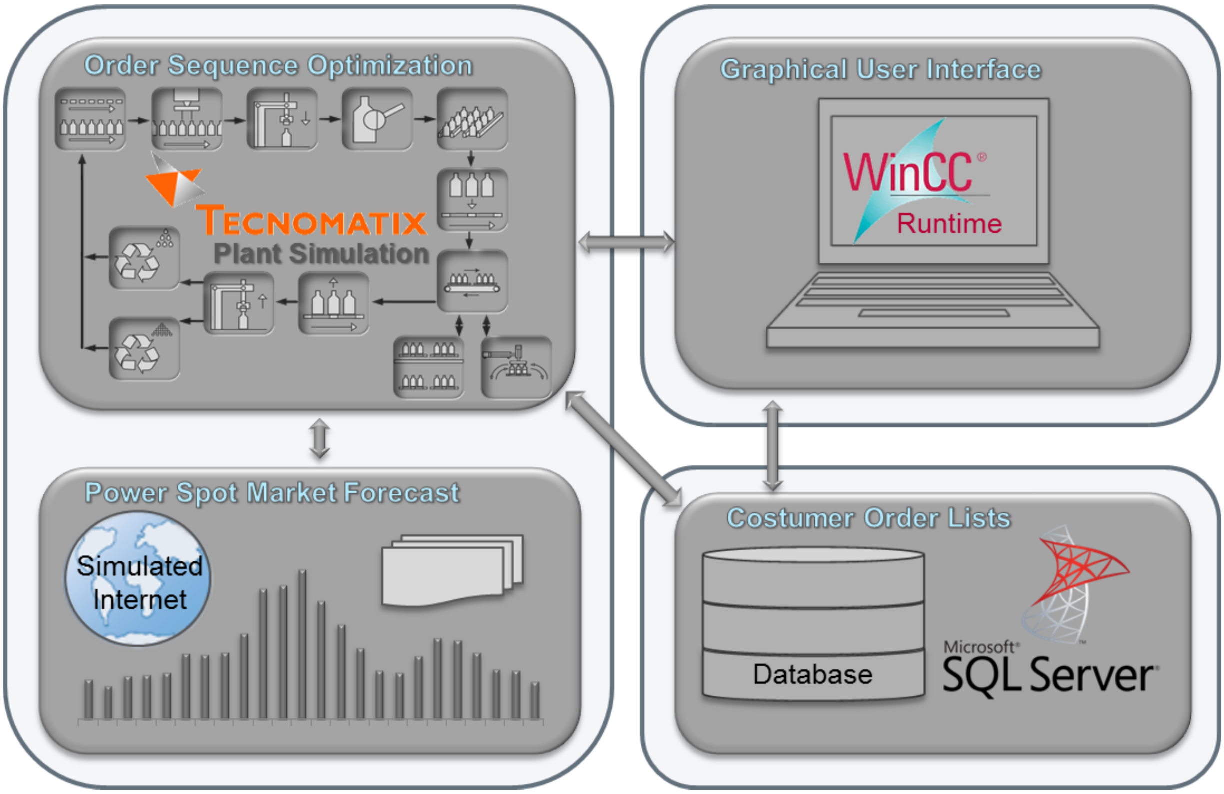

Implementation of an example application facilitating Industrie 4.0

Master's thesis in Systems, Control and Mechatronics

Selmar Allenhof

Department of Signals and Systems
Automatic control, Automation and Mechatronics
Chalmers University of Technology
In collaboration with Siemens AG
Gothenburg, Sweden 2015

Abstract

The term ‘Industrie 4.0’ is widely discussed by researchers, standardization associations and industrial companies. It is often referred to as a collective term for value-chain organization with its roots in the concepts of ‘Internet of Things and Services’ and ‘Cyber-Physical Systems’. With the development of Industrie 4.0 comes the need for engineers to adapt to new standards and expand their level of expertise in related disciplines. As this thesis was carried out at Siemens AG within the department of industrial automation and electrical drives, only for this department relevant disciplines were explored. New technical expertise is imparted to the engineers by Siemens AG in the form of technical training modules within a training institution.

This thesis explores the scope of Industrie 4.0 and subsequently explains the implementation of an industrial example application. The example application elucidates certain parts of Industrie 4.0 and is based on a local industrial test facility at the Siemens AG establishment. The purpose of this application is to present an approach to the implementation of new training modules and to show how engineers need to be educated and trained in order to incorporate the principles of Industrie 4.0. A concluding discussion is made on the relation of the example to the topic Industrie 4.0 and the possibilities of implementing the application into an engineering training module.

Keywords: Industrie 4.0, Smart Factory, Order Sequence Optimization

Acknowledgements

This master's thesis was undertaken in cooperation with Siemens AG. I would like to thank them for this interesting topic and the freedom to specify the details myself, for providing a nice work environment and all the necessary equipment to work efficiently. Thank you Georg Schoeler, Reinhard Geissel and Hans Peter Kagerer for listening to my process presentations and providing helpful feedback. I specially thank my supervisor Michael Elting for always checking on me and supporting me where he could.

I furthermore would like to thank my university supervisors Martin Fabian and Åsa Fast-Berglund. You always replied to my emails with calm and constructive answers. Thank you for being patient and checking on me, when we hadn't been in touch for a while.

Finally I would like to thank my family and partner for supporting me during the process of the thesis. Special thanks also to all the friends that assisted me in reviewing my work and helped me create a thesis that can be understood by those other than me.

Contents

List of Figures

List of Tables

Abbreviations

AGV	Automated Guided Vehicle
ANSI/SPARC	American National Standards Institute/ Standards Planning and Requirements Committee
API	Application Programming Interface
BMBF	Bundesministerium für Bildung und Forschung (English: Federal Ministries of Education and Research)
BMWi	Bundesministerium für Wirtschaft und Energie (English: Federal Ministries of Economic Affairs and Energy)
COM	Component Object Model
CPPS	Cyber-Physical Production System
CPS	Cyber-Physical System
CSV	Comma Separated Values
DB	Database
DBMS	Database Management System
DBS	Database System
DCOM	Distributed Component Object Model
DDL	Data Definition Language
DIN	Deutsches Institut für Normung e.V. (English: the German Institute for Standardization)
DML	Data Manipulation Language
DRL	Data Retrieval Language
DSL	Data Security Language
EEX	European Energy Exchange
EPEX SPOT	European Power Exchange
GMA	Gesellschaft Mess- und Automatisierungstechnik (English: Society of Measurement and Automation Engineering)
HMI	Human Machine Interface
HTML	HyperText Markup Language
ICT	Information and Communications Technology
ID	Identification
IEC	International Electrotechnical Commission
IoS	Internet of Services
IoT	Internet of Things
IP	Internet Protocol
ISO	Internation Organization for Standardization
IT	Information Technology
ITA	Industry Technical Agreements
M2M	Machine-to-Machine communication
ODBC	Open Database Connectivity

OPC	Openess, Productivity and Connectivity
OPC A&E	OPC Alarm & Events
OPC AC	OPC Alarms and Conditions
OPC DA	OPC Data Access
OPC HA	OPC Historical Access
OPC HDA	OPC Historical Data Access
OPC UA	OPC Unified Architecture
OS	Operating System
OSI	Open Systems Interconnection model
PAS	Public Available Specifications
PC	Personal Computer
PLC	Programmable Logic Controller
PLM	Product Lifecycle Management
REST	Representational State Transfer
RFID	Radio-Frequency Identification
SBO	Simulation-Based Optimization
SCADA	Supervisory Control And Data Acquisition
SMLC	Smart Manufacturing Leadership Coalition
SoA	Service-oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TIA	Totally Integrated Automation
TLS	Transport Layer Security
TR	Technical Reports
TS	Technical Specifications
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
US	United States
VDE	Verband der Elektrotechnik, Elektronik und Informationstechnik (English: German Commission for Electrical, Electronic and Information Technologies)
VDI	Verein Deutscher Ingenieure (English: Association of German Engineers)
VM	Virtual Machine
WinCC	Windows Control Center
XML	Extensible Markup Language
XML-RPC	Extensible Markup Language - Remote Procedure Call

1 Introduction

1.1 Motivation

Industrie 4.0 is a recently coined term that encapsulates the idea of the fourth Industrial Revolution. This fourth Industrial Revolution is as important as the industrial revolution that occurred during the 1800’s when new manufacturing processes transformed the industrial landscape. Industrie 4.0 has been included as a government directive in Germany, but has not yet gained widespread use throughout the rest of the world. Essentially, Industrie 4.0 is a collective term for a set of principles, concepts and technology that seeks to digitize factories so that all machinery can communicate with each other to create the most efficient system possible. Industrie 4.0 therefore also requires new skills of the engineers to work across disciplines in order to enable these massive digital changes.

This thesis is a contribution to explore what new competencies are required by engineers. A computer program, or application, has specifically been written for this thesis to explore what competencies are needed and to show how Industrie 4.0 can be used to create an efficient system of factory machines. In the first part of the thesis an overview is given of the essential ideas and scope of Industrie 4.0. Following that, an exemplary technical application is implemented in order to elucidate some core ideas and innovations brought by Industrie 4.0. The competence gaps of specialized engineers are explored in the context of arising interconnections between different technical skills and expertise. The example implementation is achieved by using existing solutions that will be complemented with future technical content.

New technical expertise is imparted to the engineers by Siemens AG staff in the form of technical training modules in a training institution. Hence, the exemplary application will represent the base upon which to derive requirements for an individual competence development in technical occupations. The concept of interdisciplinary education and training is oriented to today's current business practice, where engineers are trained in individual technical training to prepare them for the deployment in new projects. This thesis subsequently highlights interconnections between certain technical expertise and describes an approach to how new training modules can be derived from the implementation example.

1.2 Research Questions

According to Prof Dr Bauernhansl [1], Industrie 4.0 has to cope with an increasing amount of customized products. This relates especially to the final assembly, where the number of variants is the richest. Ongoing research on cognitive automation strategies as examined in [2] shows also that engineers who develop and program such assembly lines need a changed individual competence development. As stated in [3], Cyber-Physical System (CPS)-based manufacturing and service innovations are two inevitable trends and challenges of Industrie 4.0 for manufacturing industries. The paper addresses the challenges in dealing with big data issues of rapid decision-making for improved productivity. In [4] it is further argued that CPS is the technological driver; and the framework for collaborative practice is the organizational driver of Industrie 4.0. Collaboration in this sense means the coordination, cooperation and communication of information in a factory. To implement the developments brought about by Industrie 4.0 into the manufacturing industry, engineers need corresponding specialized training.

This thesis aims to answer the following research questions:

- What is the scope of Industrie 4.0 in terms of industrial production systems?
- Which technical implementation example will cover some of the core ideas of Industrie 4.0 and function as an approach for training modules to prepare engineers for an Industrie 4.0 mindset?

1.3 Scope and Delimitations

The challenge of this thesis is to investigate major and sustainable trends within ongoing research on the wide topic of Industrie 4.0. Since the thesis was carried out at Siemens AG at the Department for Industrial Automation and Electrical Drives, the focus of the thesis is set on technical changes within the environment of industrial production systems. Industrie 4.0 is described as a structure in which logistics and manufacturing systems intensively use the globally-available information and communication network and thereby form a so-called Cyber-Physical Production System (CPPS). The scientific challenge in this thesis is to focus on some specific technical novelties and implement them in an example training facility.

Due to changes in the markets in an increasingly global world economy, production businesses are put under an increasing pressure of time and costs. Today's products are pushed to launch in ever shorter periods of time while keeping the quality high and the price preferably low. The production facilities in the added value-chain of a manufacturing business are hence the core element of their value generation. To stay competitive on a global market, the facilities have to be operated as economically and insensitively to interferences as possible.

Improvement approaches for a smooth plant operation and the search for optimization potential are constantly in focus. Beneficial to this focus is the digital documentation of the plant through the whole life cycle. Therefore, a so-called 'Digital Twin' of the production facility is created. With the help of software tools, the Digital Twin synchronizes data over the whole life-cycle with the real facility. The data, in this respect all methods and changes the real facility is performing, is often referred to as 'Big Data'. A gapless flow of information between the individual subsystems is of utmost importance. This is intended to ensure that the data collection can contribute to the optimization of the system while in operating state. For this purpose, it is also necessary to exchange relevant information across all levels of planning and hence, simulation software is already applied during the planning of the facility. During the operating phase, simulation-models help compare different production scenarios allowing predictions about the future behavior of the system to be made. According to [5], the data obtained through the simulation should be evaluated so that production planning decisions can be derived. For the implementation of a Digital Twin and integration of the resulting data in the planning system, suitable interfaces for the information flow between all levels of the company have to be applied.

At the management level, so-called 'dashboards' are used. Dashboards support decision processes of the management with graphical and statistical representations. To facilitate production relevant decisions for the operators, and thereby enable cognitive automation strategies as stated in [2], dashboards are also to be deployed at the shop floor level.

To arrange the sequence of events in a production facility as efficiently as possible, material flow simulations can be used to obtain information about production time, material requirements, and expense. Shrouf and Miragliotta state in [6] that energy requirement considerations will increase as well, due to rising energy prices, changing consumer behaviors, and increasing ecological awareness. The paradigm of 'Internet of Things' (IoT) shows the potential to raise the awareness and visibility of energy consumption through the usage of smart meters and sensors at the shop floor level. Real-time energy consumption data from production line processes can therefore be collected without difficulty and then analyzed to improve energy-aware decision making. According to the VDI guideline 4499 sheet 2 'Digital factory operation', a simulation model that is running in parallel to the real production is referred to as 'Production Accompanied Simulation'. It is explained that planning and control of the manufacturing processes are supported with information from the simulation, and it is of importance that the two systems are able to constantly exchange information throughout the production. When the energy requirements of the facility are considered and related to specific products that are to be manufactured, an optimization procedure can then be applied to the planning and sequencing of the orders. The sequence of orders on the SCADA system of the facility results in a certain pattern of energy requirements for the production line. Based on the current shape of energy prices on the market, a certain pattern of resulting costs can subsequently be derived. In this specific example, the simulation model helps to optimize the sequence of orders by simulating various scenarios. The overall resulting costs for the production can be lowered while taking constraints, such as the delivery due time, into consideration.

1.4 Structure

In this thesis, an example application is implemented to serve as basis for the compilation of an engineering training module facilitating Industrie 4.0. In order to comprehend the term 'Industrie 4.0', **Chapter 2** selects four descriptive components, addresses the related technical and organizational concepts, and explains six design principles to achieve an Industrie 4.0 scenario.

The second part of the thesis deals with the conception and practical implementation of an Industrie 4.0 example application. The application is assigned to run on a local test facility and aims to manipulate the product order sequence. In order to understand the applied methods, structural aspects, and the need for such an application, **Chapter 3** presents certain preliminaries on the topic of the example application. In **Chapter 4** the required software application and tools are introduced and explained. Finally in **Chapter 5** the example application is designed and implemented.

Chapter 6 summarizes the thesis by pointing out the relationship between the example application and Industrie 4.0. Moreover, an outlook is given presenting the implementation possibilities as a training module for engineers, or as a marketable service tool.

2 Industrie 4.0

In this chapter the term 'Industrie 4.0' is defined based on a literature review. First a short overview of how the idea of Industrie 4.0 came into being and what its vision and basic goals are is given. Following that, certain requirements are presented that encouraged the industry to implement Industrie 4.0 scenarios and select potential use cases for further investigations. The chapter then presents four basic components of Industrie 4.0, provides a definition of the term and formulates six design principles. Subsequently, the need for standardization is outlined and the relevant architecture focusing on service-orientation is presented. Finally, the changing demands on engineers are explained and conclusions on Industrie 4.0 are made.

2.1 Introduction

According to [7], the German federal government follows an initiative called the 'High-Tech Strategy 2020 for Germany', that aligns with the global challenges in the five areas of climate/energy, health/nutrition, communication, mobility and security. It promotes key technologies by formulating ten future projects to realize their goals and visions. One of the government's high-tech-strategy projects is named 'Industrie 4.0' and hence numerous academic publications, conferences, and practical articles have focused on it.

According to the 'German Standardization Roadmap' [8], an innovation such as Industrie 4.0 requires a close cooperation between research, industry, and the standardization. As can be seen in Figure 1, research institutions bring the methodological foundations for a new innovation, Standardization ensures stability and investment security, and the Industry tests the new concepts on practicality and market relevance.

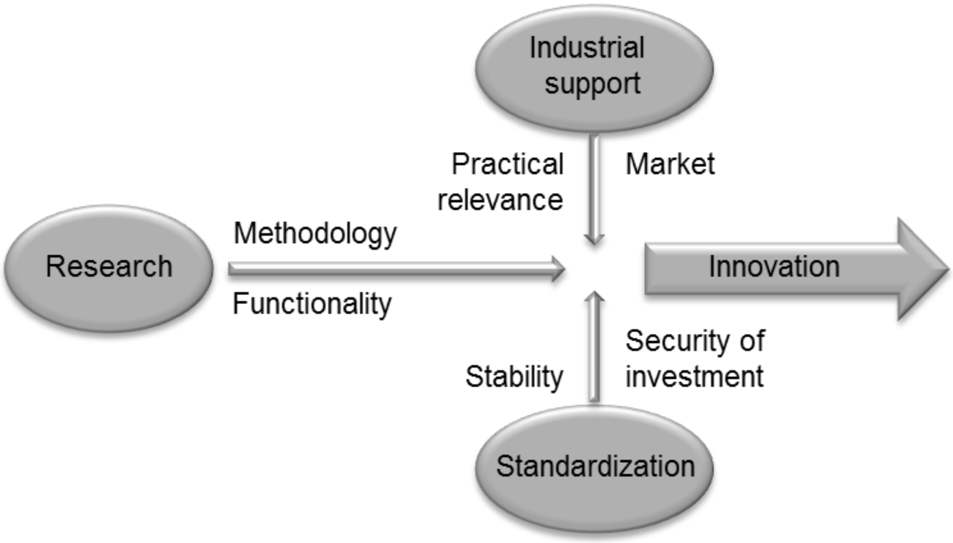

Figure 1: Three innovation driving units of Industrie 4.0

There are two reasons for interest in Industrie 4.0. First, never before was an industrial revolution announced a-priori and not observed ex-post [9]. That gives various research centers and companies the chance to actively form this revolution. The second reason for the fascination with Industrie 4.0 is the forecast of a vast economic impact. The project has potential to develop entirely new business models, products and services as well as considerably increased operational efficiency [10] [11]. As estimated in a recent study [12], developments on this project will contribute as much as 78 billion euros to the German GDP by the year 2025.

As explained in [13], Industrie 4.0 primarily promotes a computerization in the field of production technology. The project facilitates the vision of a Smart Factory, which is characterized by mutability, resource efficiency, and ergonomic design, as well as the integration of customers and business partners in value-added processes of the factory. The factory is to be extended with ideas from information technology to enable intelligent behavior. The questions of how factories should be built, organized, and structured in the future, and how existing factories can be adapted arises and is to be answered.

Even though Industrie 4.0 presently has a high importance for many research institutions, companies, and universities in the German-speaking area, a commonly established definition of the term does not exist. It is consequently difficult to debate the topic on an academic level and implement Industrie 4.0 scenarios. The various contributions of all participants during the past three years have made the term more unclear than concrete [14]. Even the organizations ‘Platform Industrie 4.0’ and ‘Industrie 4.0 Working Group’ do not provide a clear definition, only describe basic technologies, selected scenarios, and the vision Industrie 4.0 aims at. As stated in [15], companies and practitioners need a certain systematization of knowledge in order to identify and implement Industrie 4.0 scenarios. The paper [15] therefore formulates the six design principles ‘Interoperability’, ‘Virtualization’, ‘Decentralization’, ‘Real-Time Capability’, ‘Service Orientation’ and ‘Modularity’ as explained in Chapter 2.7.

Furthermore, Hermann [15] presents four trends of Industrie 4.0, which this thesis incorporates: ‘Cyber-Physical Systems’ (CPS), ‘Internet of Things’ (IoT), ‘Internet of Services’ (IoS), and ‘Smart Factory’. They are further explained in Chapter 2.5.

In the final report of the ‘Industrie 4.0 Working Group’ [10], they describe their vision of Industrie 4.0 as follows. Companies of the future will found global networks and will thereby be capable of incorporating their warehouse systems, machinery and production facilities. In the manufacturing environment, these so-called Cyber-Physical Systems consist of smart storage systems, machineries, and production facilities, which autonomously exchange information, prompt actions, and control each other independently. This automation implies essential advances for the industrial processes within manufacturing, material usage, engineering, life-cycle and supply chain management. Products of the future will be distinctively identifiable, able to be located at all times, and know their own history of events, current status, and alternative courses to reach their target state. The networks within the embedded manufacturing factory are vertically connected to the business processes, as well as horizontally networked to detached value networks while being able to communicate and make decisions in real time. To obtain this, end-to-end engineering from the moment an order is placed up to the outbound logistics is required throughout the entire value-chain.

2.2 Background

In 2011, the term ‘Industrie 4.0’ was presented to the public for the first time at the Hannover Fair by an association of representatives from politics, academia, and business as an approach to strengthening the competitiveness of the German manufacturing industry [16]. The term ‘Industrie 4.0’ refers to the fourth industrial revolution. It has been preceded by three other industrial revolutions in the history of mankind. The first industrial revolution came in the second half of the 18th century with the introduction of mechanical production systems using water and steam power. The second revolution then introduced mass production in the 1870s by the division of labor, namely Taylorism, and the use of electric power. In the third industrial revolution, the use of advanced electronics and information technology was developed to further automate production processes. This was also called the 'Digital Revolution', and occurred in the 1970s. [17]

The German federal government supported the idea of Industrie 4.0 and integrated it into their high-tech strategy program. The government subsequently formed the 'Industrie 4.0 Working Group', which published a set of implementation recommendations in October 2012. The final report of the working group was eventually presented in April 2013 [10].

