JOHANNES KEPLER
UNIVERSITY LINZ

Submitted by
Nada Ossama Ezzeldeen

Submitted at
Institute for Application Oriented Knowledge Processing

Supervisor
Prof. Josef Küng

June 2017

Connecting Small Hydro Power Stations for Decision Support

Master Thesis

to obtain the academic degree of

Master of Science

in the Master's Program

INTERNATIONALER UNIVERSITA TSLEHRGANG INFORMATICS: ENGINEERING & MANAGEMENT

JOHANNES KEPLER
UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Österreich
www.jku.at

Acknowledgment

Abstract

Contents

List of Figures

Chapter 1

Introduction

1.1 Reading Instructions

This thesis may present different interests for different readers. In this chapter, I will provide a guideline explaining what is covered in each chapter in order to facilitate browsing of the thesis and efficiently help every reader find the relevant information for him/her.

The first chapter presents the details and circumstances in which this thesis was created upon. The Problem statement of this thesis will be defined along with the motivation for solving that specific problem. Readers interested in a high level overview of the goal and the approach used for solving the specified problem, along with the original contribution brought to the existing system should refer to the *GoalAndApproach* and *OriginalContribution* sub chapters respectively.

The second chapter will discuss the most relevant concept of this thesis, namely decision support systems. They will be discussed and evaluated in terms of the foundations they are built on, their functionality, the Interfaces used for them, how they are implemented and their evaluation matrices and impact on decisions. This chapter will be most relevant for readers who would like to learn about decision support systems and understand the underlying concepts.

The third chapter will cover the Connect Hydro Project that my thesis aims to support and add to it. Connect Hydro proposes a system to connect small, private and independent hydro power plants through networked intelligent control system. In the chapter, I will also give an overview on the device they developed to collect sensor data from the power plants.

Chapter four will highlight in detail how a decision support system can bring advantage to the connect hydro project. In this chapter, I will also discuss what are the requirements for this proposed decision support system and describe the different inputs along with the expected outputs in addition to what should be the defined rules for such system. This is the chapter that my work will be based on.

The fifth chapter will cover the technical aspects of the implementation done to support this thesis. It will begin with describing the frameworks and technologies used for the implementation while explaining why they were used. Furthermore, each implemented aspect of the project will be explained in detail, namely the database model, the web portal, the data visualization and finally and most importantly the decision support system. This chapter might be of interest also for readers that want to find more details about the design and implementation of this system.

Chapter six will explain how the system implemented was evaluated, what matrices were used in its evaluation and the results. Readers interested in the results only will find this chapter the most informative for them.

The last chapter containing the conclusion and the future research will be most relevant for users interested in extending and improving the proposed system.

1.2 Foreword

Conventional energy sources based on oil, coal, and natural gas have proven to be highly effective drivers of economic progress especially with increasing improvements in the quality of life, industrialization of developing nations, and increase of the world population. However, this excessive fossil fuel consumption not only leads to an increase in the rate of diminishing fossil fuel reserves, but it also has a significant adverse impact on the environment, resulting in increased health risks and the threat of global climate change [21]. These traditional fossil fuel-based energy sources are facing increasing pressure on several environmental fronts. The increasing consumption of fossil fuel to meet the current energy demands causes alarm over the energy crisis and has generated a lot of interest in promoting renewable alternatives to meet the developing world's growing energy needs [43, 22]. Excessive use of fossil fuels has caused global warming by carbon dioxide; therefore, renewable promotion of clean energy is required [20]. An agreement called "Kyoto Protocol agreement" was made with the objective of monitoring the greenhouse emissions with an overall pollution prevention target [33]. Due to these facts, Research is directed towards utilizing already available renewable energy sources(RES) as well as finding new sources.

Renewable energy is energy that is generated from natural processes that are continuously replenished. This energy cannot be exhausted and can be used to provide sustainable energy services, based on the use of routinely available, indigenous resources. There is an enormous potential of renewable energy sources as they can in principle meet many times the world's energy demand [21]. It is becoming increasingly likely to transition to renewable-based energy systems like solar, wind or hydro power systems. Fossil fuel and renewable energy prices, social and environmental costs are heading in opposite directions. The cost of implementing renewable-based systems have dropped significantly in the past 30 years and continue to drop, while the price of oil and gas (Fossil fuel) continue to fluctuate [21]. Society is slowly moving towards seeking more sustainable production methods, waste minimization, reduced air pollution from vehicles, distributed energy generation, conservation of native forests, and reduction of greenhouse gas emissions. Changes towards environmental improvements are becoming more politically acceptable globally, especially in developed countries.

Renewable energy sources currently supply somewhere between 14% of world's total energy demand [33]. The supply is dominated by traditional biomass, mostly fuel wood used for cooking and heating, especially in developing countries in Africa, Asia and Latin America [21]. Large hydro power stations are considered a major contributer; with nearly 20 percent of the global electricity supply by them [33]. New renewable energy sources (solar energy, wind energy, modern bio-energy, geothermal energy, and small hydro power) are currently contributing about two percent. A number of scenario studies have investigated the potential contribution of renewables to global energy supplies, indicating that in the second half of the 21st century their contribution might range from the present figure of nearly 20 percent to more than 50 percent with the right policies in place [33]. RESs are also called alternative energy sources. The share of RESs is expected to increase very significantly (30–80% in 2100) [16]. Figure 1.1 shows the expected percentage for every type of renewable energy source currently and what they are expected to reach in 2040.

	2001	**2010**	**2020**	**2030**	**2040**
Total consumption (million tons oil equivalent)	10,038	10,549	11,425	12,352	13,310
Biomass	1080	1313	1791	2483	3271
Large hydro	22.7	266	309	341	358
Geothermal	43.2	86	186	333	493
Small hydro	9.5	19	49	106	189
Wind	4.7	44	266	542	688
Solar thermal	4.1	15	66	244	480
Photovoltaic	0.1	2	24	221	784
Solar thermal electricity	0.1	0.4	3	16	68
Marine (tidal/wave/ocean)	0.05	0.1	0.4	3	20
Total RES	1,365.5	1,745.5	2,964.4	4289	6351
Renewable energy source contribution (%)	13.6	16.6	23.6	34.7	47.7

Figure 1.1: Global renewable energy scenario by 2040 [33]

1.2.1 Types of Renewable Energy Sources

There exists different types of renewable energy sources. In this section we will discuss how each of them is harnessed and the different usages for the energy they produce.

- **Solar Power:** It is the energy coming from sunlight. It can be exist in two different ways. The First is solar-thermal power which is harnessed using mirrors or lenses to concentrate a large area of sunlight, or solar thermal energy (STE), onto a small area (i.e. a solar cell) which is then used for heating or domestic use. The second way involves solar cells (usually made from slices of crystalline silicon) that rely on the photovoltaic (PV) effect to absorb photons and convert them into electrons. Solar power is one of the most popular, and fast growing, sources of alternative energy.

- **Wind Power:** It is the use of air flow through wind turbines to mechanically power generators for electric power. It is considered very reliable and steady, as wind is consistent from year to year and does not diminish during peak hours of demand. Wind energy can be used to pump water or generate electricity, but it has a huge startup construction cost for constructing a wind farm.

- **Hydroelectric energy:** The kinetic energy of flowing rivers is captured and converted into hydroelectricity. It involves the usage of flowing water to turn turbines that in turn generate electricity. It is one of the renewable energy forms with the highest potential due to the availability of its creating resource; the water.

- **Bio energy:** This is the use of plant matter and animal waste to create electricity. When converted properly, it is a low-carbon source of energy with little pollution. Some of the more modern forms of biomass energy are methane generation and production of alcohol for automobile fuel and fueling electric power plants. It is very expensive as it has not advanced as quickly as other forms of renewable energy sources.

- **Geothermal power:** It is energy derived from the heat of the earth itself. This heat can be sourced close to the surface or from heated rock and reservoirs of hot water miles beneath our feet. Geothermal power plants harness these heat sources to generate electricity. On a much smaller scale, a geothermal heat pump system can leverage the constant temperature of the ground found just ten feet under the surface to help supply heat to a nearby building in the winter, or help cool it in the summer.

- **Tidal Power:** It is considered to be a potential source of renewable energy because tides are steady and predictable. Tidal power suffers from relatively high cost and limited availability of sites with sufficiently high tidal ranges or flow velocities which makes its applications limited.

The figure below explains the main usage of the different Renewable Energy sources. For Example: Hydro power is used for power generation, Solar power can be used for powering home appliances etc.

Energy source	Energy conversion and usage options
Hydropower	Power generation
Modern biomass	Heat and power generation, pyrolysis, gasification, digestion
Geothermal	Urban heating, power generation, hydrothermal, hot dry rock
Solar	Solar home system, solar dryers, solar cookers
Direct solar	Photovoltaic, thermal power generation, water heaters
Wind	Power generation, wind generators, windmills, water pumps
Wave	Numerous designs
Tidal	Barrage, tidal stream

Figure 1.2: Main renewable energy sources and their usage form [33]

1.3 Motivation

Renewable energy is the new trend that all governments are directing research into, simply because they are environment friendly and cheap. All researchers predict that the earth natural resources will run out and for the past 20 years have been trying to research new techniques to produce energy [29].

Small hydro power plants have a huge, untapped potential in most areas of the world and can make a significant contribution to future energy needs. It is largely dependent on already proven and developed technology. It has a considerable scope for development and optimization. In Austria, 9% of its power demand is supplied from small hydro power stations. They are of great significance for the security of supply and regional economy due to their decentralized character [29]. The current trend is that electricity trading prices are constantly going down and the government is directing research towards finding alternative energy and utilizing the already available ones.

1.4 Problem Statement

In countries, where many small rivers exist, the geography can be used to implement environment-friendly small hydro power plants for the generation of energy. The smaller such hydro power plants are, the higher is the impact of environmental incidents. Usually, there are more than one small hydro power plants located alongside one river, mostly operated by different owners. For more than 100 years electricity is being produced by hydro power plants on the Alm river in Upper Austria. Today 55 small and micro hydro power plants of more than 40 owners are operated on 48 kilometers [29]. To increase the overall power generating efficiency of all hydro power plants alongside the river, a good communication- and cooperating concept is needed.

The lack of good communication and cooperating is creating a problem leading to less efficient energy production, higher down time for the power plants and more maintenance. The problem can be broken into different aspects. The first would be collecting the sensor data from the different power plants; given that each power plant uses different equipment; a method should be developed to collect these data and uniform them. The second aspect would be to use the data to help the power plant owners make informed decisions, this will lead to efficient energy production and less down time. The third aspect would be to perform the actions directly on the power plants instead of informing the owner to do them, this will result in a fully automated system that doesn't need any human interaction.

In this thesis, we will tackle the second phase. We will work under the assumption that the data is collected and unified. The aim is to provide the owners with informed decisions on how to operate their power plants. A formal description of our thesis problem is defined as follows: "Given sensor data from small, private & independent hydro power plants, find decisions using the data coming from their sensors and external sources, such that the overall energy production is increased and the down time is minimized."

1.5 Goal and Approach

As mentioned in the previous section, our goal is to provide the owners of small hydro power plants with informed decision based on the sensor data collected from all the power plants such that the overall energy production is increased and the down time is minimized. In order to solve the problem specified, we propose a system prototype implementation of a decision support system for several small, private and independent hydro power plants. This system should be able to manage general events that could occur and affect all power plants as well as handle specific rules defined for every power plant by the owner.

The focus of this thesis will be split into two parts. The first is performing sound research on decision support systems, their types and techniques and processes used to develop them. The second part is implementing a web portal with an embedded simple decision support system. The Decision support system should be a combination of a knowledge-driven and data-driven DSS. The second part should include a proposed database schema for the power plants sensor data as well.

1.6 Original Contribution

Currently there exists no system connecting small hydro power plants. Owners of said power plants don't communicate with each other resulting in a need for an early warning system (Decision support system). The main goal of the Decision support system is to receive data from previous power plants along the same river and direct the owner to do some action in response to the data received.

In this thesis, a proof of concept will be developed to prove that a decision support can help small hydro power plants work efficiently and in synchronization with their neighbors to optimize their energy production. We will investigate the advantages of combining different decision support techniques to achieve the main goal as well as providing a simple prototype that can be used as a base for future research.

1.7 Outline of the Thesis

The first chapter of the thesis makes a brief presentation of the concepts related with the project and makes the reader familiar with the aims and reasons that justify the selection of this topic. The background of the problem and a short description of the social and economic context are given in order to integrate the problem into the real world.

The second chapter will discuss the concept of Decision support systems. It will investigate the foundations they are built on, their functionality, the Interfaces used for them, how they are implemented and their evaluation matrices and impact on decisions.

The third chapter will cover the Small hydro power stations as well as the Connect Hydro Project. Small hydro power stations will be discussed in terms of their different types, advantages and disadvantages. Secondly, we will discuss the different control strategies that can be used to control small hydro power plants. Finally the Connect Hydro project will be introduced and the work related to it will be discussed.

Chapter four will highlight in detail how a decision support system can bring advantage to the connect hydro project and how the prototype system was implemented and using which techniques.

The fifth chapter will cover the technical aspects of the implementation done to support this thesis. It will begin with describing the frameworks and technologies used for the implementation while explaining why they were used. Furthermore, each implemented aspect of the project will be explained in detail, namely the database model, the web portal, the data visualization and finally and most importantly the decision support system.

Chapter six will explain how the prototype implemented was evaluated, what matrices were used in its evaluation and the results.

The last chapter contains the conclusion and the future research that can help in improving the proposed system.

Chapter 2

Decision Support Systems

A decision support system (DSS) can be defined as a computer information system that analyzes complex data and solves problems by either supporting decision makers make informed decisions or suggesting decisions/actions for them [39]. DSSs are a sub-collection of information management systems that help planners, analyzers and managers in the decision making process [25]. A decision support system may present information graphically and may include an expert system or artificial intelligence (AI). It may be aimed at business executives or some other group of knowledge workers.

Much research and practical design effort has been conducted in each of the domain that comprise a Classic DSS tool design. These areas are [39]:

- Sophisticated database management capabilities with access to internal and external data, information, and knowledge
- Powerful modeling functions accessed by a model management system
- Powerful yet simple user interface designs that enable interactive queries, reporting, and graphing functions.

2.1 History of Decision support systems

DSS concept was first introduced in the early 1960s [35]. A model-oriented DSS or management decision systems was introduced as a new type of information system.Following that, the concept of decision support evolved into two main areas of research according to the two DSS pioneers, Peter Keen and Charles Stabell, the first being "The Theoretical studies of organizational decision making" done at the Carnegie Institute of Technology during the late 1950s and early '60s and the second is "The Technical work on interactive computer systems", mainly carried out at the Massachusetts Institute of Technology in the 1960s [35, 39].

In 1971, a ground breaking book "Management Decision Systems: Computer-Based Support for Decision Making" written by Michael S. Scott Morton was published. The book discussed how creating analytical models along with computers can help make key decisions. The book also highlighted an experiment where managers used a Management Decision System which was considered the first test of a model-driven decision support system [35].

By 1975, J. D. C. Little defined the four main criteria for designing and evaluating models and systems to support management decision making which are still considered relevant today. They include: robustness, ease of control, simplicity, and completeness of relevant detail [35]. In the early 1990's, some desktop online analytical processing (OLAP) tools were introduced and DSS technology shifted from mainframe-based DSS to client/server-based DSS and eventually to web-based DSS [12]. As a result of that change, Enterprise data warehouses were completed and data management and decision support companies updated their infrastructure to support the change in DSS technology.

According to Powell [34], DBMS(Database Management systems) vendors "recognized that decision support was different from OLTP(Online transaction processing) and started implementing real OLAP capabilities into their databases". In 1995, Researchers were directed towards the development of Web-Based Group Decision Support Systems(GDSS), Web access to data warehouses in addition to Web-Based and ModelDriven DSS.

According to Power [35], in the early 2000's, portals were introduced that combined information portals, knowledge management, business intelligence, and communications-driven DSS in an integrated Web environment called "Enterprise knowledge portals". This solidified the notion that the Web is the best suited platform for building future DSS.

2.2 Concept of DSS

The original DSS concept was built by combining some categories of management activity and decision problem types according to Gorry and Scott Morton [19]. The management activities were the set of decisions defined by the management to serve a specific purpose which could be strategic planning (decisions that contribute to the overall mission and goals), management control (decisions guiding the organization to achieve the specified goals), or operational control (decisions directing specific everyday tasks). The decision problem types were categorized into structured, semi-structured or unstructured problems. Structured problem types are problems that are repetitive and easily solved, they are usually solved using a computer program. Unstructured problems are problems that are difficult to solve using a computer program and relies on the decision maker's judgment [39].

