Article

Increasing System Reliability by Applying Conceptual Modeling and Data Analysis—A Case Study: An Automated Parking System

Haytham B. Ali * , **Gerrit Muller** , **Fahim A. Salim, Kristin Falk and Serkan Güldal**

Faculty of Technology, Natural Sciences and Maritime Sciences, University of South-Eastern Norway, 3616 Kongsberg, Norway
* Correspondence: haytham.ali@usn.no

Abstract: In complex sociotechnical research, companies aim to utilize data and digitalization to increase a system's reliability and to minimize a system's failures. This study exemplifies the use of conceptual modeling and data analysis to increase a system's reliability by studying a case study for a medium-sized company. The company delivers an Automated Parking System (APS). We identified, collected, and analyzed internal and external data within this context. Internal data consist of failure data from maintenance, whereas external data include environmental data, mainly weather data. Data analyses transformed the collected data into information, where conceptual modeling facilitates the understanding of information by transforming it further into knowledge. We find that the combination of conceptual modeling and data analysis aids in exploring and understanding a system's reliability. This understanding enables a company to enhance its product-development process. Conceptual modeling and data analyses guide and support each other in an iterative and recursive manner, and they both complement each other. Conceptual modeling also facilitates communication and understanding.

Keywords: conceptual modeling; data analysis; system's reliability

1. Introduction

Automated Parking Systems (APSs) are parking structures that stack cars vertically to save space. The designs of these one-of-a-kind systems allow vehicles to be transported from the entry to their parking area without a driver being present. These vertical parking structures are also known by various other titles in addition to APS, such as robotic parking garage, car parking system, Automated Parking Facility (APF), and Automated Vehicle Storage and Retrieval System (AVSRS) [1]. APSs operate mainly in urban centers. Due to land scarcity, increasing numbers of vehicles, and urban mobility, there is a need for APSs, especially in metropolitan areas [2]. Studies have shown that finding a parking lot in a metropolitan city consumed around 6 min [3].

Furthermore, the research indicated that 9–56% of traffic in urban areas was due to finding parking for the car [3]. Thus, APSs are needed to ease urban mobility. Internationally, there is a significantly increasing demand for APSs. The value of APSs globally was USD 1.23 billion in 2019, and the value is expected to grow at a compound annual growth rate of almost 11% from 2020 to 2027 [4].

However, the APSs, in general, suffer from a variety of problems. These problems include end-user (car owner) mistakes, such as pushing or forgetting to push a button, which causes the freezing of the APS. In other words, end-users who are unfamiliar with the APS (SOI) can cause SOI failures. These failures are also called use failures or Human–Machine Interface (HMI) issues. Furthermore, the SOI sometimes retrieves the incorrect car, takes a long time to retrieve a vehicle, or does not retrieve the car, especially during high-volume usage. In addition, there are mechanical failures related to the design of

the SOI. These failures decrease the SOI's availability and increase downtime. Downtime increases costs. These costs include, among others, costs for repairing parts and fees for alternative conventional parking [5,6]. Thus, a company needs to increase the system's reliability by integrating data with reliability engineering activities such as a formal reliability engineering program. In particular, when studies show that the cost of undiscovered failure in the early design phase will give a ten-time cost increase in the operation use phase, which is a later phase of the product's lifecycle [7].

1.1. Reliability Engineering

Reliability engineering is essential in systems engineering and quality management [8]. The main objective of reliability engineering is to prevent system failure. Reliability engineering includes different data sources generated by the two phases of reliability engineering: proactive and reactive phase [9]. These sources include the following:

1. Data from the design synthesis;
2. Testing and analysis from the verification process;
3. Published information (literature);
4. Expert opinion;
5. System operation data;
6. Failure data.

In this study, we aim to use, among others, data generated from the reactive reliability phase to close the feedback loop in the proactive reliability engineering phase. In other words, we use feedback data in the matter of failure data from the operation into the early phase of the Product Development (PD) process. This PD process can be an existing or future development process for the SOI. The future PD for the same SOI, e.g., a new version of the SOI, is also called New Product Development (NPD) process. In the context of this paper, we refer to the system's reliability as system's downtime, which increases the cost of the system's operation and customer dissatisfaction. This dissatisfaction is one of the main pain points for a company.

1.2. Failure Data

Failure data may mainly come from maintenance record data or service-log data. Failure data analyses play a vital role in increasing a system's reliability due to the following reasons [9]:

1. Increases knowledge regarding a system's design and manufacturing deficiencies;
2. Supports the estimation of the reliability, availability, system failure rate, and Mean Time between Failures (MTBF);
3. Improves reliability by implementing design change recommendations;
4. Aids data-driven decision making using design reviews;
5. Supports determining systems' maintenance needs and their parts;
6. Assist in conducting reliability and life cycle cost trade-off studies.

In addition to the failure data, we collected and analyzed weather data to investigate environmental factors that affect failure events.

1.3. Environmental Data

Several studies used environmental data in different domains, such as biology and ecology [10,11]. We collected historical weather data for six years as environmental data. Based on the parking systems' locations, we collect weather for the cities where these systems are allocated. The weather data include the following parameters: location, date-time, temperature, precipitation, snow, wind speed, visibility, sunrise, sunset, humidity, condition, and condition's description.

To make sense of the data, we collected and analyzed the two sources of data, i.e., internal in terms of failure data and external in terms of weather data. In addition, the two sources of data verify each other.

1.4. Data Sensemaking Using Data Analysis

The main goal of data analysis is understanding the data and the results of the analysis, which is also called data sensemaking. Data sensemaking is a conscious effort to understand data and the event(s) within its context [12]. Wieck defines data sensemaking as a two-way process that aims to fit data into a frame (mental model), such as a template and a frame around the data. This process is iterative until the data and frame unite to aid decision makers in making more data-driven decisions by increasing the understanding of both data and context. We need to conduct data sensemaking during several iterations to improve the understanding and avoid oversimplifications [13]. Sensemaking is a process including different functions such as prediction (projecting the future), forming an explanation, seeing relationships or correlations (connecting the dots), anticipatory thinking, problem identification, and detection [12]. However, data analysis also includes a process of identifying, collecting, pre-processing, analyzing, and visualizing the results of the data analysis. We used conceptual modeling to achieve data sensemaking. We also used conceptual modeling to communicate, understand, and assist in implementing more data-driven decision making based on the results and visualization of the data analysis. This type of decision making includes decision making within the early design phase of the PD and maintenance processes.

