MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„A Modeling Method for the Internet of Things"

verfasst von / submitted by

Daniel Kaiser, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Diplom-Ingenieur (Dipl.-Ing.)

Wien, 2016 / Vienna 2016

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 926
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Wirtschaftsinformatik
Betreut von / Supervisor:	Univ.-Prof. Mag. Dr. Hans-Georg Fill

Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my sincere thanks to my supervisor Univ.-Prof. Mag. Dr. Hans-Georg Fill who always had useful insights and tips right from when I started with this thesis. Dr. Hans-Georg Fill could always show me the right direction when I felt like I would be stuck in certain matters.

Furthermore, I would like to thank my family who supported me during my Studies. This would not have been possible without them.

Abstract

The Internet of Things (IoT) appears to be one of the most promising trends in IT. Extreme growth is predicted and Michael Porter, renowned professor of Harvard Business School, even regards it as the "third wave of IT-driven competition" after IT per se and the Internet. The present master's thesis first discusses the current state of the art of the IoT before presenting a modeling method tackling the changes induced by sensors and intelligent products in business processes or architecture. Furthermore, an entirely new model to represent IoT devices as well as an ontology to classify IoT sensors have been developed. The implemented models have been evaluated with a brief, notional example.

Keywords: Internet of Things, IoT, modeling method, metamodeling, ADOxx, SeMFIS, sensors, ontology

Chapter 1

Preface

1.1 Introduction and Motivation

The Internet of Things (short: IoT) has rapidly grown in the course of the past years as has been evidenced by predictions of polling firms as well as in opinion research. IoT basically means creating a network of networks and connecting anything and everything with each other, whether within a local network or actually on the internet. Big leaps in research on computation over the last decades have made it possible to transform tiny "computers" into basically any product imaginable that enables them to connect to the internet and be smart. The limits to the IoT have not been reached yet and almost everyday new ideas arise that push it even further. Connected cars such as Teslas, smart cities ([70]), refrigerators that independently re-order their inventory, or even research on Parkinson's disease in the health care sector ([47]) - the fields of application are highly diverse.

Since the IoT is still in an early phase of development, not to say in the fledgling stages, the predictions for growth in the years to come are quite massive. A recent report from research firm Gartner shows that 6.4 billion connected objects will be used in 2016, which amounts to an increase of 30 percent compared to the year before [25]. By 2020 almost 21 billion connected devices are expected to be in use globally. This does not only result in a huge number of objects on the market but also opens opportunities for companies to enter and compete on the market. Gartner's Hype Cycle 2015 attests that the IoT will reach its plateau in the next five to ten years, which further assures the promising future (figure 1.1, [26]). The IoT might even apply to the clothing sector by connecting 10 billion fashion products that, e.g., assist in finding shoes that have been lost or getting instructions on how to wash new jeans properly [25, 3].

With all this in mind, this master's thesis is composed of an analysis of the current state of the IoT, especially with a view to how businesses will have to adapt to this new situation. Subsequently, a modeling method to properly map out such complex systems and their internal business processes on the computer will be outlined.

This section is followed by the research questions on which this thesis is constituted before a conceptualization of crucial concepts is undertaken in a chapter on the theoretical basis. Chapter 2 State of the Art - Internet of Things gives a brief overview of the *things* themselves and which components they contain. Sections 2.2 Changes in Businesses and 2.3 Transformation of Companies and Competitive Advantage partially rely on [53, 52] by Porter and Heppelmann and seek to characterize the shift of conduct not only in companies but in many industries all around the world. New dogmas such as the New Technology Stack and an IoT Cloud, a general shift in the mindsets of businesses and the respective staff, or new divisions of the organizational structures are presented. Section 2.4 Possible Topologies, remaining Problems, Standardization and a brief Outlook of the IoT enumerates critical points of the IoT among other things.

Chapter 3 IoT Modeling Method - a New Approach is concerned with the accompanying modeling method. First, conceptual models will be outlined before stating its scope and giving a short introduction into metamodeling and the ADOxx platform the method was implemented on. Section 3.6 Ontology introduces an ontology for categorizing (for the IoT) essential sensors.

The last two chapters 4 Evaluation of Modeling Method and 5 Conclusion constitute the final part of the present thesis. While the former presents models created on the basis of the IoT modeling method, the conclusion identifies and synthesizes the main accomplishments of this thesis.

1.2 Research Questions

Derived from the Introduction and Motivation, the questions this master's thesis tries to answer are split in two integral parts: a more theoretical part that sums up the IoT and its smart, connected products. The basis of this section constitute two recent articles by Michael Porter and James Heppelmann on this very topic. The second part provides a brief introduction to metamodeling and the ADOxx platform after introducing the conceptual models.

While the IoT itself has been strongly researched (and still is), modeling meth-

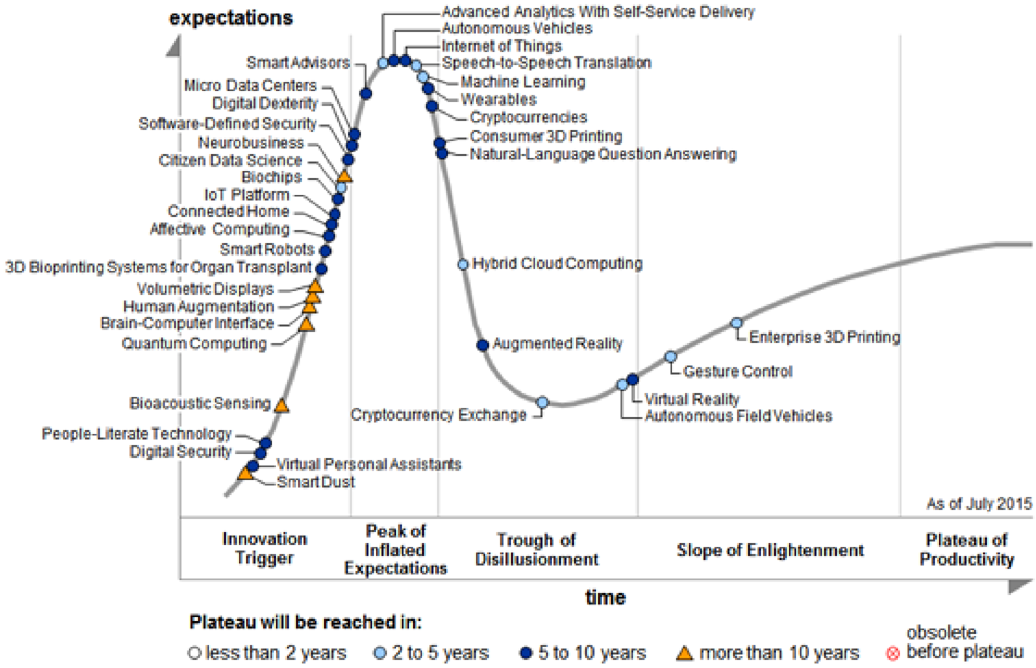


Figure 1.1: Gartner's 2015 Hype Cycle [26]

ods are still lacking. In fact, the only project similar to such an endeavor is by the IoT-Architecture group that developed a concept for modeling IoT-aware processes [56]. In other words, this thesis endeavors to close the present research gap.

The research questions are:

- What is the current state of the IoT (state of the art)? How will businesses have to change in the future to compete in this new, emerging market? What is planned for the future?

- What other projects exist in terms of modeling and IoT? How will a modeling method for the IoT look like and what is needed to realize such a method?

- How would the modeling method applied to use cases look like?

Chapter 2

State of the Art - Internet of Things

2.1 Smart, connected Products and the Internet of Things

The term Internet of Things was first mentioned in 1999 by Kevin Ashton who, at this time, held a presentation at Procter & Gamble. His idea was to link RFID technology in the supply chain of Procter & Gamble with the Internet, and with that he drew the attention of the executive members to this novel thinking [5].

In 2016, the IoT is about to change our connected world once again very drastically. Businesses will have to reconsider their whole strategy, services and products as they could become obsolete as soon as a new business enters the market with a disruptive idea that changes the current state. Smart, connected products, as Michael Porter and James Heppelmann describe the objects of the IoT in their Harvard Business Review article from November 2014, not only have a physical component with mechanical and electrical parts but also smart components such as sensors, external data sources with storage, microprocessors and software, and furthermore, connect to everything they need to. This was rendered possible by the ever increasing processing power and miniaturization of "computers" which offer new perceptions for companies and their products and services. The functions of these new products can be classified into monitoring, controlling, optimization and autonomy. Heppelmann and Porter even write that many businesses will have to ask themselves the question, "What business am I in?" and how can value be created and captured in this new world [53].

The IoT is a term that combines various new and emerging disciplines of IT: smart cities, smart homes, smart products, cyber-physical systems and many more [70, 68]. Objects with added sensors, localization or nearfield communications which therefore get smart are seen as building blocks for the IoT [40]. General surveys on the IoT are provided by Atzori et al. and Li et al. [7, 43].

According to Heppelmann and Porter, smart, connected products and the IoT are the "third wave of IT-driven competition". IT remodeled competition and strategy already twice in the last 50 years. After the automation of value chains and the Internet, the IoT will be the third major revolution. The first two waves of IT-driven competition aimed at efficiency while the products themselves largely remained the same. However, the IoT will change this situation: Sensors, external data sources and microprocessor will help build more advanced products that offer superior performance and the ability to update themselves with new features. The two researchers predict that this third wave could be the biggest yet, as it not only boosts productivity but offers large potential to innovate products and service models around those, to finally generate economic growth [53]. Ackermann further foresees a drastic change in how a product will be recognized with the IoT [1].

2.2 Changes in Businesses

2.2.1 New Technology Stack and an IoT Cloud

Before starting with the changes in business processes and the transformation of entire companies that will happen due to IoT systems in companies, a consideration of the new technology stack proposed by Porter and Heppelmann in their Harvard Business Review article on the IoT, is expedient (figure 2.1).

The core parts of this stack are constituted by the products themselves as well as a product cloud behind those. To facilitate communication between products and their respective cloud, connectivity between these components, either wired or wireless, is essential. The cloud is even more vital for such devices since wireless modules are smaller and allow for thinner and smaller products. In professional environments, however, e.g., industry in which a loss of connection would have a huge impact, large products are often preferred since space is not restricted and hence, allows for a wired connection. As already stated in section 2.1 Smart, connected Products and the Internet of Things, a product has two major parts: software (applications, OS, user interface,...) and hardware (microprocessors, sen-

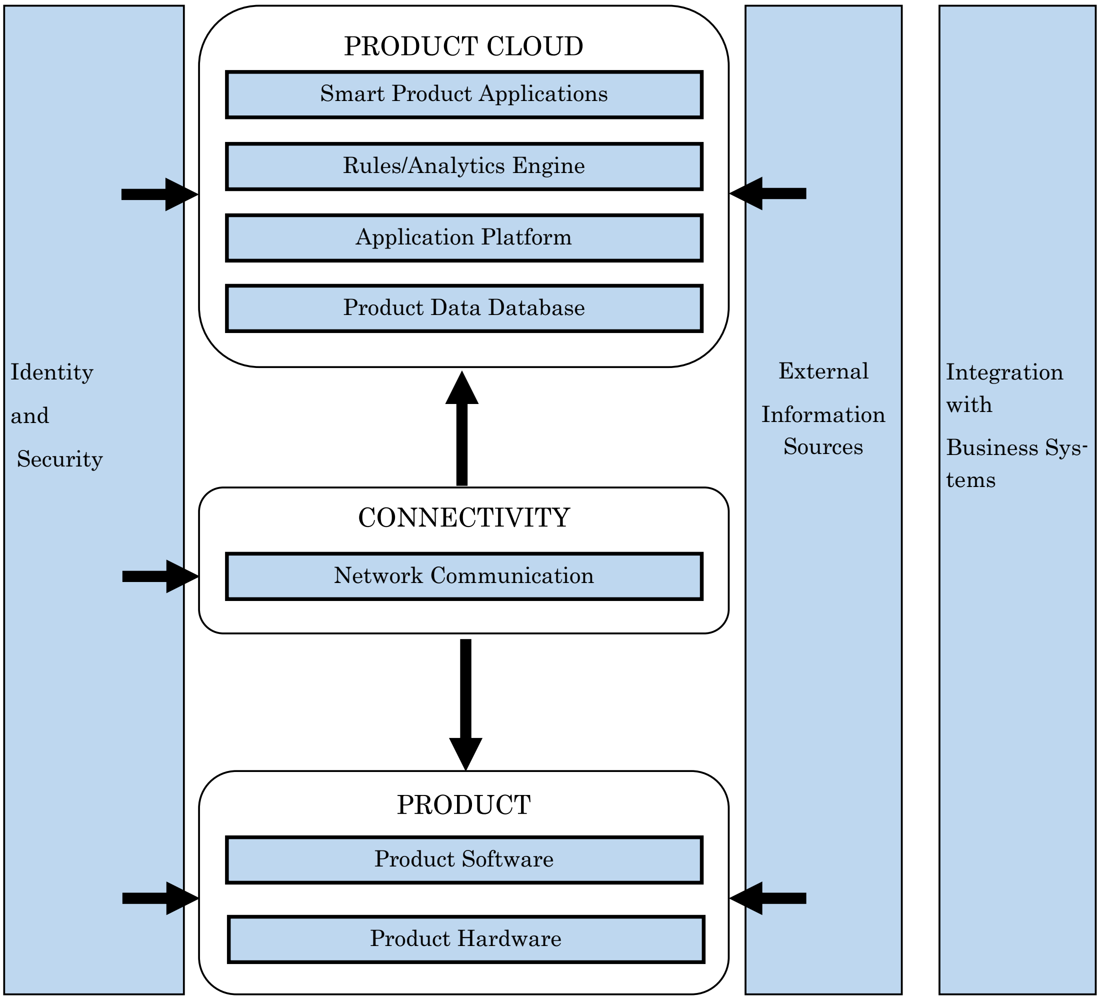


Figure 2.1: New Technology Stack according to [53]

sors, connectivity components,...) [53]. To make a product smart, a product cloud is paramount, otherwise, sensors collecting data are almost useless or at least they cannot unleash their full potential. Hwang et al. and Botta et al. explain clouds in context of the IoT in depth in their publications [30, 11]. Tao et al. propose an IoT-based cloud manufacturing service system to tackle incurring bottlenecks in this field of application [59]. According to Porter and Heppelmann, a product cloud has to offer a lot of different IT-related components that companies, which offer traditional products, such as thermostats or drilling machines, usually do not dispose. However, these products are already smart (e.g. Nest) or will become smart in the coming years. Therefore, companies will have to implement such a cloud. Within a product cloud, various issues such as applications or a database to manage incoming data in real-time have to be considered as seen in figure 2.1 [53].

Since the purpose of IoT is to interconnect everything, this new technology stack is not an enclosed system between a product and its cloud. To put it in different terms, there are a few points from the outside to remember. In such a connection-heavy environment attacks from outside or even inside can be a problem (more on security which plays an integral role in the IoT in the section on Security and Privacy Issues). The data to process by the products' applications does not necessarily only originate from the internal sensors of a product but can be derived from information sources outside a system as well. A Nest thermostat could, e.g., be connected to exterior weather stations that have a local forecast to check whether heating or cooling is needed to sustain a user's preferred temperature. To realize a properly working link between items and clouds, Kovatsch et al. researched the possibility to move application logic into the cloud, resulting in IoT objects as thin servers which only use REST resources for their range of functions [41]. Balalaie et al. depict their experience with microservices, i.e. an emerging IT architectural style to build applications that are split up in many small and easily exchangeable parts to ensure an architecture that works adequately with the concept of cloud computing [8]. The idea worth considering in Heppelmann and Porter's technology stack is the integration of IoT in business systems that are already present in companies. Data from the smart, connected devices need to be added to existing core enterprise business systems like CRM or ERP [53].

2.2.2 IoT Mindset

Value creation and value capture will change dramatically with the IoT. The former describes how a business increases the value of its offered products and services in order to get new customers while keeping the already existing ones. Value capture, on the other hand, is the generation of income on the basis of the value previously created. Both of these are of vital importance and have to be thought-out sufficiently in order for a company to be successful. However, the manner of managing value creation and capture will be altered in the near future. For instance, the value of a traditional product is created with produce that triggers the customers' needs or desires. Subsequently, the assessed price, adequate to the value, should maximize profits. With the IoT a paradigm shift within the companies' mindsets is required as the IoT opens new, progressive possibilities [29]. Dawid et al. examine smart, connected products from a managerial point of view and realize that intelligent products will have a major impact on the industry. Still, challenges on company-internal and strategic levels remain that need to be resolved [14].

If a traditional product is rendered smart by adding sensors and connecting it to other products and a product cloud, it is suddenly possible to extend its features with updates that constantly improve it. Moreover, corresponding services that customers' acquire since they meet their interest might be offered. Products with an extended lifespan due to updates or services expanding the usability, enable companies to tie customers to the product or platform, e.g., a product featuring various sensors to track the environment and recognize defects earlier (or the development of one). If these sensors reach a critical level the device automatically warns the owner and shuts down, which could prevent the product from a complete breakdown. In the next step, the device could inform its producer or service center of the need for maintenance, who then offers added services and/or sends a service employee. Due to the integrated sensors, the employee immediately recognizes the defect and is able to mend it without having to spend valuable time on finding the reason for failure. Evidently, businesses can bind their customers and make a profit with a product and create recurring revenue even if is was sold months or years before. On the other hand, since customers do not have to deal with the repair of the product, the overall satisfaction should be facilitated and assured.

The recurring revenues, be it via value-added services, applications, updates or subscriptions, can excel the original price of a product easily and according to Gordon Hui, are more appealing to venture capitalists as the model does not rely on customers that might or might not buy a second product from a company [29].