According to Gorry and Scott Morton the characteristics of information needs and models differ in a DSS environment. The unstructured nature of information needs in a DSS situations forces us to search for different kinds of database systems than those for operational environments. Flexible query languages and relational databases are needed. Similarly, the need for flexible modeling environments was arisen to handle the problem of unstructured decision process, such as those in spreadsheet packages [39].

Fig. 2.1 explains a generic model that was used and implemented in a DSS system for a decision making process where the focus is on the analysis of the problem and development of the model. It starts by recognizing a problem, then defining said problem in terms that contribute towards the model creation. Once the problem is defined, a model is designed and some alternatives are developed to find a solutions. Following that a solution is chosen and the DSS system implements it. This Figure explains a the process of a simple structured clear decision process, however, no decision process is that defined which leads to a lot of back and forth between the phases and overlapping to earlier stages as the problem becomes more defined or the solutions fail [39].

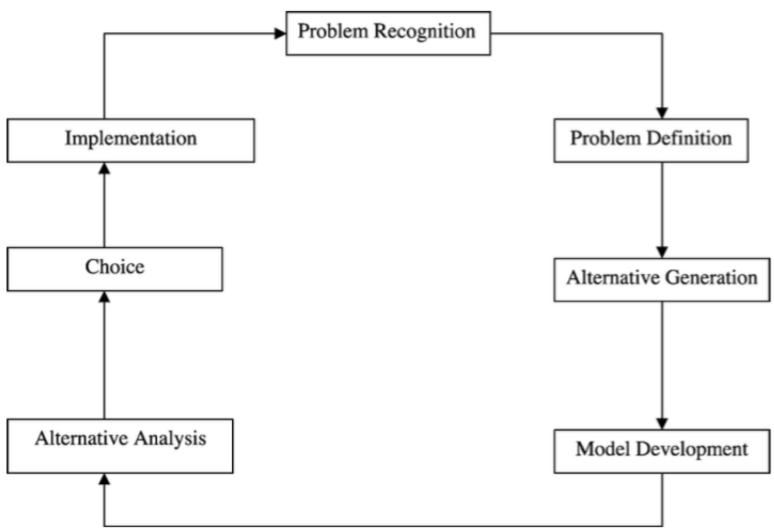

Figure 2.1: The DSS decision-making process [39]

2.3 Types of DSS

There exists a number of Decision Support Systems. These can be categorized into five types:

- Communication-driven DSS
- Data-driven DSS
- Document-driven DSS
- Knowledge-driven DSS
- Model-driven DSS

We will explore each type briefly in the following sections.

2.3.1 Communication-driven DSS

A communication-driven DSS is a type of DSS that focuses on communication as well as help two or more users collaborate, share information, co-ordinate their activities and support shared decision-making [36]. The most common technology used to deploy this type of DSS is a web or client server. A few example of a communication-driven DSS include chats and instant messaging softwares, online collaboration and net-meeting systems or a simple bulletin board or threaded email.

Communications-Driven DSS software should include at least one of the following characteristics:

- Enables communication between groups of people
- Facilitates information sharing
- Supports collaboration and coordination between people
- Supports group decision tasks

Group Decision Support Systems(GDSS) is a hybrid type of DSS. It is based on communication-driven DSS. Using various software tools, multiple users can work collaboratively in groupwork [36].
Examples of group support tools are:

- Audio Conferencing
- Bulletin boards and web-conferencing
- Document Sharing

- Electronic Mail
- Computer Supported face-to-face meeting software
- Interactive Video

2.3.2 Data-driven DSS

Data-driven DSS is a type of DSS that is able to process huge amounts of data from different sources and store them in a data warehouse system. Data Driven DSS uses on-line analytical processing(OLAP) and Data mining techniques to extract the needed data which will be discussed in the section *ToolsToDevelopDSS*. There exists two special purpose Data-driven DSS, they are: Executive Information Systems (EIS) and Geographic Information Systems (GIS). An example of a data-driven DSS is computer-based databases that have a query system.

Executive Information Systems (EIS) are computerized systems intended to provide current and appropriate information to support executive decision making for managers using a networked workstation. They focus on graphical displays and on offering an easy to use interface that presents information from the corporate database. EIS offer strong reporting and drill-down capabilities [36].

A Geographic Information System (GIS) or Spatial DSS is a support system that represents data using maps. It helps people access, display and analyze data that have geographic content and meaning [36].

2.3.3 Document-driven DSS

A relatively new field in Decision Support is Document-Driven DSS. Its focus is on the retrieval and management of unstructured documents. Documents can be Oral, written or video. They help a decision maker by keeping track of knowledge represented as documents that can affect the decisions [38]. Examples of oral documents are conversations that are transcribed; video can be news clips, or television commercials; written documents can be written reports, catalogs, letters from customers, memos, and even e-mail [36].

2.3.4 Knowledge-driven DSS

Knowledge-Driven DSS can suggest or recommend actions to managers. It contains of specialized problem solving expertise also known as a knowledge base. The knowledge base comprises of rules, facts and procedures about a particular domain. The knowledge base also provides understanding of problems within that domain, and "skill" for solving some of these problems. A related technique used in knowledge-driven DSS is Data Mining, discussed in the section *ToolsToDevelopDSS*. Intelligent Decision Support methods are used to build Knowledge-Driven DSS [38].

2.3.5 Model-driven DSS

Model-Driven DSS (MDS) are a standalone systems that performs modeling of unstructured problems with an easy to use user interface. The most basic modeling functionality; the what-if model ;can be achieved using a simple statistical and analytical tool. There can exist a hybrid DSS system that combines the modeling functionality of the MDS and the complex analysis of data of an OLAP system [36]. In general, model-driven DSS uses complex financial models, simulation models, optimization models or multi-criteria models to provide decision support [36]. Data and parameters are provided by decision makers to the Model-driven DSS to aid decision makers in analyzing a situation. Very large databases are not needed in Model-driven DSS as they are not usually data intensive [36]. Model-Driven DSS can also be called model-oriented or model-based decision support systems.

Model-driven decision support process can be divided into three stages:

- Formulation : A model is generated in a form that can be accepted in the model solver [39].

- Solution : Finding the solution of the model in an algorithm form [39].

- Analysis : Analyze and interpret 'what-if' model solution or a set of solutions [39].

Initially optimization of Model-driven DSS focused on optimizing the solution algorithm, but later the focus shifted to also finding better techniques to formulate and analyze the functions to support the DSS [39].

Tools used in Model-driven DSS include [28]:

- **Decision Analysis tools** - help decision makers decompose and structure problems. These tools aim to help a user apply models like decision trees, multi-attribute utility models, Bayesian models, Analytical Hierarchy Process (AHP), and related models [36].

- **Forecasting Support System** - A computer-based system that supports users in making and evaluating forecasts. Users can analyze a time series of data [36].

- **Linear Programming** - A mathematical model for optimal solution of resource allocation problems [36].

- **Simulation** - A technique for conducting one or more experiments that test various outcomes resulting from a quantitative model of a system [36].

2.4 Tools to Develop DSS

As Mentioned in the previous section, there exists a number of tools that are used to support the decision making process. In this section, I will explain briefly each of them.

2.4.1 Data Warehouse Systems

Data warehouse systems are systems that allow the manipulation of data by using either computerized tools customized for a specific task or general tools and operators that provide a certain functionality. A Data Warehouse is basically a database that is designed to support decision making in organizations. Data warehouses are structured to contain large amounts of data and handle rapid online queries and managerial summaries. According to Power [35], Data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data that supports the management's decision making process.

2.4.2 On-line Analytical Processing

On-line Analytical Processing (OLAP) is a technique used to support the decision support functionality especially in Data-driven DSS. It is linked to analysis of large collections of historical data [36]. OLAP software is used for manipulating data from a variety of sources that has been stored in a static data warehouse. The software can create various views and representations of the data [36]. Three main features should be available in a software product for it to be considered an OLAP application. They are:

- Multidimensional views of data

- Complex calculations

- Time oriented processing capabilities

2.4.3 Data Mining

Data Mining helps in extracting useful information by finding patterns or rules from existing data to produce data content relationships. It is based on Artificial Intelligence techniques combined with statistical tools. This information is then used to predict future trends and behaviors which also makes it a very important technique when implementing a data-driven DSS.

2.4.4 Web-based DSS

Web-based DSS can be defined as a computerized system that provides an easier and less costly way to deliver decision support information or decision support tools to a manager, business analyst or a decision maker using a Web browser. As shown in Figure 2.2, any Type of DSS can either be web-based and implemented using web technologies or local based(LAN-Based). However, the web opened a gateway that allowed for the implementation of DSS with larger scopes, access to more users and most importantly rapid access to "best practices" analysis and decision-making frameworks. The result would be well-designed DSS in a company. Using a Web infrastructure for building DSS promotes more consistent decision making on repetitive tasks [37].

	Technology	
DSS Types	LAN-Based	Web-Based
Communications-Driven and GDSS	Narrow scope	Global scope
Data-Driven	Thick-client	Thin-Client
Document-Driven	Limited, .doc,.xls	Also HTML, Search engines
Knowledge-Driven	Stand-alone PC	Shared rules
Model-Driven	Single user	Multiple users

Figure 2.2: Decision Support Systems using Web Technologies [37]

2.5 DSS Design

DSS has components and phases of development, like any other software system. No matter what kind of decision support system is being developed, there should exist four components:

- Input: Input type that will be used in the analysis should be defined.
- User Knowledge/Expertise: will user knowledge be used as part of the input as well?
- Output: The output is specific or generic?
- Decisions: The system should suggest actions, analyze data and different actions or do the action to correspond to the output.

Once these four components are considered and clear enough we can proceed to the next phase in the design and development process which is Analyzing the business decision making process.

2.5.1 Analyzing Business Decision Making Process

A key consideration in designing a decision support system is to understand the context in which the decisions will be made. These considerations include the decision type, nature of problem, people involved and eventually the decision making context [13]. An analyst is responsible for defining all the key components and creating a clear understanding of what will be required of the system.

Decisions can either be Strategic decisions, Operational decisions or Managerial decisions. It is very important to understand which type of decision should the DSS being developed support.

Strategic decisions are non-repetitive and require a lot of time to be arrived at. They involve careful analysis of the situation and consequences. Some examples of strategic decisions: evaluation of an investment proposal, decisions related to mergers and acquisitions, resource allocations, fund raising, etc [13]. Operational decisions can either be long term decisions that impact the business functionality and help the company realize their mission or short term decisions that impact day-to-day activities [13]. Managerial decisions are usually decisions taken by top management. Examples of Managerial decisions include resource allocation, talent management, research and development, new product introduction, withdraw or revamp old products [13].

After defining the decision types to be supported, the nature of the problem should be defined. Problems can either be repetitive, non-repetitive, structured or unstructured and depending on the nature of the problem we can define what type of analysis will be required by the system and if any human interaction will be needed. Another factor to be taken into consideration is weather decisions are to be taken individually or within a group.

2.5.2 Decision Making Process

After analyzing the Decision making from a business perspective, the following step would be to start the decision making process. The process starts by the following steps:

1. Defining the Problem: It begins with recognizing that a problem exists, sets a base on which assumptions can be built, collect and analyze data and finally evaluate alternatives [13]. A problem exists when:

- The expected output and the delivered output are different
- There's a divergence from the normal expected results
- An action taken is not explainable

2. Identifying Decision Maker: A problem should be sent to the right person depending on its nature [13].

3. Gathering Information : A DSS can process tons of data in just few seconds thus helping the concerned person with collecting data and identifying the factors influencing the situation [13].

4. Evaluating Alternatives and Deciding: All possible courses of action are evaluated and the most suitable action is determined, by assessing the pros and cons of each alternative. A DSS helps in justifying a particular choice [13].

5. Implementation and Follow-up: Once the decision is taken, It's time to implement the decisions. Decisions proposed should be monitored in order to determine weather it was helpful in achieving the objectives or not. If not the entire process must be repeated.

2.6 Decision Support Implementation and Development

Once There is a clear understanding of the system requirements and all the processes are defined and the design phase is complete, the development process begins. It starts with choosing a system development approach, followed by designing the user interface. The System Architecture, networking and security will be addressed in the *DecisionSupportArchitecture* and *DecisionSupportNetworkingAndSecurity* sections [13].

2.6.1 Choosing a System Development Approach

There are three common development approaches as shown below in Figure 2.3. Each of them will be discussed in detail.

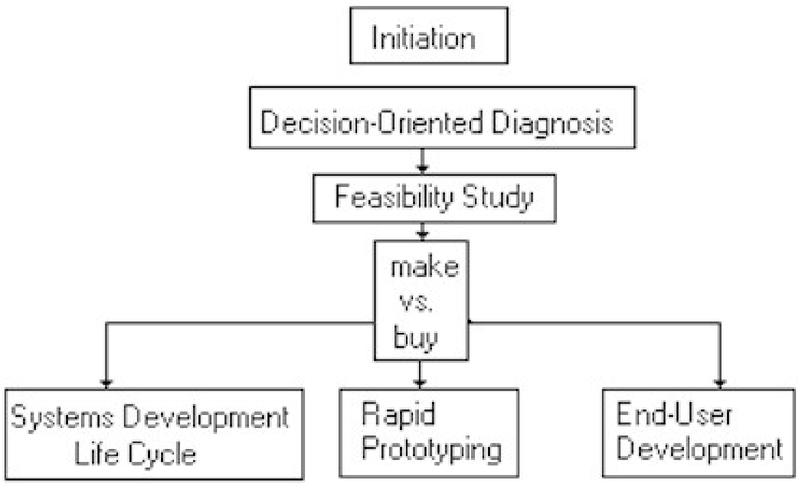

Figure 2.3: DSS Development Approaches [36]

- **SDLC - System Development Life Cycle Approach**
  The SDLC is a sequential, structured and a standardized process for a system development. It starts with identifying the system objectives (needs of end users) and goes through various stages shown in figure 2.4 [27], including
  * System analysis (technical components required)
  * System design (architecture)
  * Development (programming)
  * Testing (errors and bug fixing)
  * Implementation & Use(execution in the organization)
  * Evaluation (verification of functions and capabilities)
  * Modification (adjustments required)

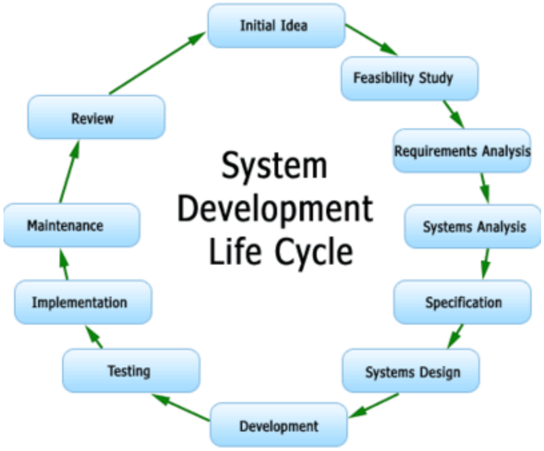

Figure 2.4: System Development Life Cycle Approach [27]

SDLC is the most commonly used and most rigid system development approach. In complex situations, it becomes difficult to use this approach, as the requirements of users are constantly changing. It doesn't promote recurring development and testing [13].

- **Rapid Prototyping Approach** Rapid Prototyping promotes a faster system development. It combines the effort of the decision makers and the analyst in charting the specific requirements. The decision maker uses general terms, the analyst uses DMS (database management system) to support rapid development of the application [13]. Rapid prototyping goes through the following steps also shown in figure 2.5:
  * Identifying objectives/ user requirements
  * Developing the first model
  * Evaluating the first model, identifying adjustments required and modification
  * Testing the developed DSS.
  * Go back to evaluation and modification, if needed
  * Implementing

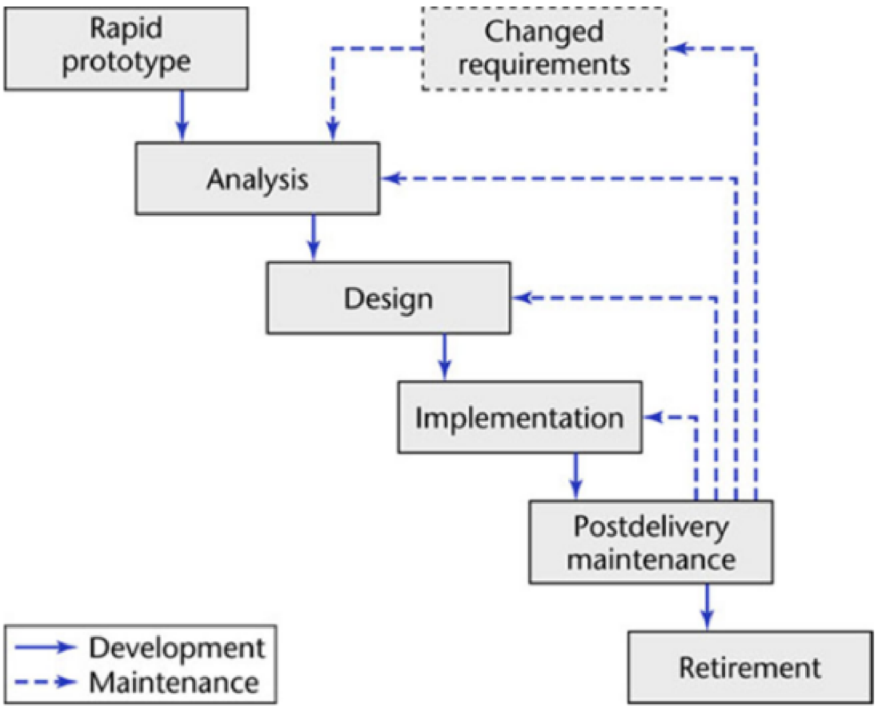

Figure 2.5: Rapid Prototyping Approach [4]

Evaluation and modification at a rapid pace is the core concept behind the Rapid Prototyping Approach as communication lines are always open. This approach is better suited than SDLC in complex situations.