1.5. Conceptual Modeling

We use conceptual modeling to deal with the complexity of analyzing the reliability of a company's System-of-Interest (SOI). Conceptual models are simple enough to share and communicate the understanding of needs, concepts, technologies, etc. These models are sufficiently detailed and realistic for guiding system development. Conceptual modeling is vital in various disciplines, such as simulations in the computer science domain, Soft Systems Methodology (SSM), and systems engineering. Various conceptual modeling types are used, such as visualization and graphing mathematical models (formulas), simulation models, and systems architecting models [14].

1.6. Introduction to the Company: Where Research Takes Place

The examined company used as a case study in this paper is a medium-sized enterprise that delivers APSs and provides maintenance services for operations. The company offers fully and semi-automated parking systems. The difference between a fully automated and a semi-automated is whether a parking attendant must drive or direct cars into the system. Fully automated parking systems do not require an attendant, and the APS's mechanical processes will transport vehicles from the garage's entrance to an available parking space.

On the other hand, a semi-automated parking system will require a parking attendant to guide the car into the machine after it arrives at a parking place [1]. The company starts begins its involvement before construction. Currently, the company is transitioning from only selling to developing, producing, and marketing APSs. The main customers for the company are building owners and land developers. The main stakeholders of the case study include company management, maintenance personnel, and car owners.

The company has around 36 parking installations in total; 35 parking installations are in Norway. In addition, the company has one parking system in Denmark. Figure 1 portrays the locations for the company's parking systems installations. However, we identified the parking system allocated in Oslo within a 30 km radius around Oslo with a different color (orange color). The other parking systems allocated more than 30 km from Oslo have a different color (purple). The parking system with an orange color is 17, which we refer to as private parking systems. Another parking system that we call a public parking system is colored in red. The other 18 parking systems are private parking systems at different locations in Norway, where also one of them is in Denmark (Copenhagen). We exclude these 18 parking systems from our data analysis mainly to increase the validity and accuracy of weather data analyses.
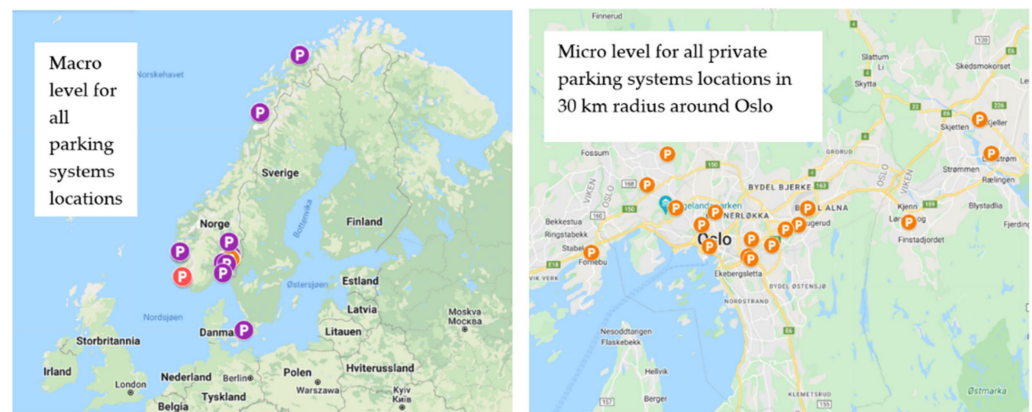
Figure 1. Map of overall company parking systems. Left: Macro-level for all parking systems locations. Right: Micro-level for all private parking systems in a radius of 30 km around Oslo.

The main difference between the private and public parking systems is that the public system does not have a permanent parking lot (platform) each time the end-users (car owners) use the system. On the other hand, private parking systems provide a permanent platform each time the car owners use the system to park their cars.

The research questions of this study are as follows:

- RQ1: How can conceptual modeling and data analysis enhance product developers' understanding of the system?
- RQ2: How conceptual modeling facilitate data identification to increase a system's reliability?
- RQ3: What are effective conceptual models for facilitating product developers' understanding of an APS?

One of the motivations is using data analysis and conceptual models to increase the system's reliability understanding. In this context, data analysis mainly utilizes feedback data from the operation phase and connects it to the PD process's early design phase. One primary source of such feedback data is failure data, also called maintenance record data. We also need other data sources to make sense of the failure data and its analysis. Other sources include, among others, in-system data and environmental data such as weather data. In this study, we collect and analyze maintenance record data and weather data. In parallel, we develop conceptual models using different abstraction levels and multi-views or perspectives. The data analysis supports conceptual modeling, and conceptual modeling guides the data analysis into an iterative and recursive process. The iterative process ends when conceptual modeling and data analysis harmonize (make sense) regarding the study's research questions.

The contribution of this study is exemplifying the use of conceptual modeling and data analysis in an iterative and recursive manner. This study concluded that conceptual modeling and data analysis complement each other. In this context, we generated several conceptual models that guide the identification, collection, and analysis of data. We collected and analyzed internal and external data sources. Furthermore, we analyzed these data sources by investigating frequencies, distributions, and correlation based on different time scopes. We investigated the anomalies among the results of the data analysis. We used conceptual modeling to understand these anomalies and enhance these models during several iterations. In this research, we used workshops, interviews, and observations to investigate the product developer's understating of the system's reliability using the study's approach. Practitioners from the industry and academia report that this approach is valuable in this context.

This paper is an extended version of the article presented at the Modern Systems Conference 2022, and it aims to apply conceptual modeling and failure data analysis to explore a company's actual needs [15]. For the conference paper, we applied conceptual modeling and data analyses in order to conduct a brief feasibility study by investigating the

value proposition of implementing Condition-Based Maintenance (CBM). We also explored actual needs behind this plan, which involve increasing a system's reliability. In this paper, we investigate a system's reliability in detail by exemplifying a more comprehensive use of conceptual modeling and data analyses. We analyze failure data in more details. Furthermore, we generate additional conceptual models to understand the system's reliability. Moreover, we analyze environmental data, mainly weather data at this extended version to investigate the environmental factors and to form an explanation and correlation between failure events and other environmental factors. The SOI for the conference and this paper is an Automated Parking System (APS).

The paper is structured as follows: Section 2 lists related studies in the form of an informal literature review. Section 3 illustrates the study's research method. Section 4 shows results from data analysis and conceptual modeling. The results of the data analysis include failure and weather data analyses. Section 5 provides a thorough discussion, and ultimately, Section 6 wraps up the study with conclusions.