Since the IoT offers a radical change in various areas of a business, a shift of the focus from traditional products to IoT supportive products is imminent. Efforts for business models that support IoT have been made by Dijkman et al., Jaehyeon Ju et al. and Bucherer and Uckelmann [15, 36, 12]. In terms of value creation, the customer's needs, offerings of a business and the role of data have to be attended to. Within a traditional product mindset, customer needs are tackled in a reactive manner, which means already existing needs are met by the product. The IoT mindset is designed to be more predictive in order to resolve real-time problems, such as applies for products that recognize problems in advance and induce measures in order not to risk total breakdown. The offering possibilities of traditional compared to IoT products differ as well. Traditional products will eventually get outdated and hence replaced with a new one. An IoT product, on the other hand, is capable of updating and upgrading to new functions due to being connected to a cloud that, e.g., releases over-the-air (OTA) updates. As for the role of data, an IoT product collects and analyzes information during usage and therefore, enables a company to get useful insights of costumers' usage to further improve the product and adjust corresponding services. Traditional products do not offer such data, which complicates the identification of requirements for future iterations of a product. Differences in the mindsets regarding value capture are apparent with a view to the path to profit, control points and capability development. As already stated above, a business offering smart, connected products makes profit by selling the initial product as well as with value-added services, applications or subscriptions that enable recurring revenue. In the traditional product mindset the path to profit is mostly selling the next product. Control points in the traditional mindset possibly comprise the brand of a products manufacturer, commodity advantages and intellectual property ownership of patents or copyrights. Because of the real-time occurrence of data, awareness of the environment and predictive manner, it is possible to offer personalized products. The last point to consider about capturing value is the capability development that includes the ability to comprehend the whole stack in an ecosystem in the IoT mindset. Since various devices are connected that do not necessarily have to be the same type or even from the same company, it is important to understand and learn from other businesses in the ecosystem the product is in and perhaps even collaborate with other companies [29].

2.3 Transformation of Companies and Competitive Advantage

2.3.1 Transformations of Companies in General

Operational effectiveness, the combination of intact supply and value chains, as well as an up-to-date infrastructure provide the basis for a successful company. However, operational effectiveness can be easily simulated by competitors and hence, does not guarantee obtaining an advantage position over competition for a long term. A business has to position itself strategically on the market by attracting customers with unique products or services that possibly turn out to be perseverative. Smart, connected products will have a massive impact on companies' internal operation. Design principles, offered services and even organizational structures might not remain the same. In a second Harvard Business Review Article on the IoT, Michael Porter and James Heppelmann focus on how the smart, connected products might transform companies internally. They further provide deeper insights in product development, manufacturing, logistics, marketing and sales, after-sale service, security and human resources – basically the whole value chain will be altered [52]. Gerpott and May establish possibilities to integrate IoT components in the product portfolios of businesses, which they describe in a business development framework. They define three possible roles of a component. While two of these roles only complement existing products (e.g. better and faster parcel tracking through tags), there is also the "innovation" role, i.e. IoT components are the main drivers of a product or service, and appeal by alluding that such components would not have been possible before (e.g. smart home or mobile health tracking) [27].

Two recent theses tried to answer whether the IoT will have a big impact on new product development and if it can help to improve it [50, 65]. In product development, an ageless design will be more important than ever as the product itself can be updated and gain new features via software updates instead of with recurring hardware refreshes as with regular products. With this long term support the quality management of products has to continue as well to keep customers satisfied while development times for updates should be kept low. Variability of products can be easily created with software, which can create products that fit a specific customer's or a certain group of customers' needs although the hardware of the product is the same, which constitutes a clear advantage for the IoT. This might even result in lower costs of the product due to a common hardware base. Mixed opinions have been identified if the IoT's incoming data helps to control development costs.

New user interfaces might emerge that require regular updating as well. GoPro, for example, a small action camera, uses smartphones for displaying current pictures, which facilitates keeping the camera itself extremely compact as well as more resilient, both important factors for the main clientele, i.e. action sports athletes. User interface and camera can both be conveniently updated with a smartphone app. In a previous subsection of this thesis, 2.2.2 IoT Mindset, new business models such as product-as-a-service with the possibility to access devices via a remote have already been discussed. Another important factor in product development is the interoperability between different systems, which might imply that manufacturers have to cooperate and co-design their products. In this way, real added value would be established for customers and consequently, they would buy both products [52].

Manufacturing as well as logistics have already started to change, too. In manufacturing, Industrie 4.0 (within German speaking areas) and Smart Manufacturing (USA) show the advantages and possibilities of smart devices that network with each other [9]. They optimize production, warn and automatically stop other connected machines if anything is amiss as well as alert staff to examine it. As has been pointed out with the GoPro example, smartphones are capable of assuming the tasks of the screen which makes the product itself simpler and for a manufacturer easier and cheaper to produce. Manufacturing does not stop after a product's manufacturing but rather has to be regarded as a continuous development. Logistics, on the other hand, initiated their becoming smart earlier with the so-called RFID tags that have helped tracking shipments since the 1990s [24]. Today, not only tracking can be done seamlessly and without scanning tags; sensors offer more useful data, for example, weather conditions and information on traffic, which makes it possible for trucks to reroute and arrive at the destinations on the fastest route [52].

Marketing and sales as well as after-sale service are affected by IoT devices, too. With the incoming data of a smart product, companies can build more accurate customer segments and create user profiles that would not have been feasible before. They identify features that customers tend to use more often and those that are not used much or those that customer's usually fail to use properly. With such information marketing specialists can address their clientele better, especially through adequate channels. Maier classifies the IoT in five subdomains and investigates the IoT and its marketing potential from a consumer marketing perspective [45]. Another thing to keep in mind is that customers might not require only one smart, connected product but rather a whole system. Home automation, for example, integrates a smart light bulb in a bigger system. Such information should be seized by marketing managers for adverts. An IoT product should be regarded the beginning rather than the end of a business – customer relationship, particularly with after-sale services in mind [52]. A more uncommon marketing concept is introduced by Jara et al. in [34] , i.e. interactive, participative marketing with the help of the IoT.

Security is another point to consider given that huge amounts of partially sensitive data accumulate that have to be secured at any point, whether during a connection or in a storage afterwards. Gaining customers' trust is crucial. Security in IoT will be addressed in more detail in the subsection 2.4.2 Security and Privacy Issues that focuses on remaining problems of the IoT and new solutions tackling them.

With more focus on software components, human resources become more important. Traditional products did not require software engineers, data analysts or specialists for product clouds. Nowadays, however, such professions are essential for progressing in the era of the IoT by keeping up with the changing market. Not only new personnel is required but the present staff of a company will have to adapt to an IoT mindset, too. Due to a shortage in IT personnel, the necessary skills are difficult to cover. To counteract that, some firms even relocate their headquarters to typical technology areas in the United States, such as Silicon Valley or Boston. Another possibility is outsourcing to IT centers in India or Pakistan. Still, specialists will be needed locally as well [52].

2.3.2 Data

According to Heppelmann and Porter data is paramount for transformations[52]. For decades, data has been gathered internally, e.g., while testing products or by external partners. Surveys, transactions with business partners or direct contact with customers provided useful insights about customers and their demands to the companies. However, information on the products themselves, such as usage and possible reasons for failure, was rather scarce. Today, data provided by the sensors of smart, connected products is highly relevant for the companies. The high amount of data produced has to be analyzed instantly. Big data and its special analysis tools are needed to generate value out of the data [2]. Compressed sensing could help decreasing the amount of incoming data by neglecting redundant data. Li et al. present a compressed sensing framework for the IoT in [44]. With the former methods of data generation, data was regularly analyzed individually. The newer, more progressive methods, however, eventually bear possibilities to generate information through the product itself. Data gathered from the device adds a whole new dimension to it.

This new resource of knowledge will force companies to not only reconsider their products but also their whole internal workflows and human resources. The analysis of data will be crucial for creating real value compared with traditional products. This endeavor, however, will not be conductible without difficulty since the total of data from IoT products as well as sales figures or service histories, will appear unstructured and multifaceted, and will have to be combined. Data mining is crucially important for coping with such problems. A survey by Tsai et al. gives further insights into the incurring IoT data and data mining [60]. Bin et al., on the other hand, introduce four data mining models specifically for the IoT [10]. A “data lake” could collect the massive amount of data in their original formats to analyze them with new programs specifically for these kind of big data tasks. Figure 2.2 from [52] shows Porter’s and Heppelmann’s intentions in this field.

2.3.3 Organizational Structure

All the imminent transformations from regular to IoT products will not be possible without changing a company’s organizational structure. Especially the software components might be a novelty to former manufacturers of traditional products; because of that, companies can learn from software industry. Since the software industry already operates with the cloud and regulates processes remotely, a similarity with smart, connected products is apparent. However, even from a software vendor’s point of view, the IoT is regarded especially challenging [54]. Shorter development cycles allow companies to react more quickly to customers’ problems and needs. The update would not be released at once but incrementally with smaller updates, which would allow for increasing customers’ satisfaction with every little update. Products-as-a-service is another major success of the software industry considering, for example, Microsoft’s Office 365, which occupies a pioneering position within such business models. There is an obvious connection between IoT products and the software industry. Jeff Immelt, CEO of General Electric, therefore, maintains that every industrial company should become a software company. Still, the transformation will not be easily realizable and will take time. Hence, the transformation process will be a mere evolution rather than a revolution and traditional and IoT products will coexist for many years [52].

To realize this evolution on an organizational structure level, Porter and Heppelmann introduce new functional units to the more commonly used divisions such as finance, manufacturing, human resources, marketing, sales, service and support, IT and R&D (figure 2.3). The latter two will have to cooperate more closely and might have to merge eventually in order to guarantee the best possible IoT products, since these two divisions overlap considerably and are major components of the IoT. Overall, the often autonomously operating divisions that only communicate occasionally will have to integrate considerably to ensure better

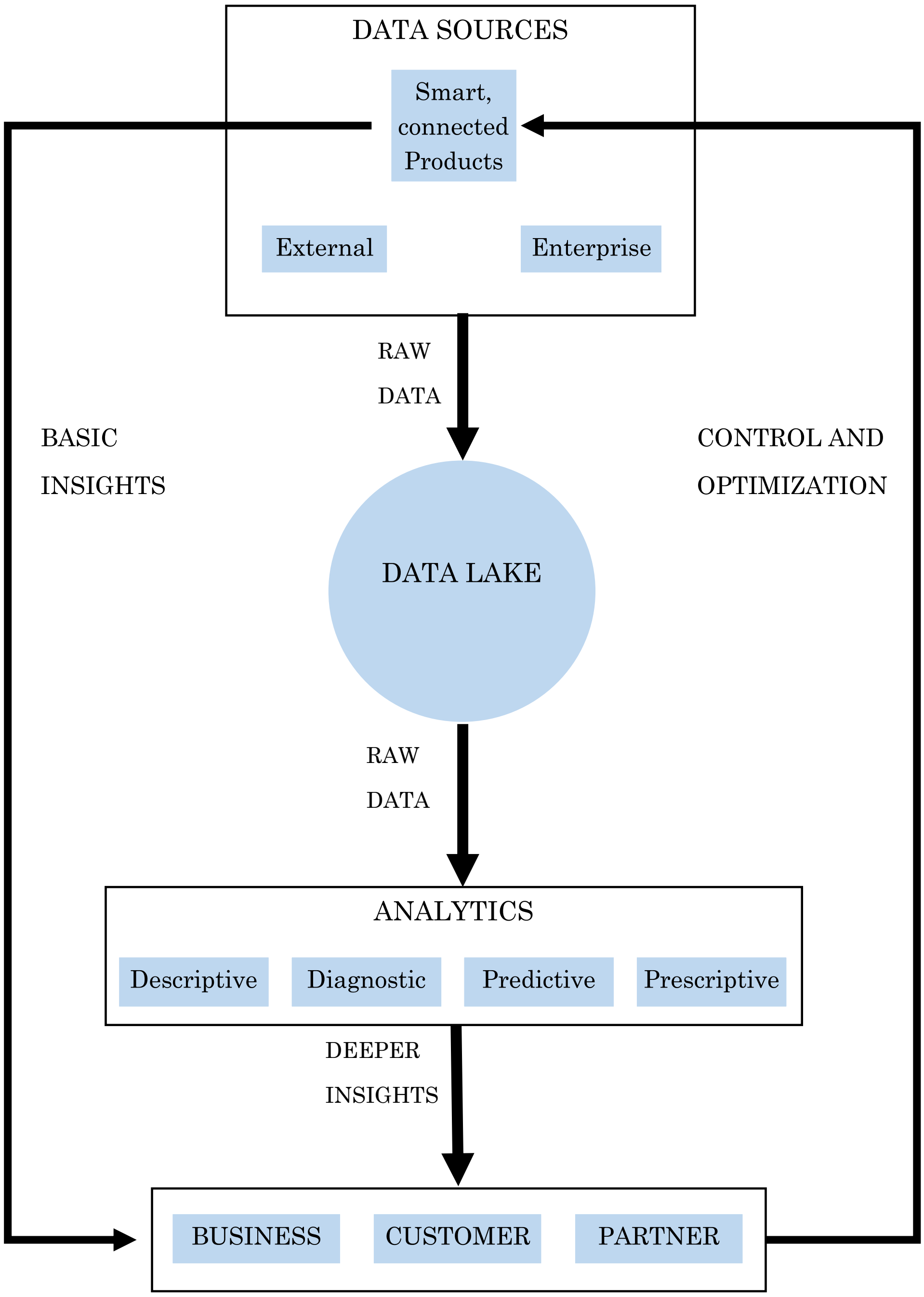


Figure 2.2: Data Lake according to [52]

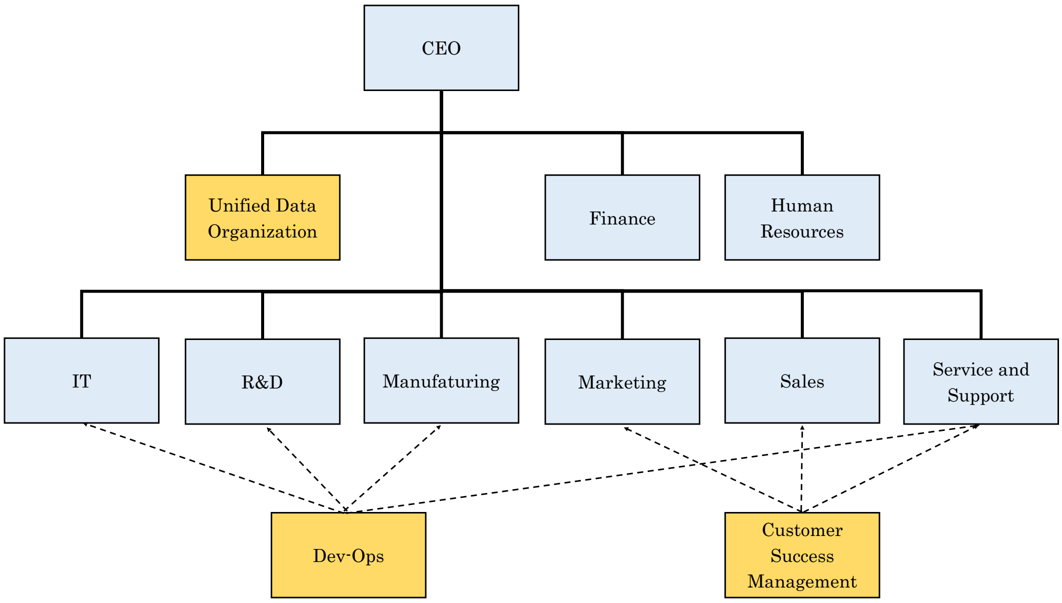


Figure 2.3: New Organizational Structure according to [52]

coordination [52]. An even more radical approach could be the boundaryless organization, which, however, has not been subject of previous research in the field of IoT [4]. The three new functional units include a unified data organization, dev-ops and customer success management. The unified data organization (already present in, e.g., the Ford Motor Company) is responsible for all the data occurring in a business and processes data for other divisions that have to work with it as well. Dev-ops is, as the word implies, composed of two groups: the dev, which comprises software engineers of ordinary, non-IoT products and the ops, personnel of manufacturing, IT and service in order to create shorter product-release cycles, new services and updates and patches for existing products. The customer success management unit is a third addition to old organizational structures. Their overall goal is to give customers the best possible product to generate recurring revenue that products-as-services have to offer [52]. Sharing knowledge (and with the IoT, data as well) in companies will be crucial to create a consistent product [67]. Knowledge networks could help overcome boundaries between functional units [28]. Inter-organizational work, for instance, to create products co-existing in a common application (e.g. smart homes) will be important as well. Saarikko et al. therefore examined the boundaries of relationships between companies [55].

2.4 Possible Topologies, remaining Problems, Standardization and a brief Outlook of the IoT

2.4.1 Possible Topologies with the IoT

With a view to IoT, three basic topologies can be identified: in a first step, the smart product connects to a single system with a product cloud. The next step and second topology is the so-called *system of systems* which involves multiple single systems that are able to interconnect through, e.g., ports and protocols. The mere task of one of such systems could be management of task executing systems. The third possible topology is constituted by single points in an IoT system that interconnect despite losing the connection afterwards since they, e.g., only needed a specific information but permanent data exchange is not necessary. This case is similar to human interaction that might also happen randomly, at a specific moment that is limited. This topology does not require a product cloud in the background but products would not only have to be smart and connected but also socially communicating. This step in the IoT will be described in subsection 2.4.4 Social IoT - a Future Outlook [52].

Thoughts on a centralized global IoT platform have been made in [61] as well. The architecture should therefore be much more data-centric instead of built around services to handle the large amount of incoming data properly for following generations of IoT products.

2.4.2 Remaining Challenges and Problems of the IoT

Although the IoT generally appears to be auspicious, there are still underlying problems and challenges that should be addressed. It is remarkably important to consider privacy and security issues of IoT since they are fundamental for achieving a foundation of trust between the new products that are connected and always exchange data, and users that initially might not feel comfortable if their refrigerator or coffee machine is connected to the Internet.

Lack of Standards

Standards of any kind (ports, gateways, protocols, platforms, etc.) help to connect things easier and are, especially in the case of IoT, necessary since connections are integral. A brief outlook on attempts to standardization of the IoT is given in subsection 2.4.3 Standardization and Eclipse IoT.

Other IT/Budget Priorities

If the IoT is not projected in an IT strategy or the general strategy of a business, mobilizing funds for testing IoT will be rather difficult. Businesses such as General Electric recognized its potential and partly aligned their strategy to a completely new direction successfully [31].

Name/Address Problems

The shift from 32-bit IPv4 to 128-bit IPv6 is imminent and has already initiated but IPv6 is not sufficient for all IoT devices, at least not at the moment. The larger space that IPv6 addresses offer is essential for the numerous new devices that will connect to the Internet but up to now IPv6 addresses are too large for small devices that do not possess the required power to transmit such addresses [33].

Miniaturization

Miniaturization of components was a key element that makes smart, connected products even possible and it will still continue to be challenging to push the IoT further. Smaller components like CPUs or sensors facilitate lower power consumption and temperature, which ultimately results in smaller units that can be attached to any product to make it connected and smart [33].

