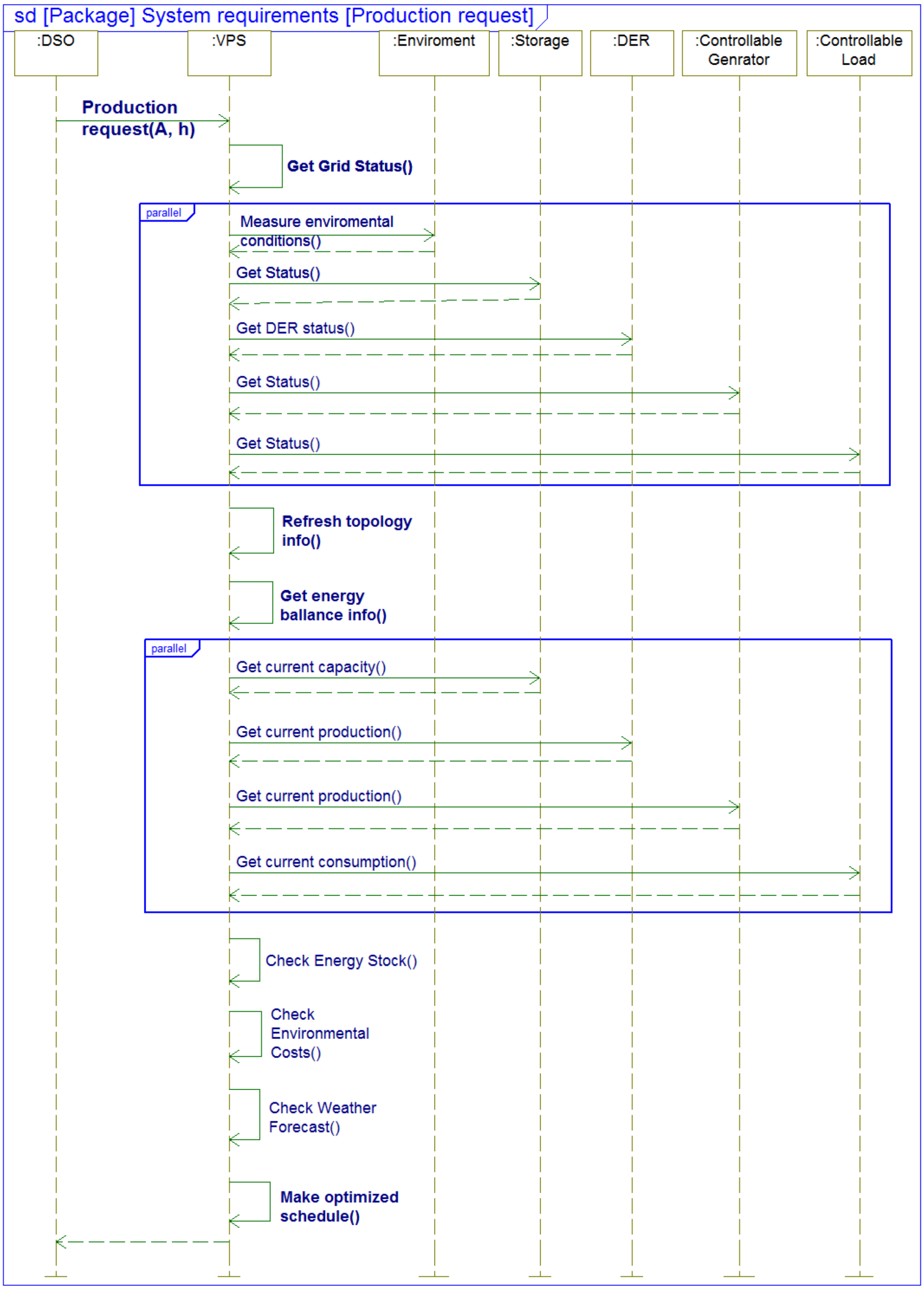


Figure 5.5: Extraction of system requirements (example)

- Checking status of local grid in terms of network status, equipment status, confirmation of elements plugged-in, maximum capacity of generators and current fuel status (if any), etc.
- Refreshing on grid topology based on data gathered in a previous step.
- Getting energy balance info in terms of current production/consumption, gen-

erators and storage elements status (Note that Energy balance info is user requirement as well).

- Checking of energy stock condition, in terms of electrical energy prices, fuel prices and so on. For this purpose VPS should interact with Energy Stock Market, but in order to keep this diagram simple as possible.

- Checking environmental cost includes additional costs and taxes for cost of using a specific fuels (like diesel, gas, etc.). For this purpose VPS should interact with Energy Stock Market and Local authorities, but for the reasons already explained diagram lacks for those element.

- Checking of weather forecast is necessary to predict production of DERs

- Based on data from all the previous steps VPS can make optimized schedule for required production. In case that required amount of energy can not be fulfilled, another negotiation will start.

- Finally new settings has to be imposed to devices.

Note that this stakeholder's requirement should be related with market requirement "price negotiation". Still, this relation is not presented in previous sequence in order to keep it as simple as possible for the demonstration purpose. A full list of system requirement is presented on diagram below (figure 5.6). One should note that some stakeholders requirements are directly transferred to system requirements without further decomposition since they can be satisfied with only one interaction that includes system under consideration. Moreover, since provided model is mostly focusing on technical aspect, ICT structure and VPP embedded devices, non-functional stakeholder's requirements will not be considered.

As shown on diagram below (figure 5.6) system requirements are divided into three packages: efficient commercial operation, efficient technical operation and reliable communication. As already explained, communication standardization for Smart Grid (that implicitly includes VPS) is ongoing process and full set of requirements is still not known. Moreover, market aspects of future power grid are still subject of academic, industrial and political argumentations and negotiation and can not be considered as know. For that reason, system requirements presented in those two packages should be considered more as descriptive ones. Besides using packaged to organize system requirements, fine-grained technical requirements grouping is done trough user defined requirement stereotypes. One should note that requirement for low pollution (that mostly referrers to low CO_2 emission) has been placed into "efficient commercial operation" package even though original (as user requirement) being in legislation package. Main reason for is this is the fact that

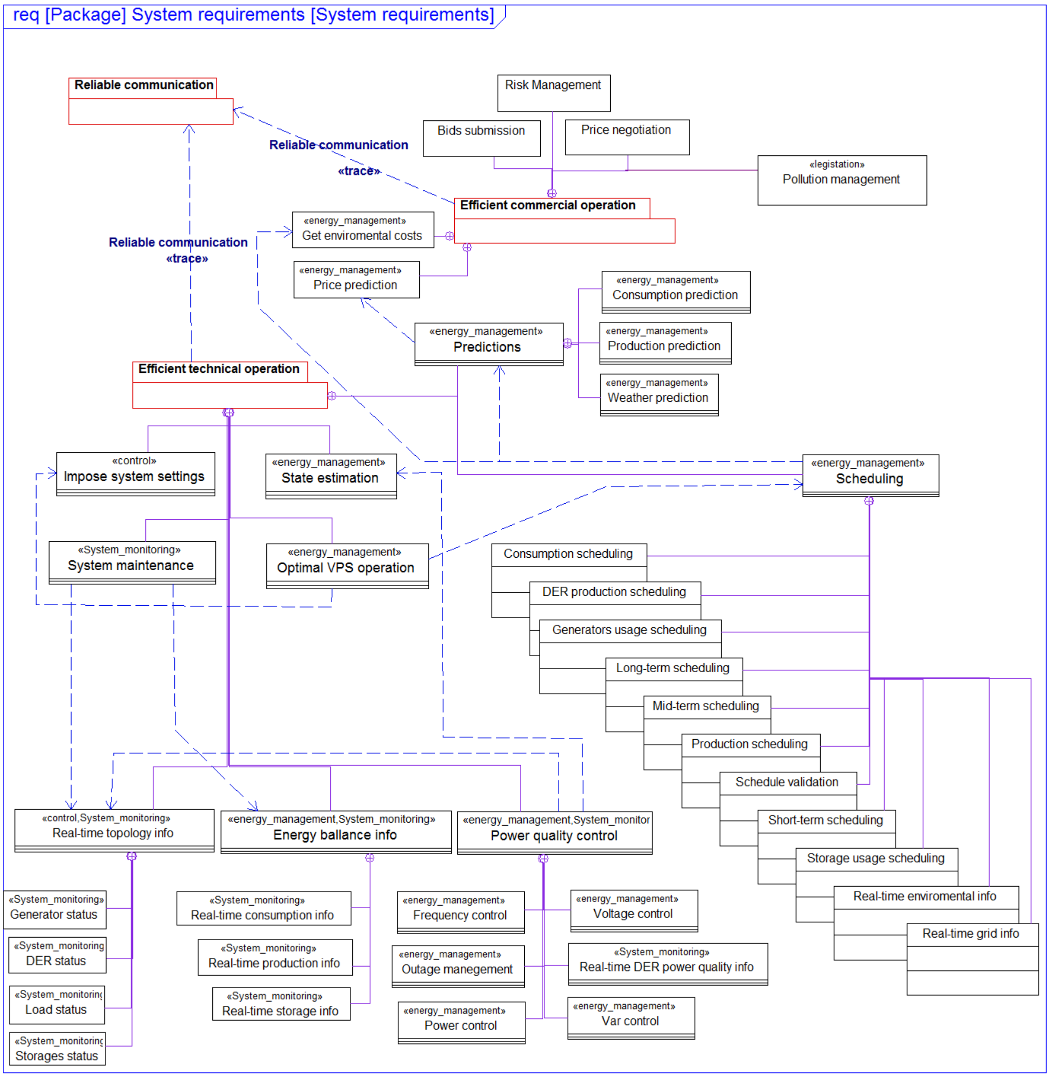


Figure 5.6: System requirements diagram

pollution is being modeled as additional fees and taxed to be paid in case of pollution margins violation. Besides inheritance relations, other relations are presented as well as already explained in section 4.2.1.