- **End-User DSS Development Approach** Designing and development a software system depending on the specific or individual needs of the decision maker is the basis of the End-User development approach. The Decision maker customizes the system on his own. This approach is rarely used as usually the decision maker lacks technical expertise and chooses inappropriate software [13].

2.6.2 Designing User Interface

A Decision support system relies heavily on its design. If its user friendly and easy to use then chances are that this DSS will be used. Although the technical expertise of analysts, designers and developers is crucial, needs of DSS users must be evaluated and met at every step of the process. The right user interface design approach is the first step in developing an efficient decision support system [13].

2.6.2.1 ROMC Design Approach

ROMC is a systematic approach for developing large-scale DSS, especially user interfaces. The approach involves designing Representation, Operations, Memory Aids, and Control mechanisms. It is user-oriented approach for stating system performance requirements. It was originally presented by Sprague and Carlson in 1982 [2].

Representation: It involves presenting information or results, in a structured way. All decision making activities in an organization take place in a certain environment or context. The representation in addition to the decision making context, provide a way to communicate information to the decision maker or user of the system about the situation [2, 13]. It provides a platform to the decision makers supported by information to help them interpret DSS outputs. It can be in the form of a table, graph, map, chart or a text document and each value on a map or a table communicates decision making context [2, 13].

Operation: Specific tasks that can be performed by a decision maker with a DSS. Example DSS operations include tracking market trends, carrying analytics or suggesting alternatives or performing all the functions [13].

Memory Aid: A data warehouse is the memory aid for a DSS. A DSS must give users a link to data warehouse. In addition, links and command shortcuts or sequences can be supplied to help users control a decision support system [13].

Control Mechanism: Allows users to effectively use representations, operations and memory aids [13].

2.6.2.2 UI design success Factors

In order to evaluate a UI design success. The following factors are considered:
Execution Time for a command given and action performed should be minimized.
Versatility of a decision support system should be flexible enough to integrate new tasks if needed.
Adaptability of a decision support system needs to remember the user's habits and adapt to them.
Learning Time should be reduced for the system.
Uniformity of Command/theme should be maintained throughout the system
Quality of Help should be available to users. The system should offer self-help manuals both online and offline.
Memory Load; Avoid too many numbers and statistical information on 1 screen. Users don't want to remember numbers

Ease of Recall; The user can remember how to use the system quickly after not using it for some time.
Fatigue; simple designs should be maintained to avoid Mental fatigue.
Errors should be managed for possible error-producing situations that might happen.

2.7 Decision Support Architecture

DSS Architecture requires full understanding of how a user will interact with the system and how the information will flow from one point to another [13].

There are four fundamental components of DSS architecture:

- User Interface : Discussed in *DesigningUserInterface*
- Database
- Model (context or situation representation)
- Knowledge

Database: A DSS accesses information directly from a database. The system architecture scheme focuses on the type of database required for a particular decision making system model, Who's responsible for different types of databases and how to maintain accuracy and security of database [13].

Model: This component of the DSS architecture handles 2 components, the DSS model and DSS model management system. A model is a representation of a context, a situation or an event that carries out some type of data analysis needed for the decision making process. A DSS model management system stores and maintains DSS models [13].

Knowledge: Information about data relationship is represented in the knowledge. This knowledge is managed by the DSS architecture and provides decision makers with alternative solutions to a problem when needed [13].

2.8 Decision Support Networking and Security

DSS Network needs to define how hardware is organized, how data is distributed throughout the system how DSS components are connected and whether the information is fed/accessed using Internet, Extranet or Intranet [13]. DSS Networking is all about connection between the components – software and hardware.

A network is defined as an assortment or a group of computers that are connected with each other or in a specific way, in order to communicate with each other. This connection facilitates the sharing of information among the connected computer systems [13].

Resource Sharing
The computer network's main objective is to share information. The most common technology for connection and resource sharing is LAN (Local Area Network). It serves hosts within a restricted geographical area. WAN (Wide Area Network) is another technology for resource sharing. The difference between LAN and Wan is that the latter is much larger and connects a group of LANs [13].

Resource Connection
TCP/IP (Transmission Control Protocol/ Internet Protocol) is a set of standard networking protocols, to enable computer systems to communicate with each other. It defines the rules and formats for the diffusion and reception of information or resources. The TCP sends data between programs using IP (Internet Protocol). It assigns a unique IP address to each workstation and sends information from one host to another in the form of packets [13].

Constant presence and cost effectiveness of the Internet make it the best way to send Information or data. One aspect that should always be taken into consideration is that data have to transferred through a secured connection to maintain security. Security related concerns are discussed in the next section.

2.8.1 Security

As a decision support system contains secret or classified information, it needs to be 100% safe and secure. It's also necessary for safeguarding employee and customer data. The Process for Addressing Security Issues begins with:

Identifying security needs: DSS users and analysts must brainstorm to identify security needs and evaluate potential threats.

Determining how important security is for your DSS.

Remedying problems: Fix the problems found that affect the DSS security. The solutions may be in the form of [13]:

* Strengthened password
* User education
* Firewalls
* Enhanced privacy
* Logging and use statistics

Implementing solutions and observing their impact: the decided solutions are im-

plemented and observed.
There may be some security holes at any given point. DSS users and analysts must Keep track of them and change the passwords and strengthen firewalls on regular basis.

2.9 Evaluation and Impact

It is difficult to determine if a DSS is successful. Therefore, some criteria have to be evaluated and decided upon them how successful the system really is. Figure 2.6 shows a few examples of how various features of a DSS can be evaluated [30]. One aspect in evaluation of a DSS can be user satisfaction. However, users may have poor introspection or (if they are not experienced in using a DSS) may not recognize good advice and may dislike being corrected by a computer system, so while it should be taken into consideration, it shouldn't be a deciding factor [30].

Various features and criteria used for the evaluation of DSS

Subject of validation	Examples of measurements
(DSS) Development process	Involvement of future users in early development phases, appropriately defined system requirements, evolutionary system development, clear definition of beneficiaries
DSS components	Precision of models, quality of data, user interface, reporting system to choice of suitable technology and management of data, complexity of DSS and data inputs
Decision process	Appropriateness of logical process followed when using DSS, number of alternatives explored by DSS, internal communication, correspondence to and appropriateness for decision organisation
Decision output	Quantification profit/loss from DSS usage, consensus achieved among decision-makers, savings of time or other resources through DSS usage, contribution to organisational efficiency, consistency of solution
User satisfaction	Degree of confidence in results derived by DSS, acceptance (willingness to change current management methods), improvement of personal efficiency, correspondence of DSS output with decision-making style, users' understanding of implemented models

Figure 2.6: DSS Process Evaluation [30]

In addition to evaluating various DSS features, there exists some tools that can evaluate a DSS, Some of them include:

- **Cost-Benefit Analysis:** Determines if a DSS is a good investment or not. This tool is used to compare the total benefits that a system is expected to produce vs the total cost of development and implementation [13].

- **Incremental Value Analysis:** The process focuses on the value offered by a proposed DSS rather than the cost incurred on it [13].

- **Qualitative Benefits Scenario Approach:** This method aims to determine if the proposed DSS will maintain the same quality and capabilities in future scenarios [13].

- **Research and Development Options Approach:** This approach aims to determine the cost of keeping the DSS flexible for future enhancements or future DSS [13].

- **Scoring Approach:** The approach separates the business and technical validation and considers other benefits of a DSS that were not considered during the analysis phase. It assigns points to each criterion/benefit upon reflecting on how well it satisfies a given factor.
  Business validation involves assessment of strategic alignment, management information support, competitive advantage and organizational risk [13].
  Technical justification involves examining technical uncertainty, strategic systems architecture and system infrastructure risk [13].

Chapter 3

Connect Hydro Project

In this chapter, We will discuss what are hydro power plants, how they work and their different sizes, their electricity generating methods(types) as well as their advantages and disadvantages. Also, We will discuss the control strategies that can be used for hydro power plants. Connect Hydro Project will be discussed in terms of its different aspects and proposed concept. And finally we will give an overview of the proposed implementation for the Connect Hydro Project.

3.1 Small Hydro Power Plants

Hydro power plants make use of the the power of moving water to produce electricity or mechanical energy. There exists two key factors for effective hydro power generation, they are "head" and "flow." "Head" refers to the height over which the water falls, while "flow" refers to the volume of water per unit time [8]. To maximize energy production, both head and flow should be high. In other words, the larger volume of water flowing over a steep gradient the greater amount of energy is generated. In small hydro power stations, it is important that a proper height of water fall is obtained naturally without building elaborate and expensive facilities [8, 23]. A well designed small hydro power system can blend with its surroundings and have minimal negative environmental impacts [8, 23].

Hydroelectricity is a term that refers to electricity generated by hydro power plants. The electricity produced may be used directly, stored in batteries, or inverted to produce utility-quality electricity. Hydro power plants generate Hydroelectricity by directing water through a turbine, which in turn drives an electric generator. It is the most widely used form of renewable energy, accounting for 16 percent of global electricity generation – 3,427 terawatt-hours of electricity production [23].

Hydro Power technology is extremely robust (systems can last for 50 years or more with little maintenance) and is also one of the most environmentally benign energy technologies available [32]. Figure 3.1 shows an example of a small hydro power plant. In the following sections we will discuss the different types of hydro power stations and how they generate electricity as well as the advantages and disadvantages of small hydro power plants.

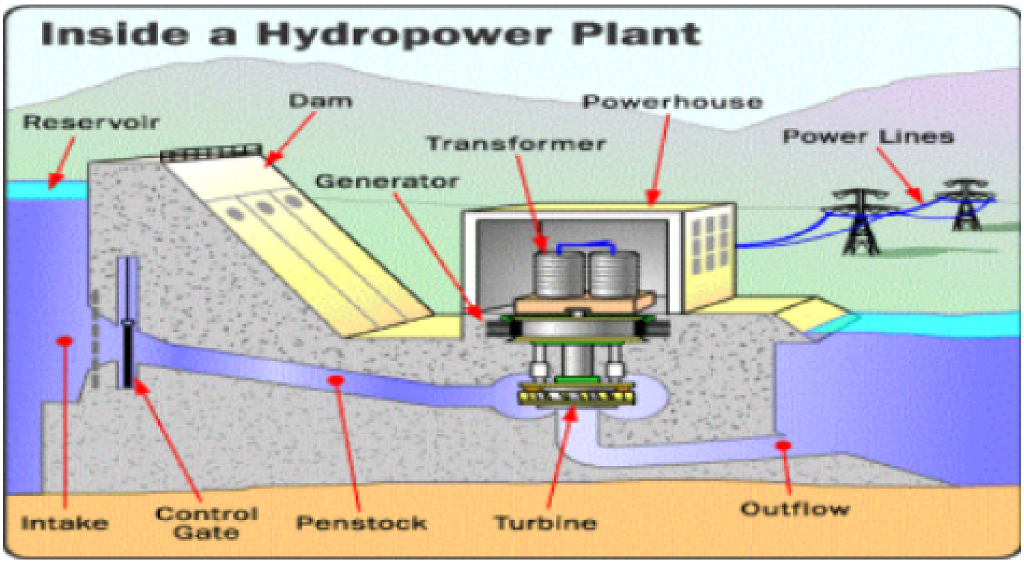

Figure 3.1: General Power Plant Structure [41]

3.1.1 Hydro Power Classification

The classification of Hydro power plants is achieved according to their average energy output, expressed in megawatts [41]. As shown in Figure 3.2, large scale hydro power plants produce more than 100 MW, Medium Scale hydro power plants produce energy between 10-100 MW, small hydro power plants produce less than 10 MW [41, 29]. Based on energy production capacity, small-scale hydro power production is broken into four size categories of pico- (<5 kilowatts), micro- (5-100 kW), mini- 100 kW-1 MW), and small (1-10 MW) [41].

There doesn't exists a common consensus among countries about the classifications of hydro power plants. For instance, some European Union countries like Portugal, Spain, Ireland, Greece and Belgium accept 10 MW as the upper limit for small-scale hydro power installed capacity, while others place the maximum capacity from 3 to 1.5 MW. Outside the EU, this limit can be much higher, as in the USA (30 MW) and India (25 MW) [41].

Type	Power Output	Applicability
Large	> 100 MW	Large urban population centres
Medium	10 - 100 MW	Medium urban population centres
Small	1 - 10MW	Small communities with possibility to supply electricity to regional grid.
Mini	100 kW - 1MW	Small factory or isolated communities.
Micro	5 - 100kW	Small isolated communities.
Pico	<5kW	1 - 2 houses.

Figure 3.2: Power Plant Size Classification [23]

3.1.2 Advantages of Small Hydro Power Plants

Hydro electric energy is a renewable electrical energy source. Hydro Power Plants don't produce any heat or toxic gases and therefore are considered non-polluting. They have a low operating, maintenance cost as well as no fuel cost which makes it inflation proof. They offer reliable and flexible operation with a long life. Many existing stations have been in operation for more than half a century and are still operating efficiently. Hydro power station produce energy with an efficiency of over 90%, it is the most efficient of energy conversion technologies. Finally, Hydro power offers a quick means of responding to changes in load demand or due to certain events [8].

3.1.3 Disadvantages of Small Hydro Power Plants

One of the drawbacks of small hydro power stations are their limitations in regard of their placement. They need to be in a location that is near the electrical power grid so that the electricity generated can be used and taken advantage of. Small power plants are limited by the river/stream water flow, they have a maximum capacity which cannot be exceeded even with high water flow. The energy supply is also disturbed by changing seasons; Yearly planning must be done to ensure constant power generation as they must be shutdown in the case of low water level [41, 29]. They are also susceptible to have problems with suspended loads (foilage, branches, waste) which requires flushing; the process of removing foilage from water by the trash racks before entering the turbine [29].

3.1.4 Small Hydro Power Plants Types

- **Conventional (dams):** Hydro electric power comes from the potential energy of water saved before a dam which is then used by the water turbine and generator to generate electricity. The electricity output relies on the volume and the difference in height between the dam and the water's outflow also known as the "Head". The higher the head, the higher the amount of potential energy in the water, they are proportional. A large pipe (the "penstock") delivers water to the turbine [41].

- **Pumped-storage:** This method produces electricity by moving water between reservoirs at different levels. At times of low energy demand, excess energy is used to pump water back into higher reservoirs. Whereas at times of higher demand, water is moved back into the lower reservoir through a turbine [41]. Pumped-storage are the most commercially implemented means of large-scale grid energy storage and improve the daily capacity factor of the generation system [41].

- **Run-of-the-river:** In run-of-river systems, river water is diverted by a weir through an opening in the river side (the 'intake') into a channel [23]. A sand trap is built in the channel to remove sand and leaves from the water. Water in directed into a small reservoir/tank known as the 'forebay' from where it is directed on to the turbines through a closed pipe known as the 'penstock' [23]. The penstock's main functionality is to direct the water in a constant stream to the turbine at a lower level. Power generated by the turning shaft of the turbine can be used to rotate a mechanical device (such as a grinding mill, oil expeller, or wood lathe), or to operate an electricity generator [23]. When electricity is generated, the 'power house' where the generator is located, transfers the electricity to a step-up 'transformer' which is then transmitted to the electricity grid. Water is returned back to the river after the electricity is produced by the turbine [23]. Figure 3.3 shows a diagram of how a run-of-the-river power plant is built.

  Most small hydro power plants are "run-of-the-river" types. The turbines are turned on to generate electricity only when the water level at the river/stream is high enough.