Large Amounts of Data and Analysis of Them

With a growing amount of devices the data transferred will increase as well. Analyzing them is exceedingly important for businesses and customers in order to have valuable information available as has already been discussed previously in this thesis. An estimated figure of 35 Zettabyte (ZB) of data by 2020 with a possible market of billions of devices render delivering important insights in device and user behavior substantially challenging for big data specialists [33].

Objects to Cloud and Scalability

Since devices might be very small, space for a large battery will be scarce compared with smartphones or larger, stationary products. Therefore, Bluetooth Low

Energy (BLE) is a more battery-friendly solution for connections which is used, e.g., in smartwatches or small fitness trackers than Wi-Fi or cell radio connections, as, e.g., in laptops. In this way, smartwatches or fitness trackers do not have to be continually connected to a smartphone since synchronization via BLE is feasible if need be to reduce the overall battery consumption. After the synchronization of data to a smartphone, the smartphone sends it through conventional ways, such as Wi-Fi or cell radio, to the cloud that processes and evaluates the data [69].

The downside of this approach is that all smartwatches and fitness trackers use a proprietary, application-specific layer to connect to a smartphone which results in numerous different ways to process data rather than unifying it. Zachariah et al. suggest a "general-purpose IoT gateway" that should solve this problem of IoT devices with a small battery capacity which establish a connection to the internet via BLE. They hope to implement a software service that in the end works better than the various application specific layers at the moment, in order to help this part of the IoT grow [69].

Security and Privacy Issues

Privacy and security issues of IoT appear to among the major concerns of people with a view to IoT. An IoT developer survey by Ian Skerrett and the IoT Eclipse project discerned similar results since the main worry for developers is the security aspect as well, especially if the deployment is planned for the following 6-18 months. The survey also finds that key concerns shift depending on the date of deployment. Interoperability is a concern in businesses which focus on Enterprise Software while connectivity is important if the focus is on Embedded Software [57].

The concerns about security and privacy are not unfounded as all the data produced by an individual can be clustered to generate a more or less comprehensive profile depending on how many IoT products are used and afterwards hacked. Especially hacks of health data collected through, e.g., new medical devices that could be straightly connected to a doctor, constitute infringement of privacy [51].

In 2009, Christian Mayer released an article on the security and privacy challenges, too. He categorizes the IoT into eight categories, for instance, sensors, devices, storage, etc., and graded their sensitivity between low, middle and high for the areas integrity, authenticity, confidentiality, privacy, availability and regulation. After an analysis, Mayer found that particularly the area of privacy lacks research and concluded that the IoT needs security and privacy mechanisms from the beginning unlike the Internet itself [46].

Jing et al. provide another in-depth analysis from a different point of view in a Springer-released paper from June 2014 [35]. The paper is about the security of the IoT and its perspectives and challenges. Jing et al. split the security of the IoT in three layers, i.e. perception layer, transportation layer and application layer. The transportation layer enables a connection, for instance, WiFi and 3G mobile networks as types of access networks. Core networks and local networks are other types that have been examined. For all these types of networks, the security issues as well as common issues that could affect any of these have been pointed out. The perception layer in this analysis consists of RFID technology and wireless sensor networks (WSNs) which were both checked regarding security flaws. For the application layer, the authors considered IoT specific applications such as intelligent transportation in the logistics industry or smart homes, as well as general problems of the application support layer such as services interruption due to DDoS attacks or security threats in cloud computing. Finally, Jing et al. found that security issues are more problematic in IoT networks than in traditional networks and enumerated open security issues of the IoT: overall security architecture for the entire IoT system, lightweight security solutions and efficient solutions for massive heterogeneous data [35].

Exploits of IoT devices and systems are similar to normal systems, they are vulnerable against attacks like DDoS, Botnets and data breaches, too. What makes it worse is that some objects were never designed to be on the Internet but will be in the future, and that an IoT environment has to cope with constraints, e.g., space and limited resources that already aggravate work for engineers. Another problem is that a system is only as strong as its weakest link, which means if an IoT device is easily exploitable, a hacker could already enter a system and access the remaining parts even if they would be more secure. Typical outcomes if an IoT devices is hacked would be that the device starts sending spam messages, the computing power of a group of hacked devices could be combined to coordinate an attack (e.g. DDoS) and it could be an entry point into a critical or corporate network which would otherwise be much harder to enter. Pierluigi Paganini even describes in [51] that smartwatches provide possibilities to enter smartphones to gather personal data of phones, or smart meters that offer potential to cause a blackout or possible fraud. To counteract all those vulnerabilities of the IoT, security and privacy have to be one of the priorities in the development to offer solutions that provide customers and businesses with the security needed. As noted in the previous section on Organizational Structure, this should ideally be a collective endeavor in individual businesses comprising all divisions and especially the businesses dealing with critical data, like IT, R&D and the data organization.

Weinberg et al. illuminate the IoT from a managerial point on convenience versus privacy and secrecy in [64].

2.4.3 Standardization and Eclipse IoT

Since the IoT represents a huge change within the IT world, standardization is fundamental in order to keep the effort to connect a huge amount of different devices manageable. The Internet Engineering Task Force (IETF) is of the same opinion and released their thoughts on that topic in 2013. Ishaq et al. give an overview about the ongoing research at IETF that first started with proprietary solutions but later focused on standardized protocols in the IPv6 Internet. However, this paper and the research is a start rather than the end of the standardization of the IoT as many challenges are still remaining, e.g., the security or scalability of the IoT [32].

Another interesting trend around the IoT is Eclipse IoT[1]. Eclipse IoT is an open source program that aims at facilitating the development of the IoT with the well-known Eclipse platform. The goal is to establish standards for key components in the IoT, namely protocols for communication between devices but also device-to-server, protocols to manage devices as well as standards for gateways and server interfaces. Tutorials, videos, presentations, server sandboxes and a growing community are available to start building an IoT system today. This project already attracts 125 developers from renowned companies like IBM, Deutsche Telekom or Intel since there is a certain urge to standardize the IoT [13].

Another Eclipse-based project for the IoT is Californium[2]. Californium as well as its five sub-projects aim at users who want to create their own IoT applications and offer a framework with implemented popular protocols[3].

2.4.4 Social IoT - a Future Outlook

Atzori, Iera, and Morabito establish an outlook of what might happen if more items are connected to the Internet in [6]. It is assumed that the IoT and social

[1] Website Eclipse IoT: http://iot.eclipse.org/ last access: 07.05.2016
[2] Website Californium: https://eclipse.org/californium/ last access: 07.05.2016
[3] Californium CoAP Framework: https://projects.eclipse.org/projects/technology.californium last access: 07.05.2016

networks approximate more rapidly than might be commonly estimated. In fact, research pursues automatically sharing sensor data to Facebook and other social networks or letting users compete by releasing their sensed sports data to provide for a comparability of performance. However, Atzori et al. divide the next steps of a possible evolution of the IoT as follows: smart objects (called "res sapiens") constitute the current state of the IoT, providing the advantages discussed earlier such as early detection of failures or interoperability across one or more systems; acting objects (res agens) should be aware of their environment and changes in that in order to react adequately to them, as well as be conscious about possible IoT "neighbors" to interact with; finally, social objects (res socialis) are supposed to create social networks between each other; not to be mistaken with social networks on which users might share their items or sensor data but real social networks created by the items themselves in which they communicate autarkical and offer human beings their services.

Nevertheless, they give a brief overlook of implementations which direct to the social networking aspect of the IoT but finally conclude that highly promising applications or business models for a social IoT have yet to be discovered although the technical requirements seem to be already available. Further open questions concerning social devices are the relationships between groups of objects, the architectural models they should be part of and, of course, the ongoing discussion of security and privacy issues pointed out in an earlier section [6].

Chapter 3

IoT Modeling Method - a New Approach

Accompanying to this master's thesis, a modeling method for the IoT has been developed. A main focus was to design a comprehensive modeling tool especially for business processes and IoT devices or products. The metamodeling platform used is ADOxx, i.e. a program to develop any modeling toolkit for the user's field of application[1]. An Ontology that classifies sensors into different classes with according specificities was generated with Protégé[2].

3.1 Introduction to Modeling

3.1.1 Motivation to Model

The characteristics of a model generally consist of three main components: *representation*, as it represents a thing or things, *abstraction*, as it focuses on important aspects and neglects insignificant features, and *pragmatics*, as it is created for a certain purpose [58]. Models per se have many purposes, for instance, to understand a situation and gaining a deeper knowledge of it, to reduce complexity, to document important aspects, or for predicting purposes to reconstruct reality and maybe eliminate a mistake that would have happened in real life without building a model first. Joshua Epstein's lists "sixteen reasons other than prediction to build models" in his article "Why model?" and provides with further insights why modeling is important [16]. While there are many reasons for models, their

[1]Website ADOxx: https://www.adoxx.org/live/home last access: 07.05.2016
[2]Website Protégé: http://protege.stanford.edu/ last access: 07.05.2016

representation is diversified as well. Possible representations range from textual, e.g. a XML document, or hierarchical to graphical; everything is possible. In the case of the IoT modeling method, the representation on the metamodel-level is hierarchical while graphical in the actual modeling environment where diagrams can be modeled. In the end, abstraction is always essential to modeling, whether to simplify complex issues or to generalize to provide a general concept through extracting common characteristics [42].

In the IoT context one purpose particularly stands out: the simulation of processes, and possibly the comparison of a process with or without involved sensors (meaning before IoT involvement and after). Modeling in advance could give insights whether or not the inclusion of the IoT in certain areas (such as automobile industry) or products of a company would change processes or manufacturing and how much could be improved (or if there is no improvement at all). Needless to say, this could not simulate the customers' or employees' real world use and how and if it would be appraised, yet it could show if a certain development is reasonable in the first place.

3.1.2 Requirements for an IoT Modeling Method

The requirements of the IoT modeling method have been outlined early on in the development stage. The goal was to unify different types of models into one comprehensive modeling method which is simple to use while documenting the key components of IoT systems. Although there will be similarities of included models to already existing ones, the IoT modeling method aims to offer a closed user experience that includes essential components without the modeler having to worry about choosing the right models.

The modeling of processes plays an integral part as they will most likely be affected by big changes induced by new insights provided by sensors and the possibility to automate (parts of) processes. Decisions could be made autonomously with the help of data at hand (even instantly). Therefore, simulation seems to be an adequate way to simulate real life scenarios to see how sensors would fit in an existing process. This sort of integration has yet not been performed which made it one of the goals of this modeling method. The extension of the basic BPMN 2.0 model with IoT tasks (which link directly to the sensors and components of an IoT Device Model) and the sensor data generator tackles exactly the integration of sensors and integrates their data to automatically calculate the distribution of decisions. BPMN Profiles would not have been sufficient for these operations as they are on the meta-level of this modeling method. As one can see, the process modeling is not only about documenting but also about simulating, which makes it so important.

UML would not be appropriate for IoT modeling as its operational purpose is too broad and not specific enough [49]. The IoT is still a sufficiently large application; hence, developing a modeling method specifically built for it is feasible. Specific features such as the deep integration of sensors make it a prerequisite for having more specific solutions at hand. SeMFIS, on the other hand, the foundation of this modeling method, is an excellent starting point since it has several important models already implemented. Moreover, the integration of semantic models, as well as ontologies, could allow for an even more automated execution of certain parts within the IoT modeling method, e.g. sensor data, due to its closeness to machine-readable input and output.

A modeling method for the IoT is a novelty even though there have been efforts to extend BMPN 2.0 for the IoT by a group at SAP [56].

Functional requirements of the IoT modeling method:

- Unified modeling method for different views of an IoT system
- Integration of sensors into BPMN 2.0
- Model to lay out an IoT product and its components
- Simulation of processes with IoT involvement

Non-functional requirements:

- Easy to become acquainted with it
- Ease of regular use (with a certain amount of modeling experience)
- Possibility to expand

3.2 Conceptual Models in Detail

In this section the models included in the IoT modeling method will be presented. These models offer the user a complete experience while challenging the needs of modeling IoT devices as well as their environment. Additionally, models which do not provide specific IoT content but extend this modeling method in a useful way, e.g., a working environment model for staff and their roles in a company, will be presented as well. The models introduced in the upcoming sections are mainly extensions of already existing models in SeMFIS or inspired by models from other modeling methods but created for IoT purposes. Before the actual implementation in ADOxx, the concepts were constructed on paper and afterwards transferred to a class diagram. For an overview of, as well as a basis for the implementation, the class diagram in figure 3.1 shows the (important) parts of the metamodel of this modeling method.

Models present in the IoT modeling method:

- IoT-Extended BPMN 2.0 Model *
- IoT Device Model +
- IoT-Extended Architecture Model +
- Company Map *
- Working Environment Model -
- Document Model -
- Semantic Annotation Model -
- Ontology Model -

(*...altered for the IoT, +...created new, -...from SeMFIS)

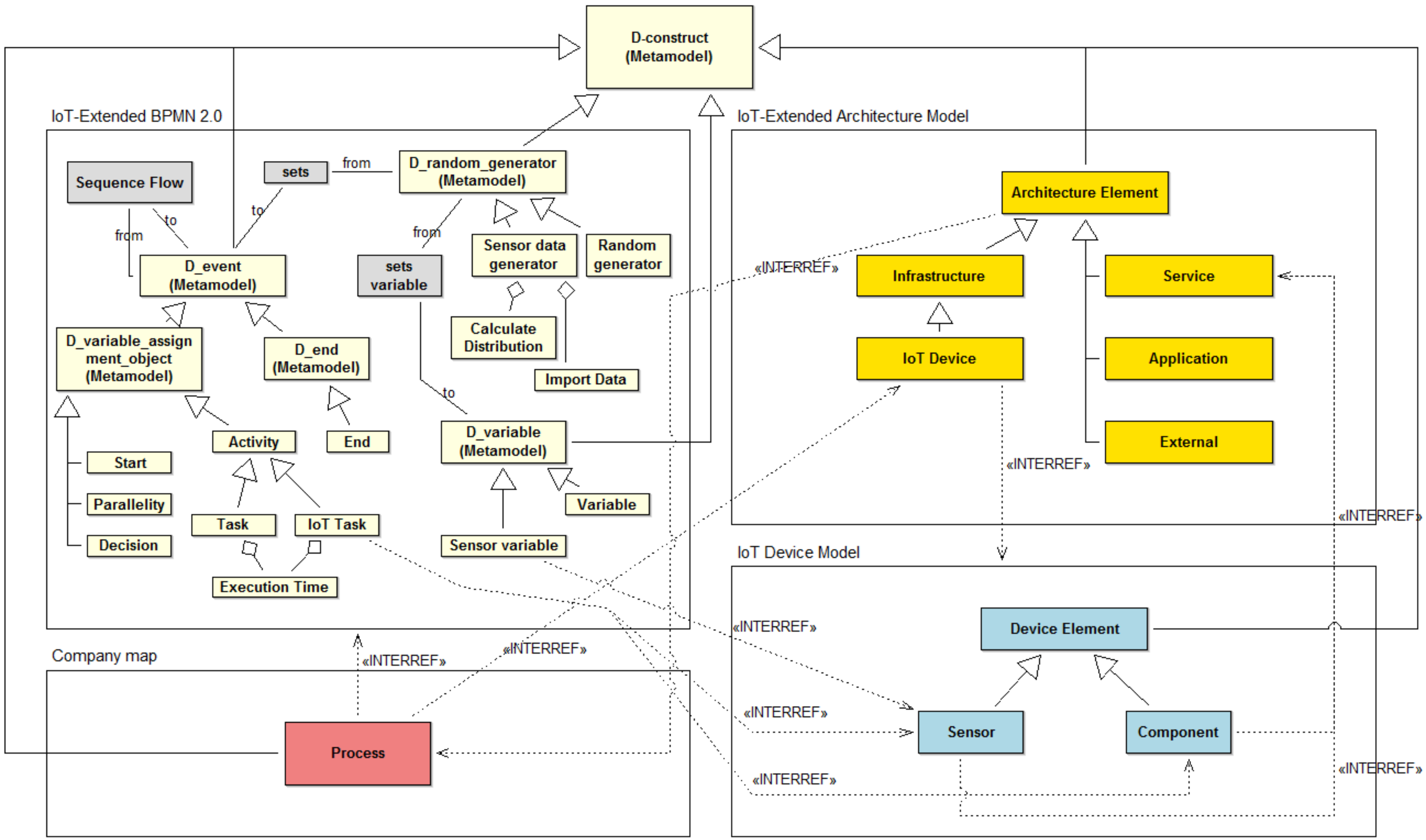


Figure 3.1: Metamodel of the IoT modeling method

3.2.1 IoT-Extended BPMN 2.0 Model

The standard BPMN 2.0 Model has been extended for this modeling method in order to support IoT devices and especially bring sensors in the context [48]. SAP and Sonja Meyer et al. also propose an extension of BPMN in their work even though in a different way than in the modeling method at hand [56]. The extended BPMN Model in this thesis was specifically designed to work with the models introduced in the upcoming sections of this chapter and therefore might not be applicable in other contexts. Three integral elements have been added: IoT Task, Sensor Data Generator and Sensor Variable.

The *IoT Task* is a direct copy of a standard *Task* and inherits the same attributes from the superclass Activity and, thus, having identical options when either task is created in the modeling method. However, they differ in small details, as an IoT Task can link to a specific sensor of the IoT Device Model characterized in the next section. Sensors as well as components of the IoT Device Model, such as parts for a wireless connection or flash storage, can be linked to these IoT activities. The appearance of the element IoT Task has also been changed to let the modeler know immediately that this modeled task is unlike a standard Task and will be executed automatically.

The elements *Sensor Data Generator* and *Sensor Variable* follow a similar concept and feature the same attributes as their regular variants *(Random generator* and *Variable)* but with changed looks and extended functions.

An overview of the most important classes of the IoT-Extended BPMN 2.0 Model is provided in table 3.1.