5.4 Use cases definition

Use Case Diagram presented on figure 5.7 describes the general VPS functionalities, their relation as well as involved internal and external actors (that can include human actor as well, but mostly referrers to component, organization, regulation, etc.). When defining a general use-cases, system requirements extracted in the previous step were taken into consideration. Further decomposition of use cases presented here has been done on figure 5.8 while relating use cases to particular system requirements. Internal actors of VPS, representing VPS components, are chosen based on previous VPS discussion and results from relevant project (see sections 2.5 and 2.7). Still, actor that are initially included can be changed later during behavioral modeling of the system. A short description of basic VPS use-cases/functionalities and key actors follows.

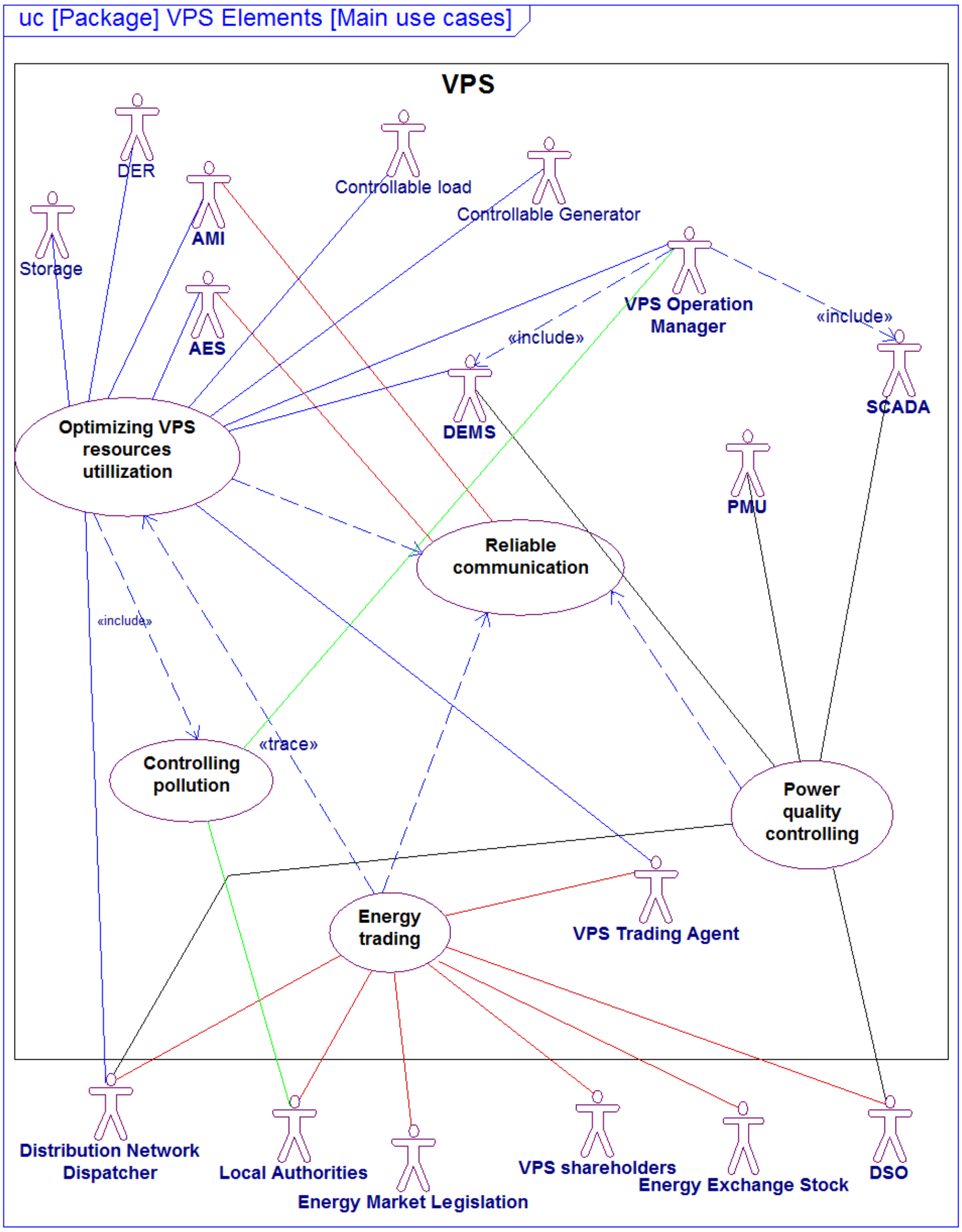


Figure 5.7: VPS operation use cases

Optimizing VPS resources utilization referees to application of optimization algorithms set achieve optimal usage of VPS resources (DERs, storages and distribution level generators) while producing energy for a local usage (energy that will be used by loads inside VPS) but also during energy trading. One should keep in mind that energy trading includes both selling of energy to power grid, and buying it when VPS production can not fulfil it's energy needs. Moreover, depending on state and local authorities regulation, as well as general energy market rules; optimization of resources usage can include buying and storing of energy when this can be approved with a low grid price. Even that this use-case has been derived while keeping in mind technical VPS requirements, as it has been already presented, optimization of resources mostly referrers to economical optimization. Basic scenario of fulfilling this functionality includes: gathering of field information (Advanced Measuring Infrastructure - AMI); getting information from the energy market (VPS trading agent); applying of advanced scheduling algorithms (VPS operation manager - SCADA and Distributed Enertgy Management System - DEMS); and finally imposing of new settings (Actuating Embedded Systems - AES). Note that Distributed Energy Resources, storages and controllable loads and generators are treated as a part of VPS even though the represent external elements that can exist on their own.

Controlling pollution functionality includes keeping CO2 emission and other kind of pollution in defined ranges. Since pollution is modeled trough additional payment that also include additional fees an taxes,it can be considered from the economical point of view. For that reason, controlling pollution is a part of optimal resources usage functionality.

Reliable communication includes all communication related VPS functionality. Various communication technologies and standards can used to provide secure, reliable and safe real-time communication between VPS and the rest of the power grid; controlled/moitored devices and embedded systems; VPS control center and field embedded devices, etc. For the reason than have already been explained for a several times in scope of this thesis, communication aspects of VPS will not be further considered. Note that reliable communication service is being used by all other VPS services.

Energy trading referrers to activity of energy selling and buying between VPS and the rest of the power grid. Besides many external actors involved in this functionality, VPS Trading agent has a major roll as the internal actor. Full description of this functionality has to wait till full development of Smart Grid energy trading legislations. For that reason, further decomposition of this functionality will not be done. Still, based on energy market trends described in section 2.3.3, it assumed that functionality of VPS resources optimization is being done in environment in which electrical energy price can be changed on hourly basis.

Power quality controlling includes full monitoring and control of power quality (voltage, active and reactive power, frequency, phase, etc.) at production place

(DERs, traditional generators, storages) and consumption place. Moreover, power quality when trading energy is being observed by DSO as well. Phaser Measurement Unit (PMU) provides full power quality monitoring while control and acquisition is being done by DMS and SCADA. To fully provide this functionality, additional mechanical elements, that are out of scope of this thesis, would be needed as well.