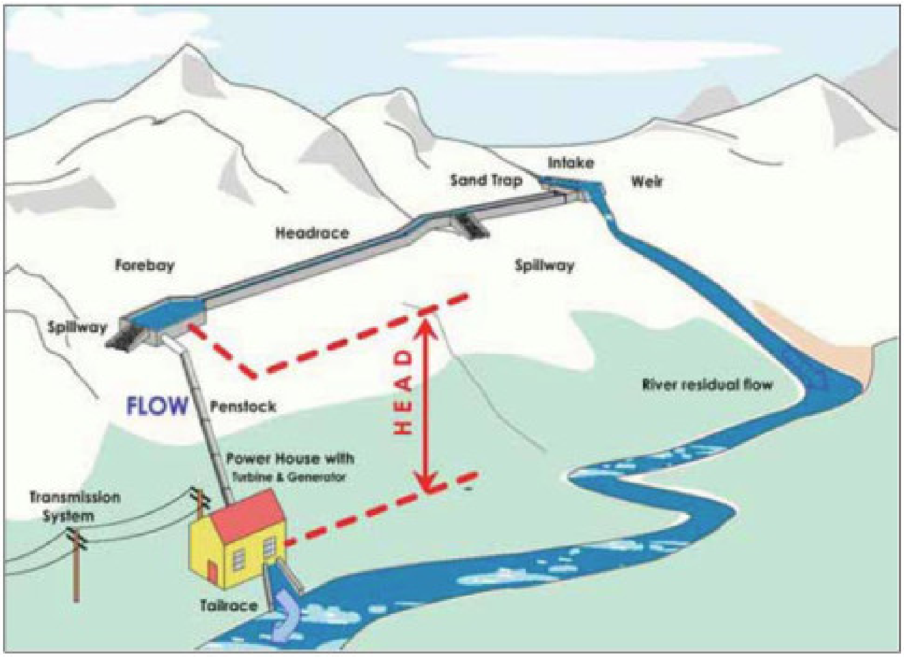

Figure 3.3: Run of the River Example [23]

- **Tidal:** A tidal power plant exploits the fact that there is a daily rise and fall of ocean water due to tides; which are highly predictable. In some situations, construction of reservoirs is permitted and can be used to generate power during high demand periods even if the tides are not high. Tidal power can only be exploited in a relatively small number of locations around the world [41].

- **Underground:** An underground power station can only be constructed in a place where there exists a large natural height difference between two waterways, such as a waterfall or a mountain lake. Water is taken from the high reservoir to the generator room that is usually built in an underground cavern near the lowest point of the water tunnel through in underground tunnel. Water is then released back to the lower outlet waterway [41].

3.2 Control Strategies

One of the problems facing small hydro power plants located along the same river is lack of communication between the owners. This lack of communication results in less efficiency in energy production. The main problem emerges when one owner does a certain action that affects the successor power plants but fails to communicate it [29]. Most research articles propose concepts and cooperation techniques to increase energy production and communication between large power plants [29]. Jäger et al. [29] discussed different control strategies that are relevant to small hydro power plants. These control strategies will be discussed here as well.

In this section, we will assume that a series of small hydro power plants operating on one river are viewed as a linear system with a defined neighborhood [14]. The neighborhood of one subsystem can be specified as a set of directly affecting subsystems of the whole linear system [14]. A control strategy can be defined as the essential characteristics of how the distributed subsystems (the plants) in such a linear system are controlled. A classification of distributed Model Predictive Control (MPC) proposed by Farina [14] based on: • the information exchange protocol, i.e., non-iterative or iterative,

• the type of the cost function to be optimized, i.e., cooperative or non-cooperative, and
• the topology of the transmission network, i.e., fully connected or partially connected.

There can exists two implementations of such distributed MPCs, a third centralized approach is introduced by Jäger et al. [29] They are [29]:

- The Local Control Strategy (LCS): A strategy based on a non-iterative and non-cooperative approach, which allows for only a partially connected topology.

- The Collaborative Control Strategy (CCS): A strategy that uses a cooperative approach, which relies on a fully connected topology of the transmission network.

- The Centralized Control Strategy (CCS): A strategy that uses information from all partners in the overall system and supplies them with the control details. A centralized service is created where every partner has to connect to this service. The service in turn coordinates all partners with respect to defined individual and the overall goals.

3.2.1 Local Control Strategy

The concept that the local control strategy is based on the fact that the control of the individual subsystem depends on data from other subsystems. All subsystems are data consumers and data providers [29]. No cooperation between the subsystems is considered, each subsystem is controlled locally.

Farina et al. [14] proposed their distributed predictive control schema for linear discrete-time systems, which focuses on non-iterative, non-cooperative, partially connected, to solve this kind of a distributed MPC problem. The approach relies on neighbors sending or receiving information at each sampling time about their future reference trajectories, and guarantee that the actual ones lie within a certain range of the reference ones [29]. Following that, each subsystem solves its own optimization problems. A two-phase approach was proposed by Matz et al. 9 to optimize the control of a single plant. The first phase involved optimizing the plant control "off-line", i.e., all predictions and decisions are based on historical data only [29]. The second phase also includes the prediction based on current, if possible real-time data and/or predictions. With this approach, additional data from the neighbors can be integrated into the individual control. This strategy utilizes the neighbor-to-neighbor communication and provides some profit to selected partners via decentralized optimization within the linear system even if not all partners in the line are contributing their information [29].

3.2.2 Cooperative Control Strategy

Unlike the local control strategy, in the cooperative control strategy, all subsystems are required to consider the effects of local control actions on them. Each subsystem is required to optimize for an objective defined for the overall system. Stewart et al. [40] propose the use of state and output feedback to improve the overall performance. The cooperative control strategy is frequently used within one plant, integrating the different systems, to achieve an optimal overall objective [29, 40].

3.2.3 Centralized Control Strategy

In the centralized control strategy, a centralized service is created that controls all subsystems, optimizing towards a centralized controller objective [29]. Control information is sent and received by all systems from the service. A powerful centralized server with strong optimization software are the main advantages of this approach. Yet its implementation is usually prevented due to organizational objections [40].

Choosing a suitable control strategy strongly depends on the subsystem owners trust in the centralized service and in their willingness and ability to consider the individual objectives [29]. In the local strategy, the subsystem maintains the full power of control but give it up completely with the centralized one. With regards to the specific problem of connecting small hydro power plants which will be explained in more details in the next section*ConnectHydroProblem*, the cooperative strategy is the best fit given good negotiation skills, when it comes to considering the effects of local control actions of all subsystems [29].

3.3 Connect Hydro Problem

For more than 100 years electricity is being produced by hydro power plants on the Alm river in Upper Austria. Today 55 small and micro hydro power plants of more than 40 owners are operated on 48 kilometers [29]. Connect Hydro is a project funded by the federal county of Upper Austria. It aims to explore a smart networking system(expert-system) which facilitates the ideal control and collaborative adjustment of small hydro power plants on a river [29].

The smart networking system consists of the collection and analysis of the latest data delivered from the small hydro power plants on real-time basis (e.g. performance data, water level, technical parameters on turbines and generators) [29]. External data such as the amount of rainfalls will be considered and fed to the smart networking system automatically. Advantages of implementing such system can include reduction of the costs of hydro power production mainly by reducing downtimes and maintenance expenditures and increasing the reliability of hydro power production. It can produce more electricity out of the river's available water supply by increasing turbine efficiency without harming nature [29].

The project involves real flowing waters and local operators of small hydro power plants. The probing is carried out with the involvement of specific stretches of running waters at established small hydro power plants involving their operators [29].

In the next section, we will provide an overview of the technical, organizational and financial aspects of the connect hydro project.

3.3.1 Technical & Financial Aspects

When discussing the **technical aspects**, the controls, the network (connection and transmission) and data integration should be considered [29]. The power plants currently use control systems ranging from analog relays over different kinds of PLCs to industry PCs. Many of these control systems are not ready to calculate complex control algorithms to optimize energy production, neither for their own system (using local control strategy), nor for the cooperative control strategy, which demands to take all models of the other subsystems, i.e. the other plants, into account [29]. Implementing a decentralized technical solution, extensive investment would be necessary with several of the plants, as each would need to implement the optimization algorithms and provide an infrastructure ready to run these sophisticated algorithms, which proves to be especially complex with the collaborative strategy [29].

The financial aspects should be considered by comparing the costs for implementation, long-term operation and maintenance including a quantified estimation of the innovative strength is important. Implementing the cooperative strategy with a decentralized infrastructure is not realistic with our scenario, as its technical complexity is too high for small operators and the anticipated costs as well [29].

As mentioned before, the main problem with the centralized control strategy is the **organizational objections** to its implementation. The partners involved fear to lose control over their own system, when it is centralized, and thus do not accept such a solution [40].

3.3.2 Proposed Concept

In the approach proposed by Jäger et al. [29], the infrastructure strategy and the control strategy were separated. This separation proves to be effective due to the size, infrastructure cost and the cost of implementing different solutions in the subsystems(partners) that are involved in the connect hydro project. Regarding the infrastructure, the idea is to create a common, centralized infrastructure with connection (and interfaces) from each partner that offers optimized services to all subsystems, regardless of the control strategy which will be implemented. This architecture will be based on a strong server infrastructure combined with a reliable and secure transmission network [29]. With a common data integration service set up in place, all data can be collected from different sources and by different interfaces without the need for local recording. The calculation is done on a highly efficient centralized resource and the results are sent back to each subsystem.

All control strategies can then be implemented since all the data is already available at the centralized server. Jäger et al. [29] propose to implement the cooperative control strategy, as all data needed is already available at the centralized server and it promises the best fit of optimized results for all partners, which is a good basis for a successful long-term collaboration. Reduced network traffic are important benefits that are considered as well in the cooperative control strategy. Changes in such a system can be easily managed e.g., new partner joining or an existing is leaving.

Despite all the advantages of this concept, some owners of subsystems may still be bothered by the use of a centralized service highlighted before as **organizational objections**. It is the provider of this common service's responsibility to build the trust needed to successfully cooperate to provide better results for all partners [29].

3.4 Proposed Implementation

In this section, we will discuss the proposed solution by Jäger et al. [29] for the challenges, which arise in the special scenario of connecting different small hydro power plants and refer to the additional ideas and concepts to increase the efficiency of generating energy. There exists a prototype implementation at several hydro power plants, which can be adapted and used for a smart connection. We will discuss the Technical and software details of that prototype.

3.4.1 Technical Level

Small hydro power plants in Austria operate independently without communication with other plants upstream and downstream. They don't provide or receive any data from other power plants or sources. Many power plants use obsolete control techniques and don't have any remote access. Owners of said power plants often live close to the power plants and take care of them. The goal of the connect hydro project is to examine the data from different power plants and explore if any real improvement can be achieved by connecting single small hydro power stations in terms of increasing electrical power and decreasing maintenance effort with low cost. An instrument should be specified to collect the data from different power plants and in return provide then with control instructions from the server. The instrument will connect to a central unit which will unify the data coming from different power plants and create useful information for the individual small hydro power plants. An example of data produced by the central server can be an early-flood warning system

3.4.1.1 Implementing a control system with external logic

Due to Power Plants having different states and types of devices, there exists a problem with setting up an overall communication system. The communication system aims to control the different parts of the power plant. The parts can be be relay control stations, small or large programmable logic controllers (PLC), or industrial computers [29]. According to Jäger et al. [29], when implementing a control system with external logic, the following steps must be considered:

- Data Gathering from several small hydro power plants
- Transferring of the gathered data to the external logic unit
- Sending the control recommendations back to the small hydro power plants

3.4.1.2 Components of the central server

Currently a prototype is installed at several stations. Data is gathered from three places in the small hydro power plants. The data is sent to the central server via C++ implementations. Connection to the server is managed via TCP/IP connection [29]. Components of the central server:

- Evaluation software: data from the sensors is received and stored/inserted into a database via a JAVA-application

- Database: responsible for data storage (MySQL)

- Web-Interface: visualization of the stored data via PHP and JavaScript

The following sensor data are gathered at the small hydro power plants:

- Opening clearance at the weir

- Water level at weir

- Water level before and after power plant

- Rack cleaning interval

- Water Turbidity

- Opening level of water turbine

- Power output from the water turbine

3.4.1.3 Components of the networking unit

Components selected for the networking units were selected according to their price, robustness, programmability, connectivity, availability, power consumption, interfaces, type of signals, compatibility with existing power plant controls, size [29]. Example of components of a finished solution/implementation [29]:

- Programmable controllers e.g. Siemens Simatic S7-1200, Mitsubishi MELSEC FX3GE, Advantech Adam 6024

- Industrial computer e.g. Siemens SIMATIC IPC227E

- Mini-PC e.g. Raspberry Pi, BealgeBone Black

- Modules for combination e.g. Arduino Ethernet, Atmega, Arduino Nano, Atmega328, Enc28J60

3.4.1.4 Central server tasks

As already mentioned, the central server is responsible for receiving and compiling all the data from the different power plants. The ideal server should be able to [29]:

- Provide a central communication component between the small hydro power plants
  - Insert data from power plants into the database
  - Send control information from database/system to the power plant
- Provide Data Storage (relational database, converting measured data to real values)
- Provide a Rule-based component (manage rules & if-then-relations, generate control information based on data and rules)
- Include Self-learning component (automatically improving rules, based on benchmarks)
- Include a User interface (web/app; manage users, rights, rules, assets, messages; visualization of facilities and parameters)
- Integrate with early-warning-system (e.g. via web service, water levels, weir opening, disturbances)
- Include a notification system (e.g. via SMS or email, necessary for the facility operators)

3.4.2 Software Level

A combination of a "Database System", a "Knowledge Processing System" and an "Early Warning System" is proposed for the central system at the software level [29]. The system should announces alarms/alerts based on the "Decision System" to be discussed in the next chapter *ChapterFour*. Figure 3.4 shows The overall communication- and regulation-schema for the proposed system is highlighted.

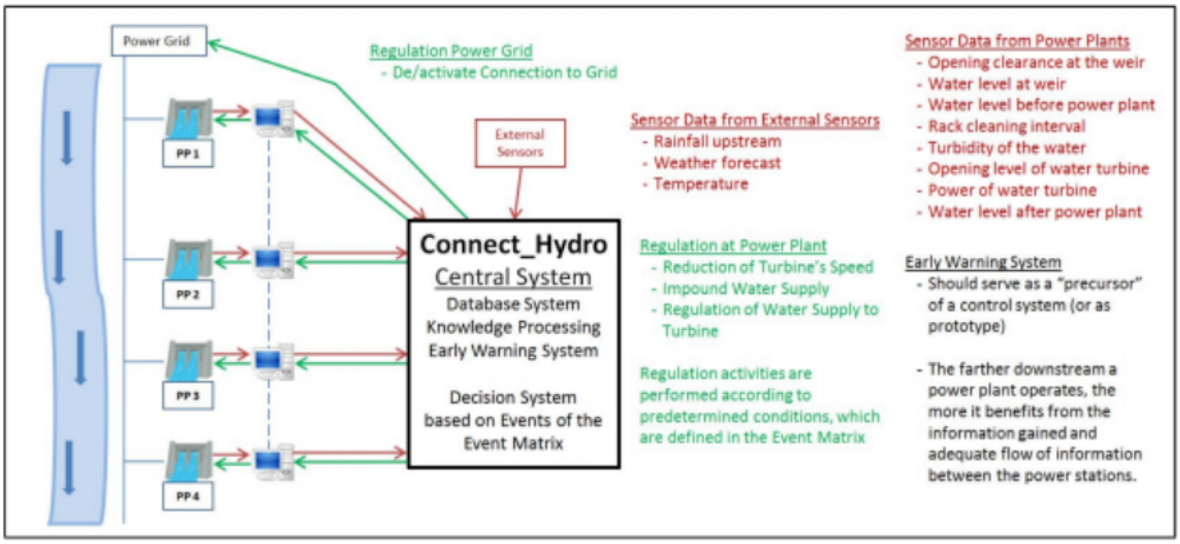

Figure 3.4: Overview technical concept "Connect Hydro" [29]

The work flow of the prototype starts by processing the input- and historical data from the database, mapping it to the events in the event matrix, correlating with previous events and knowledge from the knowledge processing system. In the next chapter we will discuss how the proposed system along with the already discussed Decision support techniques can optimize the power plants production efficiently and provide the owners with informed decisions as well as implement an early warnings to manage certain events.

Chapter 4

Decision Support in Connect Hydro

In the Last 2 chapters, We discussed the Decision support systems and all their features and the best techniques to develop them. We also introduced the Connect Hydro Project and discussed all its different aspects. In this chapter we will discuss what kind of decision support system could be implemented within the connect hydro project and how it can bring an added value. Our solution is based on the work already implemented in the connect hydro prototype. In our project, we implemented a simple DSS system integrated in a web portal.