2. Literature Review

2.1. Reliability Engineering

Reliability engineering is vital in systems engineering and quality management as it minimizes a system's failures [8]. O'Connor and Kleyner [16] define reliability as "The probability that an item will perform a required function without failure under stated conditions for a stated period of time." This conventional definition of reliability emphasizes two aspects (scientific topics): statistics and engineering. The word "probability" emphasizes the mathematics and statistics aspects, whereas the phrase "required function, stated conditions, and period of time" emphasizes the engineering aspect.

Developing a reliability program plan is crucial within the engineering aspect. This plan includes activities for reliability engineering and aids in determining which activities to perform. These activities depend on system complexity, life cycle stage, failure impact, and so forth. The systems engineering handbook includes guidelines for developing a reliability program plan [17]. There are also reliability program standards, such as ANSI/GEIA-STD-0009. This latter standard supports the system lifecycle for reliability engineering. It includes three crucial elements: (1) understanding system-level operations and environmental and resulting loads and understanding the stresses occurring throughout the structure of the system; (2) the identification of the failure modes and mechanisms; and (3) the mitigation of surfaced failure modes [18].

There are also two types of reliability engineering: proactive and reactive. Proactive reliability engineering occurs during design and development, emphasizing failure prevention. In contrast, reactive reliability engineering occurs during production, especially during maintenance and operations, emphasizing failure management [7,19]. Figure 2 visualizes these two types of reliability engineering: proactive and reactive reliability engineering.
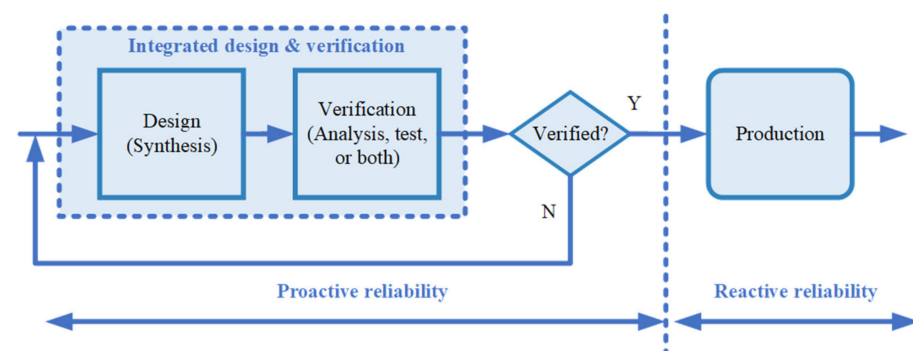

Figure 2. Proactive and reactive reliability engineering within the product lifecycle, including the Product Development (PD) process, redrawn from [7].

Figure 2 also depicts the development process. This process is an iterative one. It indicates that reliability engineering is also iterative in its nature. Systems engineering integrates reliability within the processes. Verification occurs before production proceeds and consists mainly of analysis, testing, inspection, and demonstration [7].

Shortly after reliability engineering was founded in 1957, Hollis stated a need for conventional statistical reliability. However, Hollis also emphasized that the statistical aspect is insufficient. There is also a need for other reliability activities or techniques to improve the feasibility and robustness of the designed system(s). In other words, statistical and engineering aspects complement each other [20,21]. Reliability influences other dependability attributes of the system, such as maintainability and availability. Thus, reliability affects return on investment, market share, and competitiveness. Therefore, a reliable system provides a competitive advantage and increases market shares by having a proven field design (good system reputation).

One of the statistical methods within reliability engineering involves determining the MTBF. The MTBF is the reciprocal of the failure rate: MTBF = (total system(s) operation time)/(total number of failures). The MTBF is widely used to measure a system's reliability by showing how long it operates before it requires maintenance [22,23].

2.2. The Knowledge Framework: Data, Information, Knowledge, and Wisdom

The main reason for data analyses is to make sense of the data using a two-way learning process. We use conceptual modeling mainly for this learning and understanding. The data analysis includes pre-processing, analyzing, and visualizing the results of the data analysis. One of the primary works of the literature regarding analyzing data involves the DIKW hierarchy—the data, information, knowledge, and wisdom. This hierarchy is a widely recognized concept that constructs our knowledge management model. Figure 3 illustrates four layers in the DIKW hierarchy and conceptualizes the knowledge management model. Unhelkar [24] states that this model is decisive in efficiently managing (big) data to make it exploitable in decision making, including the PD and maintenance process.
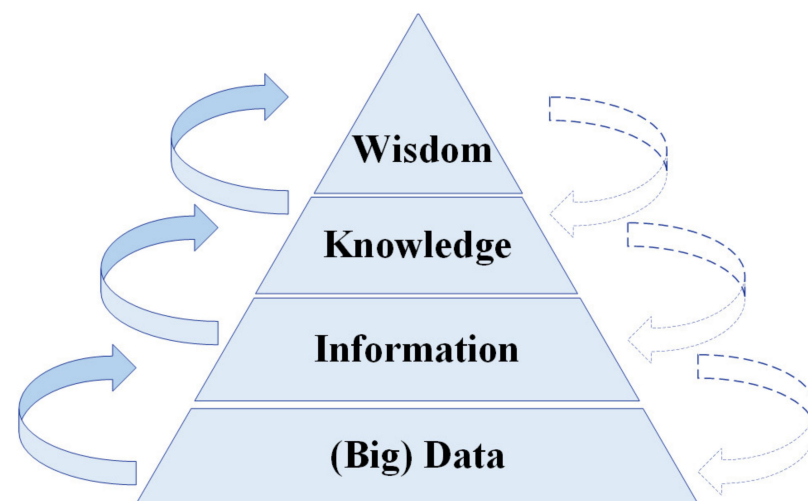

Figure 3. The DIKW hierarchy, from observation to wisdom, is based on [24–26].

The DIKW hierarchy or framework starts with the (big) data layer. These data vary from structured data (data organized in rows and columns) to unstructured data (such as images and text). Information usually answers the "who", "what", "where", and "when" questions. Information adds context to the raw data by understanding the relations. We obtain this context by analyzing the data. This analysis includes the visualization of the analysis's results using figures.

Knowledge answers the "how" question by understanding the information and its application. This understanding is also called data sensemaking. Knowledge goes be-

yond information by understanding the (hidden) patters. In other words, knowledge is information combined with judgment and experience. In this context, we use conceptual modeling to transform information into knowledge by investigating the "how" question for the anomalies, and this is visualized by information. Wisdom is the tip of the hierarchy and is about implementing knowledge within the system of interest and its context. In other words, wisdom tells us "when" to use the data and "why" the specific data are used, and it builds on the former layers, i.e., information and knowledge, by understanding principles [25,26]. We also use conceptual modeling to obtain wisdom by understanding the data's context (information) and how to use the data (knowledge). Moreover, conceptual modeling covers the gap between knowledge and wisdom, which is understanding. We obtain this understanding using conceptual modeling, which aids in appreciating the "why" and "when" in using the data by evaluating this understanding.