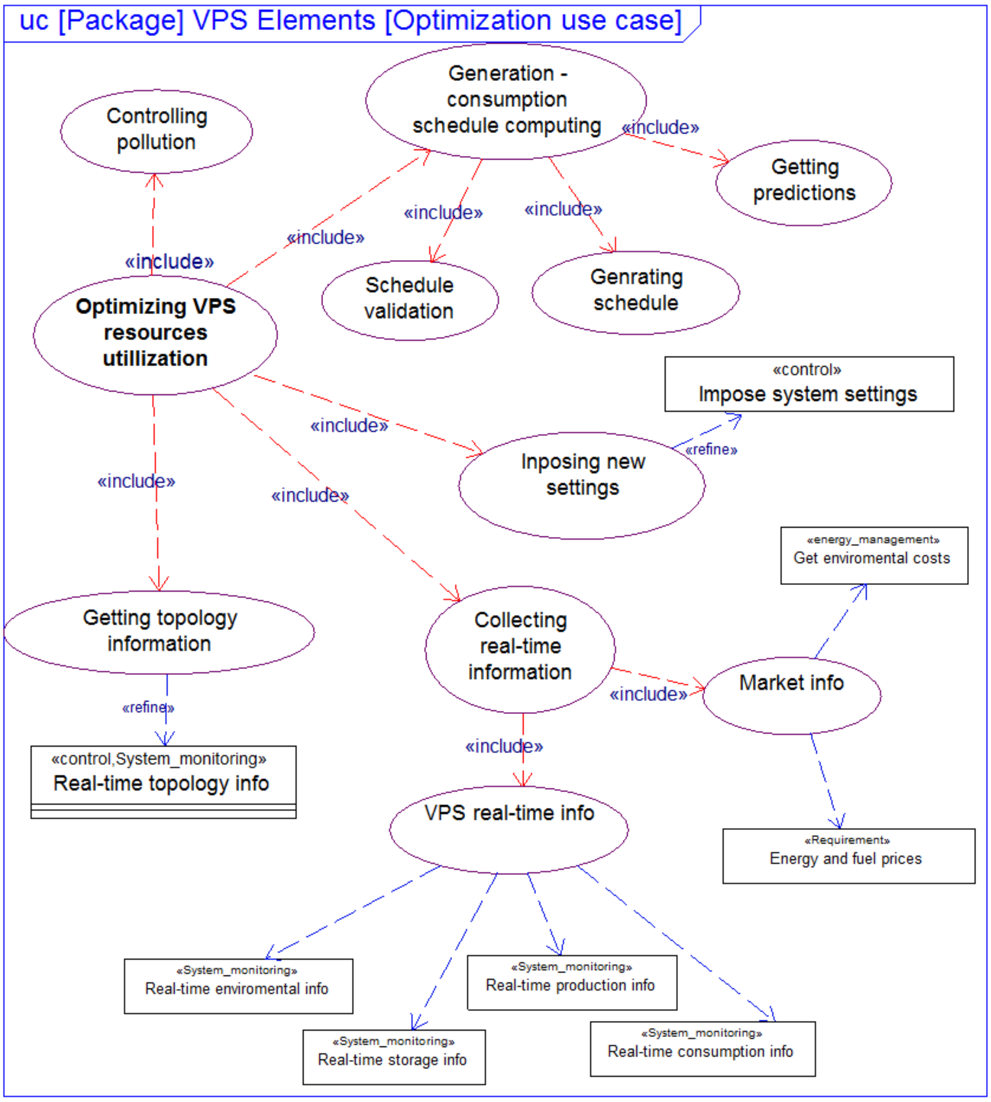


Figure 5.8: Use case decomposition and requirements relation

As example of further use cases decomposition, use case of Optimizing VPS resources utilization with all included use cases and related system requirements is given on figure 5.8 (for other use cases decomposition, as well as all other details reader is referred to original Rational Rhapsody project developed in scope of this thesis). A fine grained use cases decomposition till atomic use cases is presented on figure 5.9. One should keep in mind that not all sub uses case are necessarily atomic ones and that further decomposition is possible depending on details that one need to achieve on further behavioral and structural models. Moreover, in cases that all sub-requirements that can be derived from one, for this purpose called major requirement; are related to one use case, only major requirements is presented on this diagram. For further requirement decomposition reader is referred to figure 5.6.

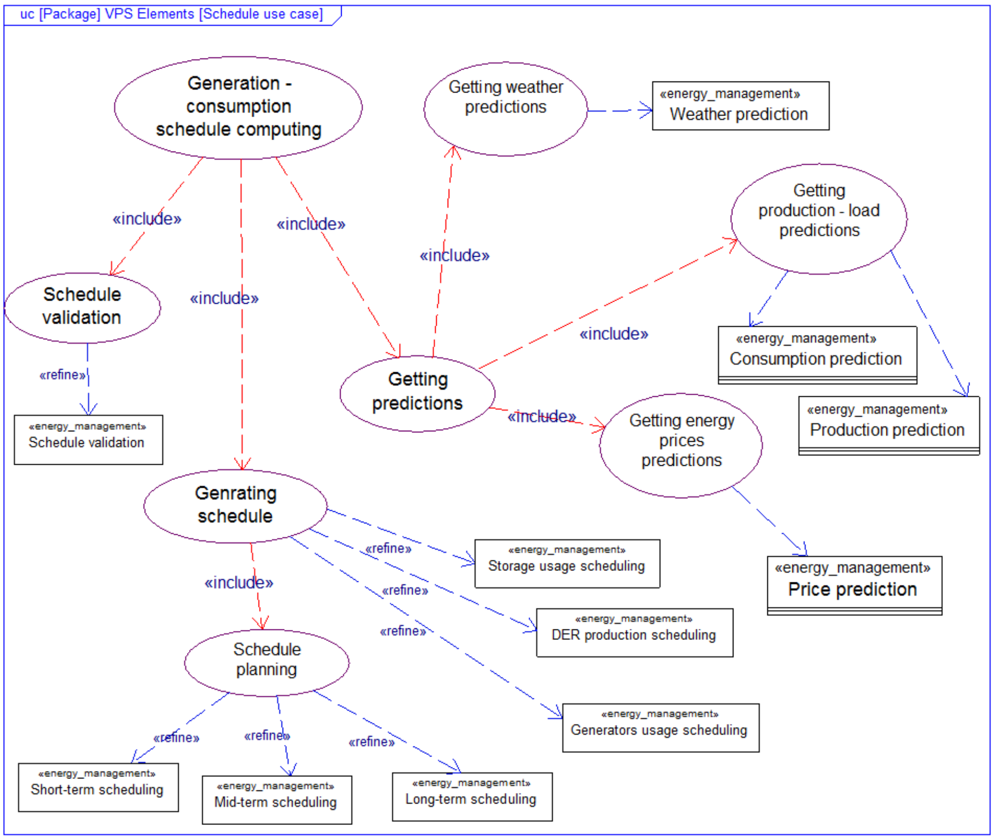


Figure 5.9: Fine grained use case decomposition

Once all uses cases have been decomposed, further behavioral system modeling can continue and structural modeling can start. In case that not all system requirements are covered by defined uses cases, additional uses cases have to be defined as a part of system remodeling.

5.5 Final system model development

So far, with excuse of actors introduction and their relation to use cases, only VPS behavior has been modeled. This part concludes high level system behavioral modeling while introducing sequence diagrams as a mean of VPS use cases realization as well as structural high level system modeling by developing system level block definition diagram and system level internal block diagram. Still, not all sequence diagrams will be presented in this section but a subset that represents interaction sequences necessary for realization of use cases presented in figures 5.8 and 5.9. A more detailed interaction for particular use cases realization including object and control flows trough activity diagrams will be presented in chapter 6 as part of concrete device functional description and detailed behavioral and structural modeling. For more details as well as other sequence and activity diagrams not explicitly presented in this section reader is referred to the source Rhapsody project.

One should keep in mind that development of behavioral and structural models presented in next subsections is iterative process. Since behavioral model describes system behavior in terms of it's components communication and interaction while realizing system functionalities, for it's development it is necessary to know the basic system structure. On the other hand, structural model shows relation between system components that are necessary for system functionalities satisfaction while new components can appear while canalizing system behavior in scope of functions realization. For that reason, development of structural and behavioral model is an iterative process that relies on waterfall process management. Diagrams presented in next two subsections describing structural and behavioral system model are final model versions containing all elements of VPS structure and high level system behavior.

5.5.1 Behavioral model

A part of system behavioral model has been already presented while introducing use cases as system functionality elements. A fully developed, fine-grained and detailed VPS model should include:

- Sequence diagrams expressing interaction between VPS structural elements for use cases realization.
- Activity diagrams for more detailed interaction description including both control and object flow while introducing the exact pots interfaces for communication between components.
- State charts to describe a state dependent behavior and develop event based model.

Still, it is not necessary to use all diagrams for developing behavioral system model. In particular case, sequence diagrams are used do describe VPS message based behavior. A flow based behavior will be used in next chapter as a part of demonstration of particular device modeling (see chapter: 6). Finally, besides additional structural diagrams, state charts can be used as a part of system level verification. System verification is out of scope of this thesis and will be briefly discussed in the last chapter (see chapter: 7).

For a demonstration purpose a realization of use case "Optimizing VPS resources utilization" is presented on figure 5.10. Note that, in order to get more details, diagram shows SCADA and DEMS as separate elements even though they, in fact, are part of VPS operation manager. It starts with realization of real-time information collecting that includes VPS and market information. AMI gathers all information from field devices including current statuses and production/conusmtion amounts, analyzes them and presents to VPS operator. Collecting of market information is done trough interaction with trading agent responsible for energy market stock data exchange. Getting topology information is a separate use case with another dedicated sequence diagram that is directly included into resources optimization (for more details regarding this and other sequence diagrams that are not presented in this section reader is referred to original Rhapsody project file). Finally, after new schedule generation and validation new settings are imposed to a field devices trough AES.