Simultaneously to the final report of the Industrie 4.0 Working Group, the 'Platform Industrie 4.0' formed out of the industry associations Bitkom, VDMA, and ZVEI. Their aim is to coordinate future activities of Industrie 4.0. Currently they are working on a reference model to structure basic ideas of the project [18]. Coordination and funding activities in Germany are borne by the Economic Affairs and Energy (BMWi) and the Federal Ministries of Education and Research (BMBF).

Although the term 'Industrie 4.0' is not as familiar outside the German-speaking area, similar activities can be found globally. In the United States, an initiative known as the Smart Manufacturing Leadership Coalition (SMLC) is also working on the future of manufacturing [19]. It is a non-profit organization of suppliers, manufacturing and technology companies, government agencies, universities and laboratories. The US government also supports research and development activities for the use of the Industrial Internet with a two billion dollar fund [20] and General Electric precisely names one of their initiatives the 'Industrial Internet'. According to [21], complex physical machinery with sensors and software connected to the network are defined in order to better predict, control and plan business and societal outcomes. Additional similar ideas to Industrie 4.0 can be found under the terms 'Integrated industry' [22] or 'Smart Manufacturing' [23].

2.3 Value-adding networks

A value-chain in the production industry represents all steps within the production of a product order as a structured string of activities. These activities create values, consume resources, and are connected to each other through processes. The concept was first published in 1985 by Michael E. Porter in his book 'Competitive Advantage'. According to the 'German Standardization Roadmap' [8], CPS will contribute to the autonomy of sub-processes within the different value-chains in a production facility. This will support both short-term flexibility as well as the medium-term variability in response to the increasingly shorter and weightier external influences and, thus, improve the resilience of production. In Figure 2 the four dimensions of the value-adding processes in industrial production are represented. Namely they are the business process, and the product, factory, and technology life cycles.

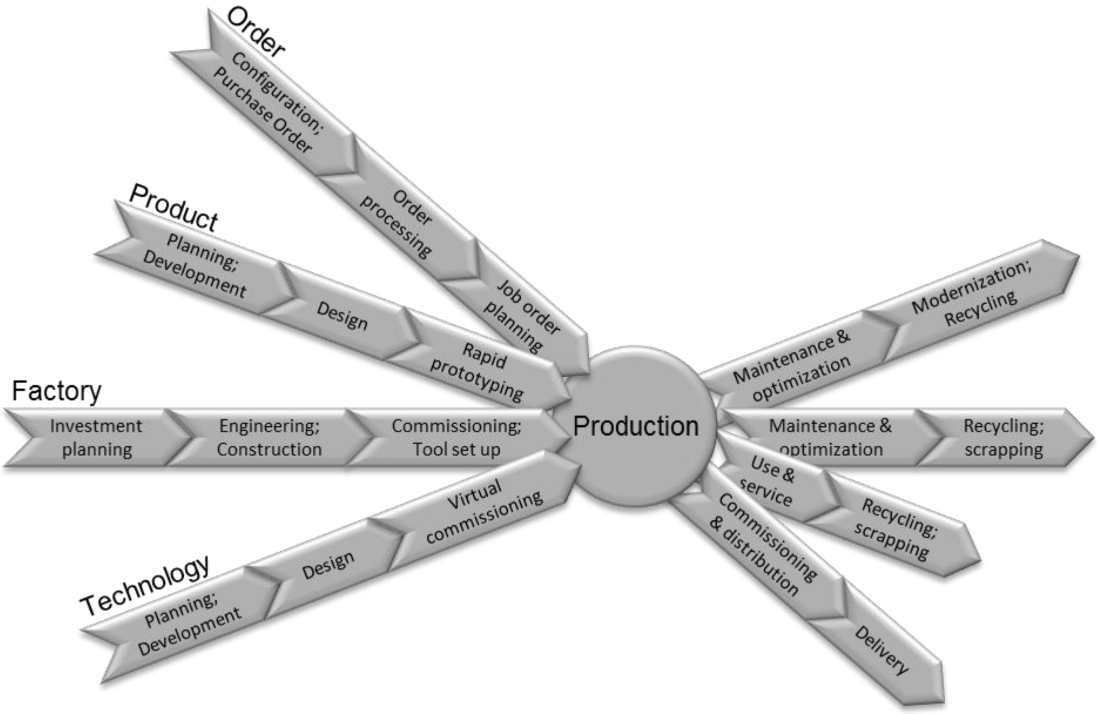

Figure 2: Four added value processes in industrial production

The foundation of Industrie 4.0 is the availability of all relevant information in real time through integration of all entities involved in the value-added processes, as well as the ability to derive the optimum value flow from the available data at all times. Dynamic, real-time optimized, self-organizing enterprise encompassing value networks are created by connecting humans, objects, and systems. They can be optimized according to criteria like costs, availability, and consumption of resources.

2.4 Industrial Requirements

The introduction of Internet technologies in the industry causes understandable concerns for many plant operators. Their facilities combine investments, know-how, production, and profitability. Many visions of Industrie 4.0 seem hardly tangible for today's production systems. The communication of production-relevant devices with a so-called ‘cloud’ is often perceived as a potential hazard. Therefore, the industry sets certain demands on Industrie 4.0. Drath identifies in [9] the following requirements for an industrial acceptance:

Security of Investment: Industrie 4.0 must gradually be introduced into existing production facilities and equipment.

Stability: Services available through Industrie 4.0 (IoS) must never jeopardize the production, neither by failure or malfunction nor by uncoordinated intervention. Production systems place increased demands on characteristics such as availability, real-time, reliability, durability, robustness, productivity, costs, security and other so-called nonfunctional properties. The demands on these properties must stay unaffected by Industrie 4.0.

Controllability: Access to plant-related data and services is a prerequisite for value creation in Industrie 4.0, but it has to be controllable. Especially write-access to production-related equipment, machines, or systems requires a special review body which ensures the validity of the procedure in the context of the total production.

Security: Prevention of unauthorized access to data or services must exist.

2.5 Components

In order to structure and specify the idea of Industrie 4.0 in a working paper from the Technical University of Dortmund Herman et al. did an intensive literature review on the topic [15]. The authors reviewed 200 publications that could be found by applying the search terms ‘Industrie 4.0’ and ‘Industry 4.0’. They subsequently identified 15 keywords that most often occurred in the context of Industrie 4.0. The paper further groups the keywords into eight representative keyword groups, which can be found in Table 1. Out of the eight groups, [15] identifies four key components of Industrie 4.0: Cyber-Physical Systems, Internet of Things, Internet of Services, and Smart Factory. The latter four groups in Table 1 are not found to be independent key components. Smart Products can be seen as a subcomponent of Cyber-Physical Systems, Machine-to-Machine communication is considered to be an enabler for Internet of Things, and Big Data, and Cloud Computing are data services which use the information inside Industrie 4.0 applications, but do not represent independent Industrie 4.0 components.

Table 1: Industrie 4.0 components (as identified in [15])

Keyword (group)	Number of Publications in which Keyword (group) occurred
Cyber-Physical Systems (CPS)	46
Internet of Things (IoT)	36
Smart Factory	24
Internet of Services (IoS)	19
Smart Product	10
Machine-to-Machine (M2M)	8
Big Data	7
Cloud	5

In the following sections, the identified Industrie 4.0 components are explained, the link to Industrie 4.0 is elucidated and application examples are given.

2.5.1 Cyber-Physical Systems (CPS)

According to [11], a substantial element of Industrie 4.0 is the union of the physical and virtual world. This fusion is realized through Cyber-Physical Systems (CPS). The technical committee 7.20 ‘Cyber Physical Systems’ and 7.21 ‘Industrie 4.0’ of the VDI-GMA describe a CPS in [24] as follows: ‘A CPS is a system that connects real physical objects and processes with virtual objects and processes through an information network, which is open, partly global, and continuously connected. Optionally, a CPS uses local or external services, uses human-machine interfaces, and offers the possibility to dynamically adapt the system at runtime.’ The CPS research agenda in [25] further explains that CPS are comprised of embedded systems, production, logistic, engineering, coordination, and management processes as well as Internet services. They immediately capture physical data through sensors and generate physical actions through actuators. The systems are connected over digital networks and thereby use worldwide available data and services.

As an example of a Cyber-Physical System, future possibilities for traffic light systems are exemplified in [9]. The real physical traffic lights log into a centralized registration system, instantiate a virtual copy of their identity and publish their planned schedule. In addition, vehicles participating in road traffic are also able to log into the registration system and download the schedules of nearby traffic lights. Based on the vehicle's location, velocity, weather data, and other traffic-sensitive information, they are able to adjust their route and velocity to optimize their journey. Emergency vehicles like ambulances or police cars could also be able to initiate a change in the scheduled behavior of the traffic lights, thereby ensuring that the traffic light is green as the vehicle approaches.

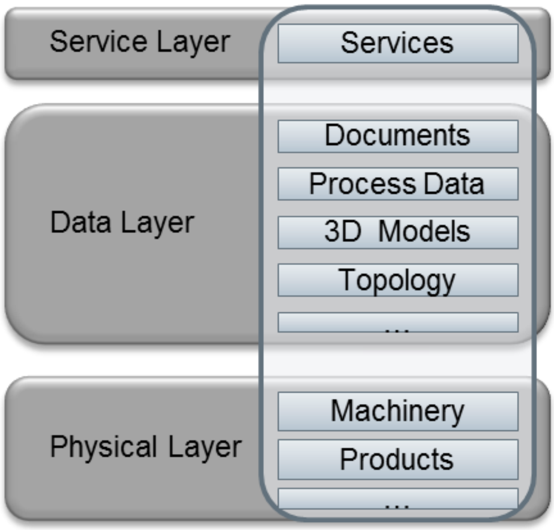

Figure 3: Three Layers of a Cyber-Physical System

Figure 3 summarizes the concept of CPS according to [9] in three layers. In the bottom layer are the physical objects, which are the intelligent, self-exploratory, and self-diagnostic assets in systems (compare to the real vehicles and traffic lights in the example). In the middle layer are the data and models of the physical objects (virtual instances of traffic lights). In the upper service layer new products and services will be developed (for instance a service that evaluates the traffic light schedules and subsequently changes the vehicles velocity to improve fuel consumption).

2.5.2 Internet of Things (IoT)

The Internet of Things (IoT) is derived from the idea that the computer will progressively disappear as a device in the future and will henceforth be replaced by 'intelligent objects and things'. Instead of being a subject of human attention itself, these intelligent, ever-smaller embedded objects aim to support human activities quietly in the background, without attracting or demanding any attention. Mark Weiser explains this vision in his article 'The Computer of the 21st Century' [26] for the first time.

The IoT designates the link of clearly identifiable physical objects (things) with a virtual representation in an Internet-like structure. In this ‘Future Internet’, explains [27], not only humans but also the representations of the ‘things’ are participants. The automatic identification using RFID is often viewed as the foundation for the Internet of Things. However, a unique identification of objects can also be obtained through a barcode or 2D code. Devices such as sensors and actuators extend the functionality to the collection of states and the execution of actions. Giusto et. al. summarize in [28] that IoT allow for ‘things and objects’ to interact with each other and with their neighboring intelligent objects through unique addressing schemas. According to the explanation given in Chapter 2.5.1, the intelligent ‘things and objects’ can be understood as CPS. Therefore, IoT is a network in which CPSs communicate through unique addressing schemas.

An example of IoT is the tracking of an ordered product over the internet. Delivery services today offer the possibility to follow the ordered product through their delivery stages. This is realized by a unique identification using barcodes or 2D codes. Other application examples are ‘Smart Factories’ as explained in Chapter 2.5.4, ‘Smart Homes’ and ‘Smart Grids’ [1].

2.5.3 Internet of Services (IoS)

According to Buxmann et al. in [29], Internet of Services (IoS) allows service suppliers to provide their services over the Internet. The IoS comprises an infrastructure for services, certain business models, the services themselves, and participants requesting the services. The services are combined into value-added services and offered by various vendors and communicated to users and customers through numerous channels. This development enables new possibilities in terms of dynamic variation in the distribution of specific value-chain activities. It is imaginable that the idea of IoS transfers from single factories to entire value-added networks (compare Chapter 2.3). Instead of offering only certain production types of their products, factories of the future could offer special production technologies. These technologies would be offered through the IoS network, enabling a customer of the services to simply manufacture a specific part, or compensate for limited production capacities [30]. Facilities with the possibility of offering such services follow a so-called Service-oriented Architecture (SoA) as is further explained in Chapter 2.9.

The concept of IoS has been applied in a project initiated by the Ministry for Economic Affairs and Energy within the program ‘Autonomics for Industrie 4.0’

[31]. In the presented project named 'Smart Face', a new distributed production control system is developed for the automotive industry. The assembly stations in the facility are of modular nature. This concept allows the flexible modification and expansion of certain stations. The transportation between the modules is realized through automated guided vehicles (AGV). Since the project is based on a SoA, both the AGV and the assembly stations publish their services to the IoS. This allows the products - in this case the vehicle bodies - to choose their course autonomously through the production process, having been pre-programmed with customer specific configurations.

2.5.4 Smart Factory

The 'Industrie 4.0 Working Group' considers so-called 'Smart Factories' a key feature of Industrie 4.0 [10]. The term constitutes the vision of a production environment in which production facilities and logistic systems largely self-organize without human intervention. The 'context-aware' Smart Factory takes into consideration the position and status of a product within the process, and assists machines and people in the execution of their tasks. As explained in Chapter 2.5.1, the technical background of Smart Factories builds on CPSs, in which the systems communicate and cooperate with each other and humans with the help of IoT (Chapter 2.5.2). The vision specifically addresses the communication between the product or work piece, and the manufacturing facility, where the product carries its production information in machine-readable form and the machines retrieve their workload from the product. The systems gather information both from the real physical and the simulated virtual facilities to determine their next production steps. Physical information in this regard often means the position of the product or the tool wear of a machine, whereas information from the virtual model are the optimal tool choice for a certain task, or the determination of optimal production schedules.

As an example of a Smart Factory, the 'Future Urban Production' facility in Fellbach, Germany is presented [32]. Among the manufactured products in the facility are gear wheels. In the past, the physical transport of goods between the various delivery and pick-up spaces have been done by an electric truck that drives around the factory once every hour. This inflexible procedure has been superseded by material supplies on demand in the framework of an Industrie 4.0 pilot project. For the implementation of a demand-driven supply, now intelligent work piece carriers are used. When a work piece is ready to be picked up, the carrier reports this status to the transportation control unit. This helps to decrease the number of transportation runs and saves personnel superfluous work.

2.6 Definition

Based on the components of Industrie 4.0 as presented in Chapter 2.5, the working paper [15] formulates a definition for Industrie 4.0 as follows: 'Industrie 4.0 functions as a collective term for concepts and technologies of value-chain organization. It is structured as modular Smart Factories, where CPS capture physical processes, generate a virtual copy of the physical facility, and make decentralized decisions. Using IoT, CPS connects and cooperates with humans and each other in real time. IoS furthermore offers both internal and cross-organizational services that are used by participants of the value-chain.'

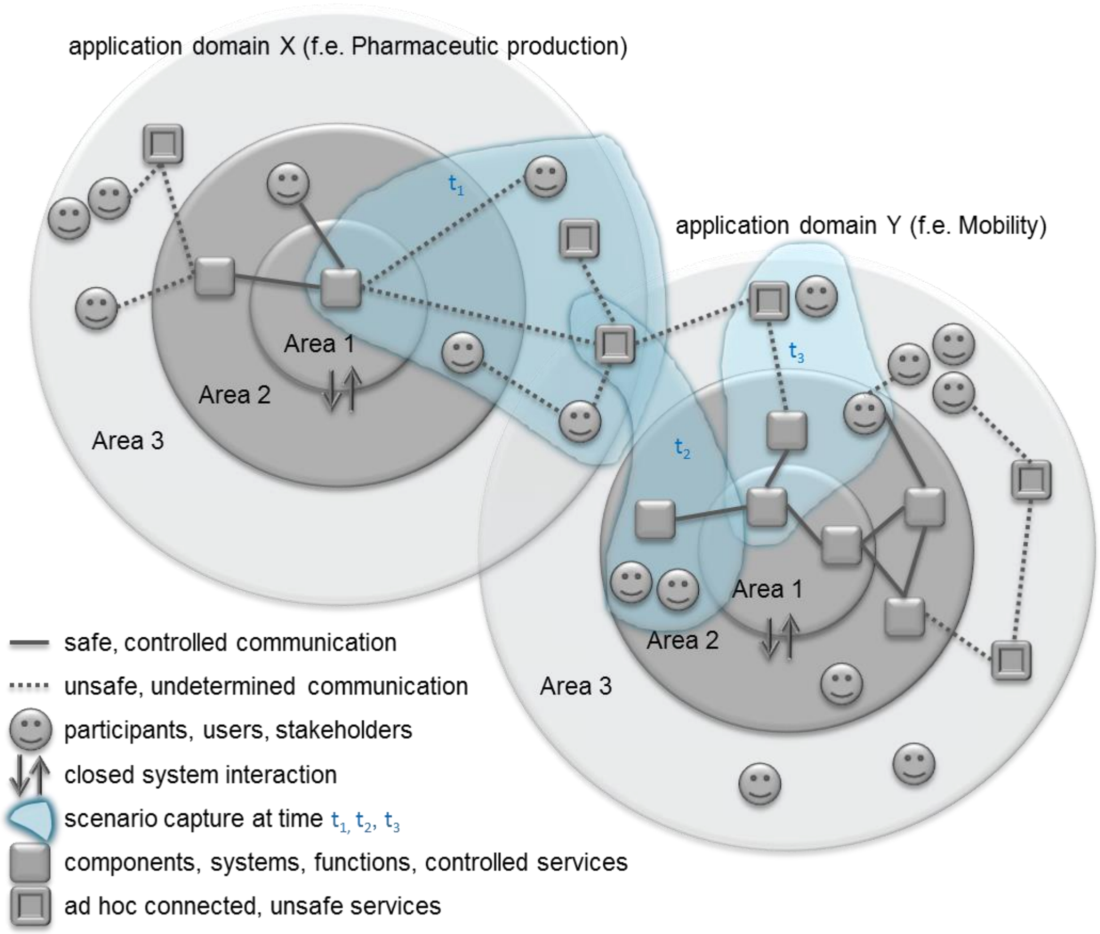

Figure 4: Cyber Physical Systems in open environments

According to Figure 4, CPS in an open environment consist of users, actors (also from social networks), services (from IoS) and information with dynamic integration, unclear reliability and availability. The challenge for the organization is that users and open systems interact ad hoc. An example is the dynamic integration of current information for traffic jams, train and flight delays into an assistance system. This application enables the planning of transport relative to the current transport course and status. As a basis for interoperable and compatible CPS, components and services with appropriate interfaces and protocols have to be established through standardized, flexible infrastructures and communication platforms (IoS).

2.7 Design Principles

In order to assist companies in identifying and implementing Industrie 4.0 pilot projects, Hermann et al. claim in their working paper [15] that specific design principles are required to be defined. The paper therefore derives six design principles as presented in Table 2. The principles are obtained by an evaluation of the literature review on the Industrie 4.0 components from Chapter 2.5. Table 2 shows which Industrie 4.0 component leads and is linked to which design principle.

Table 2: Design principles of each Industrie 4.0 component (as derived in [15])

	Cyber-Physical Systems	**Internet of Things**	**Internet of Services**	**Smart Factory**
Interoperability	X	X	X	X
Virtualization	X			X
Decentralization	X			X
Real-Time Capability				X
Service Orientation			X	
Modularity			X	

The following design principles for Industrie 4.0 are illustrated by the example of the key finder plant of SmartFactory^{KL} in Kaiserslautern, Germany [33]. The project is an initiative of the German Research Center for Artificial Intelligence for the development of a vendor independent technology operating in a production facility. The demonstration plant processes and assembles parts for key finders. All relevant data for the production process is saved on a RFID tag, which is directly attached to the work piece. The data is written on the transponder in the commissioning station of the production line. The production line further consists of a milling station, an automated assembly station and a manual workstation. At the manual workstation either the complete manual assembly of the product can be carried out, or the final manual assembly after the automated assembly station is performed. The manual station represents a context-sensitive work environment that guides people with the help of Augmented Reality through the complex manufacturing processes (see also [2]).

2.7.1 Interoperability

As described in Chapter 2.6, CPSs are connected with each other and humans over IoT and IoS. Therefore, interoperability is an important enabler of Industrie 4.0. A key success factor to enable a rule-based and situation-controlled communication between CPSs of various manufacturers will be the creation, establishment and integration of standards (see Chapter 2.8). The German Commission for Electrical, Electronic & Information Technologies of DIN and VDE has addressed this requirement in their publication 'German Standardization Roadmap' in 2013 [8]. The publication claims that the existing system landscape of technologies and structures is not yet coherent and completely defined in a globally standardized manner. It is hence not sufficient to only define the emergent behavior and additional level of integration of Industrie 4.0, but also relevant models of the classical architecture need to be overworked and integrated in addition to Industrie 4.0 itself. Further explanations and objectives of standards are given in Chapter 2.8 and requirements on an overworked architecture for Industrie 4.0 are presented in Chapter 2.9.

When addressing Interoperability in the context of the demonstration plant at SmartFactory^{KL}, all CPSs of the plant (assembly stations, work piece carriers and products) communicate with each other through open nets and semantic descriptions. The order information can for instance be transferred from the application system to the control of the picking station and stored in the digital product memory using a recently established communication standard OPC-UA (see Chapter 3.5.1).

2.7.2 Virtualization

The principle of virtualization represents the ability of CPS to monitor physical processes. The monitored data is then linked to simulation plant models and a virtual copy of the physical world is produced.