The Application flow starts as follows: Data from sensors at the power plants is gathered as well as sensor data from external sources and stored in the central system , mainly in the database. From this database, the knowledge processing system can learn, based on the events defined in the event matrix, e.g. how several sensor data combinations and occurrences will lead to which events. Based on the knowledge in the system, it is possible to predict harmful occurrences (e.g. a flooding of a power plant can damage the active turbines) or apply a set of predefined rules to the processed data, to make decisions. An example of decision to be proposed is that the turbines should be deactivated when a flooding is predicted and give signals either to the early warning system (e.g. message to the owner/operator of the small hydro power plant that it is recommended to deactivate the turbines) or an installed regulation system (e.g. which automatically can deactivate the turbines) [29].

4.1 Connect Hydro DSS Type

In *ChapterTwo* we discussed how a DSS system should be designed, starting with the problem definition in terms of input, output, user knowledge/expertise and decisions. In connect Hydro, the DSS's problem is to optimize the energy production of one power plant using the data received from downstream and upstream power plants, historical data as well as external resources. To solve this problem we would like to explore the possibility of combining two types of Decision support systems, the data-driven and the knowledge-driven systems as each separately prove extremely beneficial in solving our problem.

The data-driven approach would prove to be very efficient in managing the huge amounts of sensor data as well as the data from the external sources. The external sources can include weather forecasts, historical data of events etc..The historical data can be analyzed using an On-line Analytical Processing (OLAP) tool which will definitely contribute with producing decisions in a fast manner that were evaluated from every aspect to the power plant owners.

The knowledge-driven approach is also an effective approach for our problem. Part of the information that should be provided by the power plant owners would be a set of their rules. These rules can either be power plant specific or general rules applied to all power plants along the "Alm" river, depending on the agreement between the power plants. An example rule would be "If Water Level > 'Defined Threshold', Turn off the Turbines."

Both approaches alone are very relevant to our problem and would prove very efficient, but we think combining them would provide better results. In the next section we will discuss the event matrix that was proposed for our system.

4.2 Connect Hydro Event Matrix

In our portal, we chose only a few events to model as well as sensor data from November 2016 for 3 power plants. The sensor data were extracted from the database of connect hydro project. The sensors that were used for our prototype included:

Sensor
Water opening Size
Power output turbine
Water level at turbine
Water level after rack
Water level before rack
Canal Closed at rack
Canal Opened at rack
Cloudiness
Rack Cleaning

Using the data from these sensors for the power plants, we came up with an event matrix. The event matrix is the events that can happen if a combination of values from different sensors occur. In our prototype we considered 4 Events, water temperature, water level, Turbidity and the lack energy output. The 4 events are described below:

- **Flooding:** if the water level rises above a predefined threshold, the water channel should be closed, so that the turbines are not damaged due to the high amount of water. Reaction: Turn down/off Turbine & Activate Rack Cleaner.

- **Low water level:** if the water level in the water channel falls below a predefined threshold, the turbines cannot work properly. The water-supply should be regulated so that the turbines can work with a lower capacity and there won't be any downtime of the turbine. Reaction: Regulate Turbine.

- **Waste dump:** if the turbidity of the water is very high, it could be that there is so much dirt, which can damage the turbines. The turbines should be turned off, so the water channel can be cleared from dirt. Reaction: Turn off Turbine & Activate Rack Cleaner.

- **Ice:** if the water temperature falls below a predefined threshold, it is possible that ice will appear at the rack, which could lead into a frozen water channel. Reaction: Turn on Rack Heating.

- **No Energy output:** if the water level at the previous power plant downstream is low and they don't have any energy output, then water level will be also low at my power plant. Reaction: Turn off Turbine & Close Down Water Canal.

A visualization of the event matrix was included in the web portal as shown in the following figures.

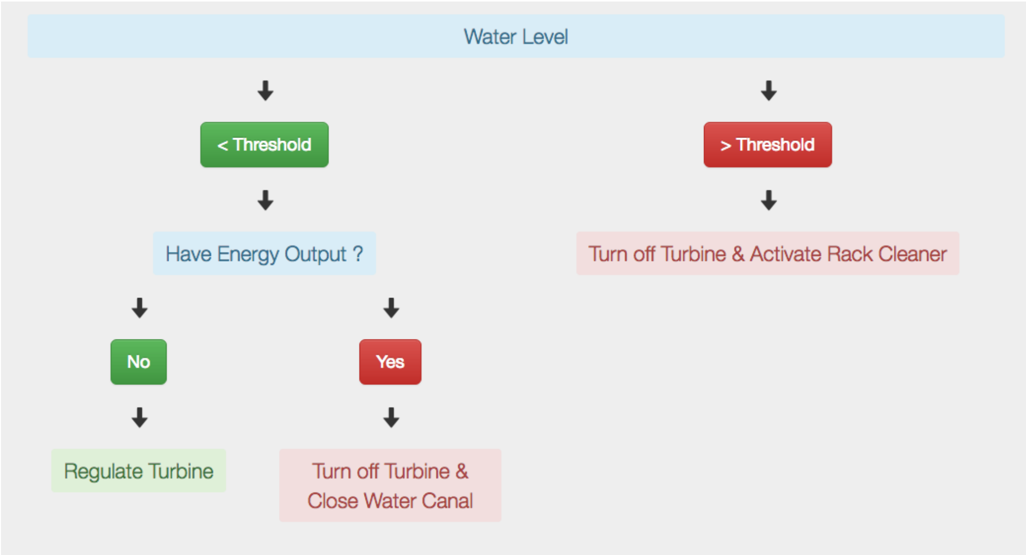

Figure 4.1: Low water Level, No Energy output and Flooding Events

Figure 4.1 explains how the events of the "Low Water Level", "No Energy Output" and "Flooding" were modeled.

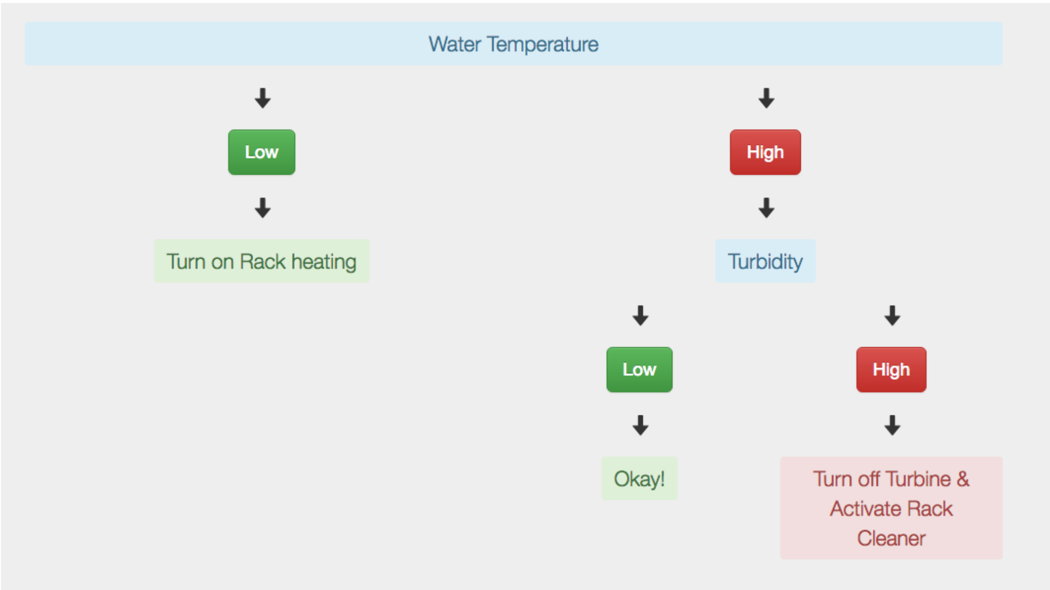

Figure 4.2: Waste Dump and Ice Events

Figure 4.2 explains how the events of the "Waste Dump" and "Ice" were modeled. In the next section we will discuss our approach in designing and Implementing the Connect Hydro DSS as well as the limitations we faced and what could be future enhancements that would make the system even more useful.

4.3 Connect Hydro DSS Design & Implementation

There should be four components properly defined before we can proceed into the designing and implementing processes.

- Input: Input for connect hydro DSS would be the sensor data coming from the power plant, historical data saved in the database and data from external resources (e.g. Weather Forecast).

- User Knowledge/Expertise: No user expertise will be needed in forming the decisions, but the users will need to implement the decisions.

- Output: The output will be specific for every power plant.

- Decisions: The system will suggest actions to power plant owners after analyzing the data and determine the needed action to correspond to the input.

After Defining the problem, the next step would be to start Analyzing the DSS from a business perspective and deciding what type of decisions the DSS will provide. In our prototype, The DSS will support operational decisions, decisions that impact the day-to-day activities as well as contribute to the overall goal of increasing the energy production for the small power plants. The decisions to be produced by our DSS will be repetitive as well as structured, the situation doesn't drastically change from 1 day to the other and the possible outcomes are limited.

The next step after analyzing the business perspective would be to define the problem solving process.

- 1. Defining the Problem: The problem is to provide decisions for power plant owners to maximize their energy production and minimize downtime for the power plant.

- 2. Identifying Decision Maker: The decision maker is the owner of the power plant.

- 3. Gathering Information: Data will be collected from sensor data at the power plant as well as the previous power plants, historical data saved in the database and finally external resources.

- 4. Evaluating Alternatives and Deciding: The system will evaluate all the data and try to see if the data matches any of the predefined events in the event matrix or if any of the rules defined by the power plant owner are applicable and decide the best action that the owner should do.

- 5. Implementation and Follow-up: There should be a system keeping track of the actions taken and if the decided actions were implemented by the owner, they should become part of the historical data be used in the future decision making.

In development of our DSS, we used SDLC - System Development Life Cycle Approach. Although SDLC is the most rigid system development approach, it fit the needs of this project as it was just a prototype. In the future, we believe the Rapid Prototyping Approach will be better and the involvement of the end users will definitely bring an added value to the project and motivate them to use the system.

The Connect Hydro DSS user Interface was a simple Interface which starts the decision making process on demand and the result was presented to the user as simple to-do list of all the actions that should be taken. Visualization of sensor data was also included as a module to further convince the owners that the decisions produced by the DSS are valid and aim to make them more profitable till they could trust the software.

4.4 Visualization for Connect Hydro

In Connect Hydro, Data Visualization was included as a tool to further convince the power plant owners that the decisions produced by the DSS were correct with the final goal of maximizing their energy production. The visualizations produced included 2 different Line charts. The first chart showed how the Energy output differed depending on the Water Level. The second chart showed the energy production of one power plant over time. Examples of the 2 charts are shown below in Figures 4.3 and 4.4.

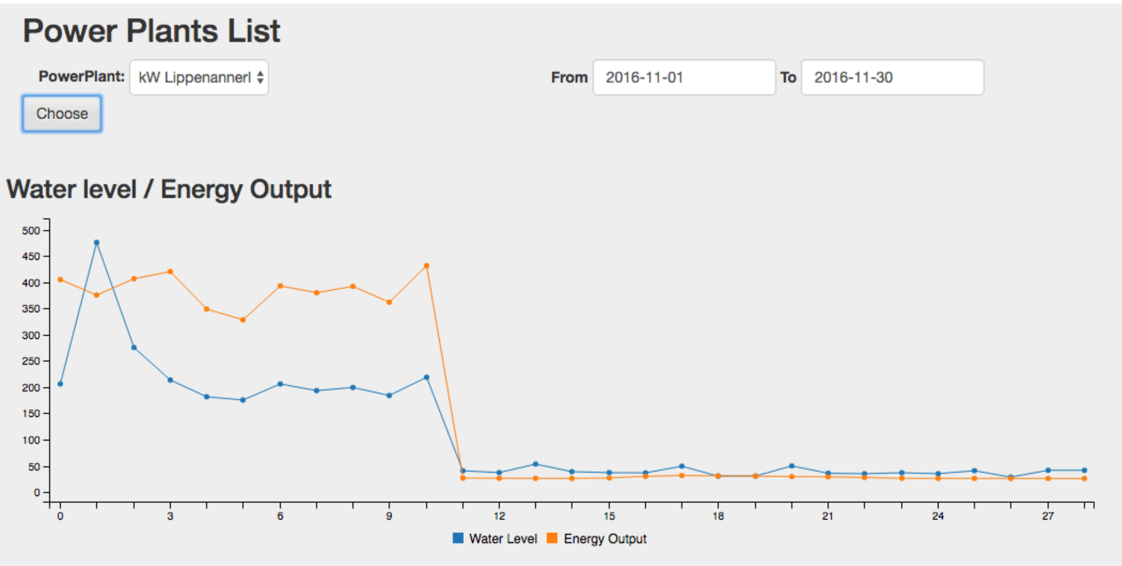

Figure 4.3: Water Level/Energy Output per day

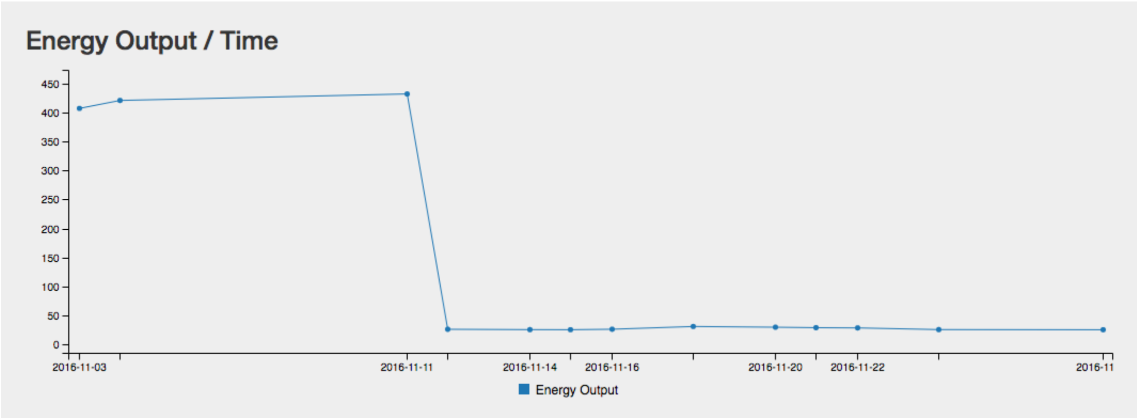

Figure 4.4: Energy Output per day

In the next chapter we will discuss the System Architecture as well as the frameworks and technologies used to develop our web portal prototype including the details of how every part was developed.

Chapter 5

System Architecture and Development Technologies

In the first part of this chapter, the web development techniques used in the development of our web portal and DSS will be introduced by giving a brief overview on web development, responsive web design and Ajax technology. Front-end and Back-end frameworks will also be discussed. In the second part, we will discuss the system architecture and explain how every part of the system was designed and developed.

5.1 Web Development Techniques

Techopedia defines Web Development as the coding or programming that enables website functionality, per the owner's requirements [9]. It refers to building, creating, and maintaining websites. It includes aspects such as web design, web publishing, web programming, database management, client-side/server-side scripting and network security configuration. Web development ranges from creating plain text pages to complex web-based applications, social network applications and electronic business applications. The web development hierarchy is as follows [9]:

- Client-side coding
- Server-side coding
- Database technology

5.1.1 Responsive Web Design

Web Technology is always developing and changing, new devices with different resolutions are created everyday. Developers nowadays need to find a way to manage these new additions and make sure their websites or web applications are always compatible. Responsive web design is an approach that proposes a design pattern that should respond to the user's behavior and environment based on screen size, platform and orientation [5]. It provides a mix of flexible grids and layouts, images and an intelligent use of CSS media queries. The website should automatically switch its resolution, image size and scripting abilities to a become compatible to the one the user is requesting on his device.

Responsive web design was originally defined by Ethan Marcotte in A List Apart [7]. The Concept behind it is that the layout should change based on the size and capabilities of the device the user is using at that particular time. For example, on a phone users would see content shown in a single column view; a tablet might show the same content in two columns. Figure 5.1 Shows how the responsive web design could handle different platforms.

Figure 5.1: Responsive Web Design [6]

5.1.2 Asynchronous Javascript (Ajax)

Ajax stands for Asynchronous Javascript. It's a combination of several technologies working together in powerful new ways. It allows parts of a web page to be updated without having to reload the entire page [17].Ajax incorporates XML or JSON, CSS, Javascript and XMLHttpRequest objects [42]. XML or JSON are text-only format used to transfer data from server to browser script. Developers are increasingly using JSON over XML because of its native JavaScript compatibility. CSS is the language used to style how the data will look on screen. Javascript is used to display the data in the browser and processes user requests/interactions like clicks. XMLHttpRequest objects are the keystone of AJAX, they actually retrieve the data with the server behind the scenes. All modern browsers support XMLHttpRequests. Ajax changed usability and the speed of web applications with its innovative concept: asynchronously exchanging small amounts of data with the server behind the scenes, without affecting the rest of the page [42].