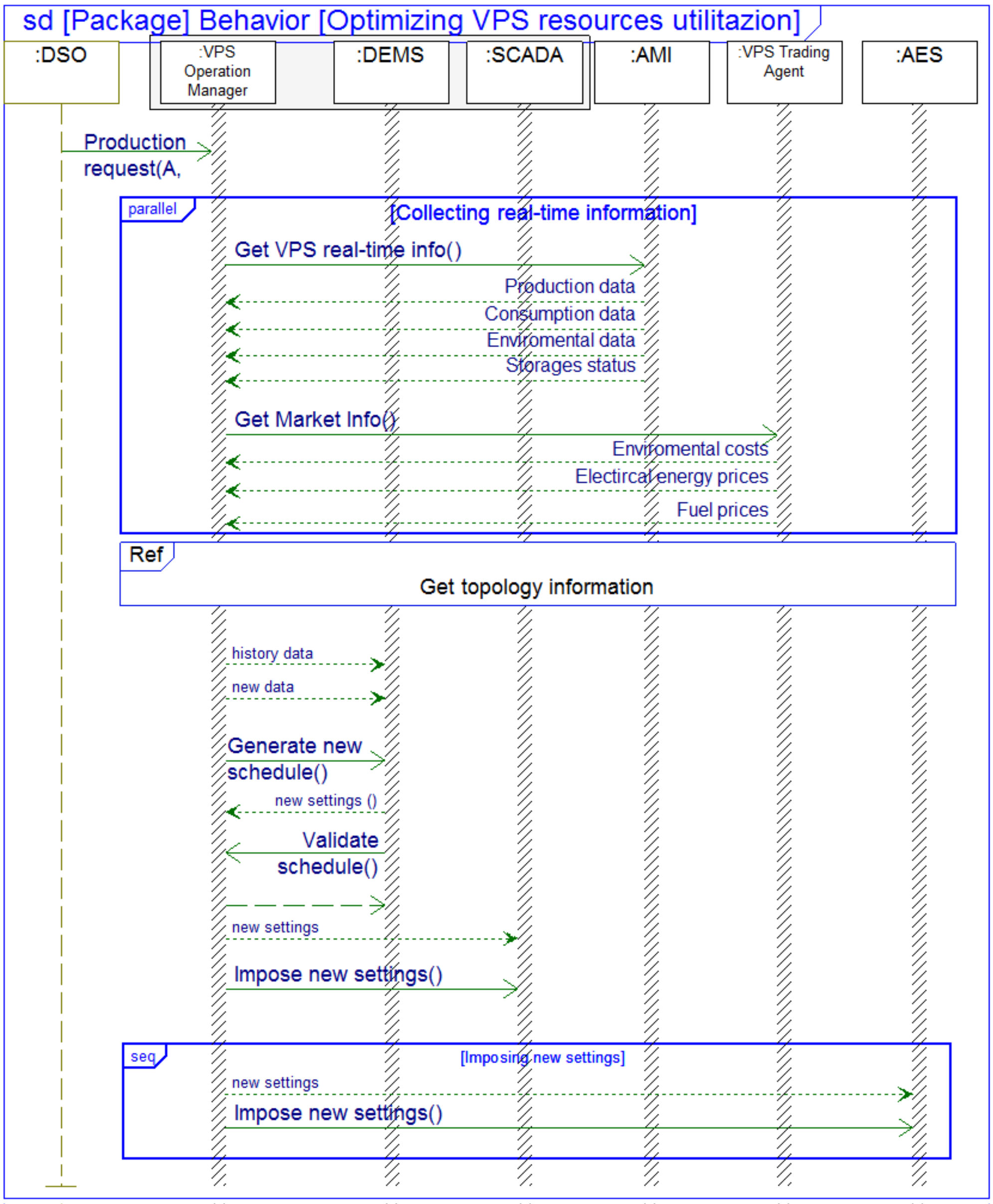


Figure 5.10: Optimizing VPS resources utilization use case realization - sequence diagram

Since VPS real-time data collection is a separate use case, not all the details are presented in a previous diagram. For all the details please see figure 5.11. Since the same diagram will be used in next chapter as a part of case study discussion it will not be explained in this section. Important point that one should note is that some additional structural elements that are previously treated as a part of AMI are appearing in this diagram.

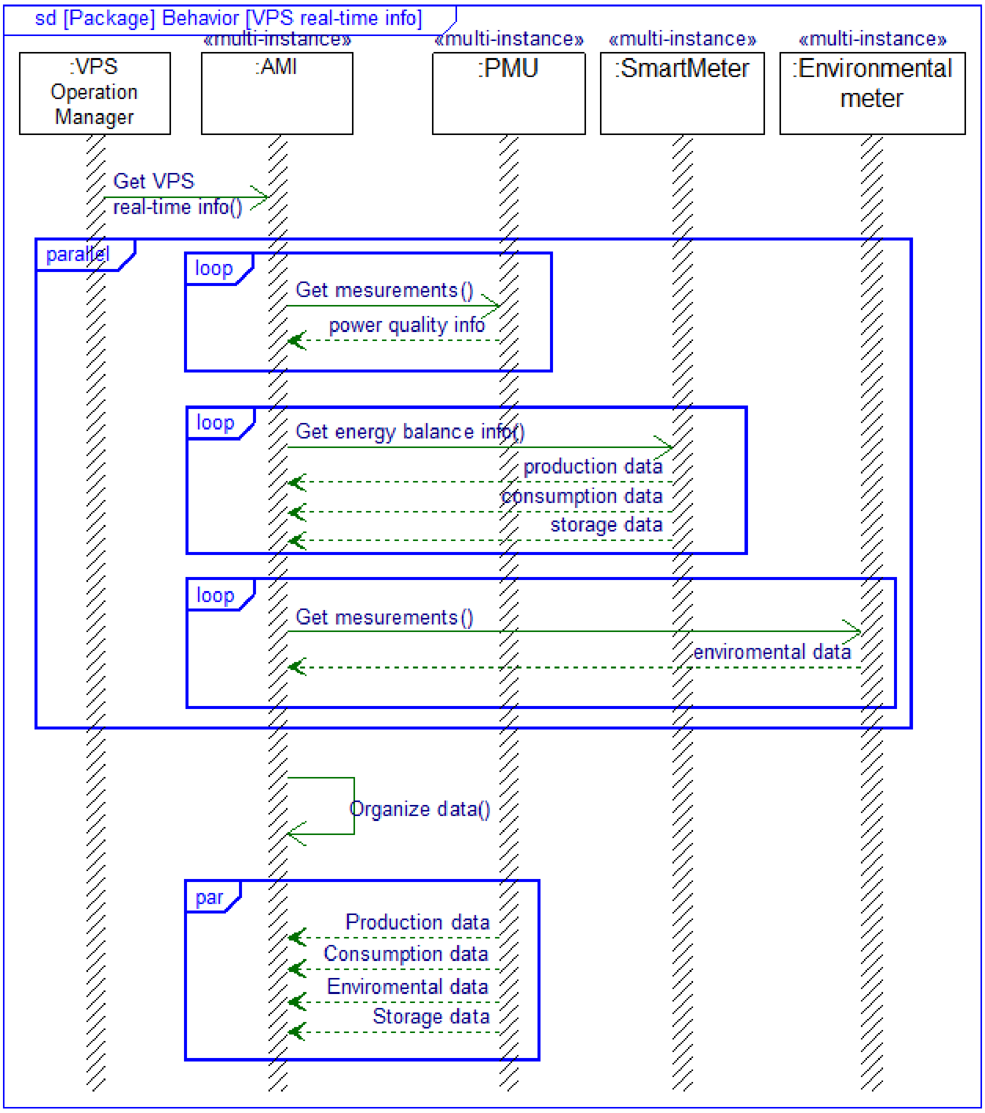


Figure 5.11: Collecting VPS real-time information - sequence diagram

5.5.2 Structural model

Based on behavioral VPS analysis and modeling whose part has been presented in previous sections, VPS structural components has been extracted. As it has been mentioned several times before, final VPS structure is not result of process being done sequently but rather iteratively. Moreover VPS structure can be observed from a various levels of abstraction representing different details levels.

One of the goals of this thesis is to present a high level abstraction model of VPS behavior and structure that will serve as a mutual understanding point for stakeholders and provide basis for further system molding in terms of components modeling and functional requirements extraction. In this chapter two high level structural models of VPS are presented. Figure 5.12 shows block definition diagram of VPS containing all it's components as well as their relation. One should note that some VPS parts that have been previously mentioned during behavioral modeling are not presented on this diagram since the are considered as parts of other components (for instance SmartMeter is a part of AMI).