The key finder demonstration plant also contains a virtual representation of itself [33] that includes the monitoring of the condition of all CPSs in the plant. If an error occurs in the facility, a human can be notified and assisted in handling the failure. Humans can furthermore be supported by the system in handling the rising technical complexity of next working steps, maintenance, or safety arrangements [2].

2.7.3 Decentralization

The progressive reduction of mass production and increasing demand for individual products gradually impedes the central control of a production system. Through developments in embedded systems, CPSs become more intelligent and are able to make decisions on their own. Only in the case of maintenance or a failure do the CPSs need to communicate to a centralized, higher level system. Hence, the production facility moves towards becoming more decentralized.

The Decentralization in the SmartFactory^{KL} plant is realized through the RFID tag in the product. All information about the individual production steps of the product can be directly retrieved from the product by the machines.

2.7.4 Real-Time Capability

To organize a factory and react to time critical events immediately, it is necessary for an Industrie 4.0 factory to collect and analyze data in real time. The increasing amount of information circulating in a factory sets sophisticated demands on the processing units. In this respect, it is crucial to determine judicious selections and cycle times for the data that will be collected for further processing.

In the key finder production, the status of the plant is continuously tracked and analyzed. Should an unpredicted event occur, the facility can react immediately and reroute products that would get stuck in the production line of another machine, as an example.

2.7.5 Service Orientation

Service Orientation is a design paradigm whose principles stress the separation of concerns in the software. It leads to components of software divided into operational capabilities, each designed to solve an individual problem. These components qualify as services [34]. According to Chapter

2.5.3, the services of humans, companies, and CPSs are accessible over the IoS and can be used by other participants. The services can be offered both internally and globally across company borders. Design examples of a service-oriented Industrie 4.0 facility is given in Chapter 2.9.

The SmartFactory^{KL} plant is also designed based on a service-oriented architecture. All CPSs in the plant post their functionalities as a web service within the factory. Subsequently, the customer order details can be retrieved from the RFID chip on the products, and a suitable production process can be composed based on the available services.

2.7.6 Modularity

Designing a factory according to the paradigms of modularity provides the advantage to flexibly adapt to requirements by replacing or expanding certain modules or CPSs in the factory. These flexible modules are often referred to as ‘plug and produce’-modules. It describes the ability of certain machines and tools in the facility to communicate its services and location in the factory. The ‘plug and produce’-ability, which also shall be vendor independent, again sets demands for necessary standardized interfaces as is described closer in Chapter 2.8.

In the key finder facility, new modules can be added or removed using the ‘plug and produce’-principle. Based on standardized hardware and software interfaces, new CPSs are recognized automatically and can be used instantaneously using the IoS. Modular systems therefore can be easily adjusted in case of changed product characteristics or seasonal instabilities.

2.8 Standardization

According to Figure 1 in Chapter 2.1, Standardization is a key component for new innovations to ensure stability, security for investments, and confidence among users and manufacturers. Standardization is understood as being the entirely consensual establishment of a recognized organization of strategies, regulations, and guidelines for general or recurrent activities. The established standards of standardization bodies like IEC and ISO are accompanied by a set of Specifications in various forms. Such Specifications can for example be the DIN Specifications (DIN SPEC), the VDE Codes of practice, the Publically Available Specifications (PAS), the Technical Specifications (TS), Industry Technical Agreements (ITA) or Technical Reports (TR). [8]

In terms of Industrie 4.0, a special difficulty arises for standardization and terminology. Before defining the behavior and additional levels of integration of Industrie 4.0, the existing system landscape of technologies and structures needs to be coherent and completely defined in an internationally standardized manner.

Although digital simulation enables an increased variety of possibilities for tests of various scenarios, there are few standards among the diversity of IT solutions available. These solutions include many data models and interface protocols. This subsequently requires enhanced maintenance, changes, and new implementations. There is also a lack of transparency in the way these changes are implemented. The classical way of programming has to be replaced by a new system of rules that sets certain specifications, as well as coherent terminology and libraries. New Industrie 4.0 terminology could for instance be added to existing specifications, like in the specification IEC 61131-3 on terminology for industrial automation and information oriented instrumentation & control technology.

Through the ever-rising information technological expenditure behind facilities in the automation environment, a steady increase in complexity and comprehensibility limits the programmability. Consequently, it leads to an abandonment of absolute control. A solution is required that ensures room for developments, while minimizing the described problems. The solutions proposed by Industrie 4.0 suggest an architecture of decentralization through a Service-oriented Architecture (SoA) as introduced in Chapter 2.9.

In summary, coherent standards are required inter alia in terminology and concepts of system architecture, communication and engineering procedures.

2.9 Service-oriented Architecture (SoA)

A Service-oriented Architecture (SoA) is an architectural pattern of information technology in the field of distributed systems that structures and uses the services of IT systems. A special focus hereby is on the orientation towards business processes whose abstraction levels are the basis for concrete service implementations. As an example, the granting of a loan by a Bank is a service abstraction at a higher level of business processes. Behind this service are a number of persons and IT-systems like ‘opening a business relationship’, ‘opening of one or more bank accounts’, ‘credit agreement’ and others. By effective orchestration of lower level services, services of a higher abstraction level can be created quite flexibly, while allowing for a maximum of reusability [35].

Simplified, SoA can be viewed as a method or paradigm to encapsulate the existing computer components such as databases, servers, and websites as services and to coordinate them so that they are combined into higher services and made available to other organizational departments or customers. A SoA therefore is a structure which enables enterprise application integration, in which the complexity of each application is hidden behind standardized interfaces.

Goals of a SoA are the long-term reduction of costs in software development and a higher flexibility of business processes by reusing existing services. The programming costs of the n^{th} with the SoA realized application should be highly reduced since all necessary services are already available and they would only have to be coordinated effectively. Thus, the remaining costs are only the cost of business analysis and software configuration.

A technical structure of the implementation of SoA is to offer services on the Internet or in a cloud (compare Chapter 2.5.3). Communication between such services can be made via SOAP, REST, XML-RPC or similar protocols. The users of these services only know that the service is offered, what inputs it requires, and what type the result is. What services are available and how they are controlled can be ascertained by a directory service or service broker such as UDDI. The World Wide Web Consortium published these relationships according to Figure 5 [36].

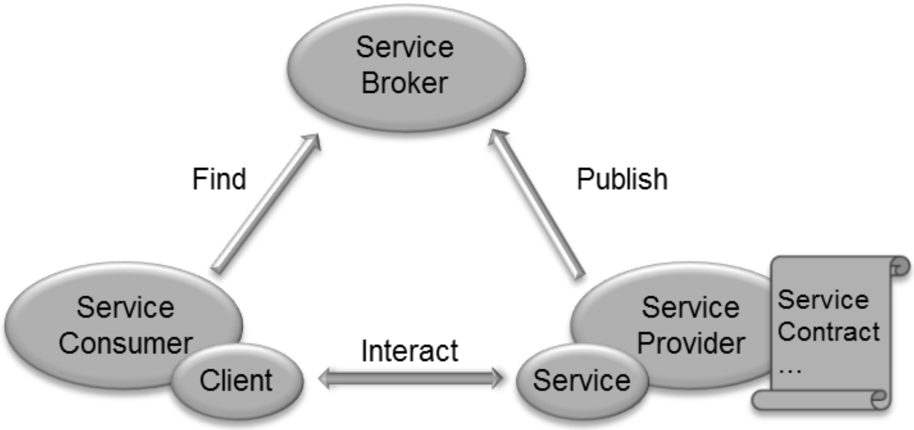

Figure 5: Structure of a Service-oriented Architecture (according to [36])

The Service Provider provides a service and its Service Contract and advertises it on a Service Broker. The Service Consumer can thereupon find a compatible Service and its Service Contract using the Service Broker and can subsequently interact with the Service Provider.

2.10 Demands on Engineers

The explanations made on the topic Industrie 4.0 largely describe the convergence of technologies that were formerly applied separately. Such a combination is for example, the fusion of mechanical engineering and electrical engineering into mechatronic engineering. An example from the production industry is for instance the development of smartphones, where the technical areas of electrical engineering, physics and IT converge. This changes competence requirements for engineers and pushes them to become more interdisciplinary. The convergences of the technologies need to be taught to students and engineers. According to Zoitl in [37], the coming industry requires engineers that are specialists in terms of having the overview over a whole system. He argues that the degree program ‘System Engineer’, for instance, is one such trendsetting course.

According to Geisberger and Broy in [25], the biggest engineering challenge is to satisfy two of the main features of CPS. On the one hand engineers must develop safe, secure, and dependable systems that can add value sustainably, and on the other hand, systems ought to be open innovation networks that facilitate new applications and networks. Geisberger and Broy explain further that the procedures necessitate an iterative approach of exploratory work from multi-vendor industries and research associations. The interconnectedness and merging of the technologies and applications necessitates an integrated approach and the development of new or at least extended engineering concepts. A substantial amount of effort needs to be devoted to the fields of integration, modelling, interdisciplinary requirement engineering and software and systems engineering. The study in [25] states engineering requirements in the following subsections.

2.10.1 Methods for Requirement Analysis, Design and Evaluation

For the requirement analysis, iterative design and evaluation of architectures and solution designs, currently two methods can be used. Either the use of interactive prototypes and realistic virtual simulations, or the mathematical modelling and simulation of a system is applied. In the context of Industrie 4.0, new challenges regarding the development of open systems arise. According to the users' needs, the systems are capable of interacting in a highly dynamic context with a global network. Meeting these requirements involves expanding the research horizon and developing new analysis methods. The new methods furthermore need to be aligned with the previously separate designed models ensuring an end-to-end verification and validation.

2.10.2 Requirements Engineering

Requirements Engineering is the key to the conception, design, validation, and verification of CPSs. The main tasks of Requirements Engineering include determining business, user, customer and process requirements, identifying potential problems, setting targets and priorities, resolving inconsistencies and conflicts and establishing requirements for the components, architecture, and communication methods of a system. With respect to the wide range of possible Industrie 4.0 scenarios, the principal Requirements Engineering concerns the identification, analysis, and specification of needed requirements.

2.10.3 Human, System and Architecture Models

To ensure that comprehensive and integrated models of humans, systems, and architectures are adequately designed, configured, and managed, the following technologies and knowledge fields are required:

- science and engineering, especially in electrical and mechanical engineering, physics, chemistry, and biology

- information and communication technology (ICT), embedded systems, sensor technology, system, and networks management and the Internet
- cognitive psychology, neuroscience and brain research on human behavior, thought, and complex problem solving
- social science and social networks

Especially in the field of engineering, research focuses on methods to reduce complexity thereby simplifying the design and management of CPS. The areas of research are:

- model-based systems engineering
- human-machine systems and human factors
- virtual engineering
- evolutionary software development
- reuse of software product lines
- generative programming and synthesis
- formal verification, validation and testing

2.11 Summary

The explanations made on the topic Industrie 4.0 largely describe technologies that already exist today. The challenge lies in the structured connection of these technologies, standardization of certain accomplished principles, and reference models. System architectures and business models have to be created and sorted into application dependent groups. The design principles formulated in Chapter 2.7 support both the scientific and practioners' communities thereby helping to implement scenarios and select potential use cases for further investigations.

As Drath says in [9]: Industrie 4.0 is a phenomenon that will come inevitably, whether we want it or not. Analogous to the introduction of the Internet in the beginning in the 1990s, which subsequently brought out an unimaginable world of online stores, auctions, online brokerage, E-Mail, Facebook, video streaming and app stores, the scope of Industrie 4.0 is still not graspable in its entirety.

Furthermore, the research on the topic Industrie 4.0 was only applied in the languages German and English. Therefore a limitation of the results persists. It is possible, that results of journals in other languages may have been left unobserved.

3 Preliminaries of the Example Application

Chapter 2 reveals an outlook of the extensive possibilities for an Industrie 4.0 use case example. An example could focus on topics like technology, logistics, management, engineering, or entire business models. The technology focus of Industrie 4.0 thereby mainly regards automation, information and communication technology, as well as production engineering. In order to meet the scope and delimitations presented in Chapter 1.3, this chapter forms the technological background for the implemented example application presented in Chapter 5.

3.1 Material and Energy Flow Simulation

According to the VDI guideline 3633 Part 8 [38], material flow simulation is a simulation related to logistic systems. It is one of several ways for performance measurement and the generation of optimization insights. It is used in production and assembly processes, for maintenance and repair processes, in warehouses, conveyors, workshops and other logistic and distribution centers. The simulation can be used at various stages along the value-adding chain of the factory life-cycle according to Figure 2.

Simulation during the planning stage of a new factory helps to verify throughput, sufficient dimensioning, throughput, capacities, interfering factors, personnel requirements, and other planning parameters. In addition, various alternatives can be evaluated and compared. Based on predefined criteria, a ranking of the alternatives can be generated. Also, existing facilities can be virtually replicated and optimized through targeted modifications within the simulation model.

The simulation model can furthermore be used by programmers to evaluate certain system controls. Consequently, a faster commissioning of the facility with fewer or no errors can be accomplished.

The simulation model also comprises advantages during the operation of the facility. The predictive test of the daily schedule of a plant can gauge necessary supply of personnel and resources, order throughput and the use of the modules in the plant. Time and use of material so far have been the prominent factors in material flow simulation to ensure the compliance of planned quality and quantity outcome. With the changing circumstances in the energy markets and increasing importance of energy-efficient production, developers of simulation software increasingly implement extensive possibilities to simulate various energy considerations. Based on the ordered energy plan, this allows forecasting of the expected running costs resulting from simulated energy requirements. An introduction on the structure, purchase, and handling of different energy plans on the energy market are given in Chapter 3.3.

If it is decided to take into account the power consumption of the factory within the material flow simulation, the model is to be extended by power requirement data of the separate items. Putz et al. integrate in [39] energy states via a process table in the chain of events of the material flow simulation. If a component arrives at a station, the stored energy requirement is registered. The energy requirement of the station can be obtained from measured data or generated from a machine model [40]. For the respective operating states of a system component, the corresponding power is deposited and recorded at every period of time. The simulation period-time is selected by the user of the simulation based on the needed precision.

The energy requirement of a plant can be represented in a load diagram over the simulation time. In Figure 6 such a load diagram is presented. The predicted power requirement allows estimation of changes in the process flow and analysis of the impact of different production parameters on the energy consumption of the system.

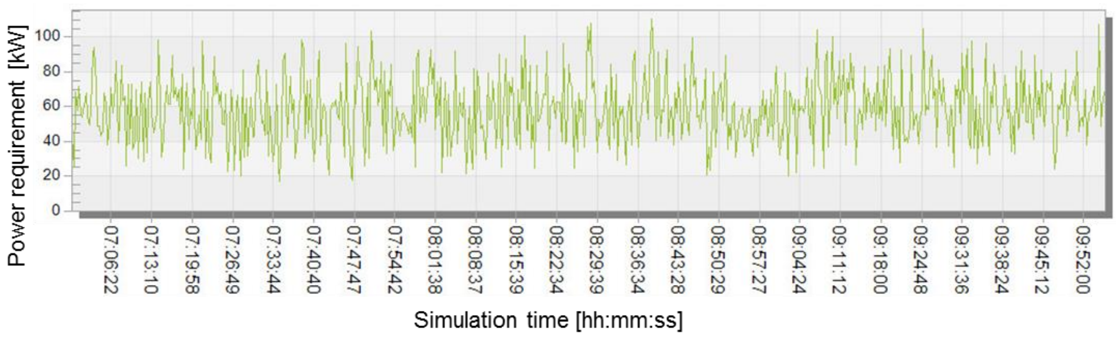

Figure 6: Simulation of the energy requirements of a production plant

Subsequent to the simulation of the production process, the obtained information can be analyzed and a simulation-based optimization on various settings executed and applied to the real facility. Further explanation on the optimization possibilities is given in Chapter 3.2.

According to the VDI guideline 3633 Part 8 [38], the cost-benefit ratio of the material flow simulation is set to 1:6. While efforts in the planning phase of the facility are high, they fall significantly in the operating phase, due to lower need for rectification on the facility.

3.2 Simulation-Based Optimization

The idea of simulation-based optimization (SBO) is to connect simulation models with an optimization component. Specific variables of a simulation model can be varied in order to minimize or maximize an objective. The basic idea of the SBO is to describe alternatives by variables of a simulation model. An SBO algorithm varies these variables, evaluates repeatedly the gained solution from the choice of variables by running a simulation of the settings, and then returns the best solution. The SBO can be realized by the optimizer controlling the simulator and vice versa.

A major difference to classical optimization problems, however, is that the 'objective function' in the SBO is stochastic, i.e. it is subject to random fluctuations, whichever occurring scenario is considered in a simulation run. The SBO therefore cannot show the optimal solution, as in the classical optimization of analytically solvable problems. The SBO only provides results to certain scenarios that are simulated. According to the objective, the solutions of the different scenarios can be ordinal ('better than', 'less than') or cardinal (absolute values) transferred into a solution ranking. The production system can be improved by adjusting the input variables until a better solution is found. The SBO therefore is an iterative process in which input variables to the simulation get adjusted, result data gets processed and evaluated and, based on the objective of the SBO, a statement about the improvement of the scenario can be made.

Examples for SBO are, for instance, to find improved integration for production and logistics in a factory or to find improved utilization of available human resources. Also the order sequence of products in advance for the day can be improved.

3.3 Energy Market

The liberalized energy market describes the market for grid-bounded energy supply by the power companies with electricity and natural gas. Many parts of the supply chain are subject to free competition. Competition should exist to ensure that consumers are supplied at the best market conditions, since a change of the power provider or gas supplier is possible at any time in compliance with the notice periods. The supply networks that are required for the energy supply cannot be meaningfully subjected to competition. Here the respective network operator has a monopoly. To prevent the network operator from taking advantage of its monopoly position, the charges for the use of the networks are regulated by the government.

The trading of electricity has a long tradition in Europe; it evolved hand in hand with progress in the use of electrical energy. Electricity is traded on both stock exchanges, such as the European Energy Exchange (EEX), and bilaterally (off-exchange trading). In this process there is a differentiation between short-term trading (intra-day, day-ahead, after- day) and long-term trading (futures, forwards). Short-term trading is characterized mainly by the fact that power is a not storable singular. Production and consumption must be done at the same time.

3.3.1 Power Spot Exchange Market

In this thesis, the stock exchange market for short-term trading is of closer interest. With more than 240 trading participants from 24 countries currently, the EEX [41] headquartered in Leipzig, Germany is the leading energy exchange market in continental Europe. Electricity, natural gas, CO2 emission rights, coal, and certificates of origin for green electricity is traded on the EEX. The EEX short-term market for electricity is carried out by the EEX Power Spot (EPEX SPOT) market [42] headquartered in Paris. The market enables intra-day and day-ahead trading of power for Germany, France, and Austria. In the day-ahead market, the power for the upcoming day is traded. The bids of the auctions for every hour of the day allow the placement of different buying and selling amounts to a different price for each hour. The quantity exported is then determined by the auction price. The auctions for Germany take place seven days a week at 12pm and the results are published at the EPEX SPOT website or through the EEX service software at 12:40pm. Figure 7 shows the power spot market results from Germany on June 26th, 2015.

Figure 7: EEX power spot market results example (retrieved from [42])

3.3.2 Electricity Price Forecasting

For the planning of production and resource allocation in a factory the question of which energy volumes are needed at what times becomes increasingly critical. With respect to the given examples of SBO in Chapter 3.2 it has also become important to estimate the energy price before the actual auctions. The EPEX SPOT market provides average hourly prices with a 7-, 30- and 200-days average. Furthermore, electricity price forecasting has become a significant area of research in the aftermath of the international deregulation of the power industry. Contrasted to electricity demand series, electricity price series can exhibit variable means, major volatility, and significant spikes [43]. Voronin and Partanen propose in [44] an iterative method for price forecasting implemented as a combination of two modules for normal price and price spike predictions. The forecast accuracy is assessed with real data from the Finnish Nord Pool Spot day-ahead energy market and delivers significant improvement compared with some of the most popular forecast methods applied for case studies of energy markets.

3.4 Database Systems and SQL

The amount of data and information increases in many systems. The system that is used to store and process these data is called a database. Due to the ever-growing amounts of data, databases are created and managed using computerized Database Systems (DBS). DBS consist of a database and a

Database Management System (DBMS) as shown in Figure 8. The DBS ensures persistent storage, consistency of user data and provides with the help of the DBMS interfaces for database applications for querying, analysis, modification, and management of these data. Today, database systems are usually implemented as a client-server architecture. A Database server handles the tasks of data management and the execution of queries, whereas a client computer takes over the dialoging to create the queries and provide the user with the processed data [45].