The classic web application model starts with a user making an action in the interface that triggers an HTTP request back to a web server [17]. The server does some processing — retrieving data, talking to various systems — and then returns an HTML page to the client. This approach makes a lot of technical sense, but it doesn't make for a great user experience. While the server is doing the processing, the user is waiting.

An Ajax application eliminates the start-stop-start-stop nature of interaction on the Web by introducing an intermediary — an Ajax engine — between the user and the server [17]. When loading a web page, the browser loads an Ajax engine — written in JavaScript and tucked away in a hidden frame. The Ajax engine is responsible for rendering the interface the user sees as well as communicating with the server on the user's behalf [17, 42]. The user interaction with the server happens asynchronously; the user is never staring at a blank window or a loading icon. Figure 5.2 shows the classic web application model vs the Ajax web application model flows.

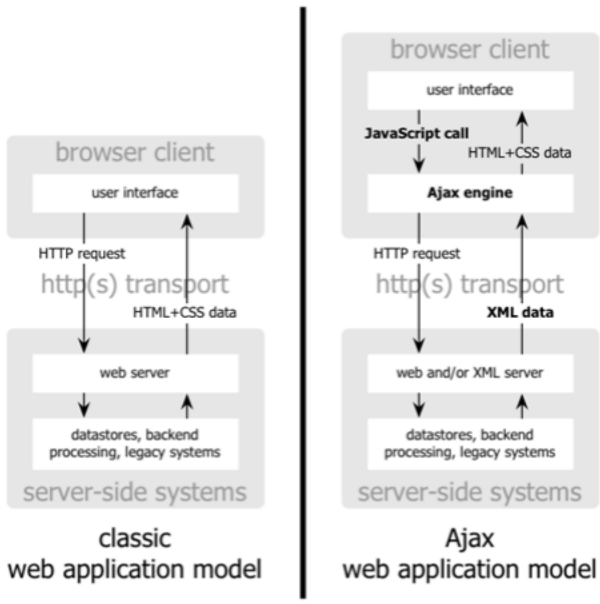

Figure 5.2: Classic vs Ajax Web Application Model [17]

5.2 Front-End Technologies

World Wide Web is always growing and changing. A few years ago the majority of users browsed the web using desktop computers with relatively similar resolutions and dimensions. Internet Explorer was the main browser used for websites browsing; Firefox, Chrome or Opera were less popular [26]. In the last few years, new devices appeared for example: smart phones, tablets, glasses and even smart TVs. These devices triggered website interfaces evolution [26]. The interfaces became more complex, dynamic yet stayed user friendly [26]. Responsive web design approach became a standard and AJAX is broadly used. This evolution led web developers to find a different way to maintain their websites responsiveness, apply code optimizations and develop websites with better performance and scalability [26]. All these aspect led to the creation of front-end frameworks.

A front-end framework is all what user sees and allows him to use the application. Framework is a platform, foundation on which ready software solutions are built, in this particular case – web interfaces. For this purpose front-end framework consists of ready components, which are used by a developer when working on a project. These components can be modified or adjusted to current needs [26].

5.2.1 Bootstrap

Bootstrap is currently the most popular front-end framework. It was created by Mark Otto and Jacob Thornton, developers working in Twitter [26]. Its main goals included supporting developers for fast development of web applications, having a consisted code, easier maintenance as well as easier further development. Bootstrap allows for rapid, responsive development that is consistent and well supported by the development and design community. It offers ready-made coding blocks which can be utilized to build a website easily. It also has a very short learning curve and is easy to use for beginners in web development. Bootstrap's most important components are [26]:

- **CSS files:** They contain the global settings that the website should have as well as define the look of the most important HTML items, i.e. text, lists, form elements, tables or images [26]. The files also comprises of a set of ready-to-use, frequently employed on today's websites, components. These components include expanding elements, buttons with advanced features (e.g. grouping), navigation menu, paginations, messages or progress bars [26].

- **JS files:** They contain ready-to-use plug-ins of popular jQuery libraries. They support creating interfaces with dynamic elements quickly. Most important are components include modals, carousel sliders, dynamic tabs, expandable lists, accordions or tooltips

[26]. Manipulation of these items can be done using HTML attributes rather than Javascript which makes it really special.

5.2.1.1 Advantages of Bootstrap

- Speed of Development: Bootstrap enables the utilization of ready made blocks of code to help the developers get started instead of writing code from scratch [18]. Bootstrap also supports cross-browser compatibility and CSS-Less functionality, thus saving many hours of development time.

- Responsiveness: Bootstrap has a fluid grid layout that dynamically adjusts to the proper screen resolution [18].

- Consistency: Bootstrap ensures consistency regardless of who's working on the project. In addition, results are uniform across platforms so output remains the same even when using different browsers [18].

- Customizable: Bootstrap can be tailored. Developers can pick and choose the features that are needed and the rest can be tossed. This is easily accomplished using the Bootstrap customize page where the developer can tick the features they need or don't need [18].

- Support: Bootstrap has a huge support community behind it and support can be found for any issues the developer may encounter [18].

5.3 Back-End Technologies

The back-end usually consists of three parts: a server, an application, and a database [15]. It is known as the server-side. It is responsible for storing and organizing data, and ensuring everything on the client-side actually works. The back-end communicates with the front-end, sending and receiving information to be displayed as a web page in a website [15].

A website needs a database to manage all it's needed information. A database stores website content in a structure that makes it easy to retrieve, organize, edit, and save data. It runs on a remote computer called a server. There are many different databases that are widely used, such as MySQL, SQL Server, PostgresSQL, and Oracle. A website is built using a language that a database can recognize. Some common back-end languages are Ruby, PHP, Java, .Net, and Python. These programming languages often run on frameworks that simplify the web development process. Spring, for example, is a framework written in Java.

5.3.1 Spring Framework

Spring is a very popular Java-based framework for building web and enterprise applications [24]. Spring framework provides flexibility to configure beans in multiple ways such as XML, Annotations, and JavaConfig which is considered the backbone of a website or an application. Spring boot was created by the spring team to address the complexity of configuration. It is very popular because of several reasons [24]:

- Spring dependency injection approach encourages writing testable code
- Powerful database transaction management capabilities
- Integration with other Java frameworks like JPA/Hibernate ORM, Struts/JSF/etc. web frameworks is simplified
- Spring utilizes some of the well-known technologies, ORM frameworks, logging frameworks, JEE, JDK timers, Quartz etc., there is no need to re-invent the wheel and developers don't have to learn any new technologies or frameworks [11]
- Spring framework provides inversion control and APIs to translate technology-driven exceptions, specifically thrown by JDBC, Hibernate or JDO, into unchecked and consistent ones [11]
- Due to modularly organized nature, Spring makes it easy for the developers to know which packages or classes are to be used and which one should be ignored [11]
- With the help of consistent transaction management interface, Spring framework easily scale down or scale up local as well as global transactions[11]
- Spring Web MVC framework for building web applications is State of the art

5.3.1.1 Dependency Injection

Martin Fowler created Dependency Injection. In object oriented design, objects have relationships with one another [31]. Dependency Injection design pattern is used to define the object dependencies between each other and try to eliminate it if possible. It exits in two major types Setter Injection and Constructor Injection.

A design based on independent classes / components increases the re-usability and possibility to test the software. A software design based on dependency injection is possible with standard Java. An example to explain DI is as follows:
We can have two classes A and B where Class A depends on Class B. DI will inject Class B into Class A by using Inversion of Control (IoC) as shown in figure 5.3 [31].

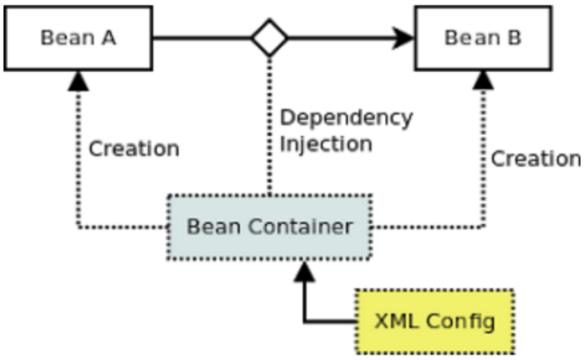

Figure 5.3: Dependency Injection [31]

5.3.2 MySQL Database

MySQL is the most popular Open Source SQL database management system. It is developed, distributed, and supported by Oracle Corporation [10]. MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL) [3]. It has a Client – Server architecture where a user can access MySQL Server using a MySQL Client the client can be the command line, desktop applications or web applications. Some of MySQL advantages include speed, reliability, ease of use and most importantly scalability.

5.4 System Architecture & Components

In the Following sections, we will discuss how different components of the application were implemented, Why they were implemented in that way and discuss any considerations that were taken during the development process. We will start by discussing the database Design and Implementation followed by the web portal, the decision support and finally the data visualization.

5.4.1 Database Design and Implementation

In this Section we will discuss the database design and Implementation that was created for our prototype and why we chose to implement it in that way.

As already highlighter, our work is based on the work already implemented in the Connect Hydro project described in *ChapterThree*.In Connect Hydro, a device was added at the power plants which had many ports, each port is connected to a sensor. The sensor data which is the voltage that a sensor is reading at a certain moment is sent to their database. The database was implemented using MySQL and data was sent to it from every device every 10 seconds. Their database consisted of two tables. One included all the sensor data retrieved from the devices in the power plants that were participating in the experiment. The second included the encoding of these sensor values, the factor that should be multiplied to the sensor values in the first table to produce the actual value. The Tables structure they used is shown below.

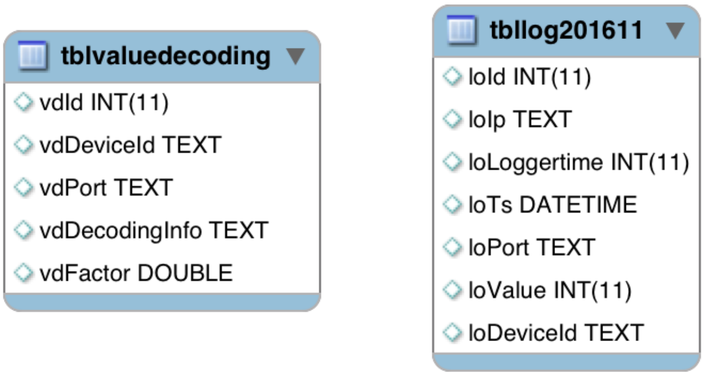

Figure 5.4: Connect Hydro Tables

The following table explains what each column in the database table tbllog201611 represented.

Column Name	Explanation
loId	Id, Primary Key
loIP	IP-Address of the Loggers at the power plant (Not Unique, always changing)
loLoggerTime	Time of when the data was collected at the power plant
loTs	Timestamp of when the value was inserted into the DB
loPort	Port ID of the device that is sending the value
loValue	Raw value collected by the device from the sensor
loDeviceId	Device ID that this sensor data is collected from (Unique)

The columns in the tblvaluedecoding are explained in the following table.

Column Name	Explanation
vdId	Id, Primary Key
vdDeviceId	Device ID that this sensor data is collected from (Unique)
vdPort	Port ID of the device that is sending the value
vdDecodingInfo	Any extra information (Always NULL)
vdFactor	The factor that should be multiplied to the raw value in the previous table to get the actual value

As shown above the connect hydro database was not well structured, In our thesis a new database design is proposed and implemented. The database design was created not only to fit the connect hydro project scope but took into consideration that it can be implemented in other locations in the future. It included information about power plants, their operators, the river that the power plant are located on. Information about Devices installed at the power plants, their ports, the connected sensors and any rules the operators defined were also included.

There were two main focuses for the database; The Power Plants and the Devices; each will be discussed separately but their relation will be clearly defined. In the rest of this section we will discuss the new database design and the relations created between the different tables.

In our scope, a power plant was defined as a location which had its own water supply (water channel) and had a power output turbine. The information collected about power plants were its name, geographical coordinates, distance between its predecessor and successor power plants. It was also given a unique ID. Multiple operators could operate one power plant. Data collected about operators included their name, Email, phone number, address and each operator was also assigned a unique ID. As Many power plants are located along the same river, that relation was also included and each river was identified by a unique ID as well as its name, Length and any other description the system admin wanted to include. The above relations are demonstrated in figure 5.5.

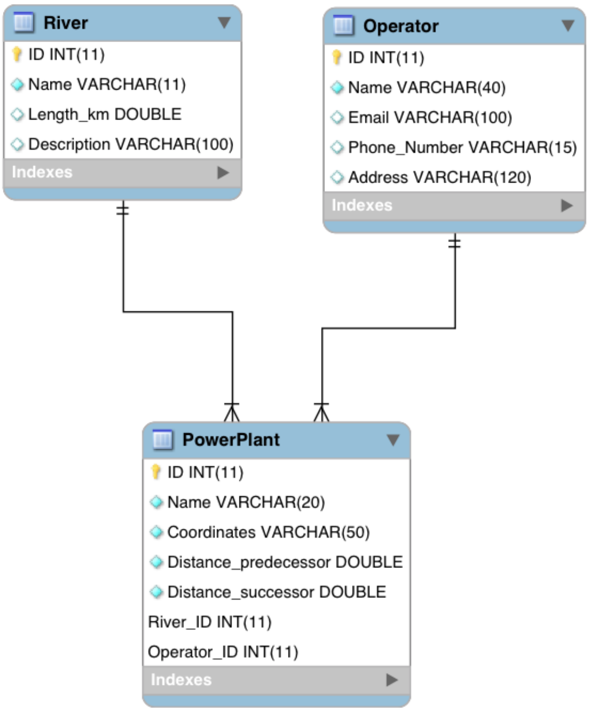

Figure 5.5: Power Plant-Operator-River Relation

Each of the three power plants that participated in the connect hydro project was fitted with a device. This device was created by NEXTSOFT IT GMBH in a contribution to the connect hydro project. It is a piece of hardware that is installed at the power plants. In some cases, one device can collect data from more than one power plant.

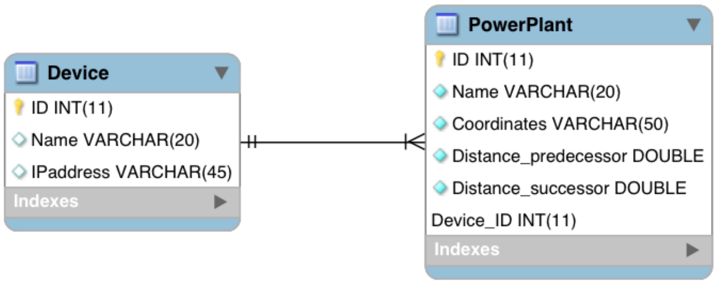

Figure 5.6: Power Plant-Device Relation

A device is identified by its unique ID and Name; an IP address is also included in the device table but it changes frequently. A device includes multiple ports and each port is connected to a sensor. Each port in the same device has a unique name.

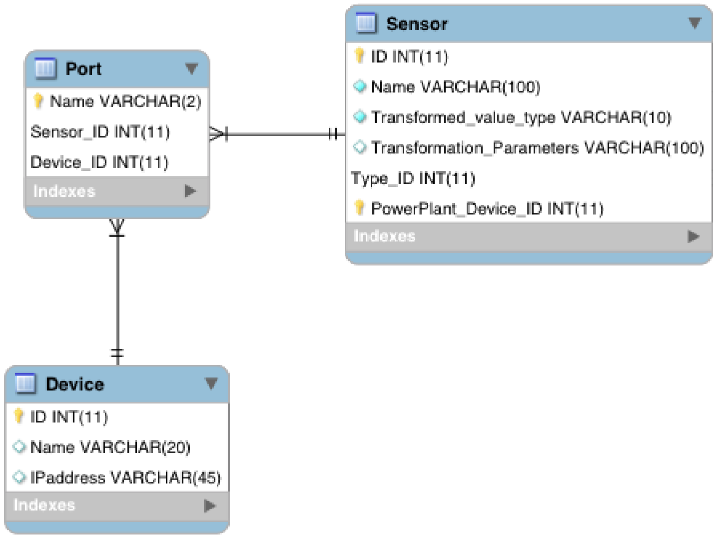

Figure 5.7: Device-Port-Sensor Relation

Each Sensor had the following specifications: Unique ID, Name, variables indicating the transformed value type and parameters and a sensor type. The transformed value parameters maps the vdFactor in the original connect hydro database. The transformed value type is added to handle future situations where a transformed value is not just a numeric value but a percentage or anything else. As already mentioned each sensor has a type, all the sensor types are added in a separate table and is referenced from the sensors table. The power plant operators can add rules specific to every sensor in their power plant that are processed and evaluated later in the decision support module. These rules are specified in a separate table as well and are identified by a name and parameter/s.