We also added stipple arrows going the other way, from wisdom to data. We transform wisdom into knowledge by identifying which implementation we want to increase our knowledge. Ali et al. [15] illustrate an example of transforming wisdom into knowledge by evaluating the value proposition of implementing a CBM system. Furthermore, we can transform knowledge into information by presenting knowledge in certain forms. This presentation is information, and this information can be further transformed into data. These data are also derived from the information or can be influenced by the knowledge process. This influence can include identifying and collecting available data. Ali et al. [27] show how we can transfer tacit knowledge into explicit knowledge in terms of data and visualization (information) using systems engineering and systems thinking.

Figure 4 shows that knowledge affects the ways we collect or create data. Data also affect knowledge. We transform data into information when we analyze and visualize the data analysis results. Furthermore, we transform information into knowledge. The latter transformation occurs by a two-way learning process. This learning process takes place by using conceptual modeling. This process triggers more issues that we aim to know more about or explore. To explore the needed issues, we collect or create the appropriate data that aid in such explorations. The entire process complies with establishing an "information lifecycle" [28,29].
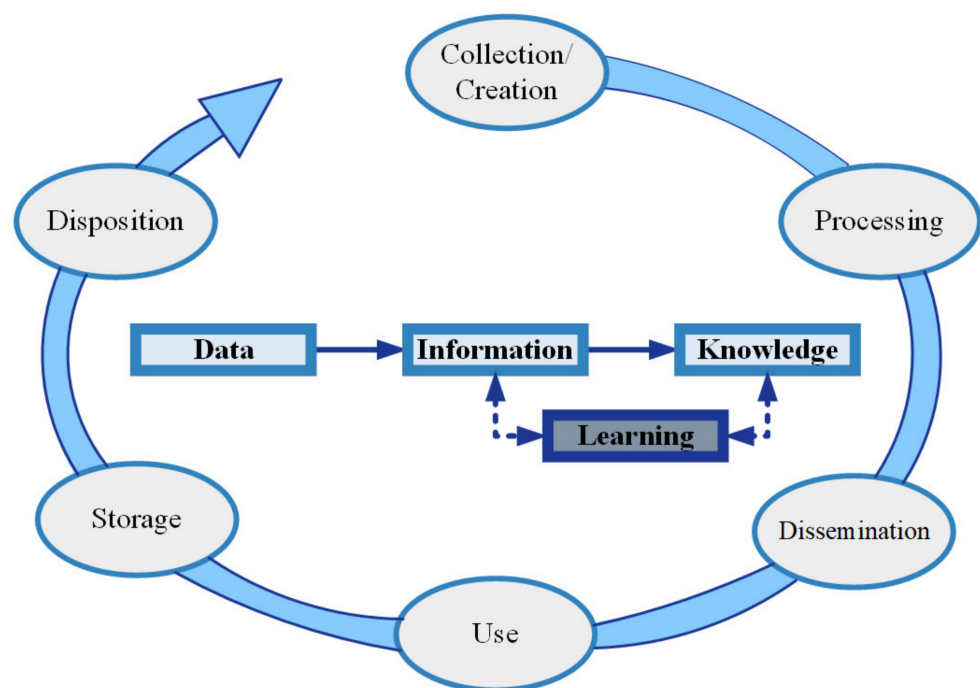

Figure 4. The process-based perspective of data, information, knowledge, and learning based on [28,29].

Learning as a concept is vital, especially for the decision-making process. Birkland [30] defines the learning process as "participants using information and knowledge to develop, test, and refine their beliefs." Leaning is essential for low probability and high consequence events, especially for shorter and rapid decision-making processes, i.e., deciding something as quickly as possible [30,31].

2.3. Conceptual Modelling

Ramos et al. [32] mention modeling as proceeding "from brain representations to computer simulations, the models, are pervasive in the modern world, being the foundation of systems' development and operation" [33]. However, conceptual models have several definitions and origins. These origins include physics, simulations, tools for co-creation or collaborative sessions, and tools that aid conceptual design in systems engineering. The common aspect among these origins is that conceptual models aid in sharing and communicating common understanding and ways of thinking.

Co-creation sessions use conceptual models in several areas, such as design thinking, by focusing on human interactions [34]. Gigamapping, from system-oriented design, enhances communication and relates strongly to design thinking in its style [35]. Neely et al. [36] suggest a workshop for co-creation sessions for interdisciplinary teams.

The scientific simulation field includes several definitions of conceptual models. Sargent [37] defines conceptual models as follows: "the mathematical/logical/graphical representation (mimic) of the problem entity developed for a particular study." Many other authors link the use of conceptual models to simulation, such as [38–40]. Robinson states that conceptual models are an essential aspect of simulation projects and mentions that conceptual models are more art than science [40,41].

Gorod et al. [42] emphasize the importance of the link between high-level conceptual representation and low-level analytics using visualization and human ability. Furthermore, Gorod et al. [42] state that conceptual models aid in capturing higher-level descriptive or textual models of the problem domain. This problem can be decomposed into lower sets of measures. These measures can be evaluated analytically [33]. On the other hand, Lavi et al. [43] mention that the conceptual model is a result of a system's presentation. The conceptual model helps design the system by providing a shared representation of the system's architecture, helping to deal with complex knowledge and resolving any ambiguities or conflicts.

Systems engineering uses various types of conceptual models. This variety results from interdisciplinary engineering, wherein each engineering field uses its domain-specific conceptual models [44]. For instance, Blanchard [45] uses several variations of conceptual models. Tomita et al. [46] consider conceptual system designs and suggest using systems thinking applications to raise the consideration from the data and information level to the knowledge and wisdom level. Montevechi and Friend [39] proposed using a Soft Systems Methodology (SSM) to develop conceptual models. Using the SSM in this case study, part of a complex sociotechnical research project makes sense; this research project aims to bridge soft aspects (including knowledge and wisdom) with hard aspects (data and information). Systems thinking plays a vital role in the development of conceptual models. For instance, Jackson [47] connects systems thinking to label complex problems. In another example, Sauser et al. [48] apply systems thinking to define a complex problem within a case study.