3.2.2 IoT Device Model

The IoT Device Model tries to combine one or more intelligent devices and their individual parts in a simple model. Decoupling of these IoT parts instead of adding more information to the BPMN model was chosen in order not to unnecessarily overload the latter. Attributes that are not present yet but might appear to be useful for users can be easily added in ADOxx retrospectively for the purpose making the method responsive to special needs. Considering [53] a smart, connected product has been categorized into three elements for this model:

- Device
- Sensor

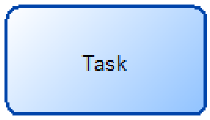
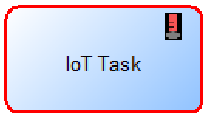
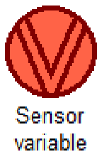
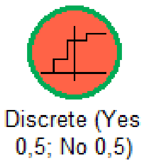
IoT-Extended BPMN 2.0 Model		
Task		Regular task as known from BPMN models; is an activity that has to be performed
IoT Task		Differs from a task as sensors will be involved; automatically or semi-automatically
Gateways		Various types of gateways to show how a process flows; types: exclusive, non-exclusive, non-exclusive (converging)
Sensor Variable		Derived from a regular BPMN variable to show that a sensor will be involved in the calculation
Sensor Data Generator		Derived from a regular random generator; reads a .CSV-file to compute the distribution from the incoming variables

Table 3.1: Elements of the IoT-Extended BPMN 2.0 Model

- Component

The *Sensor* element offers various specifications for sensors used in the real device. Sensor class (e.g. acoustic, electrical, proximity, etc.), size of a sensor, the measurement precision of it as well as its operating voltage and working temperature range can be defined individually. These specifications are in accordance with the ontology created for this project in order to have the possibility to connect them in hindsight. The import of a predefined ontology into ADOxx is done with this modeling methods' Ontology Model.

The element *Component* describes essential components that make a product smart and connected. The user, therefore, has to choose between SoC (system on chip), RAM, Flash Memory and Misc (for other parts that can be further characterized in the attribute "Description") in the notebook. The elements of the IoT Device Model can be linked, on the one hand, to the Working Environment Model for comprehending who is responsible for the different items and, on the other hand, to the IoT-Extended Architecture Model to indicate the services that

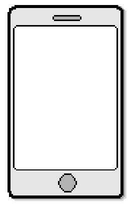
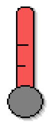
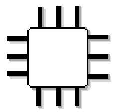
IoT Device Model		
Device	 Device	Represents a single, intelligent product; can consist of more devices (alternative representation available)
Sensor	 Sensor	A sensor which can be specified accordingly in the Notebook
Component	 Component	Can be either a SoC, RAM, flash memory or misc.; needed to make a product smart and enable connections

Table 3.2: Elements of the IoT Device Model

use IoT components or sensors. Since a product can very well be part of a bigger system of systems as referred to in section 2.4.1 Possible Topologies with the IoT, products in this model can be linked to other products to enable exactly this type of scenario. Additionally, *Components* and *Sensors* provide the opportunity to specify their costs. The elements' appearance (without the abstract symbol for *Device*) can be seen in table 3.2.

3.2.3 IoT-Extended Architecture Model

The idea of an architecture model is not new and can be found in other modeling methods as well (e.g. in [37]). However, the architecture model in the modeling method presented in this thesis extends the regular model. Based on the IT system model found in ADONIS, this architecture model adds two necessary elements to the already existing three (see table 3.3):

- Infrastructure
- Service
- Application
- IoT Device

- External

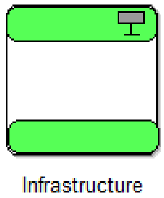
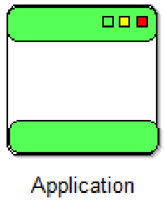
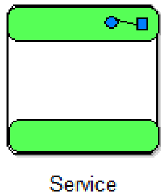
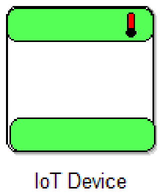
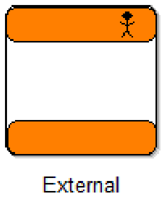
IoT-Extended Architecture Model		
Infrastructure		To model the underlying infrastructure (e.g. a business server) with responsible roles and connected processes
Application		Applications running on a previously modeled infrastructure item
Service		A service is the top object offering something for the outside world
IoT Device		A newly added item to an Architecture model to represent IoT devices in the complex IT environments
External		Another additional component for the ever increasing outsourcing of various infrastructure parts

Table 3.3: Elements of the IoT-Extended Architecture Model

The latter two have been added for this modeling method. *IoT Device* elements are linked to "IoT Device Models" introduced in the previous chapter (this was implemented with INTERREFs, a concept of ADOxx described in section 3.4). Each *IoT Device* element in this model represents exactly one IoT Device Model. The second new element, *External*, provides the opportunity to include external partners involved in the infrastructure. As outsourcing parts of the IT infrastructure became more common in the past years, it is present in this modeling method as well. Especially, cloud-based services from Microsoft or Amazon appear to be attractive for smaller businesses since they offer a flexible infrastructure that can be easily scaled if need be but the business itself does not require own servers in-house.

To show the interplay between the individual elements and how they are connected with each other, different types of arrows have been added in the ADOxx implementation. Usually an *Application* “has” one or more *Infrastructure* elements and it “is dependent on” one or more *Services*. Additionally “has External” relationships between any component present in this model (besides external itself) and an *External* element mark that these components are indeed executed by an external partner which can be further defined in the actual modeling method. The notebooks in the implemented five elements of this model all offer the option to link the modeled components to two further model types, one being the attribute “responsible Role” to connect it to the working environment model and one of its roles and the other being “referenced Process” to link it to one specific process of the Company Map model.

3.2.4 Company Map

The Company Map is a model to project the existing processes that have been modeled and connect them to the previously modeled processes in the BPMN Model. Therefore, the element *Process* denotes all the necessary functionality (table 3.4). This element has been extended with an additional attribute which links a process element to an existing IoT Device in the IoT-Extended Architecture Model.

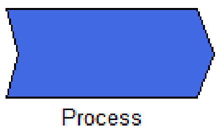
Company Map		
Process		Processes reference Tasks and IoT Tasks of the BPMN 2.0 model; this map provides an overview of a company’s processes

Table 3.4: Element of the Company Map

3.2.5 Working Environment Model

The Working Environment Model provides ample opportunities to specify a company’s workforce. Since there are no special changes necessary to enable the modeling of the recommended new divisions in businesses that deal with IoT products

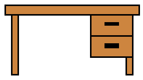
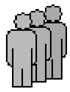
Working Environment Model		
Organizational Unit		Organizational Units group Performers in their divisions
Performer		A Performer is a single person; information about a person and their work can be entered in the notebook
Role		Roles can be adopted by performers
External partner		External Partners are modeled separately in this model

Table 3.5: Elements of the Working Environment Model

(according to [52]), the already implemented elements from SeMFIS can be reused without being altered. Four essential elements to model the organizational structure are available as can be seen in table 3.5.

Organizational Units or divisions are usually staffed with one or many *Performers* which represent a single employee. *Performers* can then take different *Roles* to model staff members who are responsible for more than a single task. Vice versa, a *Role* can be linked to more than one *Performer* if, e.g., a *Role* is generally employed as an accounting clerk of whom certainly more than one can be employed in a big firm. *External Partners* can be modeled separately in this model to have a brief overview of the involved parties. In the *Performers'* notebooks personnel costs and other things can be simulated, ADOxx itself allows for querying this model and its attributes for further analytical insights (e.g. list all workers who earn hourly wages less than x).

3.2.6 Semantic Annotation Model

The Semantic Annotation Model from the SeMFIS modeling method was extended to enable Model References for all the new elements introduced with the IoT modeling method from previous sections. It consists of four elements (see table

3.6):

- Model Reference
- Connector Reference
- Ontology Reference
- Annotator

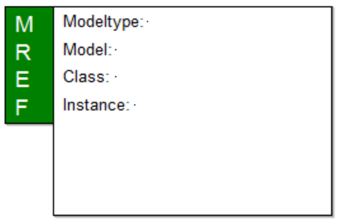
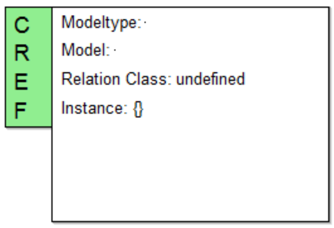
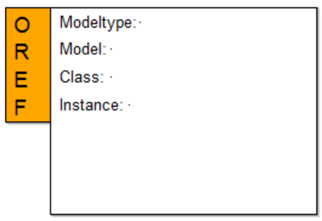
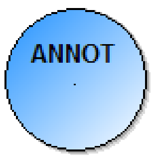
Semantic Annotation Model		
Model Reference		Model Reference references an instance of any model
Connector Reference		Connector Reference points to a model as well
Ontology Reference		Ontology Reference has an INTERREF to one modeled item of the Ontology Model
Annotator		The Annotator connects either a Model Reference or a Connector Reference and a Ontology Reference; it offers various annotation types to further specify the connection

Table 3.6: Elements of the Semantic Annotation Model

Its purpose is connecting already modeled items (*Task, Sensor, Sensor Data Generator, Decision Gateways, etc.*) to imported items of an Ontology by means of an *Annotator*. *Model Reference* or *Connector Reference* are linked (via INTERREFs in the implementation) to the model or connector of interest (e.g. one specific *IoT Task*), an imported or modeled item of any Ontology Model is linked to an *Ontology Reference* in the same manner. To connect a *Model* or *Connector Reference* to an *Ontology Reference* an *Annotator* has to be created. The relation "is Input for" then links a *Model Reference* to an *Annotator* and the relation "Refers to" points from an *Annotator* to an *Ontology Reference*. With the attribute "Annotation type" of an *Annotator* element a Semantic Annotation Parameter can be picked from a predefined list (e.g. Is equal to, Is instance of,...).

3.2.7 Ontology Model

The Ontology Model includes all the elements ontologies themselves comprise but implements them as items to model them on a canvas. Table 3.7 provides a brief overview of them.

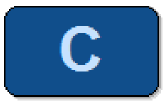
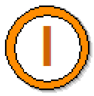
Ontology Model		
Namespace	Namespace	All of these classes represent their equivalents from Ontologies; with the import functionality adopted from SeMFIS an existing Ontology can be imported into this modeling method including all classes, properties and instances created with the Protégé Ontology editor
Class	 Class	
Property	 Property	
Instance	 Instance	
Predicate		
AllDifferent	 AllDifferent	

Table 3.7: Elements of the Ontology Model

These elements cannot only be created and altered in ADOxx, there is also a plugin for Protégé available which exports an ontology as an .XML file. This file can later be imported into ADOxx and automatically adds all the *Classes, Properties and Instances* created with Protégé as an Ontology Model in the modeling method. All the elements of this model offer further possibilities to specify them through their attributes to bring them in line with their counterparts from the ontology.

OWL instead of Frames was chosen since the ontology editor of choice was Protégé which supports exactly this standard and the compatibility with the plugin to export an ontology and import it in ADOxx had to be given [63]. However, SeMFIS supports both OWL and Frames.

The corresponding ontology for the IoT itself will be presented in the section 3.6 Ontology.

3.3 Scope of Modeling Method

The objective in mind of this modeling method was to offer a comprehensive and complete modeling program with regard to the IoT, especially considering IoT devices and business processes. This field of action has not yet been sufficiently explored and adapted into modeling methods although the IoT has a promising forecast as already discussed parts of this thesis concerned with theoretical issues. The only other project concerning modeling with the IoT in mind is “Internet of Things Architecture”, co-founded by the European Commission and developed by Sonja Meyer et al. from SAP. This group describes the modeling of IoT devices as resources in the widely used business process language BPMN 2.0 after stating that the representation of key components like sensors is basically not existent in the current modeling process. Their solution to this problem was extending BPMN with new types of activities and roles [56].

Since business processes represented by BPMN models are only one constituent of an IoT architecture and all the activities involved, the modeling method introduced in this thesis tries to capture other parts as well in regard to an enclosed experience of displaying all the fundamental models. However, an extended BPMN 2.0 model is still one of the core models. Other models offered are an IoT Device Model comprising one or more *Devices* and their *Sensors* and *Components*, an Infrastructure Model not only offering standard components like *Service*, *Application* and *Infrastructure* but also *External* for outside partners as well as *IoT device.* As known from the ADOxx platform, these models are linked at their points of intersection. For example, the extended BPMN model has an INTERREF called "referenced Sensor" which directly points to one specific sensor of the device model; another example is the created link from an *IoT Device* as a part of the IoT-Extended Architecture Model to the actual IoT Device Model where this product can be further specified.

3.4 Technical Implementation - Metamodeling and ADOxx

The modeling method for the IoT was developed with ADOxx, a relatively easy to handle, yet powerful metamodeling platform provided by the BOC Group Vienna. For further insights in metamodeling and ADOxx [22] by Fill and Karagiannis is an excellent starting point to understand essential issues such as the components of modeling methods and metamodeling in ADOxx. A concept of metamodeling itself was established by Karagiannis and Kühn in a paper on metamodeling platforms in 2002. According to their theory, a modeling method is composed of a modeling technique and mechanisms & algorithms (figure 3.2). The modeling technique is further split in two parts: modeling language and modeling procedure. Syntax, semantic and notation define a modeling language which unites elements that describe a model. The modeling procedure on the other hand involves all the steps needed for creating a model. Finally, the mechanism part of a modeling method exists to use and evaluate the created models. In the case of ADOxx, this part offers extensibility for vast implementation possibilities, which was used for import features of the IoT modeling method introduced in the following section. To create such a metamodel, a different modeling language has to be used, i.e. the metamodeling language, which complies with a defining meta-metamodel (figure 3.3). ADOxx fully relies on the concept of Karagiannis and Kühn [39].

Although mainly focusing on a definition for a domain-specific language for modeling methods, Visic et al. give insights into agile modeling method engineering, a method to let modeling methods "evolve iteratively based on changing modeling requirements and feedback loops", in other words, as if running through the phases of a waterfall model as seen in software development [62]. A similar approach was also taken for this modeling method: the basic types of models have been established on paper before they were implemented with ADOxx and gained a deeper range of functions. They were further refined after having used the first implementation of the modeling method.

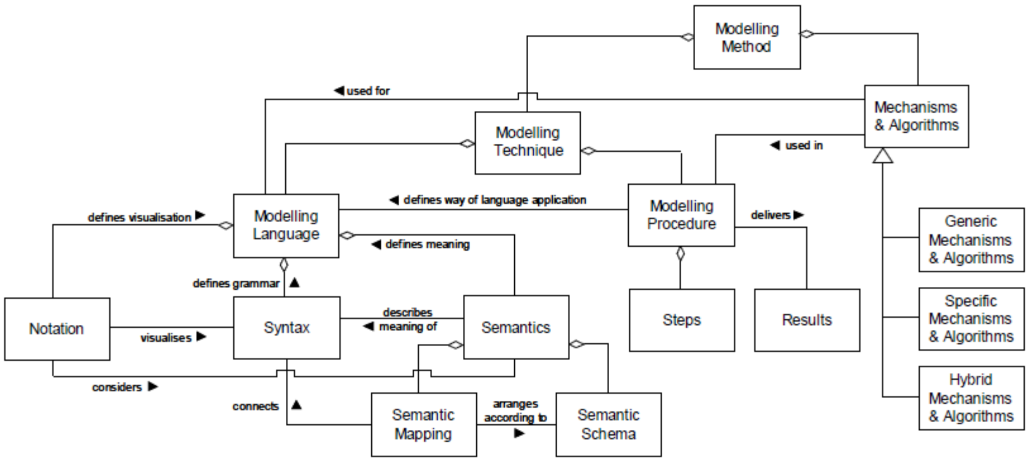


Figure 3.2: Components of modeling methods according to [39]

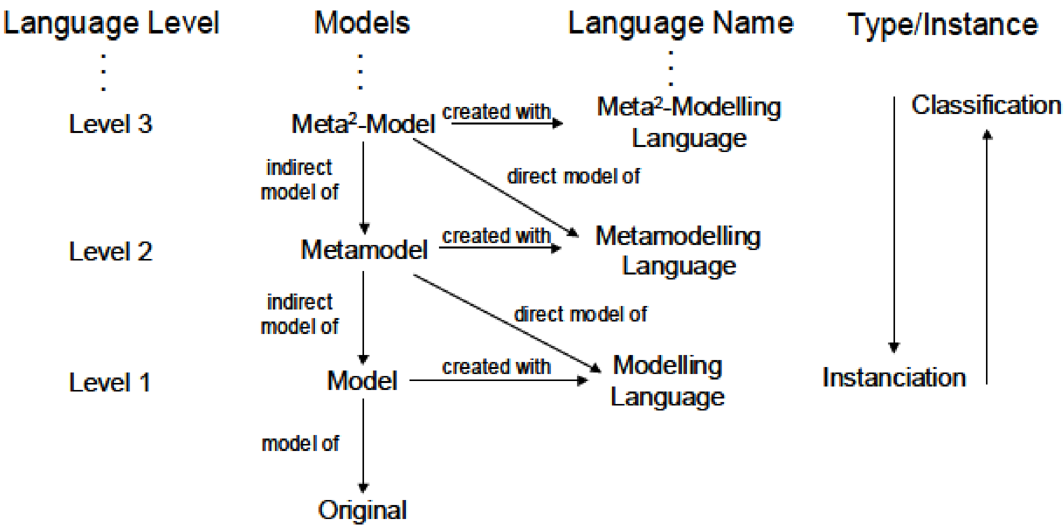


Figure 3.3: Metamodeling based on language levels according to [39]

The base for the IoT modeling method is formed by SeMFIS, a tool for managing semantic conceptual models, originally developed by Dr. Hans-Georg Fill[3,4,5] [21, 20]. Further readings regarding the conceptualization of semantic models and the application of SeMFIS can be found in [18, 17, 19, 23]. Two main reasons were decisive for the choice of SeMFIS: SeMFIS already has BPMN 2.0 models fully integrated, which were necessary to represent business processes accordingly. The second factor was the integration of semantic models and ontologies as well as the connection to the ontology editor Protégé. This connection between Protégé and SeMFIS/ADOxx is useful as it allows users to build ontologies with Protégé that can later be imported into the modeling method in which all the used classes, attributes and instances are displayed. SeMFIS does not only have BPMN, Ontology and Semantic models already integrated, it also provides Class diagrams for programming problems, a Company map that facilitates combining all the processes of a business in one model, and a Working Environment model to enumerate organizational units, workers as well as their different roles. Furthermore, ADOxx has the ability to query models and simulate business processes. These functions in ADOxx are particularly powerful, giving users the possibility to, e.g., count specific components of models or query even deeper and only release objects with specific fulfilled attributes. Simulation is another feature of ADOxx and executes, for instance, path analyses over a defined number of runs to check which branches were taken in case of decisions in business processes. Another type of simulation analysis are capacity analyses that display which roles or workers are needed to, e.g., generate a predefined quantity of goods. The underlying theory with graphs that transform a flow in a model to make it executable and enable analyses can be found in [38].