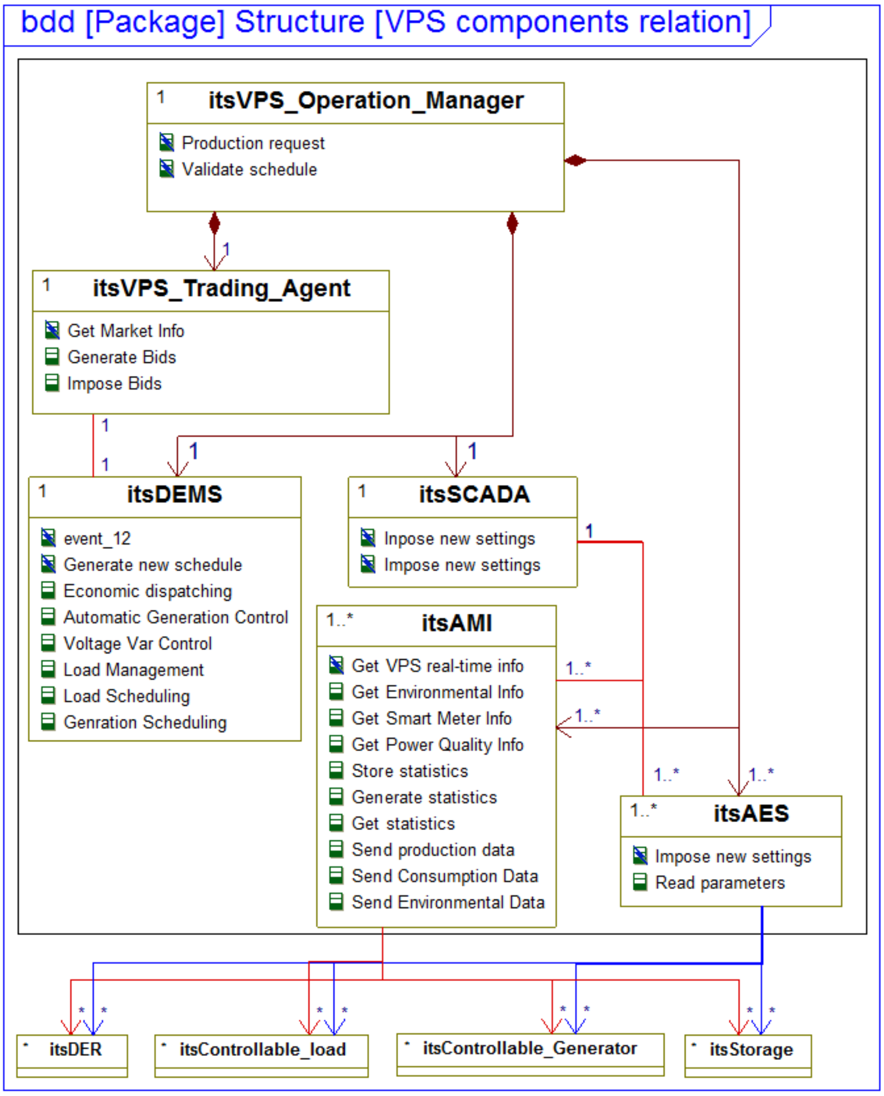


Figure 5.12: VPS block structure

Figure 5.13 shows internal structure of VPS giving a more detailed components description, communication ports and flows. One should keep in mind that VPS is not standardized concept and that some of component attributes, flows and operations defined serv for demonstration purpose only. Still, basic structure and principles as well as data flows are defined according to the previously parented behavioral analysis as well as results achieved in relevant projects (see sections 2.5 and 2.7). More details regarding AMI structure and it's flows are given in next chapter.

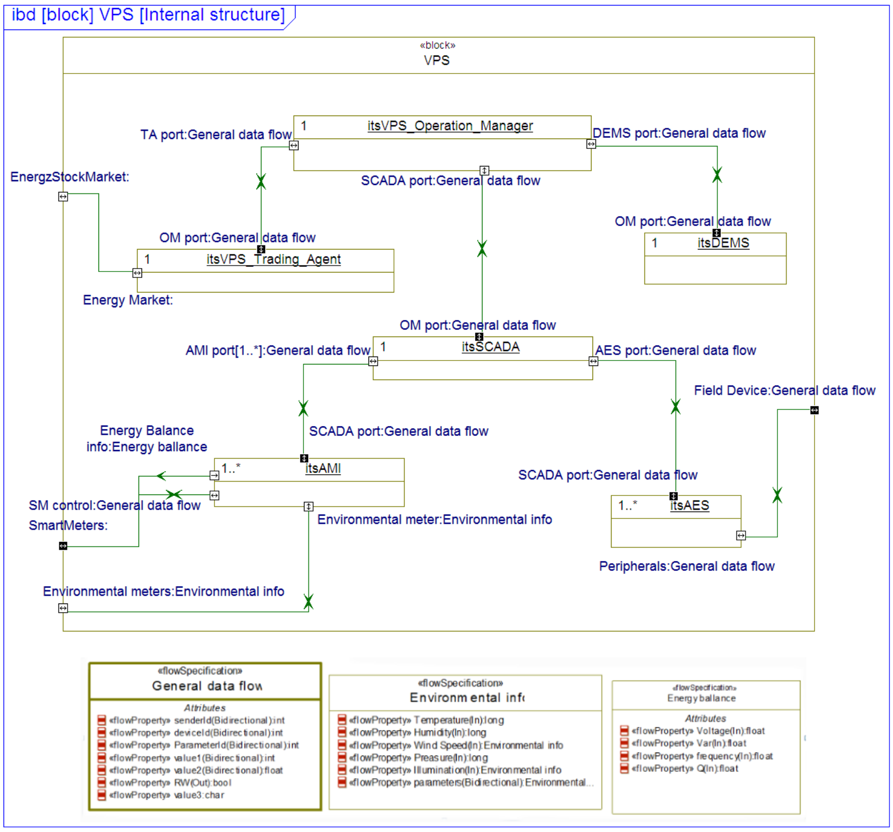


Figure 5.13: VPS internal structure

5.6 Framework for algorithms optimization

As proposed by adopted model-driven methodology presented in section 4.1, once system components and functionalities have been identified, system functionalities optimization can start. This referrers to exploration and optimization of chosen set of software algorithms, and is not, in any terms, related to system level verification that is out of scope of this thesis. For the purpose of algorithms optimization a massage based OMNeT++ (for more details regarding OMNeT++ see section 4.3.2) framework has been developed. Initially developed for the purpose of consumption scheduling algorithms optimization, this framework comes several predefined VPS component classes. Since current framework version applies only to a small set of VPS functionalities, one should keep in mind that VPS topology that can be developed with this framework, does not fully reflect topology and structure described previously in this chapter. Still, same set of classes will be fully reused once when framework reaches a more complete version able to simulate a full VPS system in terms of it's software functionalities.

Since OMNeT++ is a a message based simulator a concept of time has been introduced trough a simple clock class synchronizing operation and communication of all other components. Trough a single instance of this class user can define a clock period duration and overall simulation time. Besides clock class, current framework version comes with following classes describing VPS components or it's parts:

Consumer class is used to describe all kinds of controllable loads. During simulation time each instance of the consumer class can generate one or more tasks representing consumers consumption needs. Currents version of simulator supports one task per consumer object. Each task is being described by following parameters:

- Consumption profile - describing task energy need over time. It is given trough a separate textual file dedicated to a consumer object upon creation. Task end time is given trough this file as well.
- Tasks arrival and deadline time - given as set of arrival, deadline times trough a separate file. Since current version assumes only one task per consumer object, only first pair of arrival deadline time is considered.
- Preemptive parameter - task can be preemptive or not.

When tasks arrival time has been reached, consumer generates a task message that is being send to scheduler. Depending on applied scheduling algorithm, schedule will define when the consumer should be turned on or off. Switching consumers state is done trough a scheduler messages. For each clock period that consumer is in on state a message containing a current energy consumption is being sent to the SmartMeter.

HP predictor class describes a part of VPS operation manager module whose function is to, based on current energy balance, load predictions and weather predictions generates a DER's consumption prediction. HP predictor summaries predicted production of all VPS distributed energy resources. A production prediction is described trough a separate file dedicated to each HP predictor object. Upon simulation starting, HP predictor sends production prediction to the scheduler.

DSO predictor class describes a part of VPS Trading agent with a function of energy price prediction. This class operates similarly to a previously described one.

SmartMeter class describes a SmartMeter component or better jet a part of AMI component. On each clock period it receives a messages from a working consumers describing a current consumption. On the end of simulation Smart meter generates a VPS consuming profile.

Scheduler class describes a part of VPS operation manager component. It is a core class of the current version of framework. It receives prediction messages on the simulation start and consumption messages for each task consumption task arrival. Using a messages it controls each consumers state during simulation. Finally, it gates final results from all the classes and resent them trough a few output files describing required consumption, predicted and real production and energy prices and task schedule. Scheduler class comes with a dummy schedule that can be used only for demonstration purpose, and provides interface for a concrete scheduling algorithm application. Like this, different user defined scheduling algorithms can be applied and results easily compared.

Defining VPS network topology can be easily done trough a graphical interface and using predefined components. Components parameters have to be defied as already explained. Moreover, framework operation, simulation running, and results obtaining procedure have been briefly described during components description and will not be fatherly explained here. For more detail regarding scheduling framework, reader is referred to a source code given as appendix to this thesis. Besides, some more detail regarding operation and results obtained using this framework are given in section 6.2 as a case study.

Chapter 6

Case Studies

"My method is different. I do not rush into actual work. When I get a new idea, I start at once building it up in my imagination, and make improvements and operate the device in my mind. When I have gone so far as to embody everything in my invention, every possible improvement I can think of, and when I see no fault anywhere, I put into concrete form the final product of my brain"

Nikola Tesla

This chapter proposes how previously developed methodology, VPS model and framework can be fatherly used. Starting with structural and behavioral VPS model diagrams AMI has been used as an example of embedded devices functional requirements extraction and internal structure modeling. Moreover, demonstration of developed framework for software algorithms optimization usage has been presented using a sample VPS example containing several predefined components and profiles.

6.1 Advanced Measuring Infrastructure requirements and internal block diagram

Starting with system level structure (block definition diagrams and internal block diagrams), use cases diagrams and basic interaction sequence diagrams, a general role of Advanced Measuring Infrastructure (AMI) can be identified. This chapter gives an additional communication based diagram for a fine grained behavior description of gathering VPS information(see figure 6.1). This is the major use case in which AMI takes place. Moreover, it includes all AMI functionalities and for that reason can serve as a staring point for AMI requirements extraction and components identification. Presented activity diagram presents both, activity and data flow. The exact data flow structure has been proposed as a part of structural AMI modeling.

Since presented activity diagram already gives an idea on requirements set (requirements are related to identified actions) as well as on how requirements are related to AMI parts (using swim-lines) it would be redundant to present AMI requirements diagram here. For their reason, requirements and it's relation to particular AMI parts are given trough a simple text below.