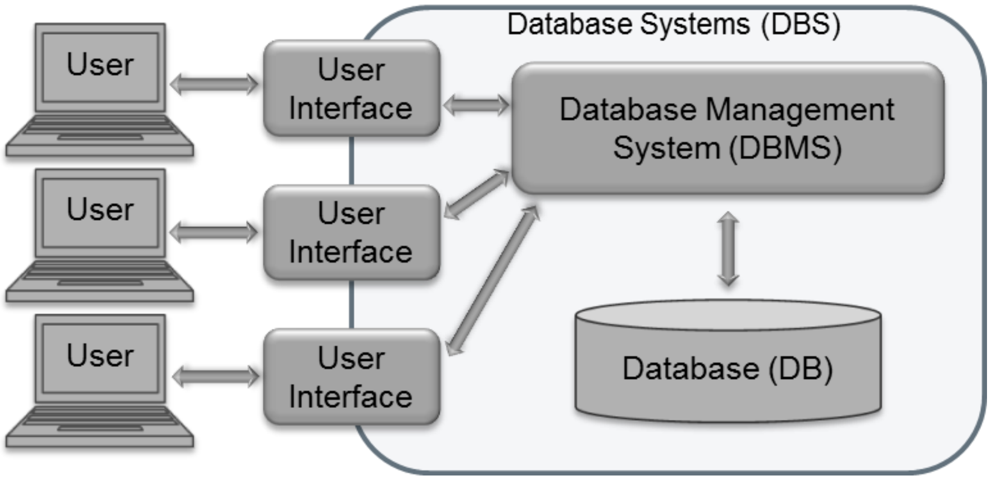

Figure 8: Database System Structure and Access

3.4.1 Relational Database Model

DBS are available in various forms. How such a system stores and manages data and thereby forms the infrastructure of the DBS is determined by the database model. The most common example of a database model is the relational database model. The corresponding DBMS is called a Relational DBMS (RDBMS). The example application implemented in Chapter 5 applies a relational DBS and hence only this DB model is of interest in this thesis. The relational DBS is based on the storage of data in two-dimensional tables, which are called ‘relations’. With the help of links, relationships between the individual relations are made. Relations can also contain entries that are references to entries in other relations. Thereby, it requires little expenditure to add or remove new tables, to link in between them, and thus apply changes to the structure of the database.

3.4.2 Database Management System

The DBMS internally organizes the structured data storage and controls and monitors all read- and write-accesses to the database. The DBMS translates the logical access of users and application software in physical access and thereby reads and writes data [46]. The functions of the management system include the user management, the command interpretation, rights management, data access, data links, and the processing of search queries.

The DBMS furthermore enables multi-user operation (compare Figure 8). Through user defined interfaces, external software is able to access the DBS. In this thesis, the standardized database interface ‘Open Database Connectivity’ (ODBC) is applied. Further explanation on the communication with ODBC is given in Chapter 3.5.2.

To query and manipulate the data through the interface, the database language ‘Structured Query Language’ (SQL) is mainly used for relational Database Models.

3.4.3 Structured Query Language (SQL)

SQL is a language for accessing databases. It has evolved steadily since its launch and has become the default language for databases [46]. In the context of programming languages, SQL follows the paradigm of declarative programming, where the programmer describes which conditions the program’s output has to fulfill. The translation of the program to source code, for instance with an interpreter, defines the method of how the results are produced. Unlike ordinary programming, SQL describes individual commands rather than entire programs. These commands are called instructions and are defined according to fixed rules. All instructions are composed out of the following components [45]:

- Keywords (UPDATE, SELECT, INSERT, FROM, …)
- Special characters (comma, space, +, *, -, / …)
- Data elements (assignment depending on the data type)
- Identifiers (e.g. name, first name, age)

The commands in SQL can basically be divided into four categories:

Table 3: Overview of SQL-instructions (according to [47])

SQL category	SQL-instruction	Description
Data Definition Language (DDL)	*CREATE TABLE*	creating a table
	ALTER TABLE	changing a table
	DROP TABLE	erasing a table

Data Manipulation Language (DML)	*INSERT INTO*	inserting a data set
	UPDATE	changing a data set
	DELETE	erasing a data set
Data Retrieval Language (DRL)	*SELECT*	querying a data set
Data Security Language (DSL)	*CREATE ROLE*	creating a role
	DROP ROLE	erasing a role
	SET ROLE	activating a role

With DDL the data structure is defined. For instance, domains and relations can be defined with the DDL. By DML the data in a database can be manipulated. This includes inter alia, entering, changing, and deleting data in a table. The DRL is responsible for the data retrieval. Thereby data can be selected by any criteria. Through DSL, the access rights on the existing data for different users are managed. Further explanation and application examples on the specific SQL-instructions can be found in [47].

3.5 Industrial Communication

The general understanding of communication is the exchange of information between two or more participants. If the term is used in an industrial environment, it is spoken of an industrial communication. The aim is to link the components of automation systems. For private applications, and also in offices, Ethernet has established itself as a communication standard. Since the requirements for private use clearly differ from the requirements in an industrial environment, a new standard was introduced with the Industrial Ethernet. The aim was to use existing standards and to supplement requirements of the industrial environment [48]. To offer an open and universal concept from the field level to the corporate management level in the context of Industrie 4.0, a promising trend for machine-to-machine communication protocols, called OPC, is presented in Chapter 3.5.1. In difference to the physical connection with Ethernet, OPC is a higher-level standard that can work on top of Ethernet using the 'Transmission Control Protocol'/'Internet Protocol' (TCP/IP). Furthermore, the standard ODBC for communication between application software and DBSs is explained in 3.5.2 and, lastly, in Chapter 3.5.3, the File Transfer Protocol is introduced, since it is used in the implemented example application in Chapter 5 to transfer a set of files.

3.5.1 Openess, Productivity, and Connectivity (OPC)

OPC is an acronym for ‘Openess, Productivity and Connectivity’ and comprises standards for data exchange in the field of industrial automation. The development of OPC is ascribed to the OPC Foundation [49], an alliance of industry vendors, software developers, and end-users. The standard can be divided into OPC UA and older OPC Classic specifications. The interface is designed for client-server and server-server communication and provides access to real-time data, alarms, events, and historical data. The advantage and aim of OPC is the networked communication of application software from different vendors.

Classic OPC

The first OPC standard was introduced in 1996 with the aim of integrating the various protocols of Programmable Logic Controller (PLC) communications in a unified interface. Based on the different requirements of industrial applications, OPC Classic is mainly composed of three OPC specifications [50]:

- Data Access (DA)
- Alarm & Events (A&E)
- Historical Data Access (HDA)

OPC DA regulates access to current process data. When using an OPC DA client, the variables that shall be read, displayed, or written are explicitly selected. The OPC client then establishes the connection to the server via the OPC server object. The OPC items on the client are gathered in groups as illustrated in Figure 9. The groups are formed based on items with the same settings, for instance the same update interval. The OPC client is just able to access the variables of a defined group. With every specified update interval, the variables from the group are examined for changes and published by the OPC server.

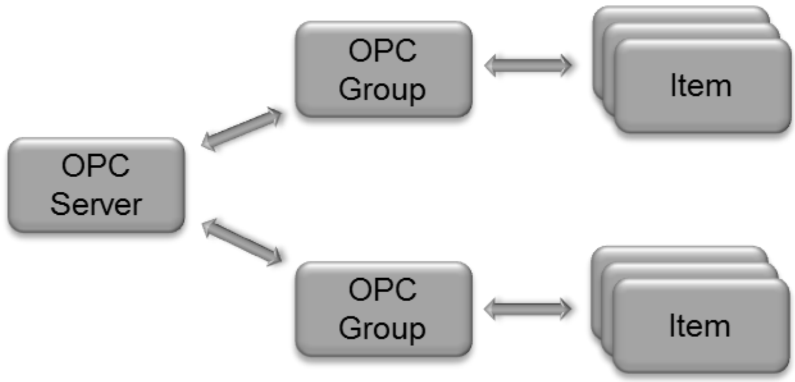

Figure 9: Objects of OPC DA clients for data acquisition

A&E describes an interface for monitoring, transmitting, and acknowledging of events and alarms. When a predefined event occurs, a notification message is passed to the user. Alarms are triggered by reaching a limit of a process value, such as a certain fill level in a tank. In contrast to events, alarms must usually be acknowledged.

HDA includes interrogation methods and analyses, which can be used to access archived data. For the data exchange, a client-server architecture is built. The OPC clients connect to the OPC server and read or write the provided data. Applications that work with or provide the data can be both server and client. The OPC client creates an OPC-HDA-Server-object on the archive server for the connection setup. Using the methods of this object, it is possible to read or update the archive data. Furthermore, an OPC-HDA-Browser-object exists to analyze, add, delete, or update certain variables on the server over different periods [50].

OPC Classic is based on the Microsoft technologies ‘Component Object Model’ (COM) and ‘Distributed Component Object Model’ (DCOM) for data exchange between participating components. Through these existing standards, the development time of products can be reduced and OPC could establish itself as a communication standard.

OPC Unified Architecture

To cope with the rising demands on Security and data modeling, the OPC Foundation is continuously working on further standardization. According to [51], the new standard ought to be platform independent, reliable, and scalable. In 2006, the new standard ‘OPC Unified Architecture’ (OPC UA) was first presented and published as IEC 62541. The DCOM has been replaced by a specially developed layer model for the used protocols, illustrated in

Figure 10. Furthermore, OPC UA is a service-oriented architecture, which consists of several layers and allows the use of different combinations of coding, security and transport protocols.

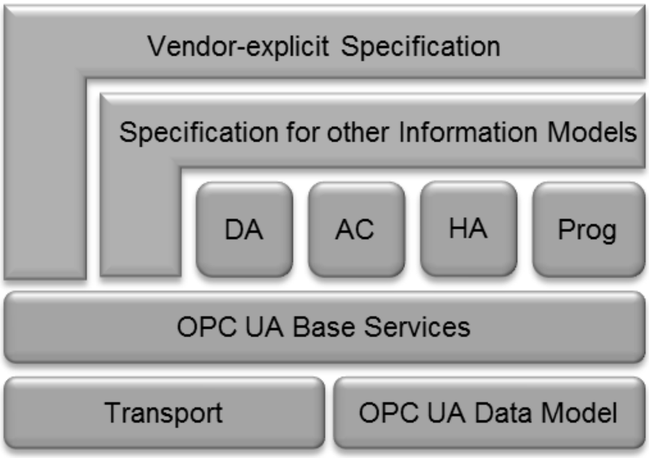

Figure 10: Architecture of OPC UA (according to [52])

In the data model, rules and basic building blocks of an information model for OPC UA are anchored. In the Transport block, methods for data exchange are gathered. For different applications, different transport protocols exist. Superimposed are the services that represent the interface between a server and the client [52]. The specifications for A&E and HDA got updated by the components 'Alarms and Conditions' (AC) and 'Historical Access' (HA), respectively. The information model is composed of objects that contain data (for DA and HA) and can trigger alarms (for AC). Data exchange with OPC UA is always in the form of a request-and-reply pattern, where the client sends a request and the server replies.

3.5.2 Open Database Connectivity (ODBC)

Open Database Connectivity (ODBC) is a standardized database interface that uses SQL as query language. It thereby offers an Application Programming Interface (API), which allows the programmer to develop an application relatively independent of the used DBMS, as long as there is an ODBC-driver available.

ODBC provides the ability to set the details for accessing the database. These are for instance the name of the database server, port for network access to the database, database name, user name of the database, and its password. Through the defined database name, the client application can connect to the database without having to know the details of the whole DBS. Modern programming environments allow uncomplicated access to a lot of different database management systems through predefined data sensitive control elements. The data access never takes place directly on a table or a database, but always through the corresponding ODBC component. For object-oriented programming languages like C++ or Java, classes are available that define the methods for handling the data of different DBS. The programmer thus does not need to take care of database specific details [53].

3.5.3 File Transfer Protocol (FTP)

The File Transfer Protocol (FTP) is a network protocol for the transfer of data over IP networks, specified in the RFC 959 [54] from 1985. It is located in the application layer of the Open Systems Interconnection Model (OSI model) and used to download files from a server to a client, upload in reverse, or to transfer files client-controlled between two FTP servers. The FTP also enables the creation of directories and the possibility to rename and delete files and directories.

The FTP uses two separate connections for controlling tasks and file-transfer tasks. An FTP session begins when the FTP-client establishes a TCP-connection to the control port of the FTP-server (the default port is 21). Through this connection, commands are sent to the server. The server subsequently replies to every command with a status code that is often extended with a text string explaining the code. The procedure of a user login can be seen in Figure 11.

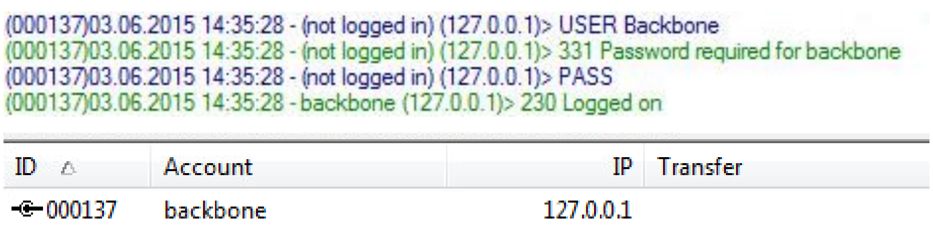

Figure 11: Communication between FTP-client and FTP-server

The FTP-client begins to log in with the Username ‘Backbone’ (first line, written in blue). The FTP-server replies with code ‘331’ and a further text string, which explains a password request for the user (written in green). After entering the password, the server acknowledges the client to be logged in.

4 Used Software in the Example Application

The Industrie 4.0 example application implemented in this thesis requires a set of software applications. In order to follow the procedures explained in Chapter 5, the used software is presented in this chapter.

4.1 DBMS SQL Server

The implemented DBMS SQL Server runs in a Microsoft Windows Server 2008 R2 environment. This is a server operating system with a graphical user interface similar to a standard Microsoft Windows operating system. The database administration tool runs on the integrated SQL Server Management Studio. Here databases, tables, and users (including their roles) are managed. With a Username it is possible to log into a database from external applications. When signing into the DBS and applying code to the database through external software, the support of all used datatypes and the functionality of the interface between the two systems must be verified. The applied interface in the example application is an ODBC interface.

4.2 Tecnomatix Plant Simulation

The software Tecnomatix Plant Simulation is a discrete event simulation tool. It is distributed by the Siemens PLM software and used for modeling logistic systems. Through the created models, the properties of a system can be examined and the performance improved. The major features of Plant Simulation version 11 used in this thesis are briefly explained.

4.2.1 Information Flow Elements

In addition to the elements of the material flow as described in Chapter 3.1, there are also information flow elements in Plant Simulation responsible for managing the data. In Figure 12 a selection of information flow elements can be seen.

Figure 12: Examples of information flow elements in Plant Simulation

The Event Manager controls the simulation. It is used to start, pause, and stop the simulation. Also, the simulation time can be retrieved from the event manager and used in code. The simulation speed can be varied and simulation duration can be set.

Tables in Plant Simulation are two-dimensional lists and can be used, for instance, to locally save a table from a DBS. The tables provide a set of methods and attributes that can be used to add, manipulate, or retrieve data in or from the table.

Plant Simulation provides multiple interfaces for the exchange of data with external software. Two of them are of interest in the implemented example application. The ODBC interface provides access to an ODBC data source and the OPC interface to an OPC DA server. Further explanation on these interfaces is given in Chapters 4.2.3 and 4.2.4.

Methods are blocks which are used to create programs using the programming language ‘SimTalk’. These are triggered by certain events or other methods and can fulfill any programmable tasks.

4.2.2 Methods and SimTalk

Methods can be edited in Plant Simulation in a dialog window. The programming language SimTalk is similar to C. Local Variables can be created, defined with a certain data type, initialized, and conditions and loops for the manipulation of the variables can be formulated. Global variables can also be queried and manipulated. Global variables are, for example, the production parameters, which are exchanged via the OPC server.

Material flow elements from the simulation, information flow elements as explained in Chapter 4.2.1, or methods themselves are able to trigger the execution of other methods. When the execution of a method is triggered, the running simulation will be paused until the execution of the method is completed. The Event Manager, for example, includes the events *Init*, *Reset*, and *EndSim*. When a simulation is started, the Event Manager triggers the method named *Init*. When the simulation is reset manually by the operator, the method named *Reset* is performed; when the simulation finishes by reaching the set simulation end time, the method *EndSim* will be started. The methods can be edited by the programmer and used to realize certain scenarios. The EndSim-method, for instance, can be used to summarize certain data and statistics about the simulation run and transmit them through the OPC Interface to other software applications.

4.2.3 ODBC Interface

The ODBC interface block in Tecnomatix Plant Simulation includes settings for specifying the name of the Database, its login data, and the alternative name of the database, which is valid inside the Plant Simulation project. Through the alternative name, the database can be manipulated with methods. For the manipulation, the interface block provides the methods *login* and *logout* as well as *sql*, to transfer a SQL-command-string to the ODBC. These three methods can be used in other methods to perform database manipulations.

4.2.4 OPC Interface

The OPC interface of Plant Simulation only operates as a client and supports OPC DA 2.0 and OPC DA 3.0. It can be used, for instance, to retrieve or set the status of certain process variables from an external control system. Similar to the ODBC interface, a project internal valid name for the interface has to be chosen in the settings. Additionally, a hostname of the Server platform and the OPC server name have to be entered. The hostname can be given in form of the symbolic computer name or as an IP address. Also the OPC-variables that are to be manipulated have to be parametrized in the settings of the interface block with an alias name and a datatype. Subsequently, these variables can be accessed within methods or the OPC-variables themselves are able to trigger a method at the time of change in value.

4.3 Totally Integrated Automation (TIA) Portal

The Totally Integrated Automation (TIA) Portal is an engineering framework from Siemens for automation solutions. It is the successor of the SIMATIC

Manager. Various engineering systems can be accessed from a single user interface.

4.3.1 Engineering tools of the Totally Integrated Automation Portal

In addition to the engineering tools for the technical control tasks, the TIA Portal also contains environments for visualization, drives, motion-control, and engine management. The engineering tools used for this purpose are SIMATIC STEP 7 as an automation tool for controllers, SIMATIC WinCC for all human-machine interface (HMI) applications, and SINAMICS Startdrive for drives and controllers [55]. Integrating all applications into a single platform results in a consistent database and can be used by all automation tasks. The TIA Portal aims to offer solutions through the whole life-cycle of added value processes (compare Figure 2) in the automation environment.

4.3.2 SIMATIC Windows Control Center (WinCC)

The TIA Portal is used in the Industrie 4.0 example application because of its capabilities in creating an HMI interface with SIMATIC WinCC. WinCC is the abbreviation for Windows Control Center. It includes solutions for panel applications as well as complex process visualization. Depending on the type of project, different programming packages are available to choose from, such as the WinCC Comfort package for panels and WinCC Professional package for complex process visualization and SCADA applications. For PC-based single-user solutions like the implemented example application, WinCC Advanced is used. The design is done through the TIA Portal user interface. To run the HMI-application, a so-called Runtime (RT) is used. The Runtime is implemented in Siemens Panels by default. As for this thesis, where the HMI is assigned to run on a PC-station, the extension 'WinCC RT Advanced' is to install on the PC. WinCC RT Advanced then performs the visualization and notification of the HMI deposited program.

To design a software program for the WinCC system as a Runtime on a single-user PC, a so-called 'PC-station' has to be implemented in the hardware configuration of the TIA Portal project. Various surface elements of the HMI, like buttons, scales, or text-string-fields are allocated with variables. These variables are assigned a name, datatype, and other properties. For example, when the value of a variable changes or reaches a certain level, they are able to trigger several events within the WinCC-program. One of the possible events is the execution of a script. Scripts in WinCC Advanced are written in the script language 'Visual Basic Script' (VBS) and expand the possibilities of the HMI in terms of programmability. Through scripts, the WinCC RT is, for example, able to login to and manipulate Databases through an ODBC Interface.

WinCC RT can further function as an OPC server and provide variables and process values to external software and the management level. Both an OPC DA server and an OPC UA server can be implemented. A distinction must be made in internal variables, which are known only in WinCC, and Tags, which are available outside of the WinCC environment.

4.4 XAMPP and FileZilla FTP Server

In the available closed test environment, the implemented Industrie 4.0 example application uses an artificial Internet. To simulate the Internet, the open source software XAMPP version 3.2.1 is used [56]. It is a cross-platform web server solution package and comprises, amongst other things, the Apache HTTP Server, MySQL database, interpreters for scripts written in Perl and PHP programming languages, and the file transfer tool FileZilla Server. In Figure 13 the control panel of XAMPP is displayed. The control panel allows the control and configuration of all software components through a single user interface.

Figure 13: Control Panel of the XAMPP software

In this thesis, the FileZilla FTP Server is used to publish a set of files through a simulated Internet. The software allows for simple user and group management, so a single test user can be added with Username and Password enabling access to the files according to Chapter 3.5.3 through a FTP Client by addressing the Servers IP (in the example application: 127.0.0.1). FileZilla Server furthermore provides SSL/TLS-encryption, the possibility to restrict certain IP addresses as well as up- and download restrictions, simulating file access through the Internet at an acceptable level.

5 Order Sequence Optimization

To demonstrate the vision of certain aspects of Industrie 4.0, an example application is designed and implemented in the scope of this thesis. According to the introductory explanations made in Chapter 1.3, the application addresses the topic of energy efficiency in production systems. In Chapter 5.1 a demonstration plant is introduced, which serves as subject for the implemented example application. As explained in Chapter 3.1, a simulation model of the plant is used and adjusted to serve an optimization goal as explained in Chapter 3.2.