ADOxx actually consists of two separate programs that meet entirely different purposes: the Development Toolkit is an application to implement a modeling method like SeMFIS or the IoT modeling method introduced in this thesis. The Modelling Toolkit, on the other hand, enables the modeling part of an implemented method.

Modeling with the ADOxx Modelling Toolkit functions accordingly (figure 3.4): the elements of choice are dragged to the canvas and linked with relation arrows to either generate a flow, as seen in BPMN models, or to reveal dependencies be-

[3]Website SeMFIS: http://www.omilab.org/web/semfis last access: 07.05.2016
[4]SeMFIS Tutorials: http://www.omilab.org/web/semfis/tutorials last access: 07.07.2016
[5]SeMFIS Case Study: http://homepage.dke.univie.ac.at/fill/semfis/SeMFIS_Case_Study.pdf last access: 07.07.2016

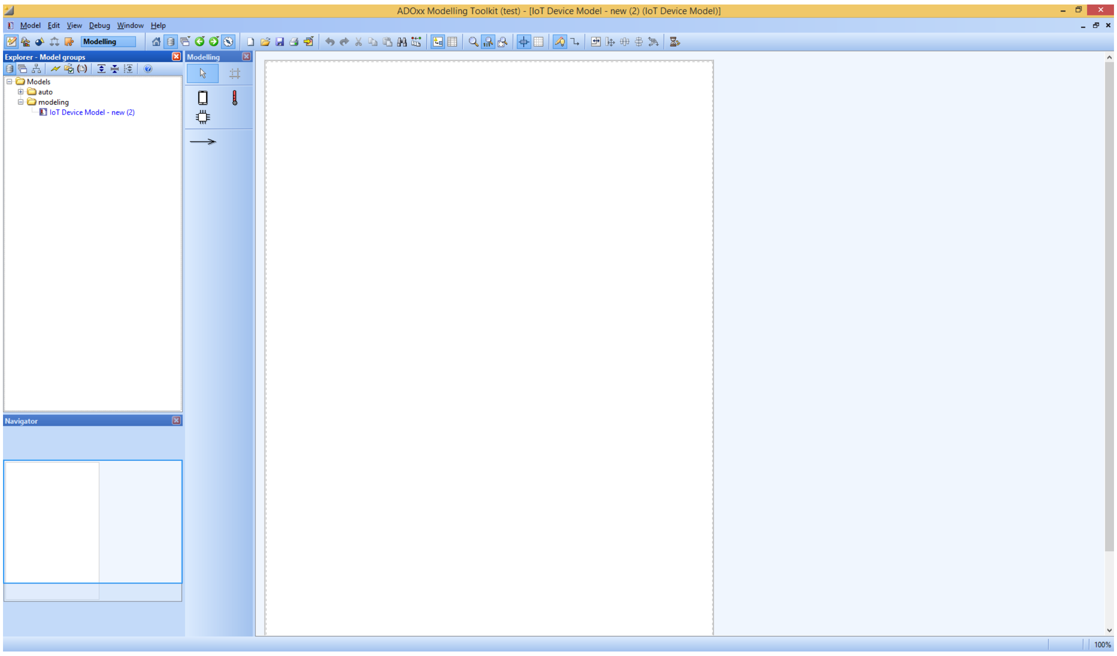


Figure 3.4: Canvas in metamodeling program ADOxx

tween certain elements. A double click on an element opens the so-called notebook, ADOxx's concept to concentrate all the defined attributes of a class in a clearly laid out factor that enables a user to describe the modeled elements accordingly. Tabs can group attributes, which share some features for easier organization of those. The underlying ADOxx Development Toolkit is where classes and their attributes are defined before they can be modeled in the Modelling Toolkit. The functionality of ADOxx can be expanded even further with ADOScript, a corresponding scripting language which was needed for the import of .CSV-files and the calculation of distributions described in the following section. Another concept in ADOxx is INTERREF which allows references from one element of a model to another element or a complete model. They have to be implemented on the metamodeling level in the ADOxx Development Toolkit.

3.5 Extensions with ADOScript

To get an advantage of the IoT to the level of modeling methods the IoT-Extended BPMN 2.0 Model introduced before has been extended with a method to automatically calculate and write the distribution of sensor readings into the designated attribute of the *Sensor Data Generator*, the central element for distributions that decides whether the execution of a process is following either path one or path two (or path three or path four, etc.). To achieve this, a .CSV-File with sensor readings has to be formatted as followed to guarantee a correct calculation: The .CSV-File should only have two columns filled with data as the script that reads in the data and writes it into the record table as well as the record table itself only provide two columns (see example table 3.8). It is suggested to fill the first column with the dates of the sensor readings while the second column *has* to contain the sensor readings the calculation should be based on. It is crucial to have the sensor readings in the second column as the underlying ADOScript file that is responsible for the computing only takes this column into account to calculate the distribution. However, the first column is not needed for calculation purposes and gets imported as-is which gives the user the possibility to fill it with other data instead of dates of sensor readings, if, e.g., dates are not relevant in a case. The very first line of the .CSV-File is not imported either. Any labeling will, therefore, not be adopted in this case, only data beginning with line two.

The import of the data and the following calculation has been split into two Scripts which results in two basic steps to use the *Sensor Data Generator's* calculation functionality:

1. Import .CSV-File by setting its path and clicking on "Import Data"

2. Calculate distribution by clicking on either:

   - Calculate Distribution (1 Critical Value)

     or

   - Calculate Distribution (2 Critical Values)

An UML Activity diagram in figure 3.5 shows the sequences of import and calculation in greater detail.

After having imported the data, the actual calculation has been split in two different calls, the first offering the comparison of sensor readings with one critical value, the second with two critical values which results in two (or three) paths if a *Sensor Data Generator* is connected with an *Exclusive Gateway.* By clicking on one of these two buttons, a dialog opens that requires to enter one (or two) critical value, which the ADOScript file then compares by means of the sensor readings imported into the record table. This results in a discrete distribution of the sensor readings, e.g., *Discrete (Critical Value 1 0.7; Other 0.3)*, which in this case would mean that 70% of the sensor readings are bigger than the entered critical value.

Date	**Data**
01.06.2016	2.6
02.06.2016	1.4
03.06.2016	3.0
04.06.2016	3.1
05.06.2016	0.5
06.06.2016	2.7
07.06.2016	3.1
08.06.2016	2.9
09.06.2016	2.0
10.06.2016	3.2

Table 3.8: Example for Sensor readings

Once the calculation is done, the outgoing arrows of an *Exclusive Gateway* have to be denominated accordingly to assure the correct execution when a business process is simulated with ADOxx.

The voltage of sensors, at least the ones available for the Raspberry Pi environment, usually reads between 0.0V and 3.3V. If a process, for instance, involves a temperature sensor that sometimes hits a critical value (say 2.5V), which would result in two different paths this process could take, in addition, a number of readings of this sensor are available, this method could automatically decide which path the process takes only with the available data and the input of this critical value. Also, the input data does not have to be necessarily in Volt, e.g., kilogram for a weight sensor or basic 1/0 (true/false) work as well.

Extracts of the ADOScript code can be found in Appendix B.

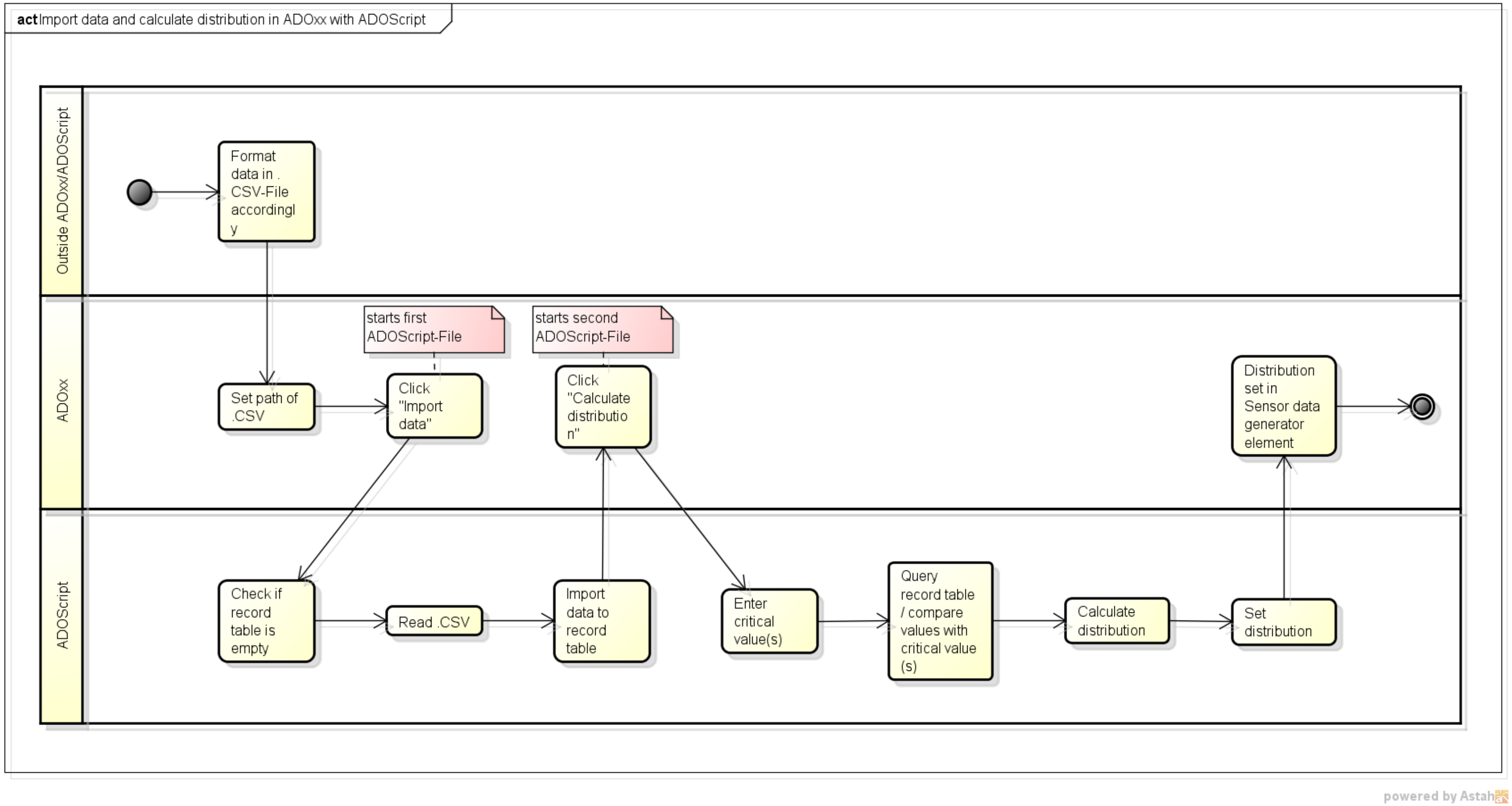


Figure 3.5: UML Activity Diagram of the Import and Distribution Calculation in ADOxx with ADOScript

3.6 Ontology

The ontology developed for this master's thesis groups sensors in their respective classes (e.g. automotive, chemical, electrical, etc.) and facilitates itemizing sensor specifications as well as what they measure. The ontology editor of choice was Protégé as SeMFIS offers the already mentioned import of ontologies created with the editor to automatically load the generated ontology into an ADOxx model. Three subclasses from the standard ontology superclass *Thing* have been created:

- Sensor_Classes
- Sensor_Specifications
- Types_of_Measure

These classes offer various subclasses to specify one or more sensors, their properties and what they output with their readings. The Sensor_Class's subclasses group sensors according to their types (e.g. navigation) and the subclasses of these types lead to the sensors (e.g. gyroscope). The classification produces a list of types and their specific sensors, see table 3.10. For the class diagram of this ontology see figure 3.6.

A comprehensive list of sensors can be found under [66] that was adapted for this ontology and its sole purpose: to fit the IoT. In order not to only classify sensors in their various groups, this ontology also seeks to specify them as well as their incoming or outgoing data. Further specifications are implemented as a subclass of Sensor_Specifications. To go into detail about sensor readings, subclasses of Types_of_Measure have been created. Specifications comprise sensors' measurement precision, their operating voltage, size, temperature range and measuring tolerance. The specifications are not kept overly minutely (for instance, the size of a sensor can be marked as large, medium or small) but will help to get an idea of how a sensor used in a product would roughly look like. Since all the different types of sensors measure different qualities, the subclasses of Type_of_Measure group readings in their types including subclasses which further specify what they output. See table 3.9 for a classification of measurements.

To connect the classes in an instance, the properties *measure* and *specify* have been implemented. This enables constructs to link a specific sensor with the specifications it has and the readings it measures.

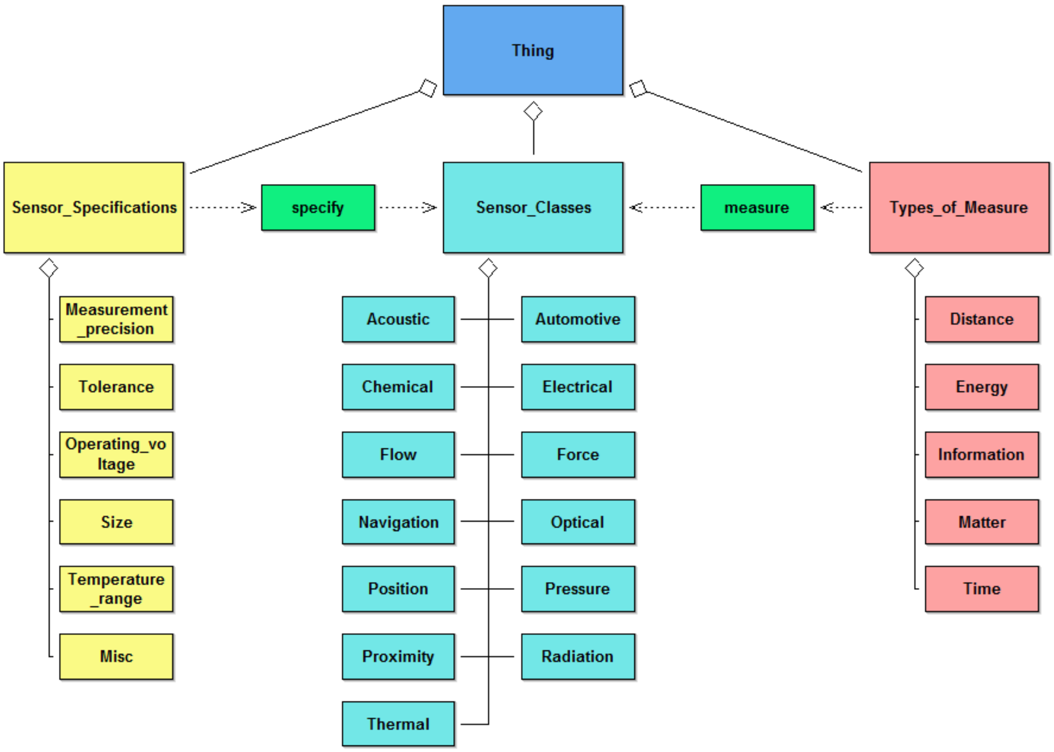


Figure 3.6: Class Diagram of the developed IoT Onotlogy

Types of Measure				
Distance	km/h	meters		
Energy	Celsius	Electrical Milliamp	Ohm	Volt
Information				
Matter	Gas Air	Liquid Oil	Water	Solid Soil
Time				

Table 3.9: Types of Measure for Sensors

Type	**Sensors**				
Acoustic	Geophone	Hydrophone	Microphone		
Automotive	Air-fuel ration meter	Blind spot monitor	Curb feeler	Parking sensor	Speed sensor
	Speedometer	Tire-pressure sensor	Torque sensor		
Chemical	Carbon dioxide sensor	Carbon monoxide sensor	Hydrogen sensor	Nitrogen oxide sensor	Oxygen sensor
	Smoke detector				
Electrical	Current sensor	Hall effect sensor	Magnetometer	Metal detector	Voltage detector
Flow	Anemometer	Mass flow sensor	Water meter		
Force	Hydrometer	Level sensor	Load cell		
Navigation	Air speed indicator	Altimeter	Depth gauge	Gyroscope	Variometer
Optical	CMOS sensor	Electro-optical sensor	Infra-red sensor		
Position	Gravimeter	Impact sensor	Shock detector	Tilt sensor	
Pressure	Barograph	Barometer	Piezometer	Pressure sensor	Tactile sensor
Proximity	Alarm sensor	Doppler radar	Motion detector	Proximity sensor	
Radiation	Geiger counter	Neutron detection			
Thermal	Pyrometer	Thermometer			

Table 3.10: List of Sensors present in Ontology

Chapter 4

Evaluation of Modeling Method

The models created for this evaluation embrace (business) processes, IoT device models and architecture models among others. Not only reproducing real-life aspects on the canvas but also the correct functionality of the calculations of the *Sensor Data Generator* as well as the simulation of the IoT-Extended BPMN 2.0 models that comes with ADOxx were important. An overview of all classes and their relations can be found in figure 4.1.