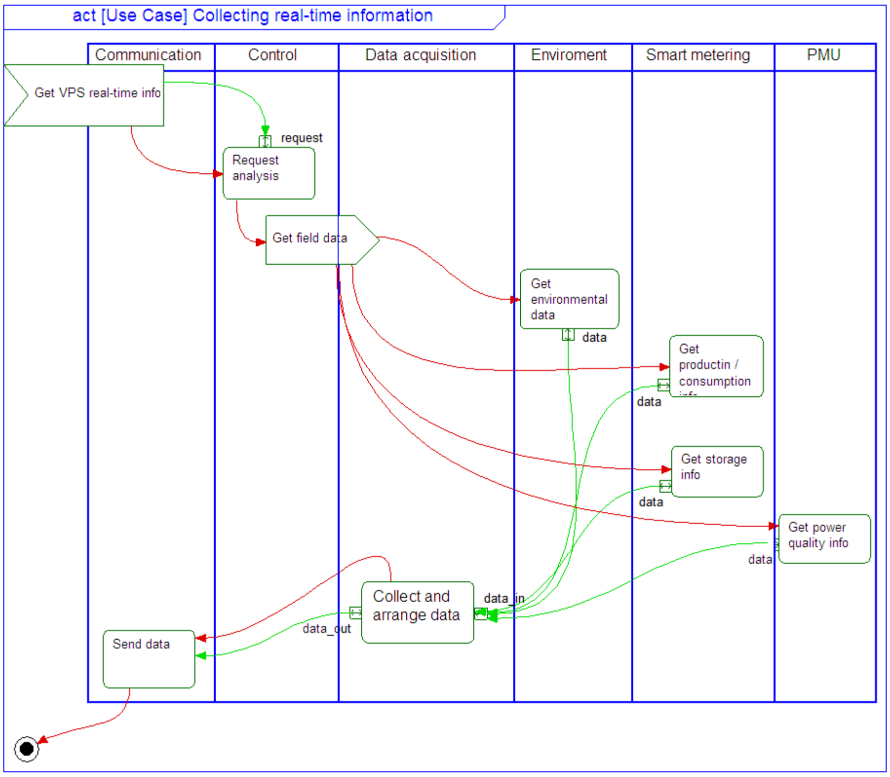


Figure 6.1: Collection real time information activity diagram

Based on activity diagram above (figure 6.1), presented actions, control and object flows and activity deduction trough swim-lines, following AMI requirements and structural components can be identified:

Communication with VPS operator in terms of request receiving and data packages sending. A separate component for this request has been identified with a main purpose of receiving commands and possibly data objects from operator and forwarding it to appropriate AMI component and providing a data sending service from AMI to VPS operator.

Request processing includes analyzing on various tapes of requests (consumption info request, power quality request, environmental data request, etc) and it's combinations. AMI control unit has been identified as a separate component that serves as request analyzer. Moreover the same unit forwards request and controls all other AMI units.

Settings imposing and reading as a part of AMI and included/attached components controlling. This request is fulfilled by communication and control unit.

Requesting measured data is done by control unit after request analysis.

Gathering production and consumption data using a smart meter as an external component requires additional AMI unit that serves as controlling ind communication interface toward Smart Meter device(s).

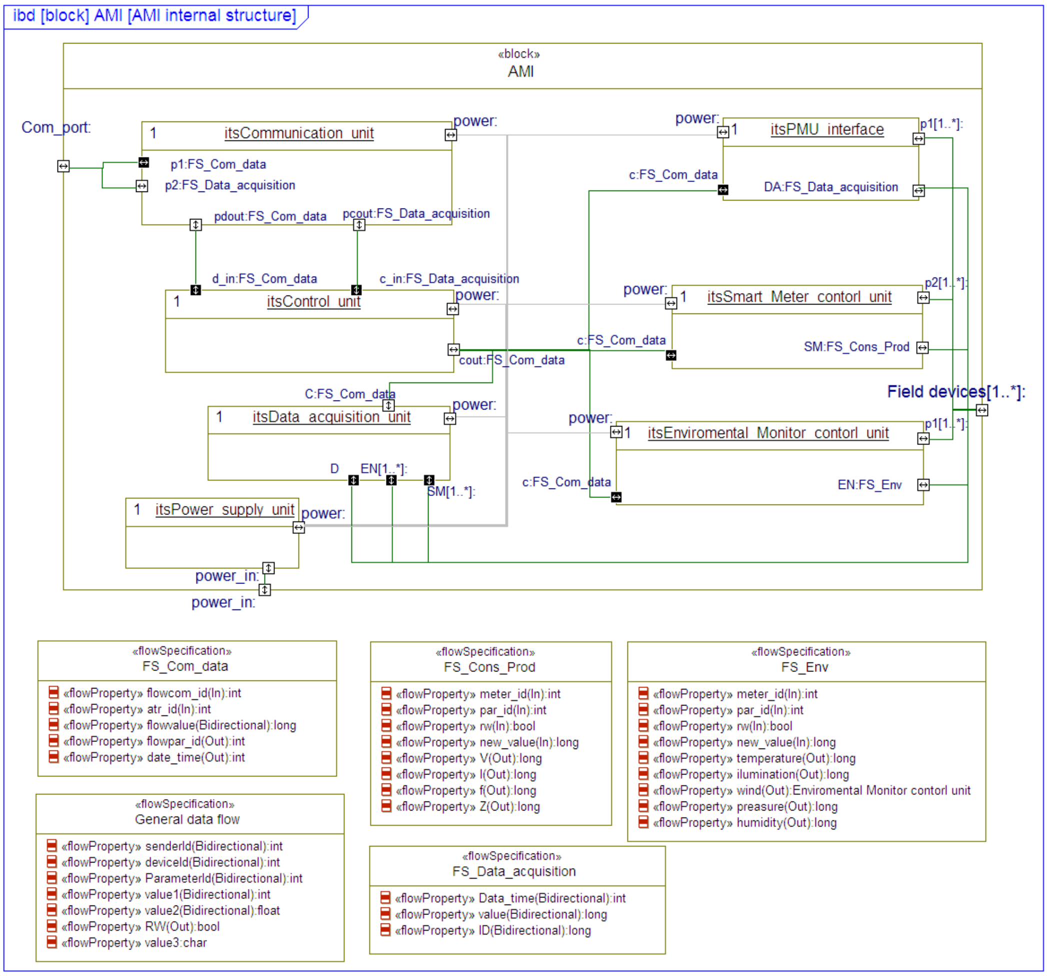


Figure 6.2: AMI internal structure

Collecting storages data is done trough Smart mattering interface unit since Smart Meters are seen to be used for measurements related to storages.

Getting power quality information requires controlling and communication interface toward PMU as an external element that handles power quality issue.

Collecting environmental information is done trough environmental interface as a separate AMI unit able to connect and control various environmental sensors and meters (including temperature sensor, humidity sensor, wind direction and speed meter, etc.).

Data acquisition includes a local data analysis and history storing but mostly referrers to data collecting and arranging in a way that VPS operation manager expect them to be. Data acquisition unit has been dedicated for this particular requirement.

Data sending is a sub-requirement of communication with VPS operator requirements that includes sending of collected and arranged data from data acquisition unit to VPS operator trough communication unit.

Finally, according to the requirements and components identified, as well as previous VPS structural and behavioral analysis, AMI internal structure can be proposed as presented on figure 6.2. One should keep in mind that proposed flow specifications are based on analysis presented here as well as information gathered form relevant projects (see section 2.7) but still should serve mostly for demonstration purpose rather that final solution proposal. Once standardization of Smart Grid protocols will be done, complete flow specification can be proposed. Based on diagram, extracted requirements can be summarized in following.

6.2 Demonstration of scheduling framework usage for algorithms comparison

As previously explained in section 5.6 a framework for consumption scheduling algorithms optimization has been developed in scope of this thesis. Purpose of this section is to give an demonstration of it's usage. For that reason, besides a dummy scheduling algorithm that is a part of schedule class, algorithm for exhaustive schedule exploration has been adopted from [24] and applied to a sample network given on figure 6.3(a). Besides clock component as necessary for time synchronization, network contains following elements:

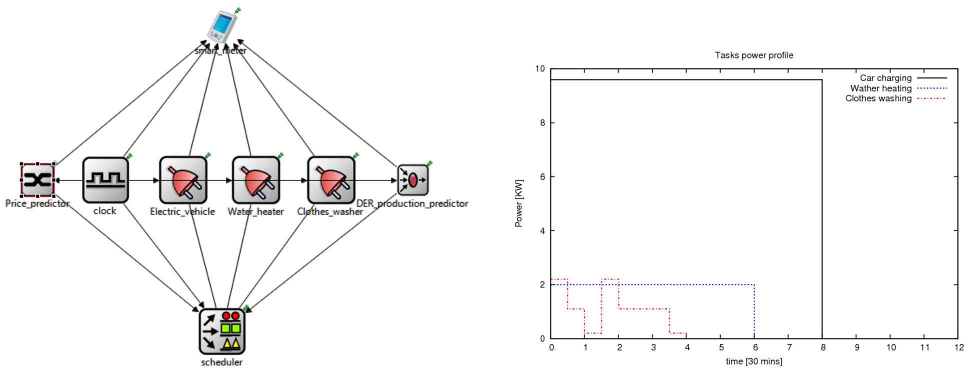
- Three consumer components using predefined consumption profiles (see figure 6.3(b)) describing clothes washer, electrical vehicle charging and water heating. Loads profiles are realistic and based on [**?**] and [10]. Tasks arrival and deadline times are given in table 6.1
- Price predictor, an instance of DSO predictor class giving an hourly based electrical energy price change (Prices are based on [29]).
- DER predictor as an instance of HP predictor class gives an prediction of VPS total production.
- Smart meter keeps tracks on consumers consumptions current price and production and generates final results.