5.1 Demonstration Plant: The 'Backbone Factory'

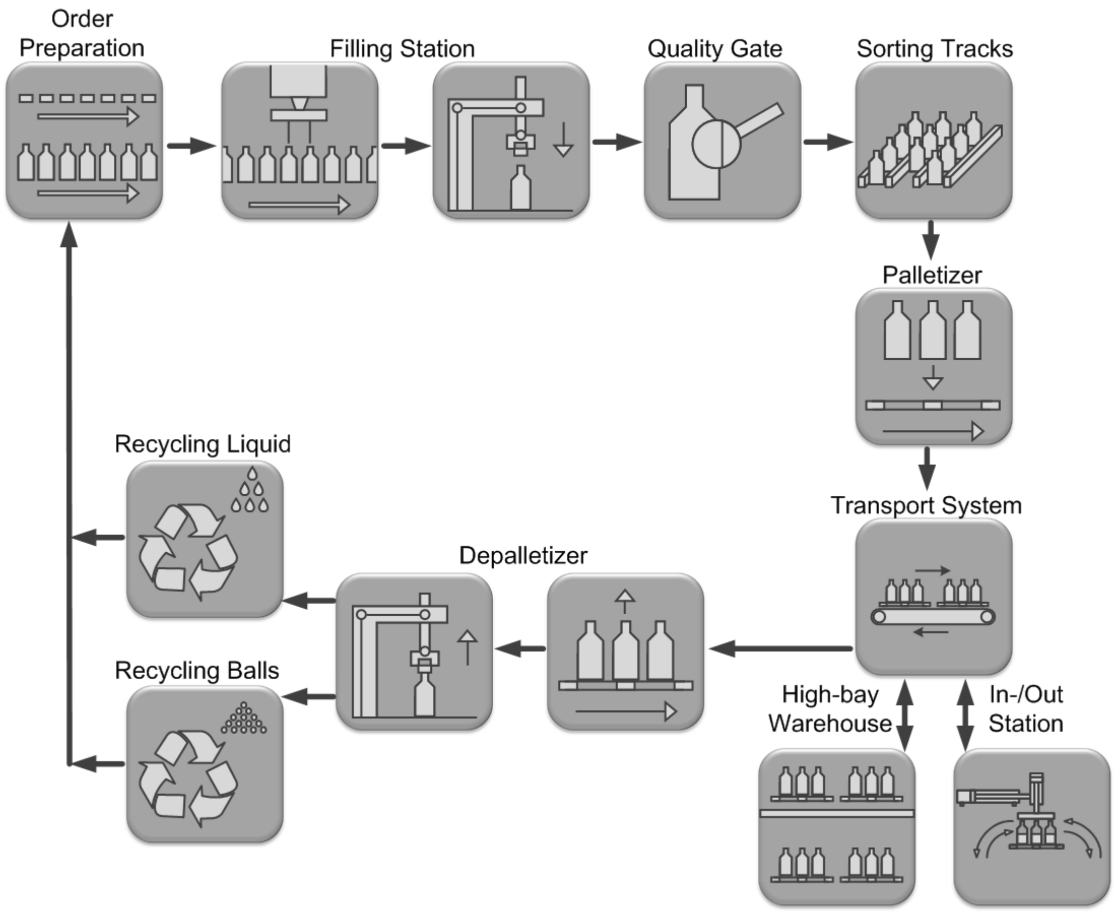

Figure 14: Scheme of the Backbone Factory

For the implementation of the example application, a demonstration plant located at the Siemens education facilities is used. The local education department is entitled ‘Backbone’, so the demonstration plant is henceforth referred to as the ‘Backbone Factory’. The Backbone Factory operates the life-cycle of bottled products. The life-cycle includes the filling of bottles with different colored balls or liquids and the storage of the bottles, up to the recycling process. The factory exemplifies a production facility for pharmaceuticals, as an example. For clarity Figure 14 shows the scheme of the entire system and Table 4 explains relevant functionality of the modules.

Table 4: Modules of the Backbone Factory and their functions

Station name	**Description**
Order Preparation	After an order is received by the factory, the according amount of bottles and lids is released to the next station.
Filling Station	Based on the received order, each bottle is filled with a combination of either different colored balls or three different liquids. The available ball colors are blue, red and yellow. After the filling procedure, the bottles get sealed with a lid. The lid also includes a RFID chip, which is used to store product related data on the product.
Quality Gate	The quality of the bottle and its filling is determined. If for instance the content of the bottle does not match the information on the RFID chip, the product gets rejected.
Sorting Tracks	The sorting tracks are used as a buffer and to sort the products based on certain criteria. The sorting is done through four parallel conveyor belts.
Palletizer	The station picks up three bottles and puts them on a pallet. The pallet can hold up to six bottles.
Transport System	On a conveyor belt, the pallets are moved between the connected stations.
In-/Out-Station	The station is used to insert new pallets or to withdraw pallets with completed orders for further delivery.
High-bay Warehouse	The module can be used for the storage of pallets with or without produced products.
Depalletizer	The station transfers bottles from the pallets to a conveyor belt and unseals them. The removed lids are transported back to the Order Preparation station through a separate conveyor belt.

Recycling of Balls and Liquids	The station empties the bottles. Balls get sorted by their color and liquids get disposed. The empty bottles get moved to the Order Preparation station.

5.2 Conceptual Preview

This chapter provides a use case description of the example application and refers to the following chapters for an explicit description on the functionality described here.

5.2.1 Energy Requirement-caused Costs of the Backbone Factory

The Backbone Factory consists of a set of modules. Each module consumes energy. Besides the base load of energy consumption, the modules cause specific energy requirements based on the active machines and actuators and their specific task on the specific product. Therefore, each bottle in the production process causes a different energy requirement of the modules according to the composition of their content. This is also referred to as product-based cost accounting.

With the help of Material and Energy Flow Simulations in conjunction with a simulation model of the Backbone Factory, the future power consumption of the factory can be estimated over a certain horizon of time (compare Chapter 3.1). The maximum time horizon depends on the amount of product orders that is provided to the simulation model.

The resulting costs for the plant operator not only depend on the power consumption of the factory, but also on the varying energy market price. Based on the energy plan the operator ordered, the energy price can become high at peak times (compare Chapter 3.3.1). Since there are multiple options for plant operators to compose an optimal energy plan, a delimitation to an exclusive use of energy from spot market auctions is applied in the example application.

In order to include the energy price into the Material Flow Simulation of future orders, future energy prices must also be known. Further explanations of the implementation of energy price forecasts and the applied delimitations on this topic are made in Chapter 5.3.2.

To obtain the resulting costs of the simulation, the energy requirement of the factory and the energy price forecast must be multiplied for any point of time.

5.2.2 Order Sequence Optimization for the Backbone Factory

The idea of the example application is to reschedule product orders for future production in order to minimize the energy costs of the Backbone Factory presented in Chapter 5.2.1.

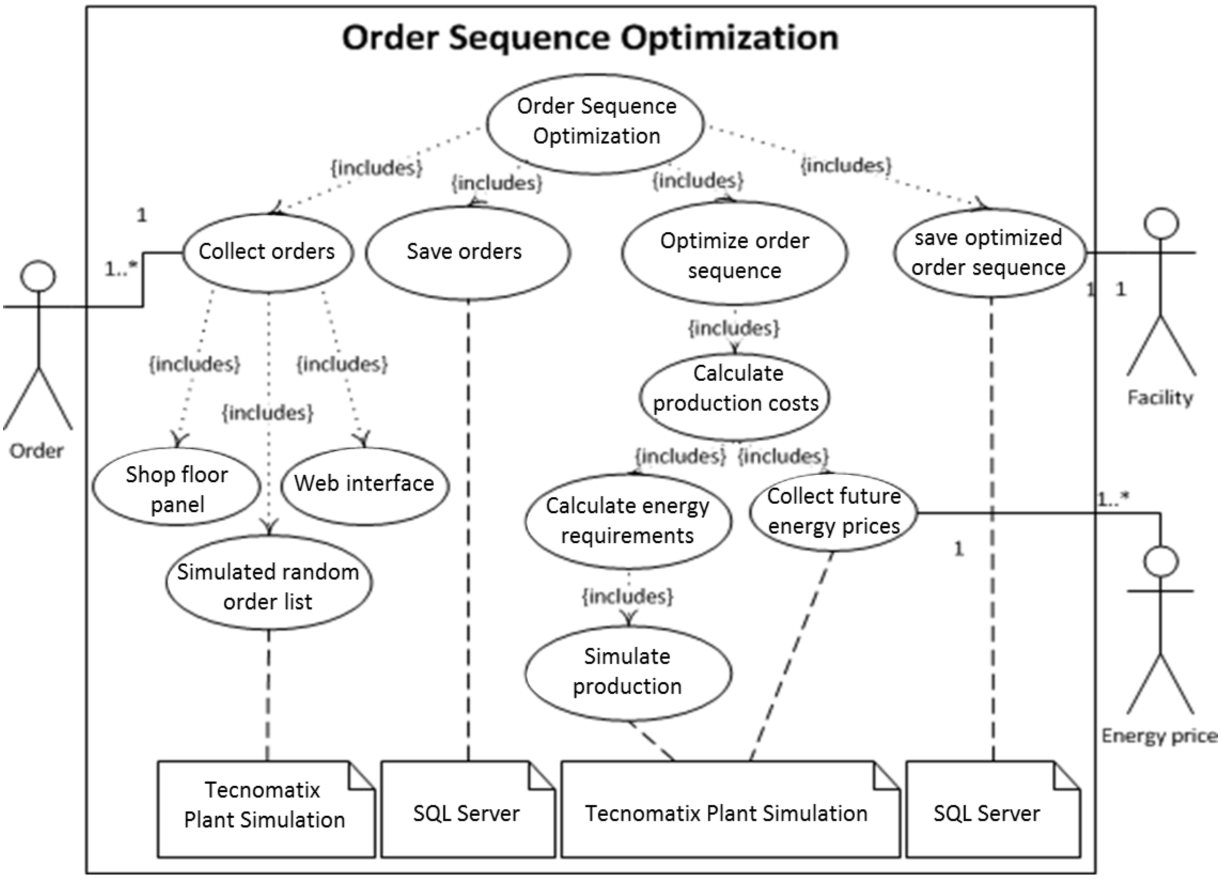

Figure 15: UML Use Case Diagram of the Order Sequence Optimization

In Figure 15 the use case diagram for the system function '*Order Sequence Optimization*' is presented using the Unified Modeling Language (UML). Starting at the left hand side of the figure with the actor '*Order*', the function '*Collect orders*' is initiated. This function includes three sub functions, which describe different methods to collect product orders for the Backbone Factory. Next to an existing '*Shop floor panel*' and a '*Web interface*', the function '*Simulated random order list*' is created for testing the functionally of the '*Order Sequence Optimization*'. The function is realized in a method in the software '*Tecnomatix Plant Simulation*' (compare Chapter 4.2) and further explained in Chapter 5.6.1. The created order list will then be stored as a SQL table on the SQL Server.

Given an order list on the SQL Server, an optimization of the order list can be initiated. To do so, the function '*optimize order sequence*' uses the sub function '*calculate production costs*'. As explained in Chapter 5.2.1, the energy requirements of the Backbone Factory have to be calculated using a simulation and the energy price forecasts have to be downloaded from the Internet in order to calculate the resulting energy related production costs. These functions are realized in '*Tecnomatix Plant Simulation*'. Subsequently, the order optimization process can be applied on the order list, and a new rescheduled order list will be created and saved on the SQL Server. Eventually the real Backbone Factory can access the updated order list and retrieve an order to initiate its production. This is illustrated through the actor '*Facility*' at the right hand side in Figure 15.

5.3 Delimitations

Since the goal of the implemented example application '*Order Sequence Optimization*' is to understand a new concept rather than implementing a fully working optimization tool, several delimitations are applied to the application.

5.3.1 Power Requirement of the Filling Station

In this thesis, only the filling of balls in the filling station of the Backbone Factory is considered for the measure of energy requirements (compare Figure 14). Further machines and modules can successively be added to the application in continuation of this work.

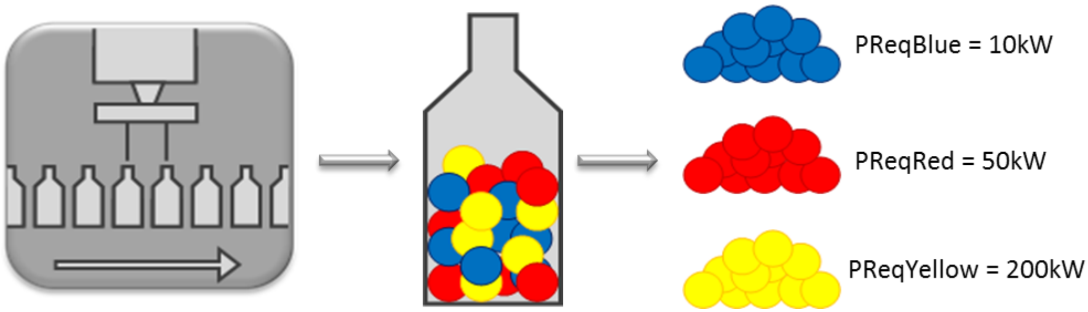

Figure 16: Example power requirements of the filling process for the three colors

Due to the assigned limitations of the considered modules, only a few actuators with low power consumption are available. To obtain varying energy requirements over the production time, the simulation model of the facility is further amended in a way such that each color has a different power requirement during the filling process. For instance, the filling process of yellow balls has a higher power requirement than the filling process of blue balls, because the process includes an extra quality check and polishing procedure. In Figure 16, example power requirements are given for the three different colors. The chosen values are unrealistically high, but serve the goal to visualize the effects of the order sequence optimization within the test environment.

To obtain a single value for the power requirement of the filling station, the following calculations are attached to the simulation model of the filling module. First, the amount of balls which are to be filled in the bottle is calculated in order to calculate the prorated percentage of the colors:

$$BallAmount = AmountRed + AmountYellow + AmountBlue$$
$$PercentBlue = AmountBlue \div BallAmount$$
$$PercentRed = AmountRed \div BallAmount$$
$$PercentYellow = AmountYellow \div BallAmount$$

Second, the power requirement of the filling station is calculated based on the color specific power requirements from Figure 16 as follows:

$$\begin{gathered} PReq = PercentBlue \times PReqBlue + \\ +PercentRed \times PReqRed + \\ +PercentYellow \times PReqYellow \end{gathered} \qquad [1]$$

5.3.2 Energy price forecasting

Plant operators apply various strategies when purchasing and combining several energy acquisition plans. The example in this thesis assumes that the entire required energy for the upcoming production is explicitly ordered from auctions of an energy spot market. Since the Order Optimization Process requires future energy prices, a forecast must be made. According to Chapter 3.3.2, there is already extensive research ongoing and certain services are available for purchase as well. Therefore, no algorithm for energy price forecasting is developed for this thesis. Instead two distinct energy price lists are included in the example application. The two lists, as shown in Figure 17, represent two different energy price forecasts over a period of 24 hours. One of the lists consists of real data, retrieved from an example provided by the EEX [42]. The other list is of random nature and aims to strain the abilities of the optimization process with random input parameters.

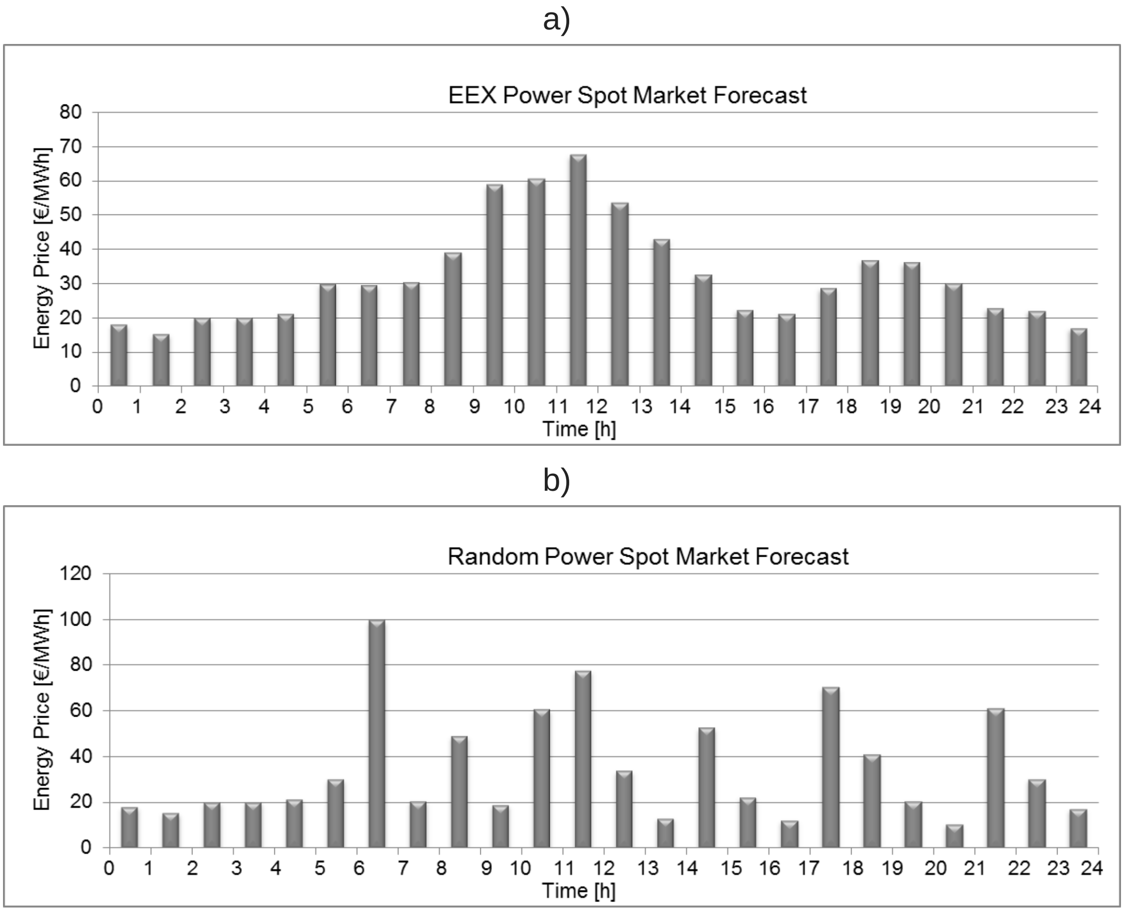

Figure 17: Power Spot Market Forecasts based on a) EEX and b) random values

The two featured lists are provided to the application using a simulated Internet introduced in Chapter 4.4. To reproduce the real EEX service, the lists are available on a FTP server as files of the datatype CSV. The FTP server interface with the two files and the structure of the energy prices in the file are shown in Figure 18. In order to retrieve the correct energy prices, the optimization application has to search for the correct country code in the CSV file. In this thesis, a German facility is assumed and therefore the prices with the lable ‘DE-AT’ are retrieved (compare Figure 18 b)).

a)

FTP-Stammverzeichnis auf 127.0.0.1

```
05/28/2015 12:05          3,165 EPEX_PowerSpotMarketAuctionResults.csv
05/28/2015 12:04          3,166 other_PowerSpotMarketAuctionResults.csv
```

```
# 2015-07-17: Prices for EPEX Power Spot Market;
#;
# Date type(ST);Delivery Date,Creation Time
# Date type(PR);Area;Unit;Hour;Price;Volume
# Date type(IL);Area;Unit;Index;Price;
# Date type(BL);Area;Unit;Block;Price;
# Date type(AL);Number of Lines
#;
ST;07.10.2014;2014-10-10T09:27:07+0200
PR;CH;EUR/MWh;23-24;47,64;2695
PR;DE-AT;EUR/MWh;23-24;16,93;27010
PR;FR;EUR/MWh;23-24;45,53;10243
PR;CH;EUR/MWh;22-23;48,79;2878
PR;DE-AT;EUR/MWh;22-23;22,02;27133
PR;FR;EUR/MWh;22-23;46,75;10464
```

b)

Figure 18: a) FTP server interface with two energy price forecast files; b) Structure of the energy price CSV-file

5.4 Interaction of Virtual Machines (VMs) and Software Applications

To implement the example application, the software presented in Chapter 4 is used. As can be seen in Figure 19, the software applications run on three different Virtual Machines (VMs) in the same local network. The figure shows the communication interfaces the software applications use to transfer relevant data. The communication interfaces are applied according to the functionality explained in Chapter 3.5.

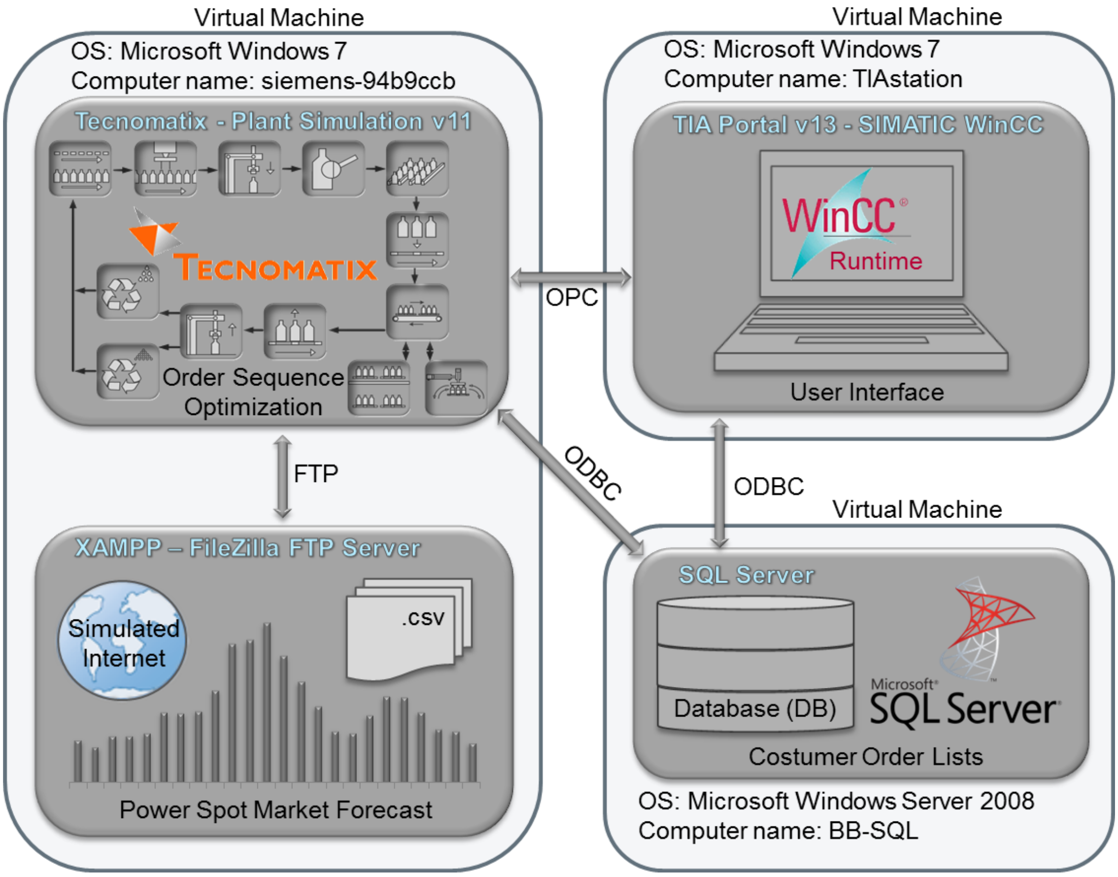

Figure 19: Structure plan of the VMs and software used for the example application

To illustrate an application optimizing the order sequence, the functional requirements presented in Table 5 can be determined.