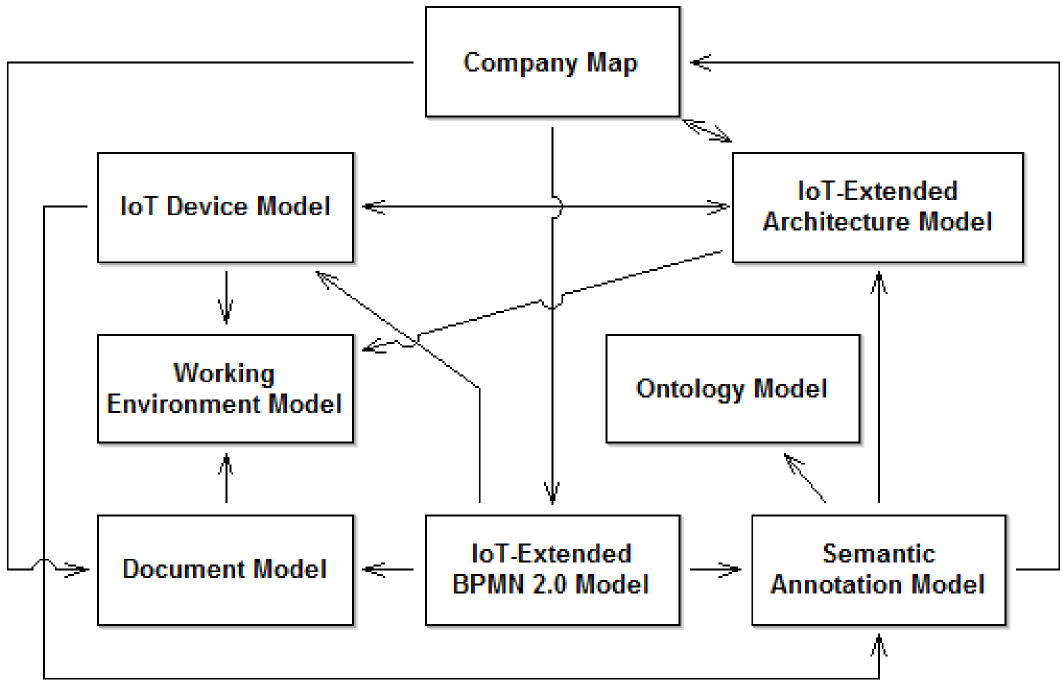


Figure 4.1: All available classes of the IoT Modeling Method and their Relations

4.1 Use Case Scenario

To evaluate the implemented IoT Modeling Method - a New Approach and the metamodels introduced in chapter 3.2 and later implemented in ADOxx, a fictitious scenario has been created in which the method was applied on models. In such an scenario, a modern car sharing company possesses a fleet of available smart cars, which are rented by customers by means of smartphone. In the following section, 4.2, the core business process of this company is shown.

4.2 Process View

The core of this evaluation is the process *Car rental*, an IoT-Extended BPMN 2.0 model that covers all the necessary steps, from the first click of a customer in an app on his or her smartphone, to having a look at the available cars, until the customer receives a receipt after the ride. This process includes both IoT tasks and regular ones but since they both inherit basic functions from their superclass, they cooperate and simulations such as path analyses operate appropriately. However, the majority of the tasks are IoT tasks since the process heavily relies on sensors and in general a smart, connected product.

The first decision in the process *Car rental* follows immediately after a car has been reserved by the user as the maximum time a car is reserved is 30 minutes (see figure 4.2). If the customer takes more than 30 minutes to reach the car, the process ends and the customer would have to restart the process. If he or she takes less time, the process continues. The distribution of the decision has been calculated with a *Sensor Data Generator* element and dummy data (a .CSV-file with waiting times in minutes) as presented in section 3.5. As the critical value was 30 minutes in this case, the ADOScript counted the frequency of the waiting time over or below the limit to finally calculate a discrete distribution and, ultimately, how often the path follows the direction which continues the process and how often the process ends. Another decision in the process has to be made when the customer enters the code that is displayed in the cars' front windshield. A .CSV-file which only contained true or false (1 for a correct entered code and 0 for false) dummy data was imported to calculate the distribution of this gateway (figure 4.3).

After a correct code has been verified, the doors unlock and the customer can enter the car. Before the actual rental starts, a non-exclusive gateway splits

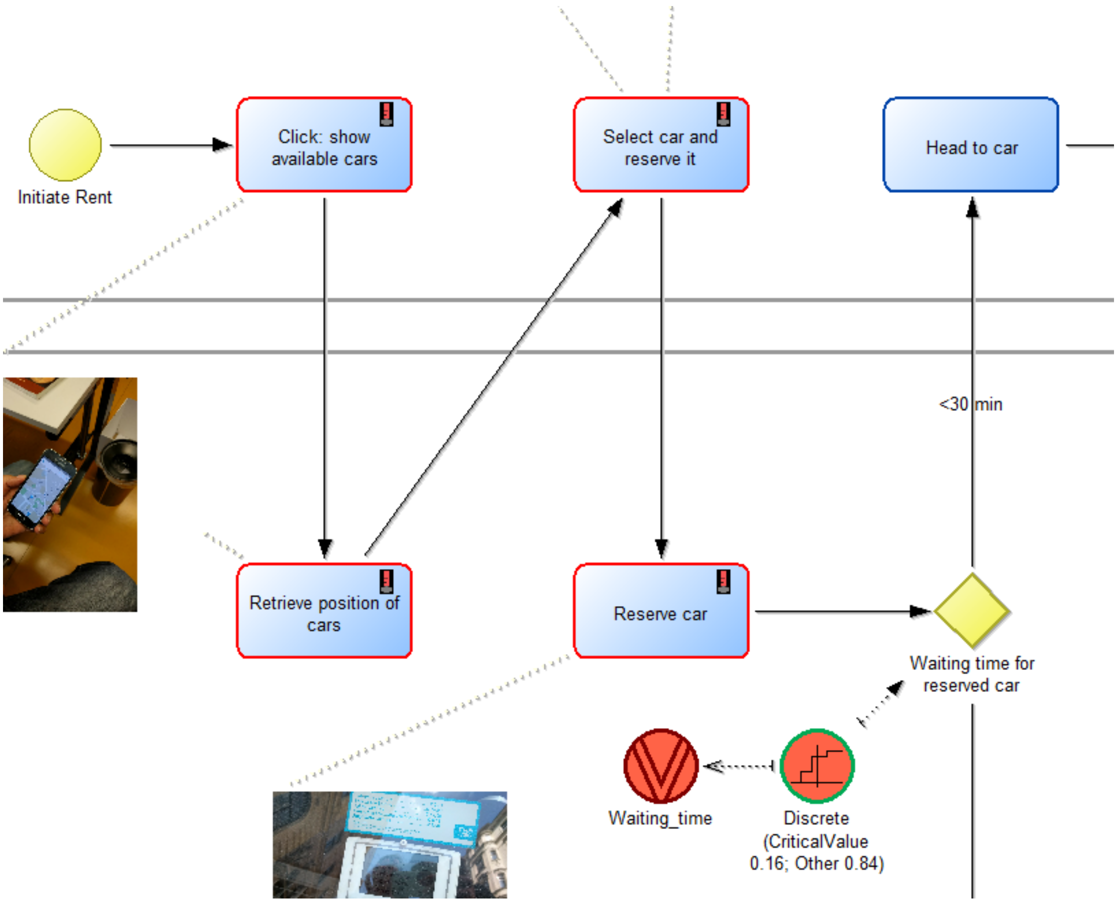


Figure 4.2: Excerpt of Process Car rental (Waiting time)

the process in two branches: a sensor checks if the key has been moved (with an attached tag) from its particular position, a second sensor checks if there is actually somebody sitting on the driver's seat (figure 4.4). If both conditions are verified, the rental and the count of the rental time start. On the basis of this time count, rental charges are calculated. After the car usage has ended and the engine is turned off, a similar non-exclusive gateway splits the process in three branches: it checks if the key is back in its initial position, whether the driver's seat is still occupied and if doors are still open or closed. For these three decisions, *Sensor Data Generator* elements once again calculate the individual distributions with dummy data (figure 4.5).

If there is still, e.g., the left door open, a message is displayed on the outside of the car until it is properly closed. The rest of the process has various IoT tasks that send back and forth data between a customer's smartphone and the rental company's cloud (click on end rental, input of car condition), or tasks between the

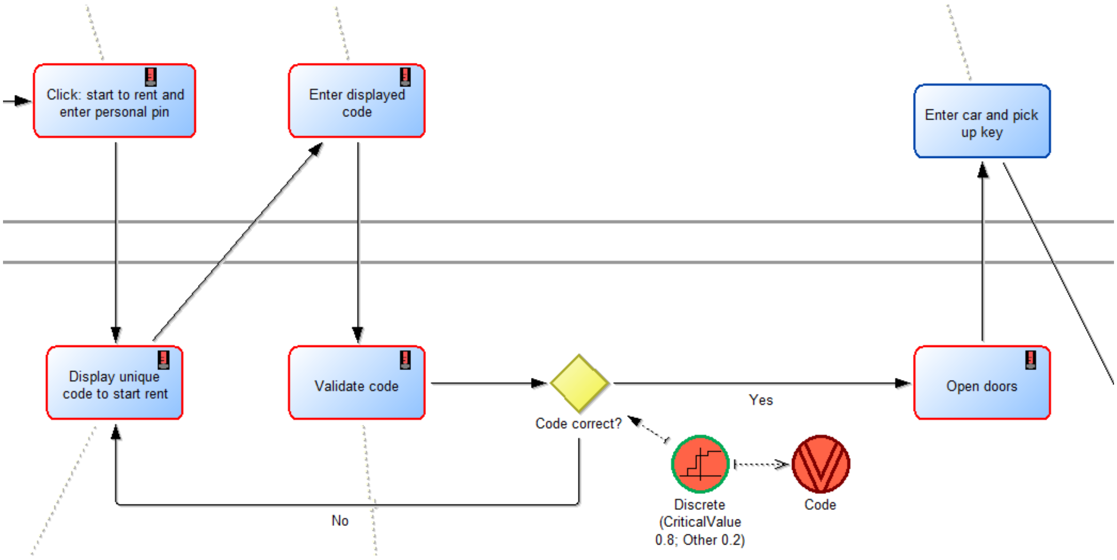


Figure 4.3: Excerpt of Process Car rental (Correct code)

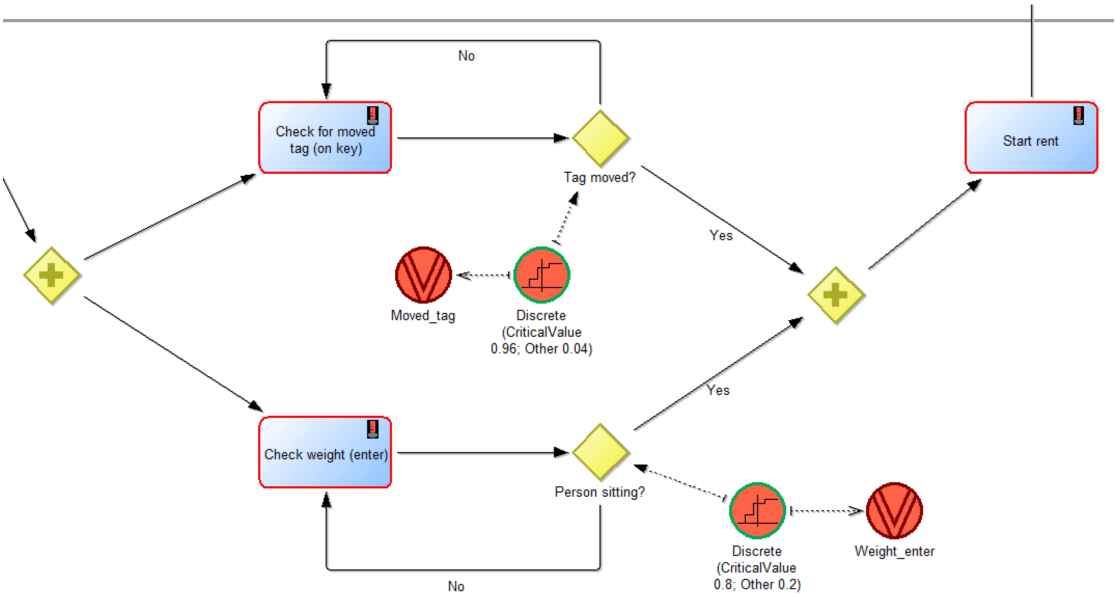


Figure 4.4: Excerpt of Process Car rental (Start rental)

car and the cloud that exchange information on the position and the current status of the car. The process finally ends when the customer receives a confirmation of his or her ride and a receipt providing information on route and fee (figure 4.6). A figure of the entire process *Car rental* can be found in the Appendix C.

Note: the sensor data in this process and the following simulation are imagi-

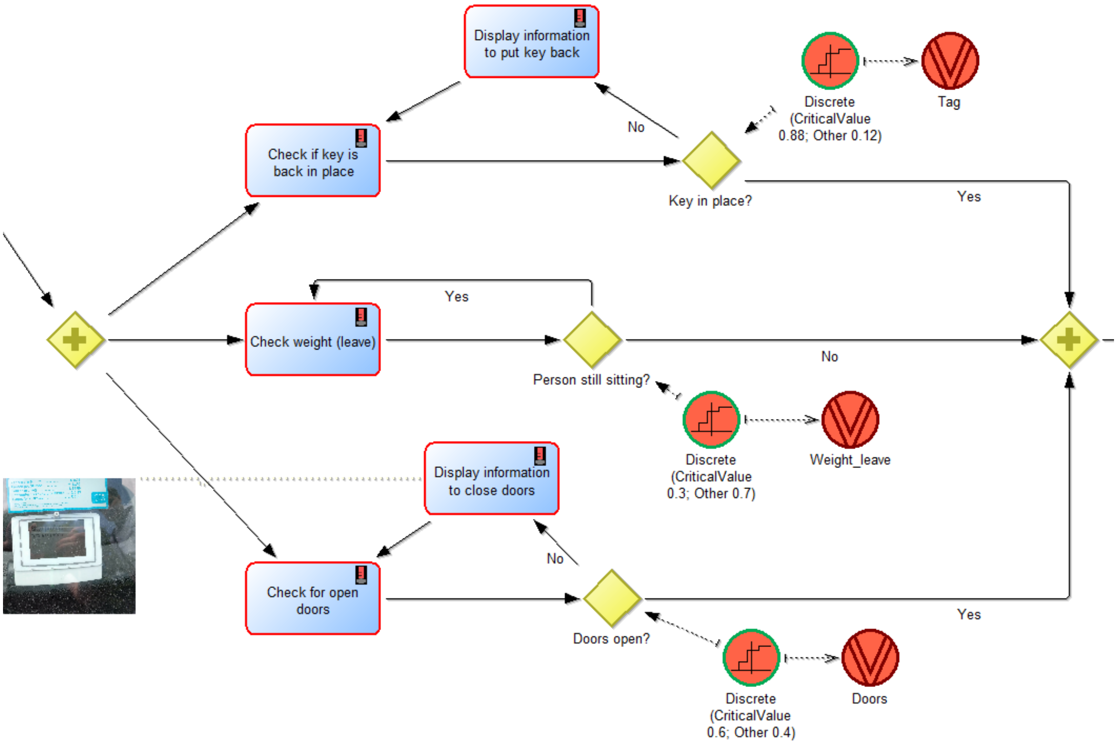


Figure 4.5: Excerpt of Process Car rental (Sensor checks)

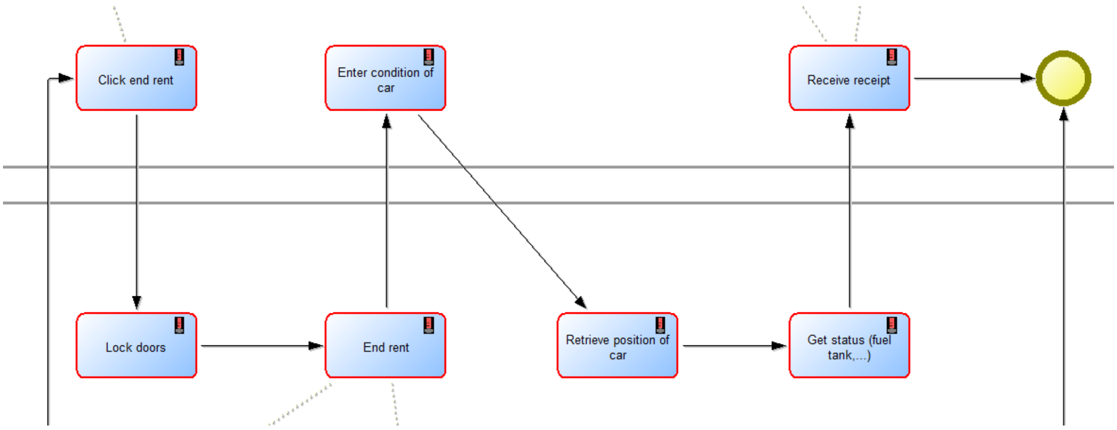


Figure 4.6: Excerpt of Process Car rental (End rental)

nary. However, it has been attempted to generate the values as realistic as possible. For instance, the .CSV-file containing bodyweight for the sensor to check whether somebody is sitting in the car or not, reads between 50kg and 101kg which constitutes a feasible weight range of a regular person in possession of a driver's license. Some other readings ranged between 0kg and 15kg to simulate that only a bag or nothing at all is on the driver's seat. With a critical value of, e.g., 45kg the distribution can be calculated easily. Data for the sensors that only read 1/0 (e.g. the moved tag) are assumptions, too. They were generated to be in a reasonable range so that the process would be executed with having to pass some decisions twice (moved tag, for instance, is 96% true but 4% not and this decision would have to be executed at least once more).

4.3 IoT Device / Sensor View

The IoT Device Model shows a device (in this case a smart car), its senors and its components (figure 4.7). Since there are numerous sensors present in a modern car, they have been grouped with *Aggregation* elements in different classes (e.g. class Proximity with motion detector and proximity sensor) to keep them clearly arranged. Moreover, components are necessary for making an item smart, therefore four basic components have been modeled.

4.4 Architecture View

To continue with the example of smart car rental service, an underlying architecture model is shown in figure 4.8. This model consists of infrastructure components such as databases and a business server (which in this case is outsourced to the Amazon cloud AWS indicated in orange), services that run on this infrastructure, applications and finally, and an IoT device which uses the applications.

4.5 Simulation

Before discussing the results of the simulation of the business process *Car rental* introduced in section 4.2 Process View, it is necessary to clarify that the execution of the simulation itself is exactly the same as in SeMFIS or other ADO* programs that include an simulatable BPMN 2.0 model. While the distribution and its calculation have been changed (the regular elements for the distribution remain as well), the simulation parts have been unaffected.

The IoT modeling method offers the possibility to simulate execution time as well as costs of activities. To get started, one has to declare time (execution time, waiting time, resting time, transport time are offered) and/or costs (activity costs and various process costs) of each previously modeled *Task* or *IoT Task* element.