- Scheduler, based on applied algorithm defines consumption schedule.

As a part of this case study two different scheduling algorithms are applied and results compared. Network structure and consumption profiles are given on figure 6.2, while figure 6.2 presents electrical energy price and DERs production predictions.

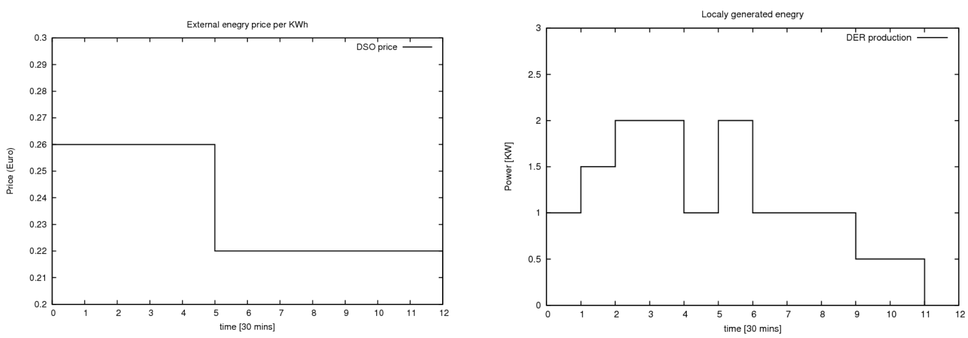
Task	**Arrival time**	**Deadline time**
Car charging	0 min	360 min
Water heating	30 min	240 min
Clothes washing	30 min	240 min

Table 6.1: Tasks arrival and deadline times



(a) (b)

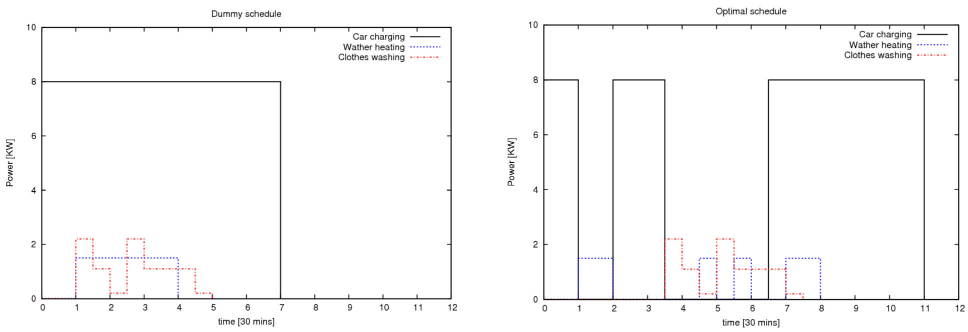
Figure 6.3: (a)Sample network (b)Consumption tasks profiles



(a) (b)

Figure 6.4: (a)Electrical energy price prediction (b)DERs production prediction

According to the energy price change as given on figure 6.4(a), calculated prices for a given set of tasks are 6.8 Euros when a dummy schedule (or no scheduling at all) has been applied (figure 6.5(a)) and 5.9 Euros when an optimal schedule (obtained as a result of exhaustive schedule exploration) has been applied (figure



(a) No scheduling (b) Optimal schedule

Figure 6.5: Consumption tasks schedules

6.5(b)). According to this, applying an optimal schedule to a given set of tasks saves 13% of VPS costs to the rest of power grid. Obtained results can serve as a reference for other consumption scheduling algorithms exploration.

Chapter 7

Conclusions and future developments

"When I am working on a problem, I never think about beauty but when I have finished, if the solution is not beautiful, I know it is wrong."

R. Buckminster Fuller

Virtual Power System emerge as solution to aggregating heterogeneous renewable resources and (un)controllable loads while leveraging their cumulative capacities and energy requests; on the other hand, this requires a strong presence of ICT supports beyond the traditional energy management ones, namely distributed and ad-hoc hardware/software solutions consisting of suitable embedded systems. These systems could represents building elements of Smart Grids being at the same time autonomous technical and commercial nodes. In sense of system structure and functionalities VPS should reflect Smart Grid solutions and in that sense solution and methodology presented here can be used as a starting point for Smart Grid modeling.

The main contribution of this thesis is development of **VPS modeling methodology** that is based on model-driven system engineering methodology, with ultimate goal to reach a comprehensive VPS description, that starting with stakeholders identification, eventually leads to the functional requirements (and structural design proposal) of components (especially embedded devices) to be deployed as enabling technologies of VPS. Due to a VPS system complexity and size (in terms of numerous stakeholders and subsystems) a full application of this methodology and development of fine-grained components models requires a large team to be employed, and so goes beyond the scope of this thesis. Moreover, a full application of methodology has to be done in parallel with Smart Grid, and so VPS, standardization, in terms of energy market standardization, communication standardization, legislation, etc. Still, a high level VPS model has been developed here. This model, as presented in case study part of this thesis, can serve for further, fine-grained components modeling.

As applying proposed methodology, **VPS stakeholders** have been fully identified and **system context** described. This served as a starting point for requirements engineering, that included users requirements collecting and analyzing and system requirements extraction. Even not complete, **requirements diagrams** and tables presented here reflect real stakeholder' and system needs. Once when standardization of particular Smart Grid parts will be done, a full requirements model can be realized by upgrading the existing one. Starting with those requirements a hight level **VPS model** have been developed. It consists of **VPS behavioral model** in terms of it's functionalities and usage scenarios modeled trough use cases and sequence diagram, as well as **model of VPS structure** presented with blocks and internal block diagrams including VPS components and communication flows structure. As a part of VPS functions optimization, a **simulation framework** (with a major goal of consumption algorithm optimization) has been developed. Finally, as a demonstration of proposed model usage, a detailed structural and functional model of Advanced Metering Infrastructure (AMI) and demonstration of algorithms optimization framework have been done. Still, models developed here should not be considered as final ones, and could be fatherly extended and adopted depending on new stakeholder requirements and standards. Figure 7.1 gives an overview on methodology steps while emphasizing parts being applied in scope of this thesis (for a detailed methodology diagram see figure 4.1).

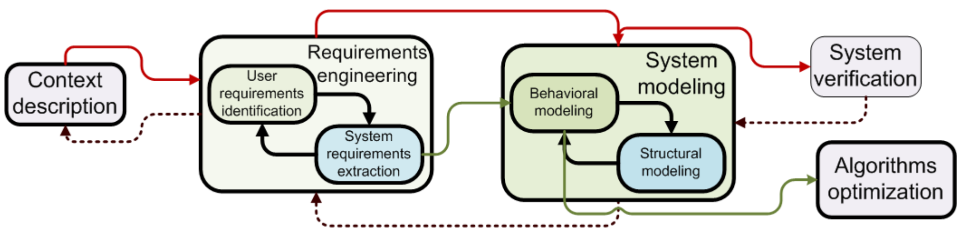


Figure 7.1: Thesis overview

In summary, in this thesis model-driven methodology has been adopted and applied on VPS case resulting in structural and behavioral VPS model that serves as a mutual understanding/developing platform among stakeholders/enginneerrs.

Being already recognized by scientistic society trough several publication (see [45], [46], [44] and [47]), author strongly believes that project presented in thesis has a great potential for further extension. Our ongoing work concerns further development of VPS model with the aim of providing finer grain components models and full identification of VPS embedded system functional requirements. Moreover, we aim at exploring (developing) and applying a suitable system model verification method as well as method of model exportation to SystemC/VHDL code.

Bibliography

[1] Alpenergy project. http://www.alpenergy.net/.

[2] European technology platform for the electricity networks of the future. www.smartgrids.eu.

[3] National energy technology laboratory. http://www.netl.doe.gov/smartgrid/.

[4] U.s. department of energy. http://www.energy.gov/.

[5] U.s. smart grid projects. http://www.smartgrid.gov/.

[6] *Telelogic Rhapsody 7.4 Systems Engineering Tutorial.* IBM Corporation, 2008.

[7] Online niagara - hydro power. http://www.onlineniagara.com/info/niagara-falls-power.htm, August 2009.

[8] *Basic systems engineering design in Rational Rhapsody 7.5.1.* IBM Corporation, 2010.

[9] Omnet++ 4.1 documentation and tutorials. http://www.omnetpp.org/documentation, 2010.

[10] Tesla car project. http://www.teslamotors.com/, 2010.

[11] Trading economic analytics. http://www.tradingeconomics.com, April 2010.

[12] *OMG Unified Modeling Language 2.2 specification.* OMG press, February 2009.

[13] *OMG Systems Modeling Language 1.1 specification.* OMG press, November 2008.

[14] International Energy Agency. Emission baselines for electricity generation projects. http://www.iea.org/papers/2000/elgen.pdf, August 2007.

[15] S. Massoud Amin. For the good of the grid. *IEEE power & energy magazine*, pages 48–59, November/December 2008.

[16] Gavin Arthurs. *Model-Based System Engineering (Elements for Deploying an Efficient Development Enviornment).* Telelogic White Paper, 2008.

[17] Steven W. Blume. *Electric Power System Basics for the Nonelectrical Professional*. John Wiley & Sons, Inc., 2007.