Muller's work illustrates bridging conceptual models with the first principle and empirical models [49]. Empirical models aid in expressing what we measure and observe without the necessity of understanding what we observe. First-principle models use theoretical science principles. These models often involve mathematical formulas and equations. Conceptual models use a selection of first principles that explain measurements and observations. Conceptual models are a combination of empirical and first-principle models. This paper emphasizes that conceptual models need to be simple enough to reason and understand the case study. Simultaneously, conceptual models need to be realistic enough to make sense. This latter description of conceptual models emphasizes the need

to balance between the simple, making sense, and practical aspects when developing conceptual models.

Muller [50] mentioned some core principles, objectives, and recommendations in applying conceptual modeling. Despite the slight modifications we made, Figure 5 visualizes these principles, objectives, correlations, and their relation between each other. In the Results section (ref. Section 4.1.6), we describe how we used these principles, objectives, and recommendations in detail.
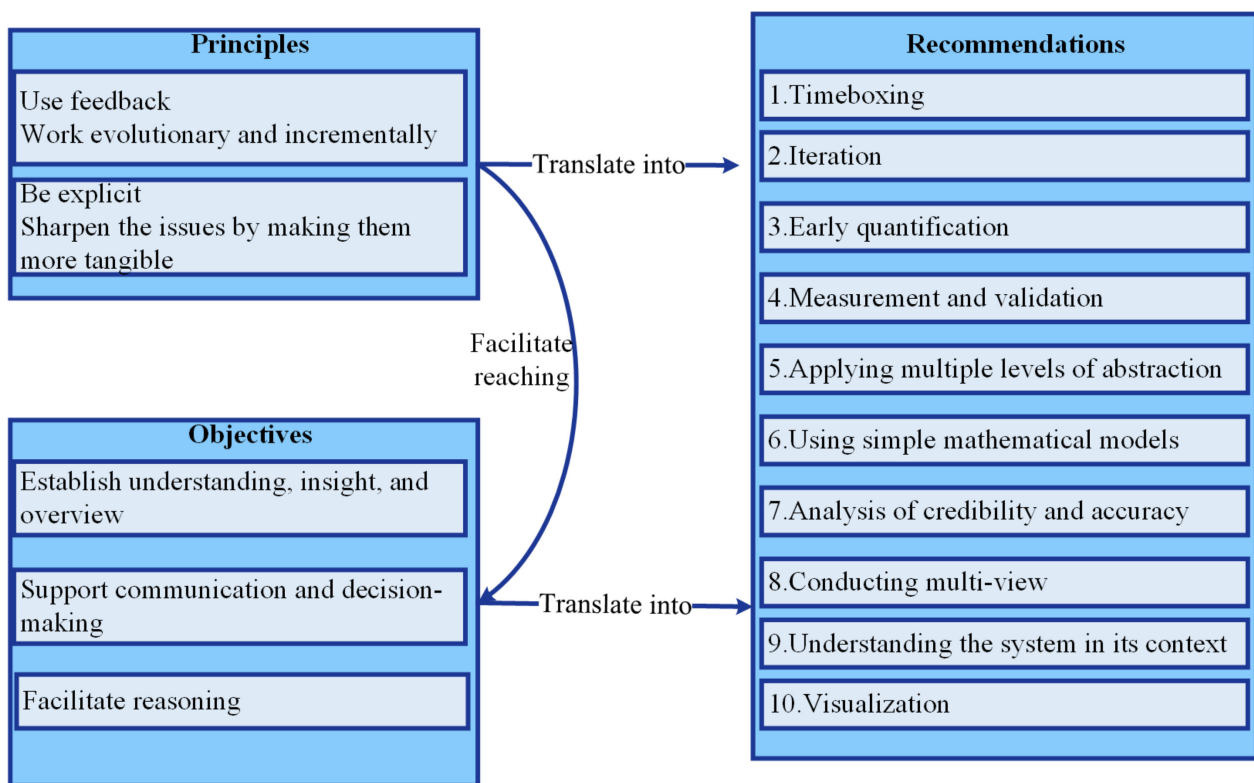

Figure 5. Recommendations, accompanied with its principles and objectives, modified from [50].

3. Methods

Figure 6 shows this study's research methodology [51]. Figure 6 illustrates the case study in the middle. The case study has its context and embedded unit of analysis [52]. We applied several iterations from the problem domain to the solution domain, applying the adapted action research cycle: Design/Plan, Test/Act, Observe, and Analyze/Reflect [53]. These iterations are from the problem domain to the solution domain. On the left of the case study, the literature review and matter-of-experts' input are the inputs for the case study. These inputs supported the study's research questions. On the right of the case study, we obtain the output, which is a new process or methodology for using conceptual modeling and data analysis to enhance product developers' understanding. These data consist of internal and external data. The internal is unstructured stored failure data within the company, while external data investigate environmental factors, mainly weather data, to explore patterns and trends with the failure events.
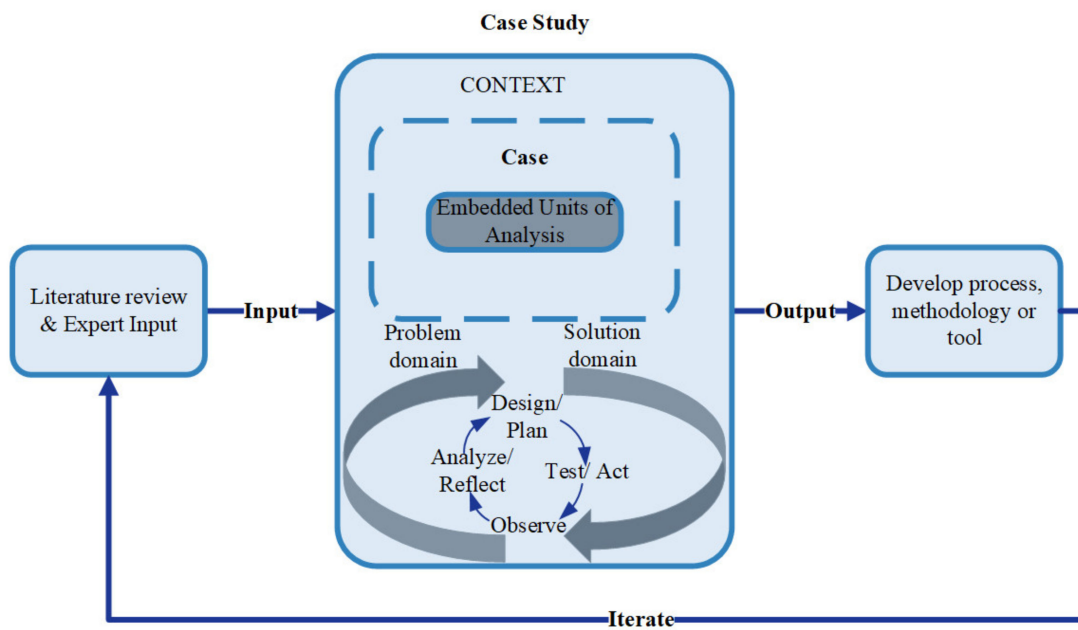
Figure 6. Case study's research methodology.