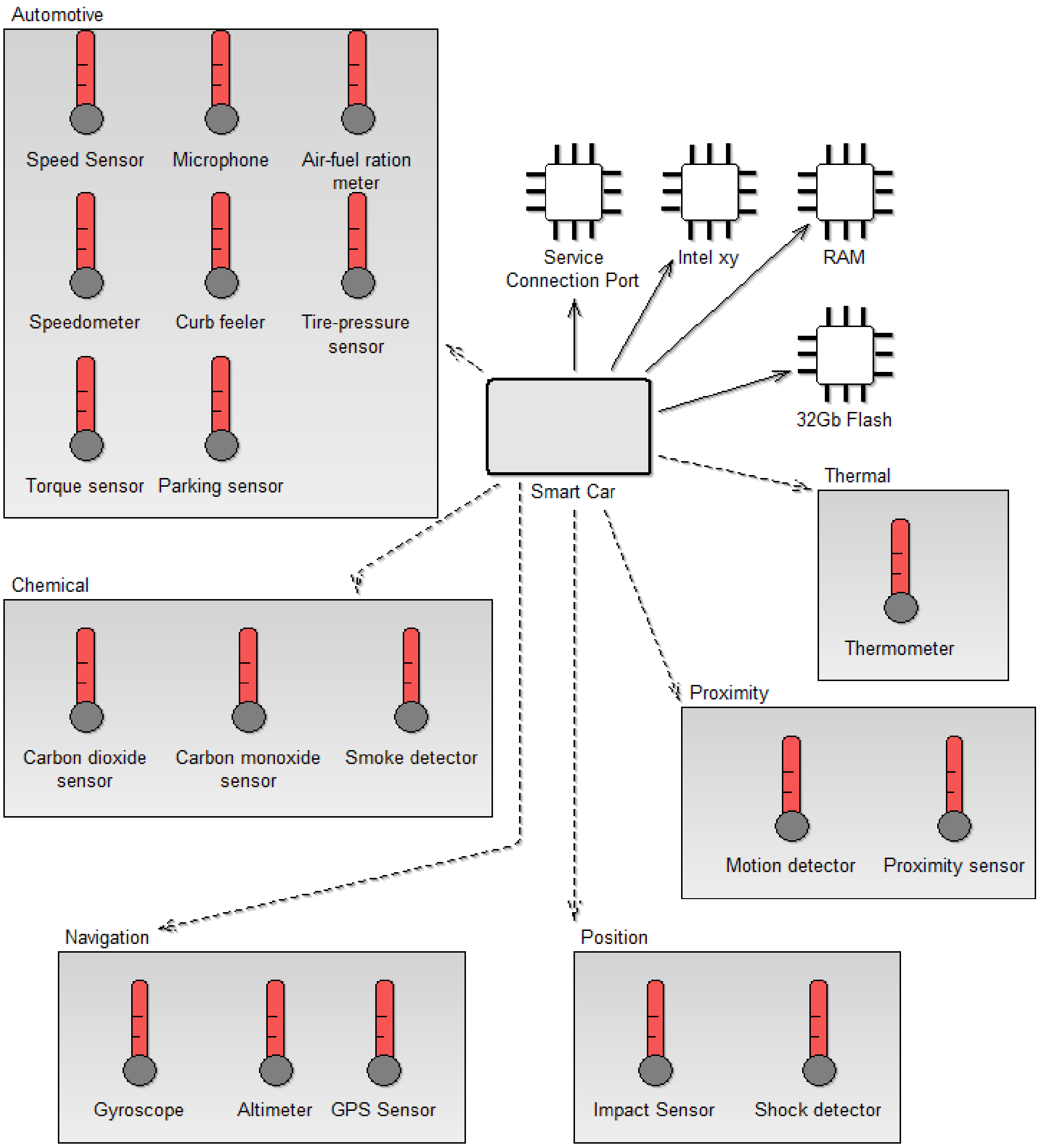


Figure 4.7: IoT Device Model for a Car Rental Service

Afterwards, one can switch from "Modeling View" to "Simulation View" in the ADOxx Modelling Toolkit to start, for instance, a path analysis and simulate which paths are chosen how often after, e.g., 1000 runs of a business process.

For the evaluation of the *Car rental* process, the execution times have been filled out accordingly to execute a path analysis. After 1000 runs of the process, the simulation provides the following results: execution time is 22:47 minutes, waiting time is 4:03 minutes and cycle time is 26:46 minutes on average. 151 different paths have been taken during this cycle of simulation which could change with the

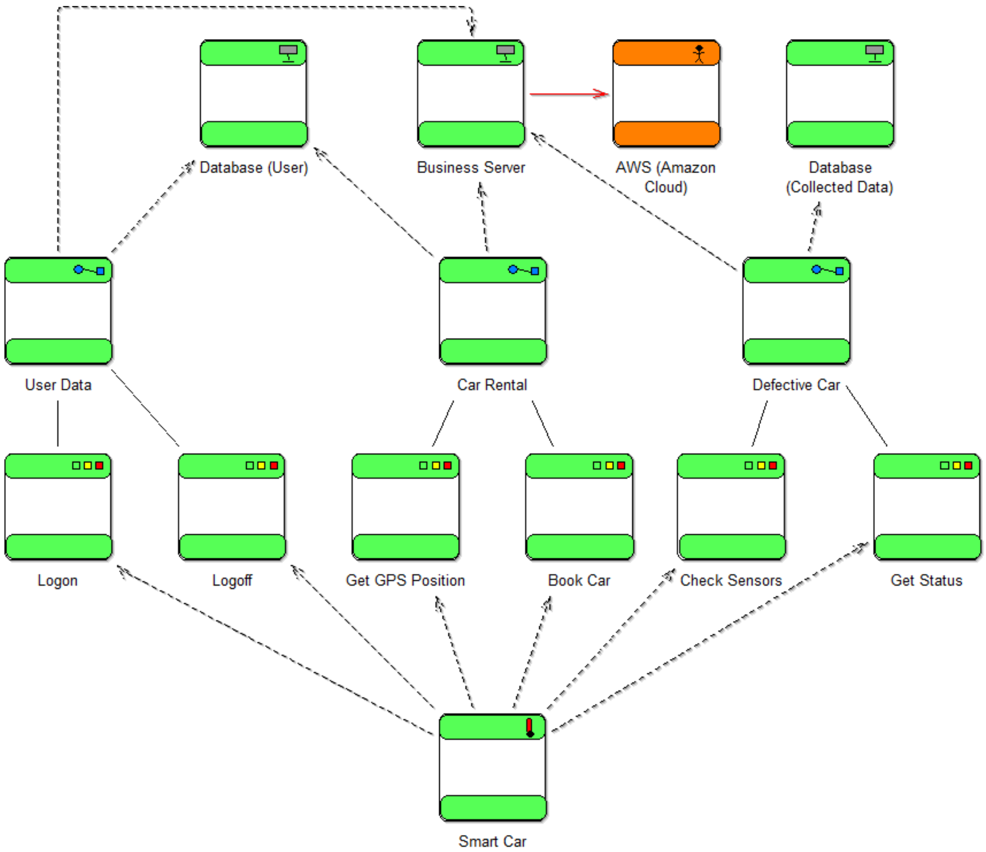


Figure 4.8: Example of an Architecture for a Car Rental Service

next 1000 runs since there are many decisions with various outcomes. Each of the 151 paths can be examined thoroughly by means of specified criteria by ADOxx (e.g. probability, cycle time or costs). By clicking on "Path results..." in ADOxx, the chosen path is listed with all the decisions that were made (see Appendix C for the entire process in which the taken path is marked in green). Since the sort criterion was "Probability" in this case, "Path 1" of 151 is most common with 18.8% outcome. All of the paths can be saved since a second simulation with 1000 runs would probably yield different results.

The simulation has shown that a view in advance can be beneficial. With 151 different paths after 1000 runs, a deeper consideration could reveal potential improvements. Generally, it showed that the process exceedingly relies on the IoT and smartphones, and enables something that would not have been feasible in such an easy way previously: contemplating available cars, renting one and driving from A to B by means of a smartphone seems rather simple, yet the involvement of IT and IoT is extremely high.

4.5.1 Advantages and Disadvantages of the IoT Modeling Method

The evaluation showed the following advantages:

- A unified modeling method including models for IoT-enhanced business processes, IoT devices, IT architecture and many more
- The integration of sensors into BPMN 2.0
- The automatic calculation of the distribution that is needed in case of decisions when simulating
- The problem-free simulation of these IoT processes (as known from ADOxx)

Disadvantages:

- Sensor data has to be pre-formatted for the automatic calculation
- Sensors could be integrated even deeper (possibly with the semantic annotation model)
- No real life testing as of yet (but a real world example with estimated data)

4.6 Future Work

Since the IoT Modeling Method has not been tested in real life circumstances yet, this would be one future goal. With the useful insights that could be yielded with real trials in the world of the IoT, improvements and extensions could be implemented easily due to the variable ADOxx metamodeling platform.

Moreover, sensor data has to be pre-formatted at the moment, which could eventually be automated with another script that filters the needed data from different sources. An even further development could be the automatic handling of sensor data on a machine-readable basis with semantic models.

Generally, further research of the IoT should pursue the sector of security and privacy. With an almost permanent connection to the internet and interconnection, one of the main goals should be to make it as secure as possible to protect it against possible threats. A secure IoT will form the foundation for the promised next revolution in the IT sector so that it can eventually initiate.

Chapter 5

Conclusion

To conclude this master's thesis, a consideration of the findings is important. After a short outline of the motivation for having chosen this topic and a description of possible future impact of the IoT, the research questions have been conceived. There appears to be a research gap especially on modeling the IoT. For this reason, the second half of the present thesis focuses on this topic.

Before a particularly for this thesis implemented modeling method was presented, the basics of the IoT were described in chapter 2. Key parts of the IoT are the *things* themselves which all represent an object that is connected with a network and other devices. Porter and Heppelmann call them smart, connected products. These products basically have a small computer attached that is responsible for measuring with integrated sensors, responding to environmental influences and allowing to remotely control them. They can also be altered with updates to gradually implement new features or simply improve the overall performance of such devices. From a business point of view, this creates longer lasting products (at least the software, a hardware failure can still occur) and, if conducted correctly, ties customers to a platform that allows for expanding their products with, e.g., value-added services. The possibilities for recurring revenues are extremely interesting for businesses as it is preferred over selling regular products over the years since customers might buy a product from a different manufacturer subsequently. To realize such smart, connected products, Heppelmann and Porter proposed the *New Technology Stack* which states that the interplay between a product, its cloud and influences of the outside world, such as external information sources, as well as integration of a business system have to be considered as well. As far as transformations inside a company are concerned, the IoT will impact the processing of incoming data and possibilities to generate the highest profit by analyzing new insights of the customers' usage. New organizational structures with new divisions might be necessary to cope with the yielded data or the work between product engineers and IT specialists for a complete and well-thought-out user experience of a smart, connected product. Another section of the first and more theoretical part of this thesis showed standardization efforts for enabling easier development of the IoT. Remaining challenges and problems were pointed out as well. Especially, security and privacy issues are of concern and the objective should be to ensure that customers do not have to worry about that.

In the second and more practical part of the present thesis, a modeling method for the IoT has been developed that allows users to model, e.g., IoT devices including sensors as well as the architecture that has to be built behind them. Business processes can be modeled in the form of BPMN 2.0 models which have been extended to fit the IoT context. Generally, some existing models were adapted for the IoT while others were newly created. This developed method is trying to tackle the research gap that is still left in the field of modeling methods for the IoT. Before work on thesis initiated, only one other project focused on modeling and IoT, and extended the task elements of the BPMN 2.0 model with sensors. However, the modeling method accompanying this master's thesis is considerably broader, offering various models that, if they are all combined, offer users a full experience to model IoT systems including products, architecture, business processes but also the entire workforce, properly. Furthermore, since it is built on the SeMFIS library for ADOxx, Semantic Annotation Models and Ontology Models are included for another way to retain the data such as incoming sensor measurements. The possibility to add ontologies to this modeling method served as inspiration to create an ontology, as well, which classifies sensors in their classes based on what they measure and which purpose they have.

The implemented metamodels have then been applied to a concrete scenario to evaluate the modeling method. Initial point was a smart car rental service with an integral business process *Car rental* that shows the interaction between a customer and a car with sensors working in the background. The architecture as well as the IoT device itself have been modeled in order to show the possibilities of this IoT modeling method.

Appendix A

Zusammenfassung

Das Internet of Things (kurz: IoT) ist einer der größten Trends der vorangegangenen Jahre in der IT-Branche. Forschungsinstitute attestieren für die kommenenden Jahre extremes Wachstum. Michael Porter, Wirtschaftsprofessor der Harvard Business School, spricht sogar von der dritten IT-getriebenen Innovationswelle, nachdem die Informationstechnologie und das Internet Unternehmen und die Geschäftslandschaft bereits nachhaltig verändert haben. Diese Masterarbeit fasst dabei zunächst den aktuellen Stand der Dinge des IoT zusammen, bevor eine Modellierungsmethode zu eben diesem Thema vorgestellt wird. Diese Modellierungsmethode wurde durch die Veränderungen von Geschäftsprozessen inspiriert, die auf Grund von Sensoren und generell intelligenter werdenden Produkten vor neuen Herausforderungen stehen. Mittels ADOxx wurde eine Methode implementiert, die mit diesem neuen Zeitalter der IT gehen möchte und auf SeMFIS von Dr. Hans-Georg Fill aufbaut. Da SeMFIS Ontologien unterstützt, wurde auch eine Ontologie, die verschiedene Sensoren klassifiziert, implementiert. Abschließend wurde die Modellierungsmethode für das IoT anhand eines fiktiven Beispiels evaluiert.

Appendix B

ADOScript Code Snippets

CC *"AdoScript"* **EDITFIELD title:** *"Enter critical value 1"* **caption:** *"Critical value 1:"* **text:** *"0.0"*
IF *(ecode = 0)* {
SET *critical1:(text)*
}

Programcode 1: ADOscript User input of critical value

```
IF (count <> 0) {
  FOR rowindex from:1 to:(count) {
    CC "Core" GET_REC_ATTR_ROW_ID objid: (objIdModified)
attrid: (datarecattrid) index: (rowindex)
  SET idRowSource:(rowid)
  CC "Core" GET_CLASS_ID objid: idRowSource SET
idClassRowSource:(classid)
  CC "Core" GET_ATTR_ID classid: (idClassRowSource) attrname:
"Data" SET idAttrData:(attrid)
  CC "Core" GET_ATTR_VAL objid: (idRowSource) attrid:
(idAttrData)
    IF (val ≥ critical1) {
      SET count1:(count1+1)
      }
      ELSE
      {
      SET count2:(count2+1)
     }
  }
  SET dist1 : (count1/number_rows)
  SET dist2 : (count2/number_rows)
}
SET distribution :
("Discrete(CriticalValue" + STRdist1 + ";Other" + STRdist2 + ")")
CC "Core" SET_ATTR_VAL objid: (objIdModified) attrname:
"Value" val: (distribution)
```