Table 5: Functions of the used Software applications

Software application	**Functions**
SIMATIC WinCC	Represents the graphical user interface to an operator. • Initiate the creation of an example order list with random orders and storage on the SQL Server • Initiate the start of the order sequence optimization • Receive optimization results in form of graphs and statistical data on the HMI
FileZilla FTP Server	Provides energy price forecasts based on daily-published power spot auction results.
SQL Server	Stores the initial as well as the optimized order list.

Plant Simulation	Controls the order optimization process and runs multiple simulations. • Download the latest order list from the SQL Server • Simulate the production of the downloaded order list and thereby obtain the energy requirement of each product while getting filled • Download energy price forecasts from the Internet • Based on the obtained energy requirements and prices, calculate the resulting arising costs of the simulated production • Apply an algorithm to reorganize the sequence of orders in order to minimize the arising costs of the production • Simulate the production of the reorganized order list • Prepare graphs and statistics showing the improvements made by the optimization process • Upload the optimized order list to the SQL Server

5.5 Implementations in the SQL Server

To create, manage, and configure databases on the SQL Server, Microsoft provides the tool SQL Server Management Studio. In Figure 20, the structure of the SQL Server as shown in the Management Studio is given. In the folder ‘*Databases*’ the database ‘*Order Lists*’ is created. As stated by the description of Table 5 and the connection through the ODBC interface as introduced in Chapter 3.5.2, WinCC and Plant Simulation can access the database and create and manipulate order lists in the folder ‘*Tables*’. Figure 20 shows two example order lists: One initial order list with the ending ‘*_Orders*’, created by the function for creating a random order list and one order list with optimized sequence marked by the ending ’*_OptOrders*’.

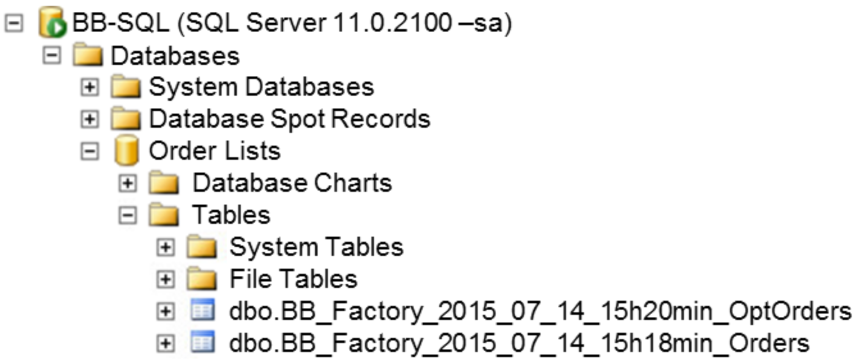

Figure 20: SQL Server structure and the Order Lists Database

5.6 Implementations in Plant Simulation

This chapter presents the implemented functionality in Plant Simulation according to Table 5. All methods in the following subsections are programmed in SimTalk as explained in Chapter 4.2.2.

5.6.1 Random Order List Creation

In Figure 21, an excerpt of an example random order list is illustrated in the form as it is stored on the SQL Server (see also the table file ending with '_Orders' in Figure 20).

OrderID	ProductID	OrderVolume	Target	Red	Yellow	Blue	Liq_1	Liq_2	Liq_3	ReleaseTime	DueTime
1	1	6	1	20	3	9	0	0	0	2015/06/09 07:00:00	2015/06/10 00:03:00
1	2	6	1	1	1	8	0	0	0	2015/06/09 07:00:00	2015/06/10 00:03:00
1	3	6	1	27	26	15	0	0	0	2015/06/09 07:00:00	2015/06/10 00:03:00
1	4	6	1	24	19	18	0	0	0	2015/06/09 07:00:00	2015/06/10 00:03:00
1	5	6	1	5	14	8	0	0	0	2015/06/09 07:00:00	2015/06/10 00:03:00
1	6	6	1	29	9	8	0	0	0	2015/06/09 07:00:00	2015/06/10 00:03:00
2	7	9	1	21	19	22	0	0	0	2015/06/09 07:00:00	2015/06/09 18:23:00
2	8	9	1	9	15	13	0	0	0	2015/06/09 07:00:00	2015/06/09 18:23:00
2	9	9	1	12	29	19	0	0	0	2015/06/09 07:00:00	2015/06/09 18:23:00
2	10	9	1	4	13	16	0	0	0	2015/06/09 07:00:00	2015/06/09 18:23:00
2	11	9	1	9	7	11	0	0	0	2015/06/09 07:00:00	2015/06/09 18:23:00
2	12	9	1	13	29	28	0	0	0	2015/06/09 07:00:00	2015/06/09 18:23:00
2	13	9	1	2	18	5	0	0	0	2015/06/09 07:00:00	2015/06/09 18:23:00
2	14	9	1	12	23	3	0	0	0	2015/06/09 07:00:00	2015/06/09 18:23:00
2	15	9	1	16	10	4	0	0	0	2015/06/09 07:00:00	2015/06/09 18:23:00

Figure 21: Example random order list on the SQL Server

The function '*Create a random order list*' is a method within Plant Simulation. The properties and initiation of the function are set by the user interface in WinCC Runtime and transmitted through the OPC interface. Further demonstration of the property settings of the method is presented in Chapter

5.7.1. Using SQL commands, the list is created and stored on the SQL Server through the ODBC interface.

The structure of the order lists on the SQL Server follow a scheme according to the control system of the Backbone Factory. The composition of one product is represented by one row in the table of Figure 21. The '*OrderID*' indicates the order number, the product belongs to. With the '*ProductID*', each specific product is characterized by continuous numbering. The '*OrderVolume*' displays the amount of products in the order. In the column '*Target*', the destination of the product is stated. In the example of Figure 21 as well as all examples used in this thesis, number 1 is set, representing the In/Out-Station (compare Figure 14). Additional columns in the table specify the amounts of each colored ball and liquid to be filled in the bottles. In the column '*ReleaseTime*', the time an order is received is stored and the '*DueTime*' specifies the deadline by which an order has to be delivered.

The created method for constructing a random order list includes a set of global variables that can be manipulated through the OPC Interface, thereby allowing a user to select the amount of products to be created in the random order list. The method then chooses a random order volume between four and ten products per order and a random amount of balls to fill in the bottle with one to thirty balls per color. A uniform distribution type is chosen for the selection of these amounts. The method also sets the release time of the order to the next day at 7am and the due time to a random value between ten and nineteen hours after the release time. The values for the target and the liquids are statically set to one and zero, respectively.

After a random order list has been created, Plant Simulation returns a pointer to the stored SQL table to the WinCC user interface in order to be used for the following optimization process.

5.6.2 Application Flow Control

According to the procedures explained in Chapter 3.2, a method is created in Plant Simulation that controls the sequence of events within the simulation-based optimization. When a user triggers the start of the order sequence optimization application through the WinCC user interface, a changing-value-recognition in the OPC interface of Plant Simulation initiates the start of the method '*Application Flow Control*'. The method is programmed according to the sequence diagram in Figure 22.

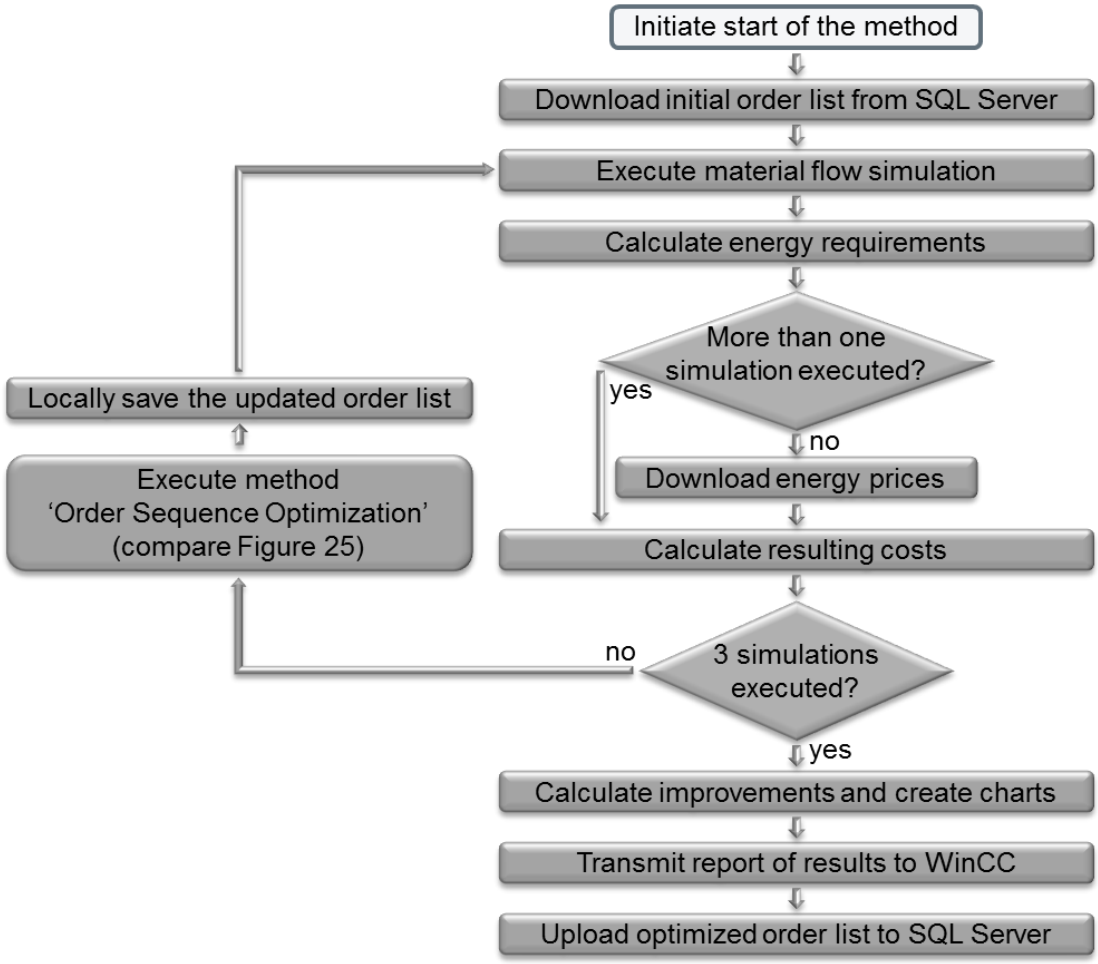

Figure 22: Sequence diagram of the Application Flow Control

After the method presented in Figure 22 is started, the initial order list is downloaded from the SQL Server through the ODBC interface, locally stored in a table, and the columns ‘*FillStart*’, ‘*FillEnd*’, ‘*PReq*’, ‘*EPrice*’, and ‘*ECost*’ are added to the local order list (see Table 6). Then the material flow simulation is executed. During the simulation, the first three added columns ‘*FillStart*’, ‘*FillEnd*’ and ‘*PReq*’ are filled. The first two columns represent the start and end time of the filling process for each product, respectively. The third column represents the power requirement of each specific product and is calculated according to equation [1] on page 47.

Once the first simulation of the whole order list is completed, the application flow control initiates the download of one of the energy price forecast files mentioned in Chapter 5.3.2. An operator at the WinCC User Interface chooses which of the forecast files is to be used for the optimization process as is further demonstrated in Chapter 5.7.2. The file is accessed using FTP. According to the point of time at which the balls get filled in the bottle, the energy price of the corresponding time is filled in the column ‘*EPrice*’ for each product. Table 6 shows an example of the newly added columns. The random power spot market forecast, as shown in Figure 17 b) is used in this example. Since the filling of the products of Table 6 takes place between 8am and 9am, the according forecasted energy price for that time slot is filled in the column ‘*EPrice*’ (compare Figure 17 b)).

Table 6: Example entries and calculations by the Application Flow Control

…	FillStart [hh:mm:ss]	FillEnd [hh:mm:ss]	PReq [kW]	EPrice [€/MWh]	ECost [€]
…	…	…	…	…	…
…	08:00:00	08:00:17	104	48,88	0,02401
…	08:00:17	08:00:30	25	48,88	0,004413

Based on the newly filled columns, the resulting energy-caused costs can be calculated according to the following equation and placed in the column ‘*ECost*’.

$$ECost[€] = EPrice[€/MWh] * \frac{PReq[kW]}{1000} * \frac{(FillEnd - FillStart)[sec]}{3600} \quad [2]$$

Figure 23 shows graphically the calculations performed over the simulated production time based on an example using an order list of 612 products. The simulated production time is 2 hours and 58 minutes, therefore three energy prices are downloaded from the energy price forecast from Figure 17 b). Figure 23 illustrates, that the products filled between 8am and 9am are in average more expensive than the filling tasks before 8am or after 9am. In total, the summed up energy requirement caused costs of the facility are 5.07€ and the caused peak cost of one specific product is 2.55 Cents.

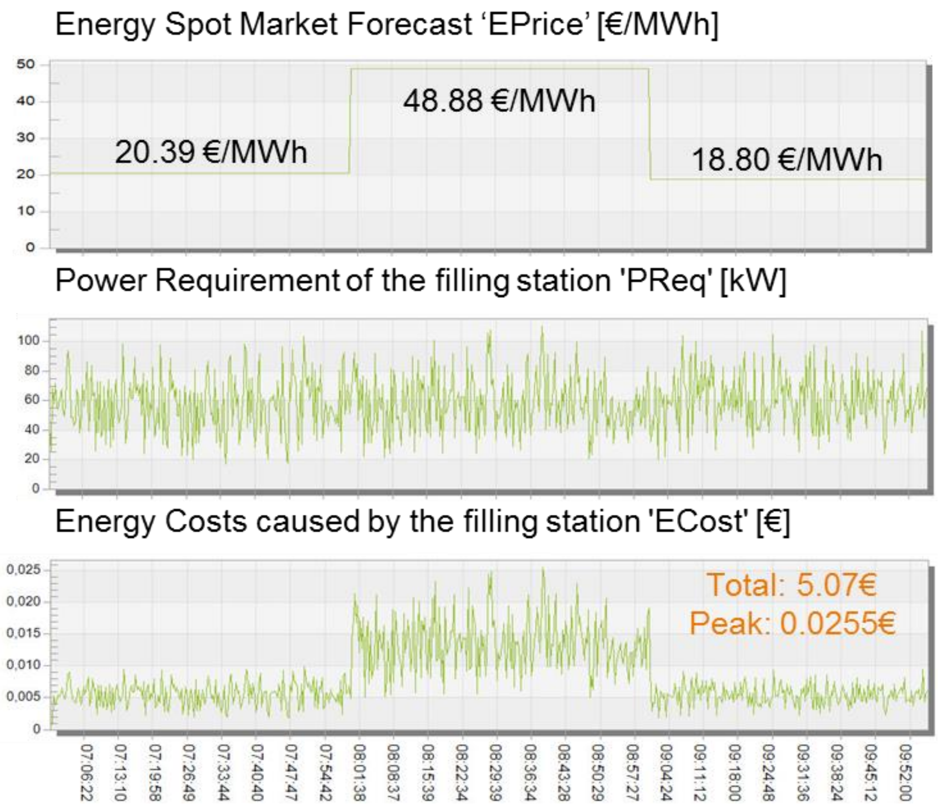

Figure 23: Example graphs for the simulation of an initial order list with 612 products

When the column '*ECost*' is filled for the whole order list, the application flow control continues with a checking the number of simulations already made. If less than three material flow simulations are made so far, the flow control executes the method '*Order Sequence Optimization*' using the current order list. This method is further explained in Chapter 5.6.4. The updated order list is afterwards stored locally in Plant Simulation. After the optimization process, another material flow simulation is made on the updated order list (compare Figure 22). This loop is performed twice and is justified in Chapter 5.6.4.

When three simulations are executed and the column '*ECost*' is calculated for the latest updated order list, the loop will be interrupted, and improvements made through the order sequence optimization are calculated. The calculations are based on the initial and updated order lists. For the purpose of visualization, statistics and graphs are created, summarized in a report and transmitted to the visual user interface on WinCC. Further explanations on the report function are given in Chapter 5.6.5.

Finally, the application flow control initiates the upload of the latest updated order list to the SQL Server. The upload is programmed in a separate method using SQL commands and is similar to the method explained in Chapter 5.6.1. After the order list is uploaded to the SQL Server, a reference link to the order list is transmitted to the WinCC user interface using the OPC interface, which thereby indicates the completion of the order optimization task.

5.6.3 Material and Energy Flow Simulation

The example application in this thesis is based on an already existing simulation model of the Backbone Factory [57]. Figure 24 shows the filling station in Plant Simulation according to Figure 14 marked by an orange circle. The simulation runs a row pointer through the order list and the filling station fills each bottle with the accordant amount of balls in the table. The filling time for each bottle is between 1 and 20 seconds depending on the amount of balls to be filled. It is calculated in a method, which gets initiated by the filling station every time an empty bottle arrives. The method furthermore includes the calculation of the power requirement for each bottle according to equation [1] on page 47. When the bottle of the last row in the order list has arrived in the In/Out-Station, the simulation is completed. As explained in Chapter 4.2.2, the method EndSim will be executed at last, which notifies the application flow control to continue its sequence of events.

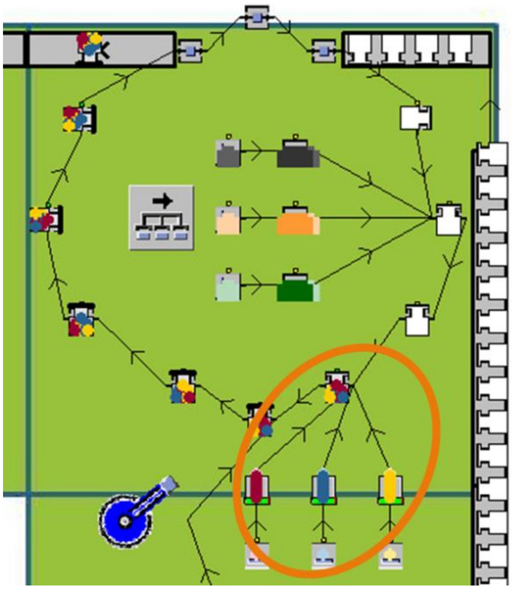

Figure 24: Filling Station in Plant Simulation

5.6.4 Order Sequence Optimization

The method '*Order sequence optimization*' reschedules the sequence of the products to be produced. The aim of the optimization algorithm is to minimize the total resulting energy costs. The objective function therefore is the sum of all product caused energy costs '*ECost*'. The minimization problem is presented in the following equation. The subscript number on '*ECost*' represents the '*ProductID*' (compare Figure 21) and *n* represents the product ID of the last product in the order list.

$$min \sum ECost = \min(ECost_1 + ECost_2 + \cdots ECost_n) \qquad [3]$$

The energy cost '*ECost*' of a specific product does not only depend on the power requirement '*PReq*' of the product, but also on the energy price '*EPrice*' at the time of day at which the product is getting filled. Therefore '*ECost*' also varies based on the time at which the product is getting filled. A minimization of the energy cost '*ECost*' for the specific products can consequently be obtained, when products of high power requirement '*PReq*' get filled in times of low energy price '*EPrice*'. The method '*Order Sequence Optimization*' hence follows the sequence diagram of Figure 25.