We iterated this study's research methodology process at least three times, inspired by action research [54]. We conducted these iterations after accepting the risk of concerns and limitations of the output in the form of a methodology we explain in Figure 7. These concerns and limitations include limited access to the all-embedded units of the analysis of the case study, the lack of in-depth knowledge of the context or company or both, and the lack of in-system data for the same period of failure data as it was not technically possible to collect them. The majority of these concerns and limitations may refer to one main reason: We are not employees in the company. However, we had an office in the company of research.
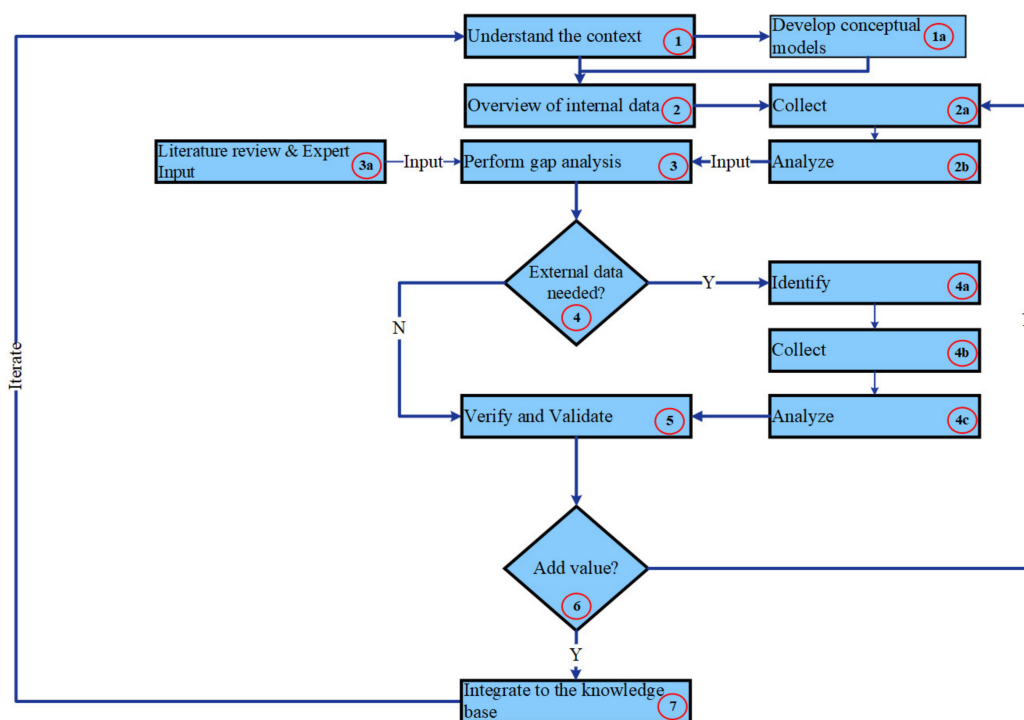

Figure 7. The performed steps visualize the research approach.

Since we investigate sociotechnical systems within this research, we needed to explore the research's technical and social aspects. We covered the technical aspects by examining the study within the Design, Test, Observe, and Analyze dimensions; in contrast, we covered the social aspects by Act and Reflect. Furthermore, we developed a process, or a methodology (output), based on these iterations. Figure 7 visualizes this process or methodology by using a workflow. This workflow illustrates the research approach by explaining the steps conducted in this study. Figure 7 portrays these steps as red circles containing their numbers. These steps are:

1. Understand the context. We started by understanding the context. This understanding includes an understanding of the SOI and its life cycle, including the PD process, focusing on the early phase. This step includes the following sub-step:

    a. Develop conceptual models. In this step (1a), we developed conceptual models to communicate the context understanding and validate it at the first iteration using interviews and workshops. We developed other conceptual models in successive iterations based on the results of the data analysis. These conceptual models also guided the data analysis in an iterative and recursive manner. In other words, we used conceptual modeling as an input for the data analysis and vice versa. We conducted all processes in several iterations until conceptual modeling and data analysis were harmonized. For instance, we developed conceptual modeling such as end-user's workflow and maintenance process functional analyses to achieve a shared understanding of the context, including a system's reliability and its consequences in terms of time and cost. The focus on systems' reliability is a result of the data analysis, which explored inferior system reliability.

2. Overview of internal data. In this step, we aimed to have an overview of the stored internal data (physical artifacts) within the company. We identified all possible data sources within the PD process. This identification results from developing conceptual models alongside the research data collection. The data sources we identified include failure data, design data, and in-system data. This step includes the following sub-steps:

    a. Collect. In this step (2a), we collected the available stored internal data. In this context, we collected the failure data as feedback data.

    b. Analyze. In this step (2b), we analyzed the collected internal data, mainly failure. The data analysis's results are inputs for the gap analysis.

3. Perform gap analysis. In this step, we performed a gap analysis. The gap we identified is the need for feedback data in the early phase of the PD process, such as failure data. This step includes the following sup-step:

    a. Literature review and expert input. The failure data analysis results, alongside the input from the literature review and subject-matter expert are the inputs for the gap analysis. The experts include practitioners from the industry and domain scholars from academia.

4. Decision gate: External data needed? After analyzing failure data (internal data), we have a decision gate to decide if we need external data to investigate patterns and trends or form an explanation with failure events. The decision was yes at this step. Thus, we performed the following sup-steps:

    a. Identify. In this step (4a), we identified a need for weather data based on internal data analysis's results and subject-matter experts from the company and domain scholars.

    b. Collect. In this step (4b), we collected weather data for the same period for the failure data analysis and for the cities where the parking systems installations exist.

    c. Analyze. In this step (4c), we analyzed weather data and observed possible trends and patterns or any explanation. In this context, conceptual modelling aided in understanding possible explanations with the failure events.