Programcode 2: ADOScript Query of data / calculation of the Distribution

Appendix C

Entire Process *Car rental*

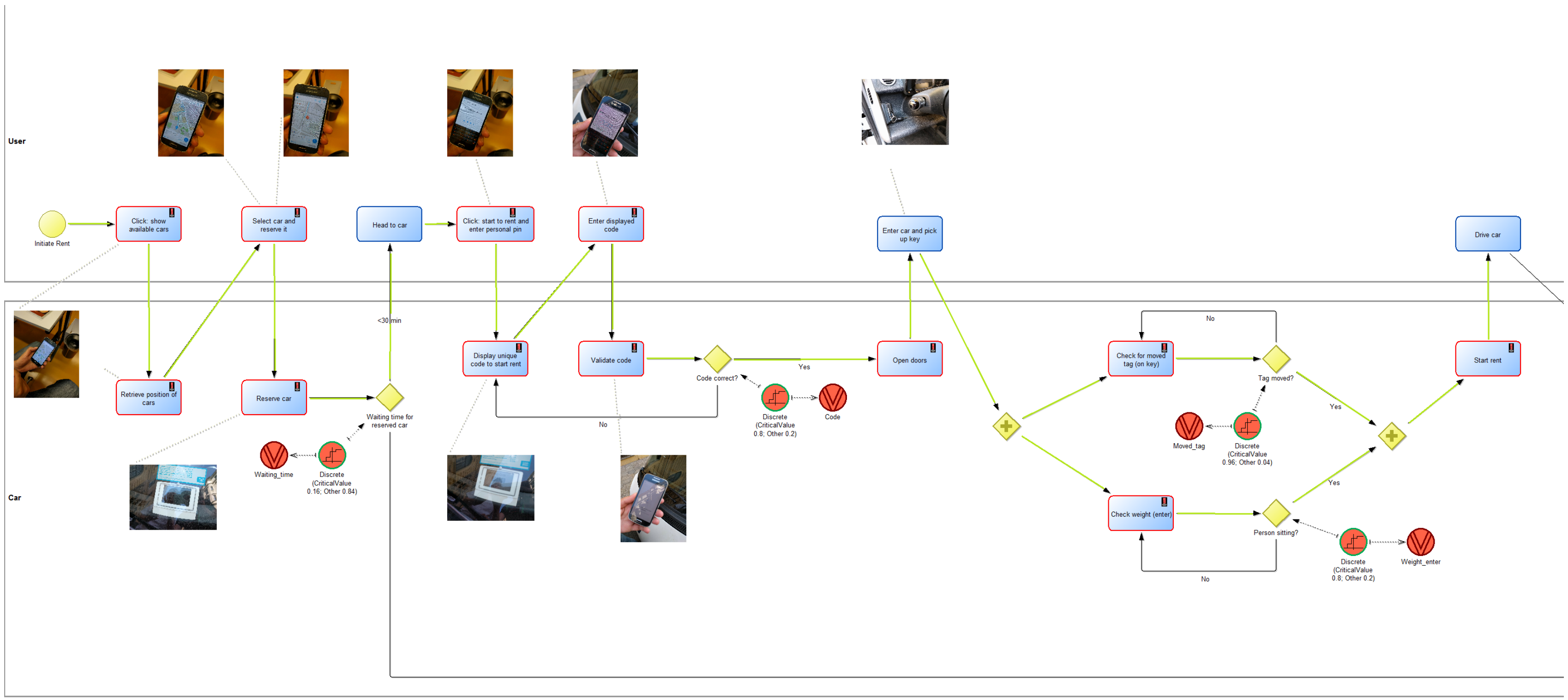


Figure C.1: Entire Process Car rental Part 1

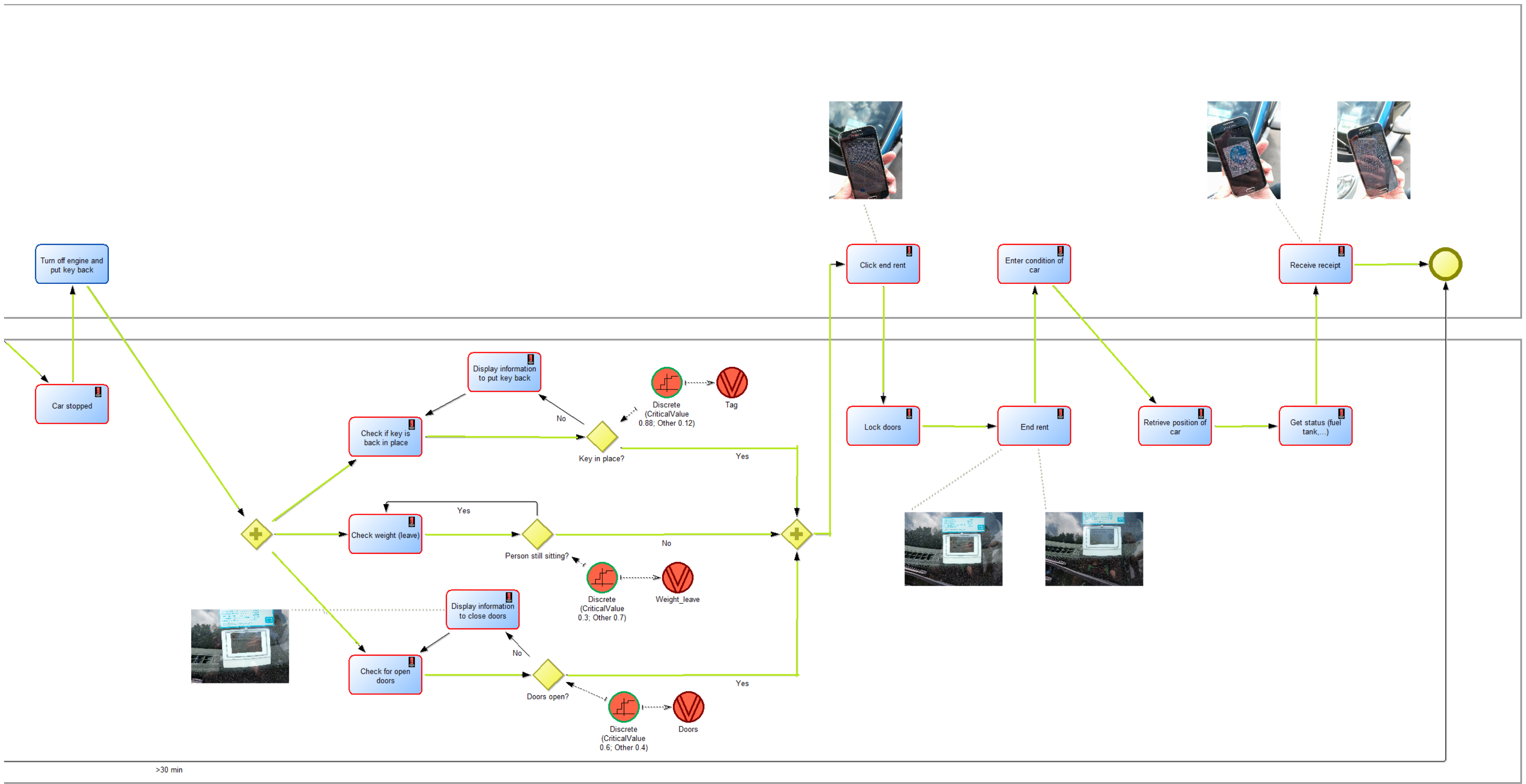


Figure C.2: Entire Process Car rental Part 2

Bibliography

[1] Malte Ackermann. “Internet of Things Changes the Definition of What a Product is”. In: *embeddedworld2015 Conference* (2015).

[2] Charu C. Aggarwal, Naveen Ashish, and Amit Sheth. “The Internet of Things: A Survey from the Data-Centric Perspective”. In: *Managing and Mining Sensor Data*. Ed. by C. Charu Aggarwal. Boston, MA: Springer US, 2013, pp. 383–428. ISBN: 978-1-4614-6309-2.

[3] Rachel Arthur and Forbes. *10 Billion Items Of Connected Clothing: The Internet Of Things Just Became A Lot More Fashionable*. 2016. URL: `http://www.forbes.com/sites/rachelarthur/2016/04/21/10-billion-items-of-connected-clothing-the-internet-of-things-just-became-a-lot-more-fashionable/#7a61c32d4d66` (visited on 05/18/2016).

[4] Ron Ashkenas, Dave Ulrich, Todd Jick, and Steve Kerr. *The boundaryless organization: Breaking the chains of organizational structure*. John Wiley & Sons, 2015.

[5] Kevin Ashton. “That ‘internet of things’ thing”. In: *RFiD Journal* 22.7 (2009), pp. 97–114.

[6] Luigi Atzori, Antonio Iera, and Giacomo Morabito. “From” smart objects” to” social objects”: The next evolutionary step of the internet of things”. In: *Communications Magazine, IEEE* 52.1 (2014), pp. 97–105.

[7] Luigi Atzori, Antonio Iera, and Giacomo Morabito. “The internet of things: A survey”. In: *Computer networks* 54.15 (2010), pp. 2787–2805.

[8] Armin Balalaie, Abbas Heydarnoori, and Pooyan Jamshidi. “Migrating to cloud-native architectures using microservices: An experience report”. In: *European Conference on Service-Oriented and Cloud Computing*. Springer. 2015, pp. 201–215.

[9] Thomas Bauernhansl, Michael Ten Hompel, and Birgit Vogel-Heuser. *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung· Technologien· Migration*. Springer-Verlag, 2014.

[10] Shen Bin, Liu Yuan, and Wang Xiaoyi. "Research on data mining models for the internet of things". In: *2010 International Conference on Image Analysis and Signal Processing*. Apr. 2010, pp. 127–132.

[11] Alessio Botta, Walter De Donato, Valerio Persico, and Antonio Pescapé. "On the integration of cloud computing and internet of things". In: *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*. IEEE. 2014, pp. 23–30.

[12] Eva Bucherer and Dieter Uckelmann. "Business Models for the Internet of Things". In: *Architecting the Internet of Things*. Ed. by Dieter Uckelmann, Mark Harrison, and Florian Michahelles. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 253–277. ISBN: 978-3-642-19157-2.

[13] Benjamin Cabé. *Growing an Open Community: 3 interesting Facts about Eclipse IoT*. 2016. URL: `https://blog.benjamin-cabe.com/2015/08/07/growing-an-open-community-3-interesting-facts-about-eclipse-iot` (visited on 05/07/2016).

[14] Herbert Dawid, Reinhold Decker, Thomas Hermann, Hermann Jahnke, Wilhelm Klat, Rolf König, and Christian Stummer. "Management science in the era of smart consumer products: challenges and research perspectives". In: *Central European Journal of Operations Research* (2016), pp. 1–28. ISSN: 1613-9178.

[15] R.M. Dijkman, B. Sprenkels, T. Peeters, and A. Janssen. "Business models for the Internet of Things". In: *International Journal of Information Management* 35.6 (2015), pp. 672–678. ISSN: 0268-4012.

[16] Joshua M. Epstein. "Why Model?" In: *Journal of Artificial Societies and Social Simulation* 11.4 (2008), p. 12. ISSN: 1460-7425. URL: `http://jasss.soc.surrey.ac.uk/11/4/12.html`.

[17] Hans-Georg Fill. "An Approach for Analyzing the Effects of Risks on Business Processes Using Semantic Annotations". In: *ESADE/AIS* (2012).

[18] Hans-Georg Fill. "On the conceptualization of a modeling language for semantic model annotations". In: *International Conference on Advanced Information Systems Engineering*. Springer. 2011, pp. 134–148.

[19] Hans-Georg Fill. "On the social network based semantic annotation of conceptual models". In: *International Conference on Knowledge Science, Engineering and Management*. Springer. 2014, pp. 138–149.

[20] Hans-Georg Fill. "SeMFIS: A Flexible Engineering Platform for Semantic Annotations of Conceptual Models". In: *Semantic Web Journal, IOS Press* (2015).

[21] Hans-Georg Fill. "SeMFIS: A Tool for Managing Semantic Conceptual Models". In: *Workshop on Graphical Modeling Language Development* (2012).

[22] Hans-Georg Fill and Dimitris Karagiannis. "On the conceptualisation of modelling methods using the ADOxx meta modelling platform". In: *Enterprise Modelling and Information Systems Architectures-An International Journal* 8.1 (2013).

[23] Hans-Georg Fill, Daniela Schremser, and Dimitris Karagiannis. "A Generic Approach for the Semantic Annotation of Conceptual Models using a Service-oriented Architecture". In: *International Journal of Knowledge Management* 9.1 (2013). URL: `http://eprints.cs.univie.ac.at/3056/`.

[24] Klaus Finkenzeller. *RFID-Handbuch: Grundlagen und praktische Anwendungen von Transpondern, kontaktlosen Chipkarten und NFC*. Carl Hanser Verlag GmbH Co KG, 2015.

[25] Gartner Inc. *Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015*. 2015. URL: `http://www.gartner.com/newsroom/id/3165317` (visited on 05/18/2016).

[26] Gartner Inc. *Gartner's 2015 Hype Cycle for Emerging Technologies Identifies the Computing Innovations That Organizations Should Monitor*. 2015. URL: `http://www.gartner.com/newsroom/id/3114217` (visited on 05/18/2016).

[27] Torsten J. Gerpott and Sebastian May. "Integration of Internet of Things components into a firm's offering portfolio – a business development framework". In: *info* 18.2 (2016), pp. 53–63.

[28] Morten T Hansen. "Knowledge networks: Explaining effective knowledge sharing in multiunit companies". In: *Organization science* 13.3 (2002), pp. 232–248.

[29] Gordon Hui. "How the internet of things changes business models". In: *Harvard Business Review* (2014), pp. 1–5.

[30] Kai Hwang, Jack Dongarra, and Geoffrey C Fox. *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan Kaufmann, 2013.

[31] Marco Iansiti and Karim R Lakhani. "Digital Ubiquity: How Connections, Sensors, and Data Are Revolutionizing Business (Digest Summary)". In: *Harvard Business Review* 92.11 (2014), pp. 91–99.

[32] Isam Ishaq, David Carels, Girum K Teklemariam, Jeroen Hoebeke, Floris Van den Abeele, Eli De Poorter, Ingrid Moerman, and Piet Demeester. "IETF standardization in the field of the internet of things (IoT): a survey". In: *Journal of Sensor and Actuator Networks* 2.2 (2013), pp. 235–287.

[33] Raj Jain. *Internet of Things: Challenges and Issues*. `http://www.cse.wustl.edu/~jain/talks/iot_ad14.htm`. 2014. (Visited on 05/07/2016).

[34] Antonio J. Jara, María Concepción Parra, and Antonio F. Skarmeta. "Participative marketing: extending social media marketing through the identification and interaction capabilities from the Internet of things". In: *Personal and Ubiquitous Computing* 18.4 (2014), pp. 997–1011. ISSN: 1617-4917.

[35] Qi Jing, Athanasios V Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. "Security of the internet of things: Perspectives and challenges". In: *Wireless Networks* 20.8 (2014), pp. 2481–2501.

[36] Jaehyeon Ju, Mi-Seon Kim, and Jae-Hyeon Ahn. "Prototyping Business Models for IoT Service". In: *Procedia Computer Science* 91 (2016). Promoting Business Analytics and Quantitative Management of Technology: 4th International Conference on Information Technology and Quantitative Management (ITQM 2016), pp. 882–890. ISSN: 1877-0509.

[37] Stefan Junginger, Harald Kühn, Robert Strobl, and Dimitris Karagiannis. "Ein geschäftsprozessmanagement-werkzeug der nächsten generation—ADONIS: konzeption und anwendungen". In: *Wirtschaftsinformatik* 42.5 (2000), pp. 392–401.

[38] Dimitris Karagiannis, Stefan Junginger, and Robert Strobl. "Introduction to business process management systems concepts". In: *Business process modelling*. Springer, 1996, pp. 81–106.

[39] Dimitris Karagiannis and Harald Kühn. "Metamodelling platforms". In: *EC-Web*. Vol. 2455. 2002, p. 182.

[40] Gerd Kortuem, Fahim Kawsar, Vasughi Sundramoorthy, and Daniel Fitton. "Smart objects as building blocks for the internet of things". In: *IEEE Internet Computing* 14.1 (2010), pp. 44–51.

[41] Matthias Kovatsch, Simon Mayer, and Benedikt Ostermaier. "Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things". In: *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on.* IEEE. 2012, pp. 751–756.

[42] Jeff Kramer. "Is Abstraction the Key to Computing?" In: *Commun. ACM* 50.4 (Apr. 2007), pp. 36–42. ISSN: 0001-0782.

[43] Shancang Li, Li Da Xu, and Shanshan Zhao. "The internet of things: a survey". In: *Information Systems Frontiers* 17.2 (2015), pp. 243–259.

[44] Shancang Li, Li Da Xu, and Xinheng Wang. "Compressed Sensing Signal and Data Acquisition in Wireless Sensor Networks and Internet of Things". In: *IEEE Transactions on Industrial Informatics* 9.4 (Nov. 2013), pp. 2177–2186. ISSN: 1551-3203.

[45] Max V. Maier. *The Internet of Things (IoT): what is the potential of Internet of Things applications for consumer marketing?* June 2016. URL: `http://essay.utwente.nl/70001/` (visited on 08/13/2016).

[46] Christoph P Mayer. "Security and privacy challenges in the internet of things". In: *Electronic Communications of the EASST* 17 (2009).

[47] Emily Mullin and Forbes. *Pfizer And IBM Launch Ambitious 'Internet Of Things' For Parkinson's Research.* 2016. URL: `http://www.forbes.com/sites/emilymullin/2016/04/07/pfizer-ibm-launch-ambitious-internet-of-things-for-parkinsons-research/#3382766b24d6` (visited on 05/18/2016).

[48] Object Management Group (OMG). *Business Process Model and Notation 2.0 Specification.* 2011. URL: `http://www.omg.org/spec/BPMN/2.0/` (visited on 08/10/2016).

[49] Object Management Group (OMG). *Unified Modeling Language Specification.* 2015. URL: `http://www.omg.org/spec/UML/` (visited on 08/10/2016).

[50] Rubina G.T. Oliana. *The Internet of Things : the next big thing for new product development?* June 2016. URL: `http://essay.utwente.nl/69980/` (visited on 08/13/2016).

[51] Pierluigi Paganini. *Internet of Things - Privacy and Security issues.* `http://de.slideshare.net/paganinip/io-t-42674396`. 2014. (Visited on 05/07/2016).

[52] Michael E Porter and James E Heppelmann. "How Smart, Connected Products Are Transforming Companies". In: *Harvard Business Review* 93.10 (2015), pp. 96–114.

[53] Michael E Porter and James E Heppelmann. "How smart, connected products are transforming competition". In: *Harvard Business Review* 92.11 (2014), pp. 64–88.

[54] Dominik Rüchardt and Christoph Bräuchle. "A large software vendor's view on Cyber Physical Systems". In: *2016 3rd International Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems (EITEC).* Apr. 2016, pp. 29–34.

[55] Ted Saarikko, Ulrika H Westergren, and Tomas Blomquist. "The Inter-Organizational Dynamics of a Platform Ecosystem: Exploring Stakeholder Boundaries". In: *2016 49th Hawaii International Conference on System Sciences (HICSS).* IEEE. 2016, pp. 5167–5176.

[56] SAP, Sonja Meyer, Klaus Sperner, Stefan Meissner, Carsten Magerkurth, Stefan Debortoli, and Matthias Thoma. "Internet of Things Architecture IoT-A Project Deliverable D2. 2–Concepts for Modelling IoT-Aware Processes". In: *IoT-A* (2012).

[57] Ian Skerrett. *IoT Developer Survey.* `http://de.slideshare.net/IanSkerrett/iot-developer-survey-2015`. 2015. (Visited on 05/07/2016).

[58] Herbert Stachowiak. *Allgemeine Modelltheorie.* Springer-Verlag, 1973.

[59] Fei Tao, Ying Cheng, Li Da Xu, Lin Zhang, and Bo Hu Li. "CCIoT-CMfg: cloud computing and internet of things-based cloud manufacturing service system". In: *IEEE Transactions on Industrial Informatics* 10.2 (2014), pp. 1435–1442.

[60] Chun-Wei Tsai, Chin-Feng Lai, Ming-Chao Chiang, and Laurence T. Yang. "Data Mining for Internet of Things: A Survey". In: *IEEE Communications Surveys Tutorials* 16.1 (2014), pp. 77–97. ISSN: 1553-877X.

[61] Rajesh Venkatesan, Madhava Venkatesh Raghavan, and K Satya Sai Prakash. "Architectural Considerations for a Centralized Global IoT Platform". In: *Region 10 Symposium (TENSYMP), 2015 IEEE.* May 2015, pp. 5–8.

[62] Niksa Visic, Hans-Georg Fill, Robert Andrei Buchmann, and Dimitris Karagiannis. "A domain-specific language for modeling method definition: From requirements to grammar". In: *Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on*. IEEE. 2015, pp. 286–297.

[63] W3C. *OWL Web Ontology Language Overview*. 2004. URL: https://www.w3.org/TR/owl-features/ (visited on 08/10/2016).

[64] Bruce D Weinberg, George R Milne, Yana G Andonova, and Fatima M Hajjat. "Internet of Things: Convenience vs. privacy and secrecy". In: *Business Horizons* 58.6 (2015), pp. 615–624.

[65] Dennis Westerbeek. *The Internet of Things as a source for improving the new product development process*. June 2016. URL: http://essay.utwente.nl/69966/ (visited on 08/14/2016).

[66] Wikipedia. *List of sensors*. 2016. URL: https://en.wikipedia.org/wiki/List_of_sensors (visited on 08/10/2016).

[67] Annick Willem and Marc Buelens. "Knowledge sharing in inter-unit cooperative episodes: The impact of organizational structure dimensions". In: *International Journal of Information Management* 29.2 (2009), pp. 151–160.

[68] Charlie Wilson, Tom Hargreaves, and Richard Hauxwell-Baldwin. "Smart homes and their users: a systematic analysis and key challenges". In: *Personal and Ubiquitous Computing* 19.2 (2015), pp. 463–476.

[69] Thomas Zachariah, Noah Klugman, Bradford Campbell, Joshua Adkins, Neal Jackson, and Prabal Dutta. "The internet of things has a gateway problem". In: *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. ACM. 2015, pp. 27–32.

[70] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. "Internet of things for smart cities". In: *Internet of Things Journal, IEEE* 1.1 (2014), pp. 22–32.