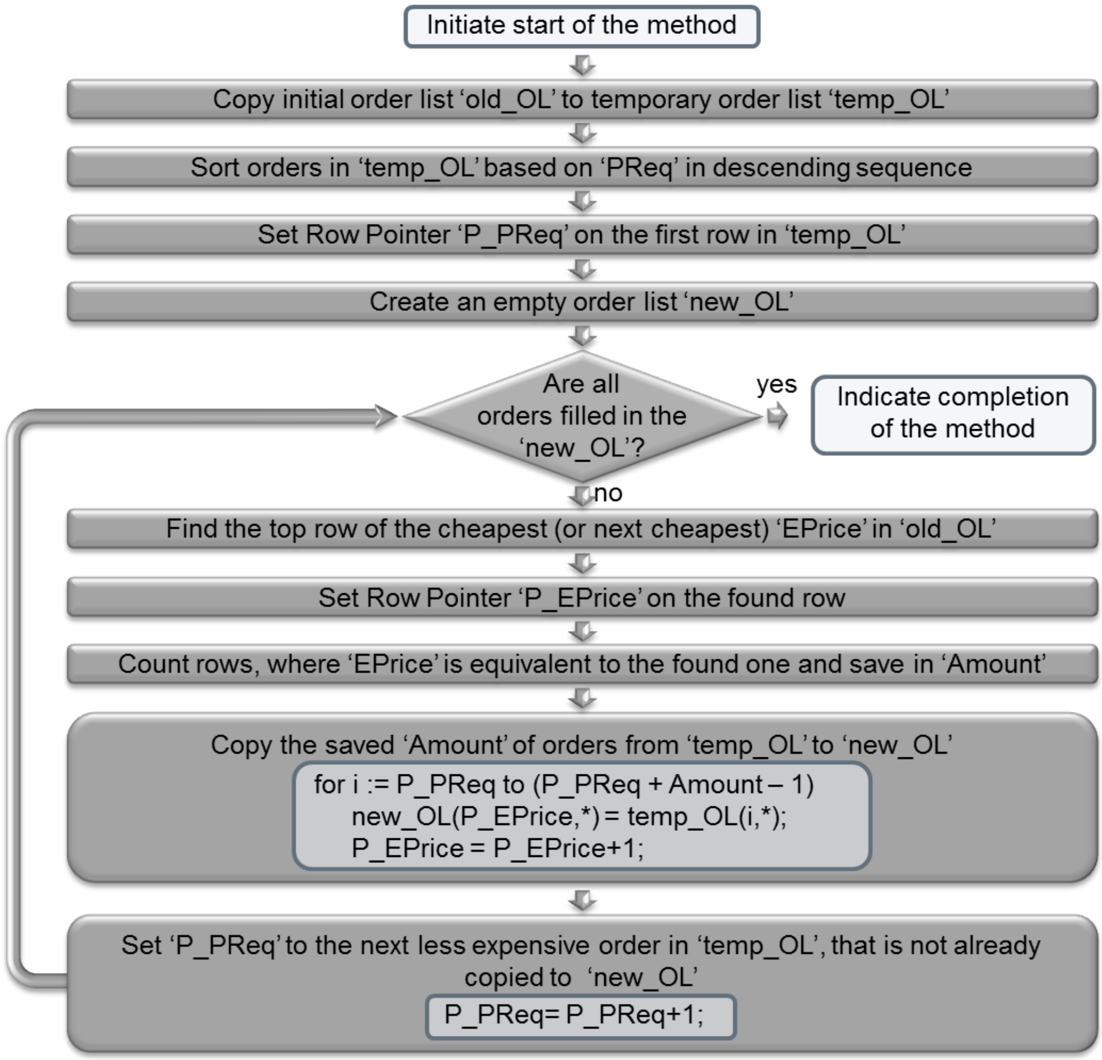

Figure 25: Sequence diagram of the Order Sequence Optimization

The optimization methodology of Figure 25 is applied on the example of Figure 23. The results of the optimization process are illustrated in Figure 26.

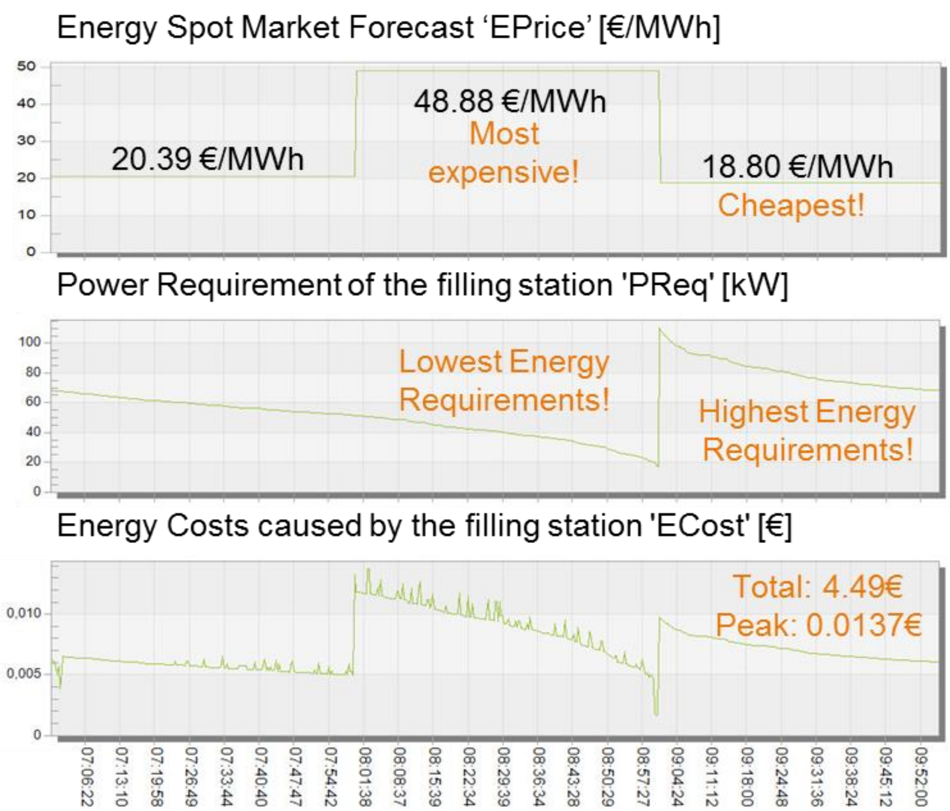

Figure 26: Example graphs for the simulation of an improved order list of 612 products

The cheapest energy price '*EPrice*' during the simulated production in Figure 26 is between 9am and 10am with a value of 18.80 €/MWh. Consequently the optimization method shifts the products with the highest power requirements '*PReq*' to a position in the order list, where they get produced between 9am and 10am. According to the loop in Figure 25, the algorithm runs from the cheapest energy price, through the next more expensive one of 20.39 €/MWh in Figure 26 up to the most expensive one at 48.88 €/MWh. At the latter one, the products with the least power requirement get filled. According to equation [2] on page 55, the resulting energy cost '*ECost*' is lowered compared to the initial order list. In total, the summed up energy requirement caused costs of the facility are 4.49€ and the caused peak cost of one specific product is 1.37 Cents. By comparing Figure 23 and Figure 26, the following improvements can be obtained. In total 0.58€ (11.51%) are saved and the peak price is lowered by 46.14%.

When the order optimization task is started by the user interface of WinCC, the optimization algorithm is executed in total two times, before an optimized order list is returned by the application. The reason is due to a time shift in the material flow simulation of the products.

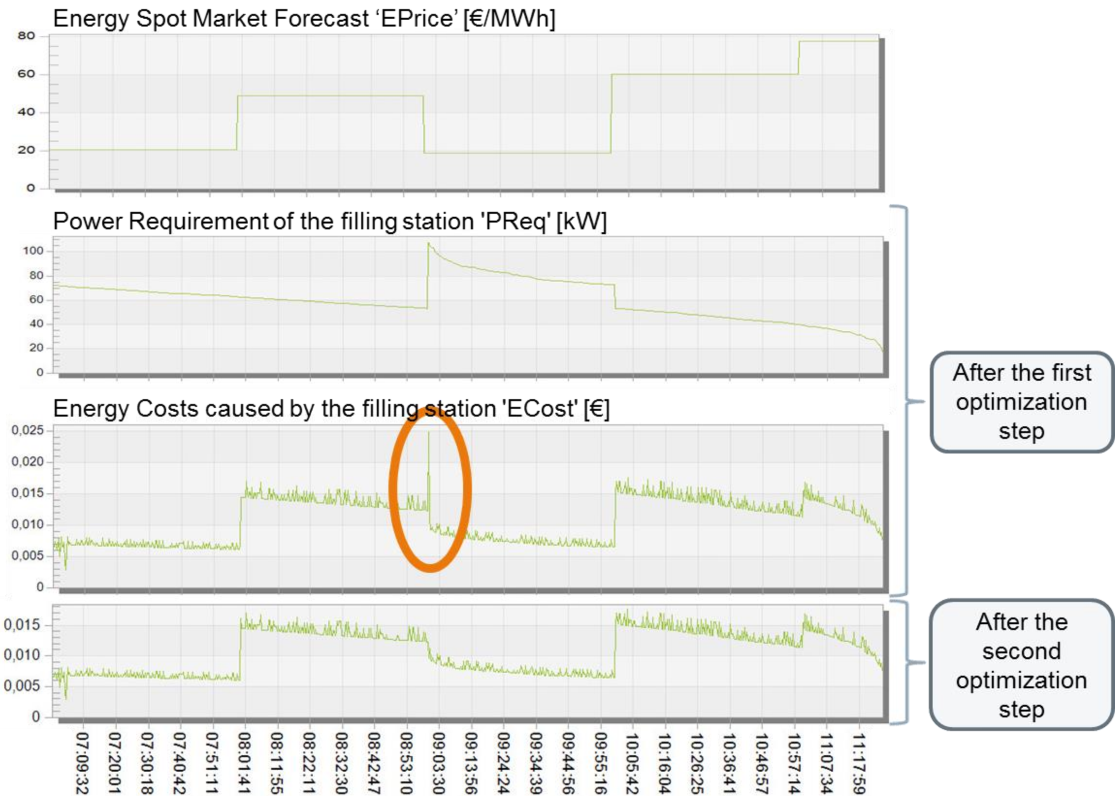

Figure 27: Example graphs illustrating the need for two optimization steps

When an order list is rescheduled by the algorithm in Figure 25, the times at which the filling process for specific bottles takes place are only estimated, but cannot be known before the next simulation is executed. In the example of Figure 27, the product with the highest power requirement '*PReq*' is meant to be shifted to an order list position where it gets filled at 9am, since the '*EPrice*' is the cheapest between 9am and 10am. Instead the simulation reveals, that the product is filled just before 9am, where the '*EPrice*' is more expensive, hence causing a peak in the '*ECost*'-graph highlighted by the orange circle in Figure 27. The reason for the misplacement of the order is due to the technique the algorithm uses to find the desired position in the list. The algorithm sets the product order exactly on that position, where the 9am-filled-product of the initial order list was placed. However, this estimate assumes that all previous product orders (from 7am to 9am) of the initial as well as the optimized order list take the same time in total. If the production of the previous orders takes less time in the optimized list than they did in the initial order list, as in the example of Figure 27, the product with the highest power requirement gets filled too early.

To correct the optimized order list, the method '*Order Sequence Optimization*' is applied a second time after the simulation has completed a second time (compare Figure 22). Since just minor changes are applied to the order list in the second optimization step, a misplacement of product orders will not repeat there. After the second optimization step, the peak in Figure 27 disappears.

5.6.5 Report of Results

After the optimization and simulation steps are completed and an improved order list is obtained, the application flow control initiates the calculation of certain graphs and improvement statistics to fill in a report. Figure 23, Figure 26 and Figure 27 are part of the generated graphs. Furthermore, distribution graphs of the separate colors to be filled over the production time are created (presented in Figure 28), and improvement statistics are calculated (as shown in Figure 29).

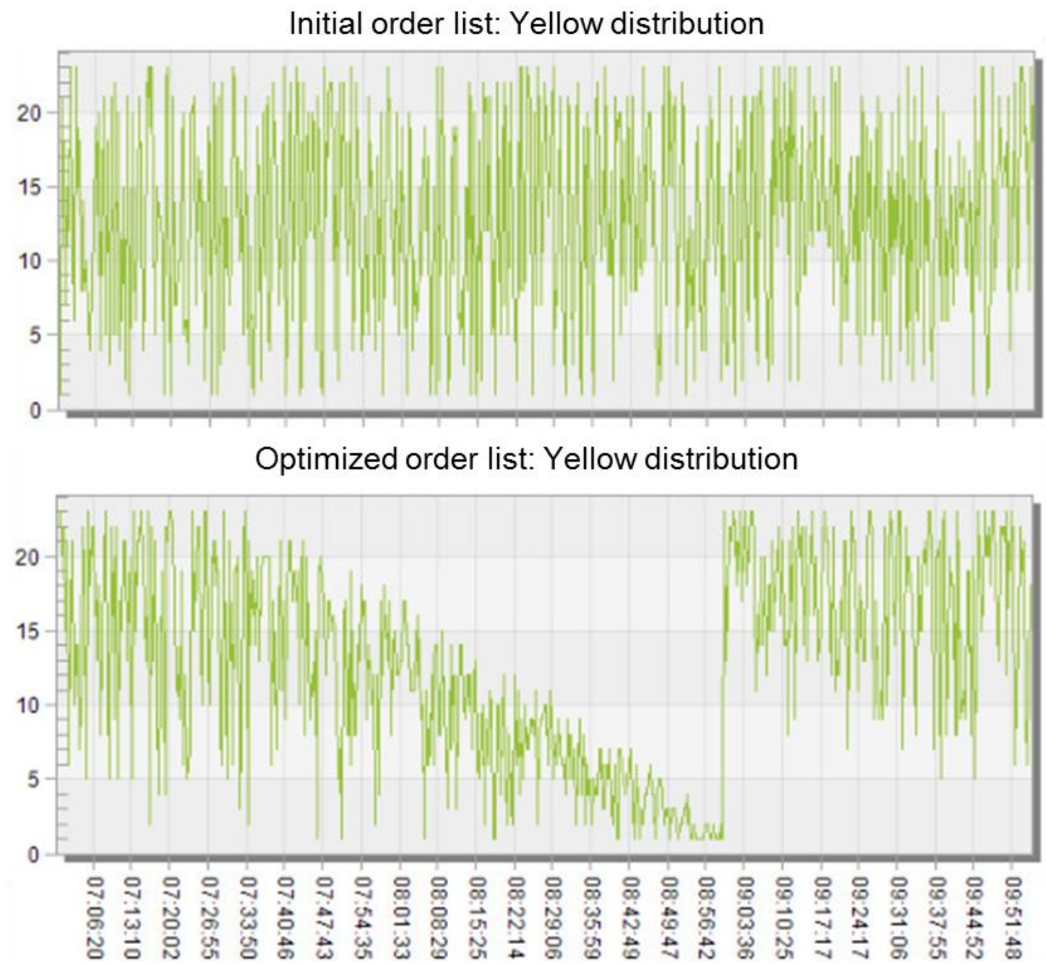

Figure 28: Example graphs illustrating the color distribution of yellow balls over the production time

All figures shown in this chapter belong to the example already used in Figure 23 and Figure 26, where 612 product orders are to be optimized. The bottom graph in Figure 28 shows a decrease of yellow proportion between 8am and 9am in the simulation of the optimized order list. This effect is due to the highest power requirement setting for the filling process of yellow balls (200kW according to Figure 16) and the most expensive energy price forecast between 8am and 9am (according to Figure 26). Consequently, bottles with a high share of yellow balls in their required color combination are preferably filled at another time and consequently avoided by the optimization algorithm between 8am and 9am.

- **Total Energy Costs [€]**

  Initial OL: 5.07468226181262
  OL after the 1. optimization step: 4.49036130202699
  OL after the 2. optimization step: 4.46589319755259
  Totally saved (between initial OL and OL after the 1. opt. step): 0.584320959785635
  Totally saved (between the OL of the 1. and the 2. opt. step) : 0.0244681044744013
  Totally saved (between initial OL and OL after the 2. opt. step): 0.608789064260036
  Totally saved (between initial OL and OL after the 1. opt. step) [%]: 11.514434394892
  Totally saved (between the OL of the 1. and the 2. opt. step) [%] : 0.544902800212441
  Totally saved (between initial OL and OL after the 2. opt. step) [%]: 11.996594719658

- **Total Peak Energy Cost [€]**

  Initial OL: 0.0254792227789887
  OL after the 1. optimization step: 0.0137232542681316
  OL after the 2. optimization step: 0.0137232542681316
  Totally saved (between initial OL and OL after the 1. opt. step) [%]: 46.139431382309
  Totally saved (between the OL of the 1. and the 2. opt. step) [%] : 0
  Totally saved (between initial OL and OL after the 2. opt. step) [%]: 46.139431382309

Figure 29: Example improvement statistics

After the presented graphs and statistics are generated, they are embedded in a report function of Plant Simulation, which allows implementation of the figures in an HTML page. Figure 30 shows the structure of the page created for the example application as well as a summary of the current optimization task.

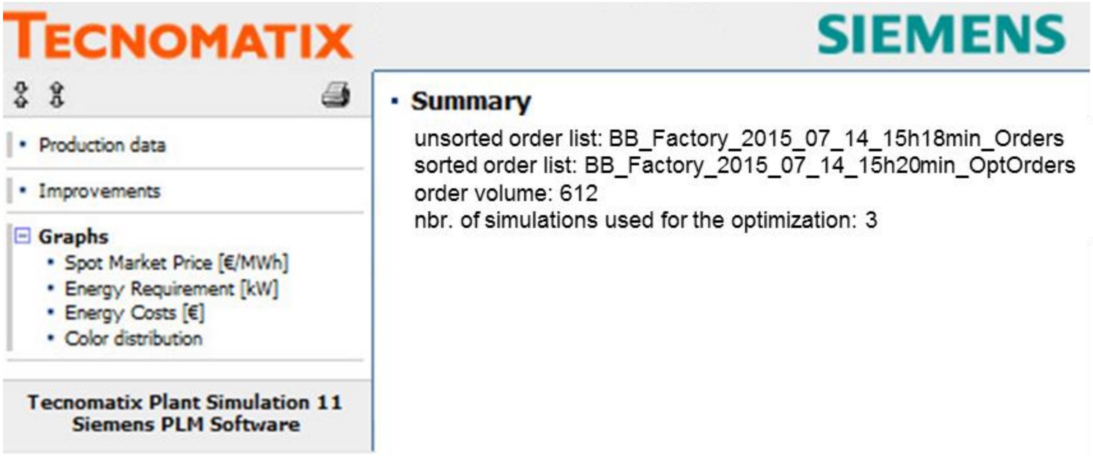

Figure 30: Report Layout of the application example

The created report is subsequently submitted to the WinCC user interface thereby indicating the completion of the optimization task.

5.7 Implementations in WinCC

The interface that is created for WinCC Runtime represents the operator's Human Machine Interface (HMI). Through the interface, the operator can create order lists, control their optimization processes, and test different scenarios by changing various settings.

5.7.1 Random order list creation

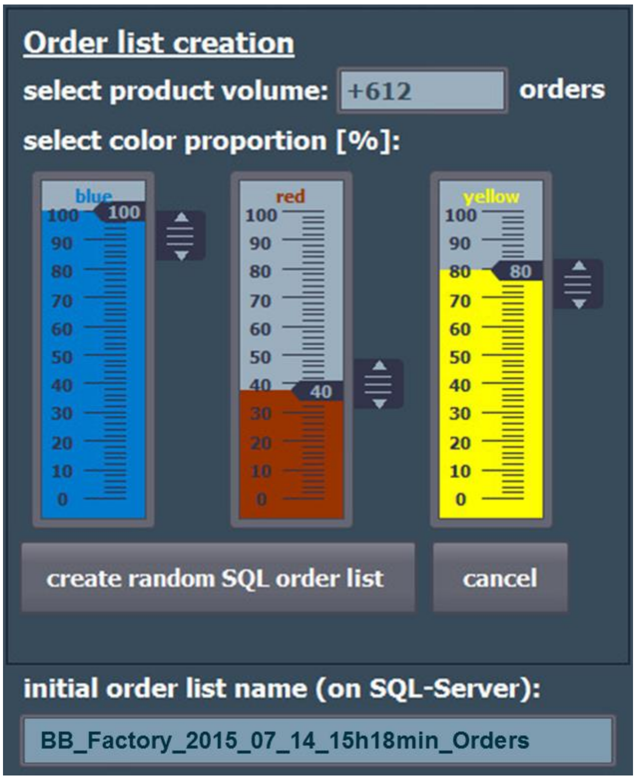

Figure 31: Interface for the random order list creation

Figure 31 presents the interface for the creation of a random order list. The interface allows for selecting specific amounts of products to be filled in the order list, as well as the proportion of colors. If the slider is set to 100 percent of a specific color, the application will randomly select between 1 and 30 balls of that color for each product. If the slider is set to 50 percent, between 1 and 15 balls are selected. The creation of the random order list can be initiated by pushing the button '*create random SQL order list*'. In the notification window (see Figure 33) the status of the process is continuously updated. When the order list is created and stored on the SQL Server, the order list name on the server is returned to the user interface by filling the textbox at the bottom of Figure 31.

5.7.2 Order Sequence Optimization

The interface for the control and input of specific properties of the Order Sequence Optimization is presented in Figure 32. In the upper part of the interface, the energy price forecast list can be chosen out of a selection box. Further property settings allow varying the simulated power requirement for the filling process for each color. By this means, the optimization algorithm can be tested under different conditions. The order sequence optimization in

Plant Simulation is initiated by pushing the button ‘*start order optimization*’. The process can be followed through the notification window and when it is completed, the optimized order list name is displayed in the textbox in the lower part of Figure 32. The completion of the task furthermore affects the HTML report page to become visible on the user interface.