5. Verify and validate. The other direction for the decision gate is to proceed with verification and validation of the internal data analysis in case the need for external data is not identified. We verify internal data analysis using the literature and participant observations. Otherwise, external data as another data source were used to verify the internal data analysis's results. We conducted the validation using, as mentioned, workshops and interviews within the company. We conducted these workshops physically and digitally. The participants of these workshops included company management, the maintenance head of the department, maintenance personnel, project leader, and engineers involved in the early phase of the PD process.

6. Decision gate: Add value. Based on the interviews and workshops, we could evaluate if the finding, including the data analysis's results accompanied by conceptual models, added value to the company in terms of whether it makes sense or not. However, we have one more decision gate after verification and validation. In this decision gate, we need to decide if there is no value added, and then we need to return and collect more internal data stored in the company for further analysis. We conducted some iterations and collected some more internal data such as more failure data for an extended period and some maintenance cost data. We also discovered that pre-processing the data and understanding it were also vital for achieving value.

7. Integrate to the knowledge base. After obtaining feedback and verifying and validating the value, we aim to integrate this value into the knowledge base. This step is still a work in progress as this integration depends on the company's working processes. One of the suggestions we plan to use is A3 Architecture Overviews (A3AO) for the findings. Another suggestion is to integrate the findings as a parameter within the engineering activities in the PD process's early phase. These activities could, for example, be the Failure Modes and Effects Analysis (FMEA).

Even though we visualize the research method's steps as a linear and rigid sequential step, we conducted these steps using several iterations and time boxing. Time boxing varies in time from minutes to days to weeks. The iterations, alongside time boxing, aimed at enhancing the data analysis and conceptual modeling results by using different time boxes for each step.

However, we also recommend iterating the entire process or methodology, including other available stored data other than the analyzed one, that is, failure data in this context.

4. Results

The Results section starts with the case study's subsection. This subsection includes a description of the SOI and failure data description. The Results section continues with failure data analysis and then with weather data descriptions followed by weather data analyses. The results section concludes with a conceptual modeling subsection. The conceptual modeling subsection includes descriptions and visualizations of the most significant conceptual models that we generated for this case study, alongside the principles, objectives, and recommendations we used for developing these models.

4.1. Case Study

In this case, study, we exemplify the use of conceptual modeling and data analysis in an iterative process. Data analyses guided conceptual modeling, while conceptual modeling generates more questions than answers. We used data analysis to answer these questions and increase the understanding of the system's context. Conceptual modeling facilitated the product developer's understanding and communication of the system and its reliability.

4.1.1. Description of the System

Figure 8 portrays the SOI: the semi-automated parking system and its configuration. Figure 8 visualizes a drive-in indication. The figure also visualizes a variety of SOI configurations. These configurations vary from $4 \times 2 \times 2$ to $9 \times 1 \times 3$ and $11 \times 2 \times 3$. The first

number is the width, the second is the depth, and the third is the height. Depending on the building architecture, the car entrance can be a straightforward or an inclined plane, as with the SOI's configurations.
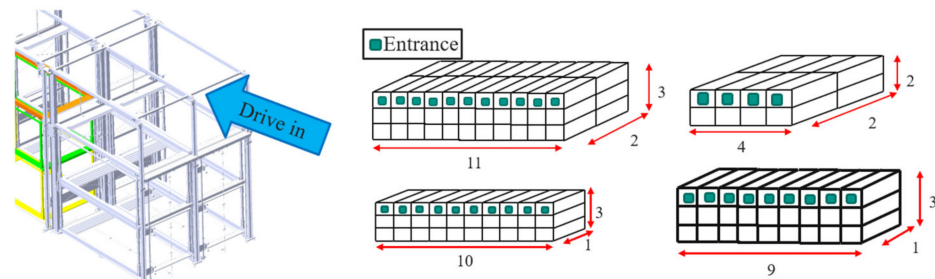

Figure 8. The system-of-interest (SOI): semi-automated parking system (**left**) and its configurations (**right**).

The SOI consists of numerous hardware parts. We mention here some of the main parts to better understand the SOI. We visualize these parts in Figure 9, which contains the following:

- The gate: The gate is the entrance and exit before and after parking. A collection of two up to three gates is called a segment. Figure 8 shows a segment including two gates.
- The control unit: The control unit is a touch screen with operating instructions, including a key switch and emergency stop. The control is connected to the power unit through cables and fixed on a wall.
- The platform: The platform carries the car to the correct position.
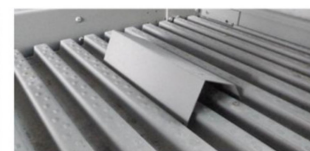- The wedge: The wedge helps the driver park the car in the correct position.

Gate

Control unit   Platform   Wedge

Figure 9. The System-of-Interest (SOI): main parts.

4.1.2. Failure Data Description

This subsection illustrates the failure data analysis and its results. The failure data, in the form of maintenance record data, are unstructured. The maintenance personnel manually log failure events using Excel. In the Excel file, there are several sheets. Each sheet belongs to a specific semi-automated car parking garage system. Each sheet's content includes the following columns, which are also called parameters:

- Date (for a maintenance event);
- Time;
- Telephone number (for the maintenance personnel who investigated the failure event);
- Tag number (for the user (car owner));
- Place number (for which parking lot the failure event occurred);
- Reason (possible reasons for the failure event);
- Conducted by (includes initials of the maintenance person's name that performed the failure or maintenance event);
- Invoiced yes/no (if the failure event is invoiced as it is not included within the maintenance agreement with the company or not). The company's agreement includes only planned maintenance. In other words, most maintenance tasks should be invoiced. However, the company rarely invoices its customers.

The main parameter within the failure data is the description of the failure event, which is the reason why the parameter is mentioned above. The description is a free-format text that can include, among other, data about what part of the SOI failed, possible reasons, and maintenance actions to fix the failure event. As it is a free format text, it can include one of the mentioned or more as it lacks a guideline to enters this description. This inclusion depends on the maintenance personnel and their time availability. We collected data for private and public parking systems for the period 2016–2021. The amount of failure data for these two types is around 2000 rows times 8 columns.

4.1.3. Failure Data Analysis

Since the maintenance personnel manually log the failure data and include descriptions, especially the failure events, we employed an introductory Natural Language Processing (NLP) method to analyze them [55]. NLP is a well-known method for analyzing text entry fields, such as the description field within the company's failure data. Manual pre-processing is necessary for data analyses and data quality. We divided this subsection into failure data collection, failure data pre-processing, failure data analysis, and failure data visualization.