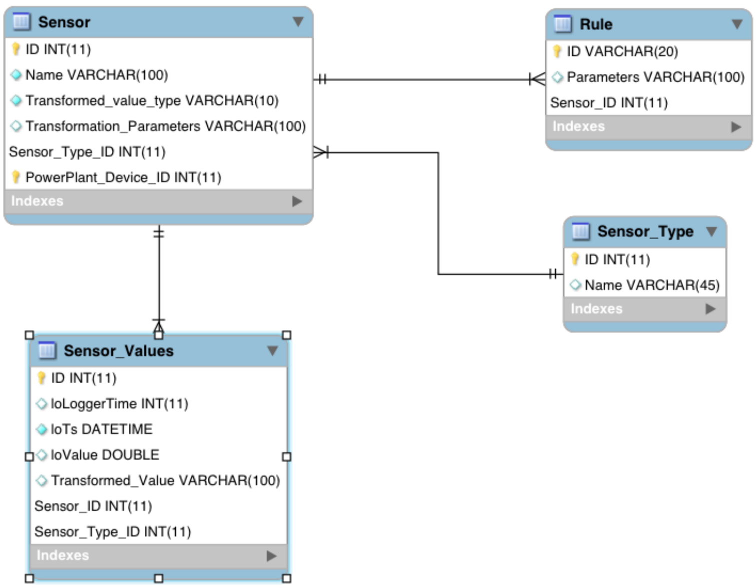

Figure 5.8: Sensor Relations

The sensor Values table is the table that maps the data in the tbllog201611 to our database. It contains some columns with the same names as the original table like loLoggerTime, loTs and loValue. It also contains the Transformed_Value column which is the raw value (loValue) multiplied by the transformation parameters.

The database included some other tables that were not connected any tables. They included a table saving the relation between two successor power plants, a table saving all the emails and passwords of the users who will use the web portal and finally a table with Actions and warnings messages to the power plant operators; its function will be discussed later.

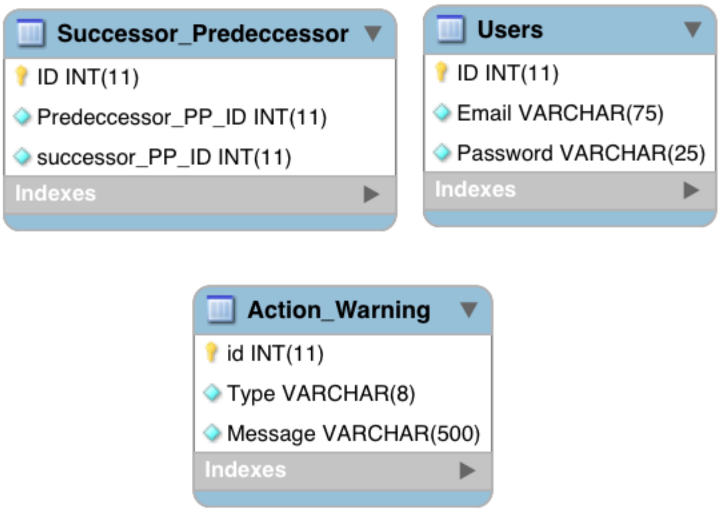

Figure 5.9: Separate Tables

For the scope of our project, we extracted one month data from the original connect hydro database and using a series of manual SQL queries added them into our database. We chose September 2016 for the demo data as it highlighted a few interesting events which were later used for the decision support and the data visualization. The Data extracted corresponding to that month included 1062321 records which were more than enough to simulate the different actions and visualization.

5.4.2 Web Portal

A web portal was developed for our prototype in this thesis. It included the MySQL database discussed in the previous section, the decision support system module and the data visualization which will be discussed in the next few sections. For our development we chose Spring MVC (Model-View-Controller) framework mainly because it's flexibility and ease of integrating with all the technologies we used in the prototype. It is based on

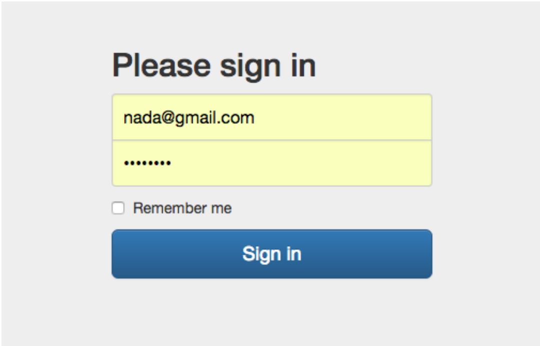

Figure 5.10: Login Page

Java which was the preferred language for the connect hydro team. In spring every Model represents an object and in our case we mapped it to our database tables. Each Model had an associated DAO class. DAO should perform raw database operations and translate them to some higher level constructs (i.e. return them to the controller as Java objects).

Earlier in this chapter we discussed bootstrap; In our prototype we used bootstrap for the front-end development because of its flexibility, responsive designs as well as its mobile compatibility. Our website can be opened from any browser and any mobile device once hosted.

The first functionality of the portal is the log-in functionality. Whenever the user first opens the website (web portal), he is directed to log in page. The Login HTML page sends a request to the controller along with the email and password the user entered. A request is sent to the database using the UserDAO class to verify the credentials he entered. If the user exists in the database in the Users table and the credentials entered is correct, he is directed to the homepage else he is not allowed to proceed. A screen shot of the login view as shown in figure 5.10.

5.4.3 Decision Support

In an ideal situation, a decision support system should be triggered when the sensor data is entered into the database or every defined interval. In our case that was not possible as we were using static data for the month of September 2016 and couldn't have direct access to the connect hydro database. Our decision support system was triggered on demand.

As mentioned before in *ChapterFour*, there are 2 different types of DSS implementations included in our system. The first is the event matrix which was already explained and the second are the rules for the knowledge-driven part of our DSS. The power plant operators come up with rules that given certain conditions, the DSS should propose an action such that the defined rule is applied.

- If water Level < 27, Turbines should be Turned Off.
- If Water Turbidity is high > 25, Activate Rack Cleaning.
- If No Energy Output and water level < 25 , Turn off turbines.

The above are a few examples of rules that can be defined by owners. In our implementation we wanted to be flexible and allow for future additions of rules, therefore the rules logic was implemented in Java. Power plant operators only had to define a rule name and its parameters and add it to the database, then in a Java class a rule with the same name is added by the developers and the logic is implemented where the message that would be sent to the operator or the actions proposed are added. During our investigation we found that when the rules are applied, it is better and provide better results if the sensor data is retrieved from the previous power plant rather than the one applying the rules. For example: Given two Power plants A & B, If B wants to apply the rules and check what kind of actions the system proposes to him, the system should check the data from A. If water level is low at power plant A then it will definitely be low at power plant B or if the water is unclean it will not clean itself by the time if reaches power plant B. Another scenario would be that power plant A does rack cleaning, Power plant B will need to be informed to close its canal to avoid all the leaves and dirt coming into his power plant and ruining his turbines.

In our decision support system, the average value of last 30 minutes data of a specific sensor was calculated and used in the rules (i.e. compared to the parameter defined by the operator in the rule). Ajax calls and JSON data was used to send the requests to the server and send the results back to the view in our website. The following Images show different actions proposed to different power plants in our portal after the rules and the decision matrix events are applied to the sensor data received.

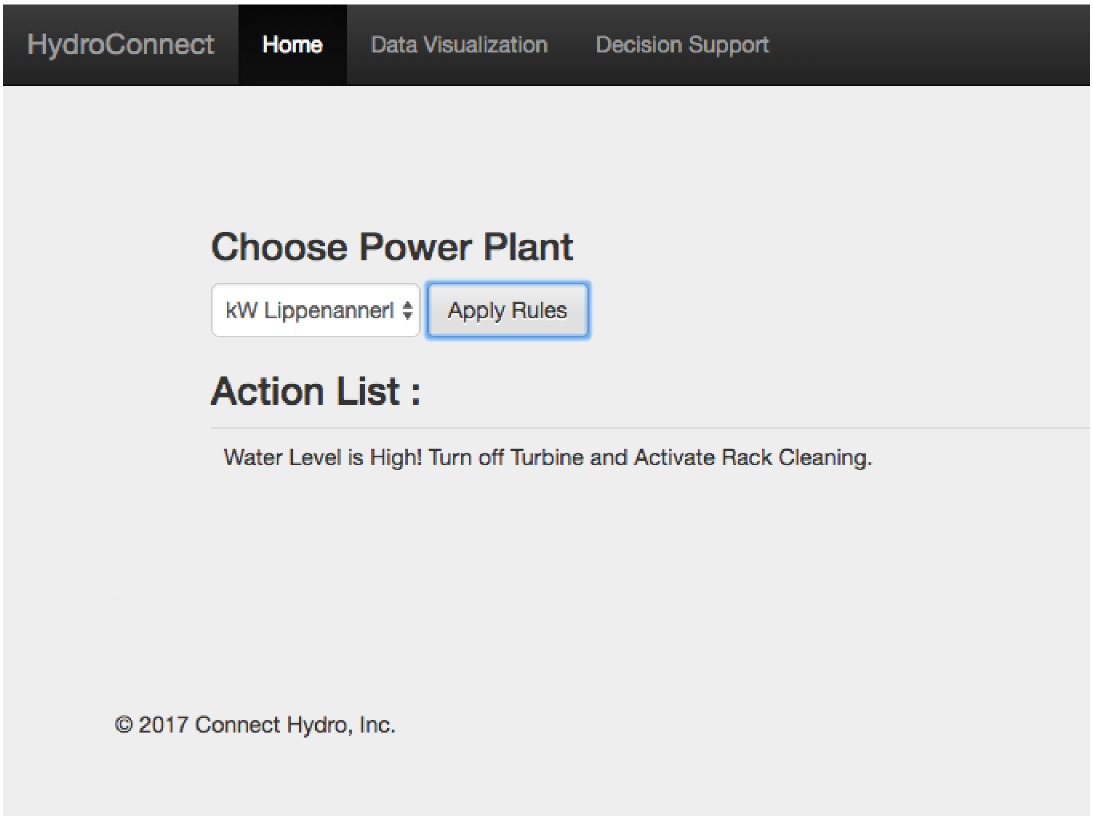

Figure 5.11: Power Plant 1

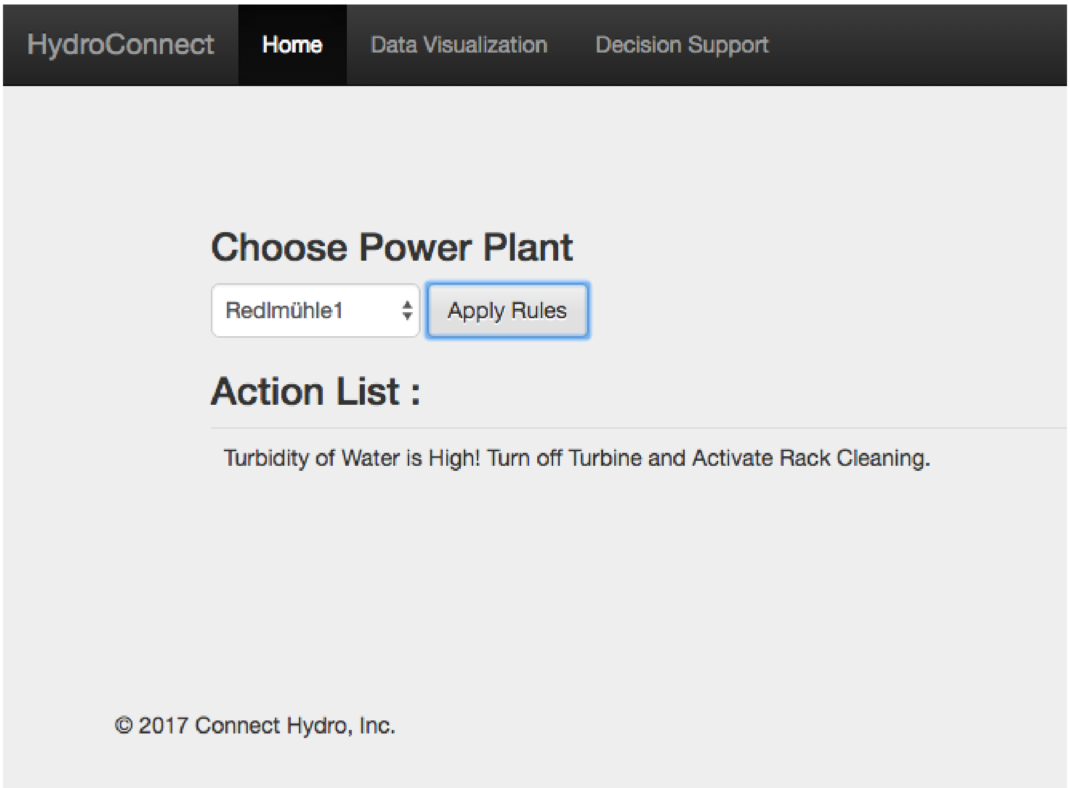

Figure 5.12: Power Plant 2

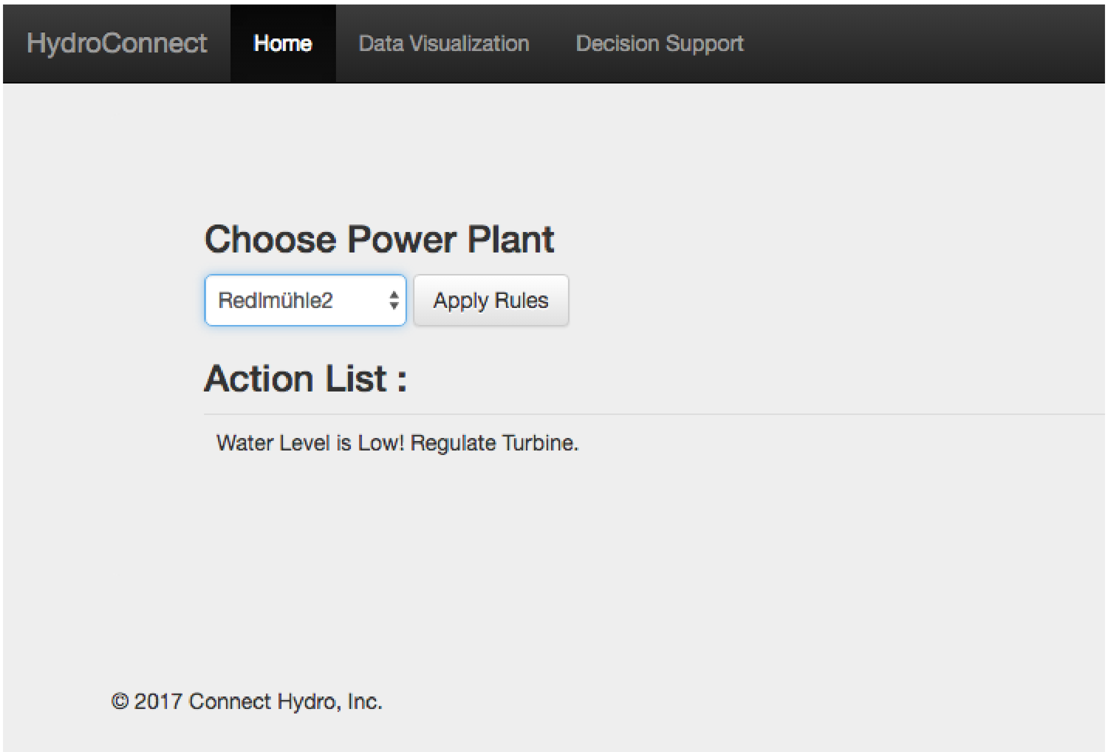

Figure 5.13: Power Plant 3

5.4.4 Data Visualization

In our web portal, two types of visualization were included, a decision tree for the event matrix and line charts for the sensor data visualization. Both were highlighted in *Chapter-Four* but in this section we will explain their contribution towards our final goal and how they were implemented.

Decision tree are defined as the classification or regression models in the form of a tree structure [1]. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data [1]. In our scope the decision tree is added as a tool to further convince the power plant owners that the decisions produced by the DSS were contributing to the final goal of maximizing their energy output.

The Decision tree in our prototype was developed using HTML and CSS scripts. It was included to convince the power plant operators that the decisions proposed to them are well informed.

The second visualization added were two line charts showing the "water level vs Energy Output" and "Energy Output per day" for a specific power plant in a specific date range. The user would start by choosing a power plant, start date and end date then clicking choose as shown below.