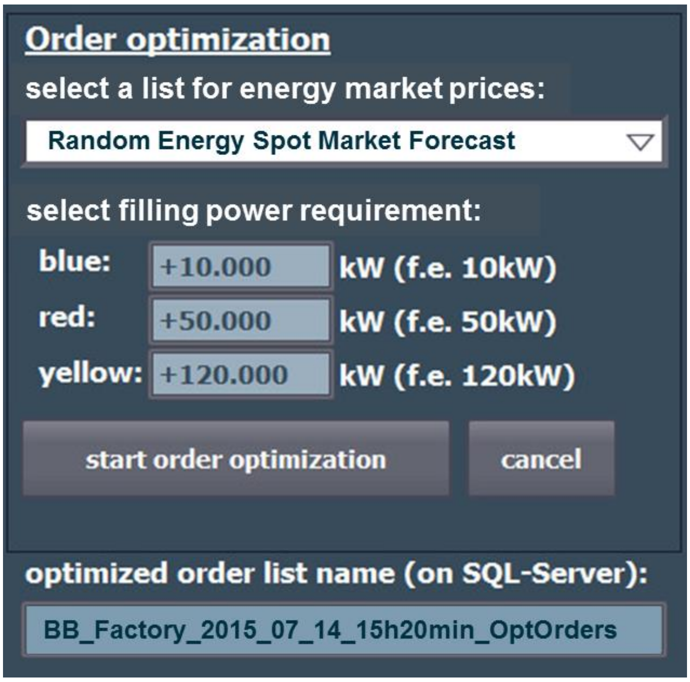

Figure 32: Interface for the order sequence optimization

5.7.3 Notification Window

	Nr.	Time	Date	State	Text
$	6000	15:20:26	14.07.2015	K	Plant Simulation message: Optimized order list: BB_Factory_2015_07_14_15h20min_OptOrders
$	6000	15:20:26	14.07.2015	K	Plant Simulation message: Optimized order list with 612 orders uploaded to SQL server.
$	6000	15:20:24	14.07.2015	K	Plant Simulation message: Optimization results will be prepared and transmitted to the HMI.
$	6000	15:20:23	14.07.2015	K	Plant Simulation message: Simulation 3 finished.
$	6000	15:20:18	14.07.2015	K	Plant Simulation message: Simulation 3 started.
$	6000	15:20:17	14.07.2015	K	Plant Simulation message: Order Optimization algorithm for Simulation Nr. 2 completed.
$	6000	15:20:17	14.07.2015	K	Plant Simulation message: Simulation 2 finished.
$	6000	15:20:12	14.07.2015	K	Plant Simulation message: Simulation 2 started.
$	6000	15:20:12	14.07.2015	K	Plant Simulation message: Order Optimization algorithm for Simulation Nr. 1 completed.
$	6000	15:20:12	14.07.2015	K	Plant Simulation message: Energy prices downloaded from Power sport market forecast.
$	6000	15:20:12	14.07.2015	K	Plant Simulation message: Simulation 1 finished.
$	6000	15:20:07	14.07.2015	K	Plant Simulation message: Simulation 1 started.
$	6000	15:20:07	14.07.2015	K	Plant Simulation message: Order Optimization started.
$	6000	15:18:01	14.07.2015	K	Plant Simulation message: Order list created: BB_Factory_2015_07_14_15h18min_Orders
$	6000	15:17:58	14.07.2015	K	Plant Simulation message: A new random SQL order list with 612 orders will be created ...

Figure 33: WinCC notification window

The notification window in WinCC shows all alarms and notifications relevant for the application to run. This includes errors, interface or communication problems, as well as status information. The example in Figure 33 shows the Plant Simulation notifications for a random order list creation (the two bottom lines), and a complete order sequence optimization run. The presented notifications for the latter are coherent with the described sequence of events in the Application Flow Control method presented in Figure 22.

5.7.4 User Interface Layout

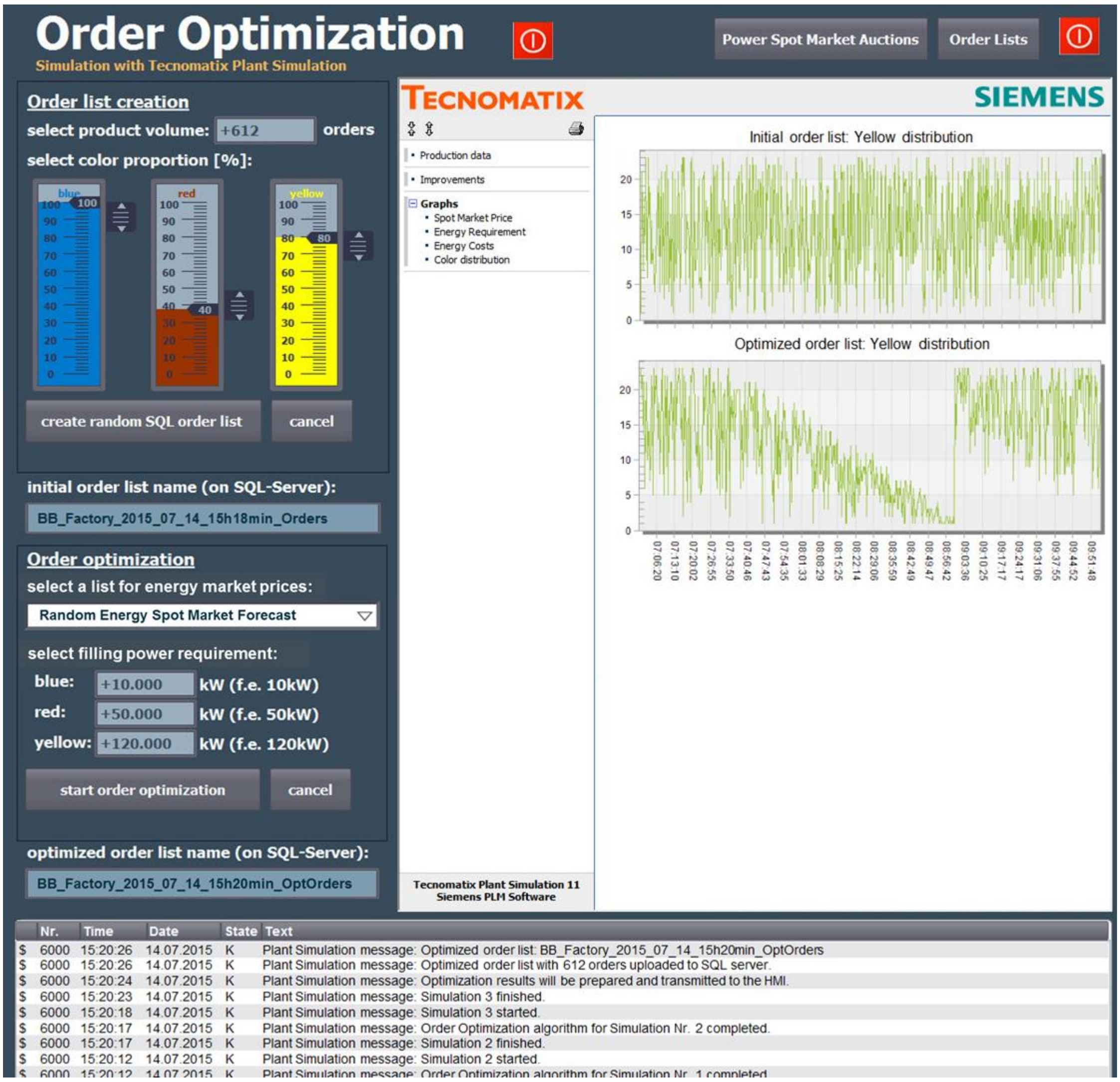

Figure 34: WinCC layout of the example application

Figure 34 shows the Layout of the WinCC Runtime user interface. It combines the presented control interfaces, the notification window, as well as the Plant Simulation report page. In the upper part of the figure, two additional buttons are available. One button enables the user to review the energy price forecast lists, and the other allows for reviewing the created order lists.

6 Conclusion and Outlook

Due to continuous changes in the business environment and competition on the global market, production systems are dependent on continuous improvement processes. Based on the current state of a production plant, it is necessary to streamline operations and make necessary changes in the production process. These changes must also take into account the energy flows in a production system. With the help of energy management systems the production planning of a facility can be operated in an energy-efficient manner. To contribute to this development, an example application using energy requirements of the facility as well as energy market prices from the Internet in order to optimize the facilities production plan is presented in this thesis. The example application aims to implement new design paradigms according to the trend of Industrie 4.0. Chapter 6.1 elucidates the relation between the example application and Industrie 4.0, and Chapter 6.2 gives an outlook on the implementation possibilities of the application in engineering training modules or as a marketable service tool.

6.1 Relation between the example application and Industrie 4.0

According to Chapter 2.5.1, the simulation model of the Backbone Factory represents a virtual copy of the physical facility and thereby enables the possibility of making decentralized decisions. Due to the availability of energy price forecasts, a new service can be provided to the facility. The service 'Order Sequence Optimization' can be implemented locally on the shop floor level in an operator panel, as a system wide service in the facilities Product and Plant Lifecycle Management, or as a service in the globally available Internet of Services (IoS), as explained in Chapter 2.5.3. The energy price forecasts could also be obtained through the IoS.

The optimization process of the example application can be manually executed by managers and operators or implemented in the automated decision processes of the facility resembling a Smart Factory as described in Chapter 2.5.4.

6.2 Outlook on the adaptation of the Example Application

Depending on the further adaptation of the example application, it can be implemented to serve the following two purposes.

6.2.1 Application as an Engineering Training Module

As stated in the motivation chapter of this thesis, an example implementation was designed to serve as the basis for an engineering training module using Industrie 4.0. A training module in this sense should elucidate the technical and conceptual changes and developments brought by the industrial revolution. According to the demands on engineers as explained in Chapter 2.10, an engineering training module could be created using the optimization application of this thesis. It would teach engineers on the one hand to develop safe, secure, and dependable systems using secure interfaces like OPC UA and ODBC, and on the other hand to create a system that resembles an open innovation network facilitating new applications and networks. The new ideas behind the example application are the use of Internet services into the automated decision making processes of the system. The example could furthermore be an Internet service itself. By implementing the application within a training module, engineers would perceive the idea of these new structural concepts.

6.2.2 Application as a Marketable Service Tool

To revise the example application in order to become a marketable service tool, some technical changes are necessary.

The delimitations applied in Chapter 5.3 have to be adapted to correspond to reality. This means changing the simulated power requirement of the filling station to the real power requirement. To make further decisions based on the power requirement of the whole production plant all other modules of the facility also have to be considered in the optimization process.

Further necessary changes of the application are the use of real energy price forecasts and real customer product orders instead of using simulated lists. Additionally, the delivery due date must also be considered as a constraint of the optimization process; however, because the due date was set wide enough to avoid violations, this particular constraint was not included in the optimization algorithm.

An additional improvement could be implemented in the optimization process by estimating the needed number of optimization steps. In this thesis, the number was statically set to two, but in some instances, just one optimization step would have been appropriate.

Bibliography

[1] P. D. T. Bauernhansl et al., „Industrie 4.0, Whitepaper FuE-Themen,“ 2014.

[2] Å. Fast-Berglund, M. Åkerman, M. Karlsson, V. G. Hernández and J. Stahre, „Cognitive Automation Strategies,“ *Variety Management in Manufacturing. Proceedings of the 47th CIRP Conference on Manufacturing Systems,* 2014.

[3] J. Lee, H.-A. Kao and S. Yang, „Service innovation and smart analytics for Industry 4.0 and big data environment,“ *Product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems,* 2014.

[4] G. Schuh, T. Potente, R. Varandani, C. Hausberg and B. Fränken, „Collaboration Moves Productivity To The Next Level,“ *Variety Management in Manufacturing. Proceedings of the 47th CIRP Conference on Manufacturing Systems,* 2014.

[5] A. Sproedt, J. Plehn, P. Schönsleben and C. Hermann, „A simulation-based decision support for eco-efficiency improvements in production systems,“ *Journal of Cleaner Production,* 2015.

[6] F. Shrouf and G. Miragliotta, „Energy management based on Internet of Things: practices and framework for adoption in production management,“ *Journal of Cleaner Production,* Bd. 100, pp. 235-246, 2015.

[7] „Förderschwerpunkte Industrie 4.0,“ Federal Ministry of Education and Research in Germany, [Online]. Available: http://www.softwaresysteme.pt-dlr.de/de/industrie-4-0.php. [Accessed on 17 06 2015].

[8] „The German Standardization Roadmap Industrie 4.0 - DKE,“ 04 2014. [Online]. Available: https://www.dke.de/de/std/documents/rz_roadmap%20industrie%204-0_engl_web.pdf. [Accessed on 23 06 2015].

[9] R. Drath, „Industrie 4.0 – eine Einführung,“ 2014. [Online]. Available: http://www.openautomation.de/fileadmin/user_upload/Stories/Bilder/oa_2014/oa_3/oa_3_14_ABB.pdf. [Accessed on 17 06 2015].

[10] H. Kagermann, W.-D. Lukas and W. Wahlster, „Final report of the Industrie 4.0 Working Group,“ *Recommendations for implementing the strategic initiative Industrie 4.0,* 2013.

[11] H. Kagermann, „Chancen von Industrie 4.0 nutzen,“ in *Industrie 4.0 in Produktion, Automatisierung und Logistik. Anwendung, Technologien und Migration*, 2014, p. 603–614.

[12] W. Bauer, S. Schlund, D. Marrenbach and O. Ganschar, „Industrie 4.0 – Volkswirtschaftliches Potenzial für Deutschland,“ 2014. [Online]. Available: https://www.bitkom.org/files/documents/Studie_Industrie_4.0.pdf. [Accessed on 17 06 2015].

[13] „Zukunftsprojekt Industrie 4.0,“ Federal Ministry of Education and Research in Germany, [Online]. Available: http://www.bmbf.de/de/9072.php. [Accessed on 17 06 2015].

[14] T. Bauernhansl, M. ten Hompel and B. Vogel-Heuser, Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien und Migration, Springer Vieweg, 2014.

[15] M. Hermann, T. Pentek and O. Boris, „Design Principles for Industrie 4.0 Scenarios,“ Technical University Dortmund, Dortmund, 2015.

[16] H. Kagermann, W.-D. Lukas and W. Wahlster, „Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution,“ 11 04 2011. [Online]. Available: http://www.vdi-nachrichten.com/Technik-Gesellschaft/Industrie-40-Mit-Internet-Dinge-Weg-4-industriellen-Revolution. [Accessed on 17 06 2015].

[17] „Der Weg zur Industrie 4.0,“ Plattform Industrie 4.0, 2015. [Online]. Available: http://www.plattform-i40.de/hintergrund/rueckblick. [Accessed on 17 06 2015].

[18] „Plattform Industrie 4.0,“ [Online]. Available: http://www.plattform-i40.de/. [Accessed on 18 06 2015].

[19] „Smart Manufacturing Leadership Coalition,“ [Online]. Available: https://smartmanufacturingcoalition.org/. [Accessed on 18 06 2015].

[20] P.C.o.A.o.S.a. Technology, „the White House,“ 2014. [Online]. Available: https://www.whitehouse.gov/sites/default/files/microsites/ostp/PCAST/amp20_report_final.pdf. [Accessed on 18 06 2015].

[21] P. C. Evans and M. Annunziata, „Industrial Internet,“ 26 11 2012. [Online]. Available: http://www.ge.com/docs/chapters/Industrial_Internet.pdf. [Accessed on 18 06 2015].

[22] T. Bürger and K. Tragl, „SPS-Automatisierung mit den Technologien der IT-Welt verbinden,“ in *Industrie 4.0 in Produktion, Automatisierung und Logistik. Anwendung, Technologien und Migration*, Springer Vieweg, 2014, pp. 559-569.

[23] S. Dais, „Industrie 4.0 – Anstoß, Vision, Vorgehen,“ in *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien und Migration*, Springer Vieweg, 2014, pp. 625-634.

[24] „Glossar Industrie 4.0 des Fachausschuss VDI/VDE-GMA 7.21 "Industrie 4.0",“ [Online]. Available: http://www.iosb.fraunhofer.de/servlet/is/48960/Begriffsdefinitionen_VDI_GMA_FA7-21_v4.pdf?command=downloadContent&filename=Begriffsdefinitionen_VDI_GMA_FA7-21_v4.pdf. [Accessed on 19 06 2015].

[25] E. Geisberger and M. Broy, „Living in a networked world: Integrated research agenda Cyber-Physical Systems,“ 03 2015. [Online]. Available: http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Publikationen/Projektberichte/acaetch_STUDIE_agendaCPS_eng_WEB.pdf. [Accessed on 19 06 2015].

[26] M. Weiser, „The Computer for the 21st Century,“ 09 1991. [Online]. Available: https://www.ics.uci.edu/~corps/phaseii/Weiser-Computer21stCentury-SciAm.pdf. [Accessed on 22 06 2015].

[27] CERP-IOT, „Internet of Things: Strategic Research Roadmap,“ 15 09 2009. [Online]. Available: http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2009.pdf. [Accessed on 22 09 2015].

[28] D. Giusto, A. Iera, G. Morabito and L. Atzori, The Internet of Things, Springer Vieweg, 2010.

[29] P. Buxmann, T. Hess and R. Ruggaber, „Internet of Services,“ *Business & Information Systems Engineering,* pp. 341-342, 2009.

[30] A.-W. Scheer, „Industrie 4.0. Wie sehen Produktionsprozesse im Jahr 2020 aus?,“ IMC AG, Saarbrücken, 2013.

[31] „Autonomik für Industrie 4.0," [Online]. Available: http://www.autonomik40.de/_img/article/AUTONOMIK_Broschuere_barrfrei.pdf. [Accessed on 22 06 2015].

[32] J. Schlick, P. Stephan, M. Loskyll and D. Lappe, „Industrie 4.0 in der praktischen Anwendung," in *Industrie 4. 0 in Produktion, Automatisierung und Logistik. Anwendung, Technologien und Migration*, Springer Vieweg, 2014, pp. 57-84.

[33] „SmartFactory: Keyfinder production line," SmartFactory, 2015. [Online]. Available: http://www.smartfactory.de/. [Accessed on 23 06 2015].

[34] „The Service-Orientation Design Paradigm," Arcitura™ Education Inc., 2015. [Online]. Available: http://serviceorientation.com/serviceorientation/the_service_orientation_design_paradigm. [Accessed on 24 06 2015].

[35] J.-Y. Chung and K.-M. CHao, „A view on service-oriented architecture," *SOCA: Service Oriented Computing and Applications,* Bd. 1, Nr. 2, pp. 93-95, 06 2007.

[36] H. Haas, „Service-Oriented Architecture," World Wide Web Consortium, 2003. [Online]. Available: http://www.w3.org/2003/Talks/0521-hh-wsa/slide5-0.html. [Accessed on 03 07 2015].

[37] R. Heinze, „Der Ingenieur im Zeitalter von Industrie 4.0," 2015. [Online]. Available: ftp://ftp.ni.com/pub/branches/germany/2015/artikel/03-march/07_Der_Ingenieur_im_Zeitalter_der_Industrie_4.0_Ronald_Heinze_ETZ_1-2_2015.pdf. [Accessed on 03 07 2015].

[38] *Machine-oriented simulation: Simulation of systems in materials handling, logistics and production,* VDI, 2007.

[39] M. Putz, A. Schlegel, S. Lorenz, E. Franz and S. Schulz, Gekoppelte Simulation von Material und Energieflüssen in der Automobilfertigung, Chemnitz, 2011.

[40] J. Hesselbach, Energie- and klimaeffiziente Produktion: Grundlagen, Leitlinien und Praxisbeispiele, Wiesbaden: Spreinger Vieweg, 2012.

[41] „European Power Exchange: Company Information," European Power Exchange (EEX), 2015. [Online]. Available: http://www.eex.com/en/about/eex. [Accessed on 26 06 2015].

[42] „EPEX SPOT: Company Information," European Power Exchange (EEX), 2015. [Online]. Available: http://www.epexspot.com/en/company-info/cooperation. [Accessed on 26 06 2015].

[43] D. W. Bunn, Modelling Prices in Competitive Electricity Markets, New York, NY, USA: Wiley, 2004, pp. 1-17.

[44] S. Voronin and J. Partanen, „Price Forecasting in the Day-Ahead Energy Market by an Iterative Method with Separate Normal Price and Price Spike Frameworks," *Energies,* Bd. 6, pp. 5897-5920, 12 11 2013.

[45] M. Unterstein and G. Matthiessen, Rationale Datenbanken und SQL in Theorie und Praxis, Berlin: Springer Vieweg, 2012.

[46] E. Schicker, Datenbanken and SQL: Eine praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL, Bd. 4, Wiesbaden, Germany: Vieweg+Teubner Verlag, 2014, p. 3.

[47] R. Steiner, Grundkurs Relationale Datenbanken: Einführung in die Praxis der Datenbankentwicklung für Ausbildung, Studium und IT-Beruf, Wiesbaden, Germany: Springer Vieweg, 2009.

[48] M. Metter and R. Bucher, *IT in der Industrieautomatisierung,* Erlangen, Germany: Siemens Public Corporate Publishing, 2003.

[49] „What is OPC?," OPC Foundation, 2015. [Online]. Available: https://opcfoundation.org/about/what-is-opc/. [Accessed on 29 06 2015].

[50] W. Mahnke, S.-H. Leitner and M. Damm, OPC Unified Architecture, Berlin, Germany: Springer, 2009.

[51] K. F. Früh and W. Ahrens, Handbuch der Prozessautomatisierung: Prozessleittechnik für verfahrenstechnische Anlagen, München, Germany: Oldenburg Industrieverlag, 2004.

[52] „OPC Unified Architecture - Pioneer of the 4th industrial (r)evolution," 2014. [Online]. Available: http://www.opcfoundation-events.com/uploads/media/OPC-UA-Pioneer-of-IE40-US-v2.pdf. [Accessed on 29 06 2015].

[53] L. Liu and M. T. Özsu, „ODBC," in *Encyclopedia of Database Systems*, Springer, 2009, p. 1947.

[54] J. Postel and J. Reynolds, „File Transfer Protocol (FTP)," 10 1985. [Online]. Available: https://tools.ietf.org/html/rfc959. [Accessed on 02 07 2015].

[55] „Totally Integrated Automation Portal: One integrated engineering framework for all automation tasks," 2013. [Online]. Available: http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_tia_portal_en.pdf. [Accessed on 01 07 2015].

[56] „XAMPP: Apache + MySQL + PHP + Perl," Apache Friends, [Online]. Available: https://www.apachefriends.org/about.html. [Accessed on 01 07 2015].

[57] T. Kamleiter, *Konzeption einer Vorgehensweisezur automatisierten Eingabe von Betriebsparametern für eine kombinierte Material- und Energieflusssimulation in Plant Simulation; Master's Thesis,* Erlangen, Germany, 2015.

[58] P. Brown, J. A. Estefan, K. Laskey, F. G. McCabe and D. Thornton, „Reference Architecture Foundation for Service Oriented Architecture Version 1.0," 04 12 2012. [Online]. Available: http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/soa-ra-v1.0-cs01.html#_Toc343761541. [Accessed on 03 07 2015].