[18] Alec Brooks, Ed Lu, Dan Reicher, Charles Spirakis, and Bill Weihl. Demand dispatch. *IEEE power & energy magazine*, pages 20–29, May/June 2010.

[19] Sustainable Energy Europe Campaign. www.sustenergy.org.

[20] A. Clark, Pavlovski C.J., and J. Fry. Transformation of energy systems: The control room of the future. In *IEEE Electrical Power & Energy Conference (EPEC)*, Montreal, QC, USA, Oct 2009.

[21] F.M. Cleveland. Cyber security issues for advanced metering infrasttructure (ami). In *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, Pittsburgh, PA, July 2008.

[22] IBM corporation. Ibm rational rhapsody documentation. http://publib.boulder.ibm.com/infocenter/rhaphlp/v7r5/index.jsp, 2010.

[23] IBM corporation. Ibm rational rhapsody oficail site. http://www-01.ibm.com/software/awdtools/rhapsody/, 2010.

[24] Onur Derin and Alberto Ferrante. Scheduling energy consumption with local renewable micro-generation and dynamic electricity prices. In *CPSWEEK/GREEMBED, First Workshop on Green and Smart Embedded System Technology: Infrastructures, Methods and Tools*, Stockholm, Sweden, April 2010.

[25] K. Dielmann and A. van der Velden. Virtual power plants (vpp) - a new perspective for energy generation. *Modern Techniques and Technologies, MTT*, pages 18 – 20, April 2003.

[26] Michel dos Santos Soares and Jos Vrancken. Model-driven user requirements specification using sysml. *JOURNAL OF SOFTWARE*, (6):57–68, June 2008.

[27] Mohamed E. El-Hawary. *Introduction to electrical power systems*. John Wiley & Sons, Inc., 2008.

[28] Mohamed E. El-Hawary and Yaoyu Wang. *Communication and Control in Electric Power Systems: Applications of Parallel and Distributed Processing*. John Wiley & Sons, Inc., 2003.

[29] Eurostat. Electricity - domestic consumers - half-yearly prices. http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=nrg_pc_204&lang=en, 06 2010.

[30] Hassan Farhangi. The path of the smart grid. *IEEE power & energy magazine*, pages 18–28, January/February 2010.

[31] IEA Report for the Clean Energy Ministerial. Transforming global markets for clean energy products. www.iea.org/papers/2010/global_market_transformation.pdf, 2010.

[32] Sanford Friedenthal, Alan Moore, and Rick Steiner. *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann OMG Press, 2008.

[33] Sanford Friedenthal, Alan Moore, and Rick Steiner. *OMG Systems Modeling Language Tutorial*. INCOSE, September 2009.

[34] M. Galster and E. Bucherer. A taxonomy for identifying and specifying non-functional requirements in service-oriented development. In *IEEE Congress on Services*, Honolulu, HI, July 2008.

[35] C. Guille and G. Gross. Design of a conceptual framework for the v2g implementation. *IEEE Energy 2030 Conference*, 17-18 November 2008.

[36] D.G. Hart. Using ami to realize the smart grid. In *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, Pittsburgh, PA, July 2008.

[37] Maik Hollmann, VDE, and Jürgen Voss. Modeling of decentralized energy supply structures with "system dynamics". In *International Conference on Future Power Systems*, Amsterdam, NL, March 2006.

[38] ARTHUR D. LITTLE inc. Distributed generation:system interfaces, white paper.

[39] H. Kaindl, S. Kramer, and R. Kacsich. A case study of decomposing functional requirements using scenarios. In *Third International Conference on Requirements Engineering*, Colorado Springs, CO, August 1998.

[40] I. Kaitovic. Decentralized energy modelling. *ALaRI internal report*, pages 1 – 6, November 2008.

[41] I. Kaitovic. A quick practical guide on running simulations under ibm ration rhapsody. *ALaRI internal report*, pages 1 – 10, November 2009.

[42] Kenji Kobayashi. Forecasting supply and demand up to 2030 (iea report). http://www.iea.org/speech/2005/kk_melbourne.pdf, October 2009.

[43] Lessons learned from European research FP5 projects. Towards smart power networks. http://ec.europa.eu/research/energy/pdf/towards_smartpower_en.pdf, 2005.

[44] S. Lukovic, I. Kaitovic, and U. Bondi. Adopting system engineering methodology to virtual power systems design flow. In *CPSWEEK/GREEMBED, First Workshop on Green and Smart Embedded System Technology: Infrastructures, Methods and Tools*, Stockholm, Sweden, April 2010.

[45] S. Lukovic, I. Kaitovic, M. Mura, and U. Bondi. Functional requirements of embedded systems for monitoring and control structure of virtual power plants. In *IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems, EESMS*, Crema, Italy, September 2009.

[46] S. Lukovic, I. Kaitovic, M. Mura, and U. Bondi. Virtual power plant as a bridge between distributed energy resources and smart grid. In *43rd Hawaii International Conference on System Sciences (HICSS)*, Honolulu, HI, USA, January 2010.

[47] S. Lukovic, I. Kaitovic, M. Mura, U. Bondi, F. Kulic, and Popovic D. Functional model of virtual power plant (vpp). In *CIGRE 2010*, Paris, August 2010.

[48] G. Martin. Uml for embedded systems specification and design: motivation and overview. In *Design, Automation and Test in Europe Conference and Exhibition*, August 2002.

[49] Dragan Milicev. *Model-Driven Development with Executable UML*. Wiley Publishing, 2009.

[50] Brian Nolan, Barclay Brown, Laurent Balmelli, Tim Bohn, and Ueli Wahli. *Model Driven Systems Development with Rational Products*. IBM Redbooks, 2008.

[51] University of Delaware. Vehicle to grid technology. http://www.udel.edu/V2G/, September 2009.

[52] ADDRESS project. www.addressfp7.org/.

[53] DISPOWER project. www.dispower.org.

[54] E-Energy project. www.e-energy.de/en/.

[55] Fenix project. www.fenix-project.org/.

[56] INTEGRAL project. www.integral-eu.com.

[57] AlpEnergy project report. Virtual power systems white book. http://www.alpenergy.net/images/stories/White_Book_VPS.pdf, May 2009.

[58] Nist Draft Publication. Nist framework and roadmap for smart grid interoperability standarts 1.0. September 2009.

[59] D. Pudjianto, C. Ramsay, and G. Strbac. Virtual power plant and system integration of distributed energy resources. *Renewable Power Generation, IET*, pages 10–16, March 2007.

[60] H.B. Puttgen, P.R. MacGregor, and F.C. Lambert. Distributed generation: Semantic hype or the dawn of a new era? *IEEE power & energy magazine*, January/February 2003.

[61] Victorian Auditor-General's Report. Towards a "smart grid" - the roll-out of advanced metering infrastructure. download.audit.vic.gov.au/files/111109_AMI_Full_Report.pdf, November 2009.

[62] V. Sadana and Xiaoqing Frank Liu. Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements. In *Computer Software and Applications Conference*, Beijing, July 2007.

[63] Enrique Santacana, Gary Rackliffe, Le Tang, and Xiaoming Feng. Getting smart. *IEEE power & energy magazine*, pages 41–48, march/april 2010.

[64] Karlheinz Schwarz. Iec 61850, iec 61400-25, and iec 61970: Information models and information exchange for electric power systems. www.scc-online.de, 2004.

[65] Y. Serizawa, E. Ohba, and M. Kurono. Present and future ict infrastructures for a smarter grid in japan. In *Innovative Smart Grid Technologies (ISGT), 2010*, Gaithersburg, MD, January 2010.

[66] Eko Adhi Setiawan. *Concept and controllability of Virtual Power Plant, PhD thesis*. Kassel University press, 2007.

[67] SmartGridNews. The traditional grid. http://www.smartgridnews.com, January 2010.

[68] A. Tsadimas, M. Nikolaidou, and D. Anagnostopoulos. Handling non-functional requirements in information system architecture design. In *Software Engineering Advances*, Porto, October 2009.

[69] Tim Weilkiens. *Systems Engineering with SysML/UML: Modeling, Analysis, Design*. Morgan Kaufmann OMG Press, 2007.

[70] Tim Weilkiens. Compact sysml reference card. http://www.oose.de/downloads/sysml.overview.oose.pdf, 2008.

[71] DOE/NETL white book. Understanding the benefits of the smart grid. http://www.netl.doe.gov/smartgrid/referenceshelf/whitepapers/, June 2010.

Appendix A

Project files and source code

Virtual Power System model being presented has been fully developed under IBM Rational Rhapsody Developer for System Engineers using SysML profile. For simulation framework development, OMNeT++/C++ has been used. Both, Rhapsody project files and OMNeT++ source code are available by addressing request to the author. Once the model reaches a more mature version it will be downloadable from ALaRI platform as well.

Appendix B

Business plan

"If you're not a risk taker, you should get the hell out of business"
Ray Kroc

Business plan that describes an idea and analysis on how to start a new business that relies on usage of embedded devices is a mandatory part of master thesis for ALaRI Business Track students. Still, it is not required to directly relate this business plan to a major subject of the thesis.

Business plan that has been developed in scope of my studies has been done in collaboration with my friend and college Christianos Nikolaos. Since it is not related to nighter VPS or model-driven methodology it has been decided to present it not as a part of the thesis but as a separate document. Moreover, the document is publicly available trough ALaRI platform.