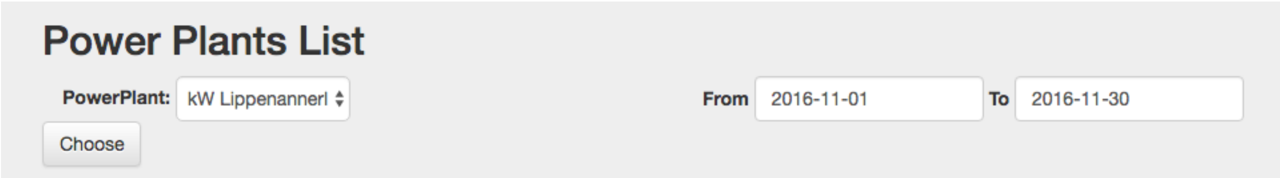

Figure 5.14: Requesting Visualization Data

An Ajax request is sent to the controller to retrieve the required data. The controller queries the database for the required sensor data. In our implementation we queried the database for the average value of each sensor involved (i.e. Water Level and Energy Output) per day. A list is returned to view in a JSON object. This JSOn object is then parsed and its data bounded to the charts. For the charts visualization we used the "D3" and "C3" libraries, they are JavaScript libraries for manipulating documents based on data and provide an option to bind to JSON data directly without needing anymore data parsing. One Interesting feature provided by the C3 library is the value displaying when the user hovers on the chart as show below.

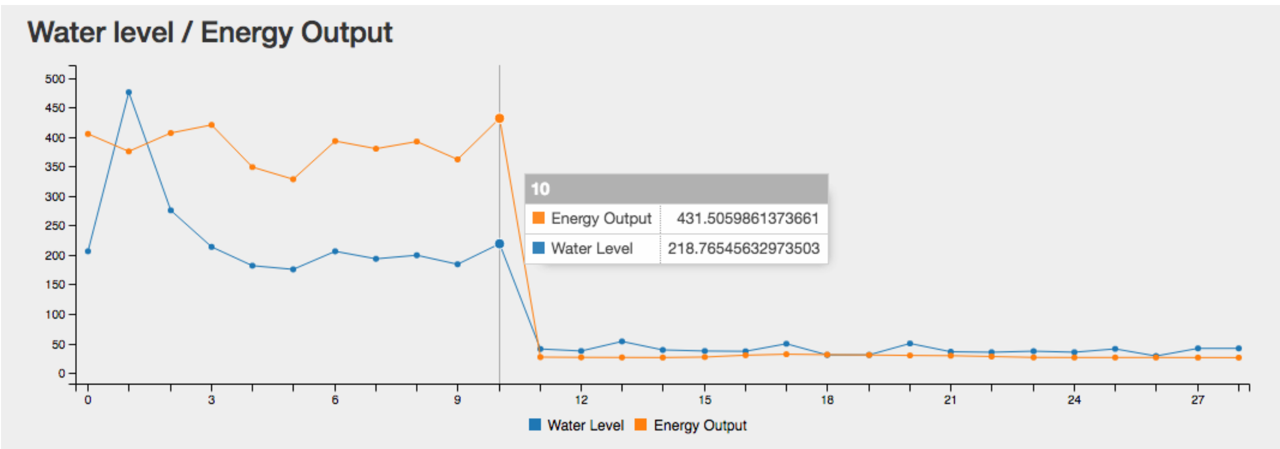

Figure 5.15: Display Data on Mouse Hover

Chapter 6

Evaluation

In this chapter we will discuss how our system could be evaluated. In *ChapterTwo* we discussed the different methods used to evaluate decision support systems. Unfortunately they couldn't be applied in our scenario as our prototype was developed as a proof of concept to prove that a decision support system could benefit the connect hydro project and that combining two types of decision support can be advantageous and optimize power plants energy production.

SInce our system didn't have any users, there was no possibility to perform evaluation. Therefore, evaluation was reduced to checking that all the functionalities are included and are performing as expected. The functionalities include:

In Connect Hydro, a qualitative benefit study was conducted through meetings with experts and power plants operators. In the study a few situations were studied and discussed that if a decision support system could prevent these events from happening then it will be a great interest to them and will motivate them to implement the system and help convince other operators to join the project.

The Data visualization could be an interesting tool that could help our system predict future trends in energy production based on the history as well as some external resources like weather forecasting

Chapter 7

Conclusion

7.1 Summary

7.2 Lessons Learned

7.3 Future Research

7.3.1 Machine Learning

7.3.2 No-SQL Database

NoSQL encompasses a wide variety of different database technologies that were developed in response to the demands presented in building modern applications:

Developers are working with applications that create massive volumes of new, rapidly changing data types — structured, semi-structured, unstructured and polymorphic data.

Long gone is the twelve-to-eighteen month waterfall development cycle. Now small teams work in agile sprints, iterating quickly and pushing code every week or two, some even multiple times every day.

Applications that once served a finite audience are now delivered as services that must be always-on, accessible from many different devices and scaled globally to millions of users.

Organizations are now turning to scale-out architectures using open source software, commodity servers and cloud computing instead of large monolithic servers and storage infrastructure.

Relational databases were not designed to cope with the scale and agility challenges that face modern applications, nor were they built to take advantage of the commodity storage and processing power available today.

Launching an application on any database typically requires careful planning to ensure performance, high availability, security, and disaster recovery – and these obligations continue as long as you run the application. With MongoDB Atlas, you receive all of the features of MongoDB without any of the operational heavy lifting, allowing you to focus instead on learning and building your apps. Features include:

- On-demand, pay as you go model
- Seamless upgrades and auto-healing
- Fully elastic. Scale up and down with ease
- Deep monitoring & customizable alerts
- Highly secure by default
- Continuous backups with point-in-time recovery

7.3.2.1 NoSQL Database Types

- Document databases pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.
- Graph stores are used to store information about networks of data, such as social connections. Graph stores include Neo4J and Giraph.
- key-value stores are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or 'key'), together with its value. Examples of key-value stores are Riak and Berkeley DB. Some key-value stores, such as Redis, allow each value to have a type, such as 'integer', which adds functionality.
- Wide-column stores such as Cassandra and HBase are optimized for queries over large datasets, and store columns of data together, instead of rows.

7.3.2.2 The Benefits of NoSQL

When compared to relational databases, NoSQL databases are more scalable and provide superior performance, and their data model addresses several issues that the relational model is not designed to address:

- Large volumes of rapidly changing structured, semi-structured, and unstructured data
- Agile sprints, quick schema iteration, and frequent code pushes
- Object-oriented programming that is easy to use and flexible
- Geographically distributed scale-out architecture instead of expensive, monolithic architecture
- Selecting the appropriate data model: document, key-value & wide column, or graph model
- The pros and cons of consistent and eventually consistent systems
- Why idiomatic drivers minimize onboarding time for new developers and simplify application development

Abbreviations

RES Renewable Energy Sources

DSS Decision Support System

JDBC Java Database Connectivity

DBMS Database Management Systems

OLAP Online Analytical Processing

OLTP Online transaction processing

EIS Executive Information Systems

GIS Geographic Information Systems

GDSS Group Decision Support Systems

MDS Model-Driven DSS

AHP Analytical Hierarchy Process

SDLC System Development Life Cycle Approach

LAN Local Area Network

WAN Wide Area Network

TCP/IP Transmission Control Protocol/ Internet Protocol

IP Internet Protocol

MPC Model Predictive Control

LCS Local Control Strategy

CCS Collaborative Control Strategy

CCS Centralized Control Strategy

Ajax Asynchronous Javascript

JSON JavaScript Object Notation

SQL Structured Query Language

XML eXtensible Markup Language

DI Dependency Injection

IOC Inversion of Control

Bibliography

[1] Decision tree - classification. http://www.saedsayad.com/decision_tree.htm. Accessed: 2017-06-22.

[2] Designing and evaluating dss user interfaces. http://www.dsc.ufcg.edu.br/~garcia/cursos/SAD/Notas/Cap5DSSResources.pdf. Accessed: 2017-06-12.

[3] Mysql. http://searchenterpriselinux.techtarget.com/definition/MySQL. Accessed: 2017-06-21.

[4] Rapid prototyping approach. https://www.slideshare.net/delaco/1-rapid-prototyping-model. Accessed: 2017-06-12.

[5] Responsive web design definition. https://developers.google.com/web/fundamentals/design-and-ui/responsive/. Accessed: 2017-06-18.

[6] Responsive web design image. http://www.business2community.com/web-design/5-reasons-business-needs-responsive-website-01640405#dPBrxMkk7MrjOeWx.97. Accessed: 2017-06-18.

[7] Responsive web design origin. https://alistapart.com/article/responsive-web-design/. Accessed: 2017-06-18.

[8] Small scale hydropower. http://www.small-hydro.com/about/small-scale-hydrpower.aspx. Accessed: 2017-06-13.

[9] Web development definition. https://www.techopedia.com/definition/23889/web-development. Accessed: 2017-06-18.

[10] What is mysql? https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html. Accessed: 2017-06-21.

[11] Aswani. Spring framework advantages and disadvantages. http://www.aksindiblog.com/spring-framework-advantages-disadvantages.html. Accessed: 2017-06-21.

[12] Bhargava, H., and Power, D. Decision support systems and web technologies: a status report. *AMCIS 2001 Proceedings* (2001), 46.

[13] Experts, M. Decision support systems. https://www.managementstudyguide.com/decision-support-systems-articles.htm. Accessed: 2017-06-12.

[14] Farina, M., and Scattolini, R. Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. *Automatica 48*, 6 (2012), 1088–1096.

[15] FERGUSON, N. What's the difference between frontend and backend? https://careerfoundry.com/en/blog/web-development/whats-the-difference-between-frontend-and-backend/. Accessed: 2017-06-20.

[16] Fridleifsson, I. B. Geothermal energy for the benefit of the people. *Renewable and sustainable energy reviews 5*, 3 (2001), 299–312.

[17] Garrett, J. J., et al. Ajax: A new approach to web applications.

[18] Gimmer, C. Top 5 reasons to use bootstrap. https://bootstrapbay.com/blog/reasons-to-use-bootstrap/. Accessed: 2017-06-20.

[19] Gorry, G. A., and Morton, M. S. S. A framework for management information systems. *MIT Sloan Management Review 30*, 3 (1989), 49.

[20] Hall, D. Cooling the greenhouse with bioenergy. *Nature 353* (1991), 11–12.

[21] Herzog, A. V., Lipman, T. E., and Kammen, D. M. Renewable energy sources. *Encyclopedia of Life Support Systems (EOLSS). Forerunner Volume-'Perspectives and Overview of Life Support Systems and Sustainable Development* (2001).

[22] Hiemstra-van der Horst, G., and Hovorka, A. J. Fuelwood: The "other" renewable energy source for africa? *Biomass and bioenergy 33*, 11 (2009), 1605–1616.

[23] José Luis Carrasco, Andrea Pain, D. S. Hydropower (small-scale). http://www.sswm.info/content/hydropower-small-scale. Accessed: 2017-06-13.

[24] Katamreddy, S. P. R. Why spring boot? https://dzone.com/articles/why-springboot. Accessed: 2017-06-21.

[25] Khodashahri, N. G., and Sarabi, M. M. H. Decision support system (dss). *Singaporean Journal of Business Economics and Management Studies, Vol1*, 6 (2013).

[26] lawomir Wilczyn ski, S. Front-end frameworks. introduction (part 1). http://www.merixstudio.com/blog/front-end-frameworks-introduction-part-1/. Accessed: 2017-06-20.

[27] LLC, T. System development life cycle. https://shaparo-mis.wikispaces.com/System+Development+Life+Cycle. Accessed: 2017-06-12.

[28] Makowski, M., and Wierzbicki, A. P. Modeling knowledge: Model-based decision support and soft computations. In *Applied decision support with soft computing*. Springer, 2003, pp. 3–60.

[29] Markus Jager, Markus M. Schwarzb, D. A. B. P. J. K. Connecting small, private and independent hydro power plants to increase the overall power generating efficiency. *Procedia Computer Science 00* (2016), 000–000.

[30] Mysiak, J., Giupponi, C., and Rosato, P. Towards the development of a decision support system for water resource management. *Environmental Modelling & Software 20*, 2 (2005), 203 – 214. Policies and Tools for Sustainable Water Management in the European Union.

[31] Nare, H. Spring dependency injection. http://nataraz2java.blogspot.co.at/2013/11/spring-dependency-injection.html. Accessed: 2017-06-21.

[32] Paish, O. Small hydro power: technology and current status. *Renewable and sustainable energy reviews 6*, 6 (2002), 537–556.

[33] Panwar, N., Kaushik, S., and Kothari, S. Role of renewable energy sources in environmental protection: a review. *Renewable and Sustainable Energy Reviews 15*, 3 (2011), 1513–1524.

[34] Powell, R. Dm review: A 10 year journey. *DM Review, February* (2001).

[35] Power, D. J. A brief history of decision support systems.

[36] Power, D. J. Decision support systems types. http://dssresources.com/dsstypes/. Accessed: 2017-06-11.

[37] Power, D. J. Web-based and model-driven decision support systems: concepts and issues. *AMCIS 2000 Proceedings* (2000), 387.

[38] Power, D. J., and Kaparthi, S. Building web-based decision support systems. *Studies in Informatics and Control 11*, 4 (2002), 291–302.

[39] Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J., Sharda, R., and Carlsson, C. Past, present, and future of decision support technology. *Decision support systems 33*, 2 (2002), 111–126.

[40] Stewart, B. T., Venkat, A. N., Rawlings, J. B., Wright, S. J., and Pannocchia, G. Cooperative distributed model predictive control. *Systems & Control Letters 59*, 8 (2010), 460–469.

[41] Timofeyev, M. Hydro power plants. *National Research Tomsk Polytechnic University.*.

[42] Wodehouse, C. How ajax (asynchronous javascript + xml) works. https://www.upwork.com/hiring/development/how-ajax-works/. Accessed: 2017-06-20.

[43] Youm, I., Sarr, J., Sall, M., and Kane, M. Renewable energy activities in senegal: a review. *Renewable and Sustainable Energy Reviews 4*, 1 (2000), 75–89.

Nada Ossama

Date and Place of Birth: July 23rd, 1990 in Cairo, Egypt
Gender: Female
Nationality: Egyptian
Marital Status: Single
Address: 10 Julius Raab Strasse, 4040 Linz, Austria
Mobile Phone: +43 688 64246359
Email: nadaossama90@gmail.com

Education

Sep 2016 - Present : Master in Informatics,Johannes Kepler University (JKU)- Graduate July 2017.
Sep 2008 – Jul 2012: Bachelor of *Science,* GPA: 2.76**, German University in Cairo (GUC)*
Sep 2005 – Jul 2008: British IGCSE High school certificate, Grade: 110%, *Egyptian Language School, Cairo*

**GPA on a scale of 1-5.0, with 1 being the highest GPA possible and 5.0 being the lowest GPA possible.*

Work Experience and Internships

July 2012 – July 2016: **Customization Engineer**, *Amadeus IT Group*

- Participated in Workshops with Customers to determine their needs and propose a solution
- Wrote functional/technical specification and Solution Architecture documents
- Wrote User Guide and Deployment Guides
- Developed applications as per functional specifications document using .NET framework and Amadeus Web Services, followed by testing and delivering the applications
- Provided second level support and training
- Trained new Members and offered them support

July 2010 – Sep 2010: **Database Administration Intern**, *Vodafone Egypt*

- Developed an Enterprise Manager application using Java
- Learned different Types of support by rotating with the support Team

Technical Skills

Programming Languages: C#, Java, VB.NET
Web Technologies: HTML, JavaScript, Spring, ASP.NET, XML, WPF, Web Services
Databases: MS SQL Server, SQLite, MySQL
MS-Office: Excel, Word, Access, Powerpoint

Courses

Dec 2012: Delivering the Extra Mile, *Logic Training & HR Development*
Nov 2012: Amadeus Cryptic Basic Course, *Amadeus IT Group*
Sep 2012: Presentation Skills Course, *Logic Training & HR Development*

Extracurricular Activities

Feb 2012 – May 2012: Junior Teaching Assistant, *Introduction to computer science II for 2nd semester management students, German University in Cairo*

Feb 2012 – May 2012: Junior Teaching Assistant, *Introduction to computer programming for 4th semester engineering students,, German University In Cairo*

Jul 2011 – May 2012: Vice-president, *GUC Theater and Cinema School , German University in Cairo*

Feb 2011 – May 2011: Junior Teaching Assistant, *Introduction to computer science for 1st semester management students, German University In Cairo*

Feb 2010 – Mar 2010: Head Of Ushering Team, *Google I/O Event, German University in Cairo*

Feb 2010 – Mar 2010: Planning Team Member, *Google I/O Event, German University in Cairo*

Jul 2007 – Jul 2009 : Events Committee Member, *Flying Colors Team, British Council*

Language Skills

Arabic: Mother Tongue
English: Fluent
German: Basic (Learning)

Awards and Achievements

Sep 2016: Scholarship at the Johannes Kepler University

Sep 2007: Scholarship at the German University in Cairo

Hobbies

Sport activities
Reading
Bike Riding
Travelling and discovering new places

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Masterarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder inhaltlich entnommenen Stellen deutlich als solche kenntlich gemacht habe.

Linz, Juni 2017 Nada Ossama