Failure Data Collection

We collected the failure data from the company. The company's employees download this data in terms of an Excel file. The file is a living document. First, we obtained part of the data for a short period of time to conduct the first iteration of the data analysis. Furthermore, we collected all stored data for the period 2016–2021 and conducted additional data analysis iterations as the first one added value according to the matter-of-experts from the company and domain scholars.

Failure Data Pre-Processing

We conducted the pre-processing iteratively and recursively. For instance, we started by generating a template including the most significant parameters. We identified these significant parameters based on state-of-the-art input and matter-of-experts from the company and scholars. Furthermore, we collected the data from different sheets since each sheet belongs to a parking system. We conducted this process manually.

Additionally, we used a code to generate this template using manual pre-processing as an input. We discovered several issues using the code or automation to pre-process the data since the maintenance personnel manually logs the data. These issues include but are not limited to empty data cells within the Excel rows, a different format for the parameters

such as time and date, and a different arrangement for the parameters (columns) within the different sheets. We fixed these issues manually and automatically depending on the nature of the issue and possible capabilities for pre-processing it. In addition, we added two columns (parameters) to this template: parking system name and parking system type, i.e., private or public, based on the criteria we determined before in the Introduction section. We went back and forth during several iterations between the data points (parameters) to the template and vice versa until the data and its template came into unity. The pre-processing consumed approximately 80% of the data analysis time, i.e., about one year.

Furthermore, we conducted a pre-processing of the text-free format that describes the failure event as part of the NLP method. This pre-processing step was conducted after we imported the data from the generated template as the data framework for a specific parking system type, i.e., private or public data. The pre-processing step involves the following sub-steps:

- Tokenization: We divided the text (sentences) into words, commas, and so forth, called tokens.
- Removing the numbers: We removed numbers that stick to the word without a full stop.
- Removing stop words: We removed stop words. Stop words are words that do not add a significant meaning in the natural language, such as is, the, this, and so forth. We removed the stop words using the "corpus" module from the Natural Language Toolkit (NLTK) Python library [56].
- Stemming: Stemming means that we reduced the words to their root by removing the word's prefixes and suffixes that affixes its root, also called a lemma. We used a stemming algorithm called Snowball Stemmer from Python's NLTK library [57]. We also lower casing the letters for all text.
- Translation: We translated the results from Norwegian to English. We used Google-trans Python library, which uses Google Translate API (Application Programming Interface) [58].

Failure Data Analysis

We conducted the data analysis by using several sub-steps:

- Tokens (words) frequency: We determined the most common words. We calculated the frequency of each word in the free text (reason column) in number and percentage. We chose to save the ten most repetitive (frequent) words. We also determined the frequency of the unique words. We calculated the frequency of the word per row in number and percentage. Each row represents a failure event. In addition, we conducted an n-gram analysis showing the most frequent three words, four words, and five words. We noticed that the three-word phrase made the most sense in a way that provided an understandable phrase. We chose to show only the most repetitive word in the failure events as the point in this context is to investigate the most critical subsystem.
- Failure events frequency versus other columns (parameters): We calculated the frequency of the failure events versus other parameters. These parameters include the year, month, day of the week, and time of day (hour). In this calculation, we saved the numbers and percentages. The calculation of percentages was performed due to the company's confidentiality.
- Failure rates: We calculated the failure rates by determining the number of failure events divided by the total system operation time.
- Mean Time between Failures (MTBF): We determined the MTBF by taking the reciprocal of the failure rate values (total operation time/number of failures).

Failure Data Visualization

We visualized the data analysis's results by using several methods for visualization. These methods include, among others, bar charts and dot charts. We show the most significant visualizations results based on the data analysis's results.

Figure 10 visualizes the data analysis results for the public and private parking systems. The figure shows the most repeated words in percentage. We observe that the "gate" is the most repeated word in the public parking systems, with approximately 80% of roughly 2000 failure events. The "gate" is also the most repeated word, with around 50% of the failure data for the private parking systems. However, when we manually examined those failures, we observed that the maintenance personnel used different terms, such as "system" or "segment" for the gate. Part of these terms, especially "system," is used by the maintenance personnel for many things other than the gate. These things include the entire parking system, the control unit, etc. This usage of different terms meaning one thing is a natural result of manually logging the description of the failure data as humans are complex and messy, and we use additional terms all the time for the same thing.
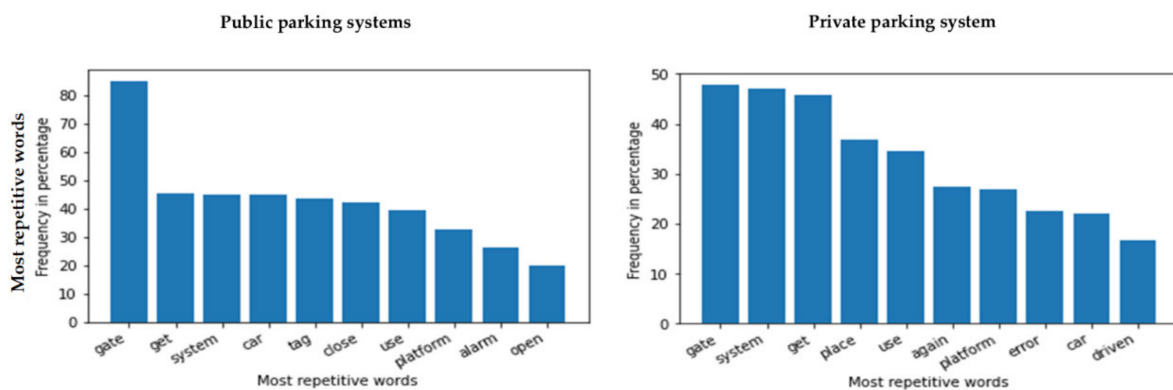

Figure 10. The most repetitive words in percentage in the entire description field entry (reason column) in failure data for public (**left**) and private (**right**) parking systems.

Figure 11 depicts the MTBF for the public (left) and private (right) parking systems, which is the inverse of the failure rate. We calculated the MTBF per year for six years. The MTBF average is approximately 66 h for the public and 37 h for the private parking systems. We noticed that the MTBF for the public parking system is almost double that of the private parking systems. One of the reasons could be that the public does not have a permeant platform, whereas the private has a permeant platform for each end-user. However, we also observed that parts of the private parking systems lack proper documentation of failure data. This lack of documentation can be another reason for the different MTBF values. The MTBF values (66 h and 37 h) indicate that the SOI has an inferior system's reliability as it means that approximately every day and a half up to three days, there is an issue or failure within the semi-automated parking systems.
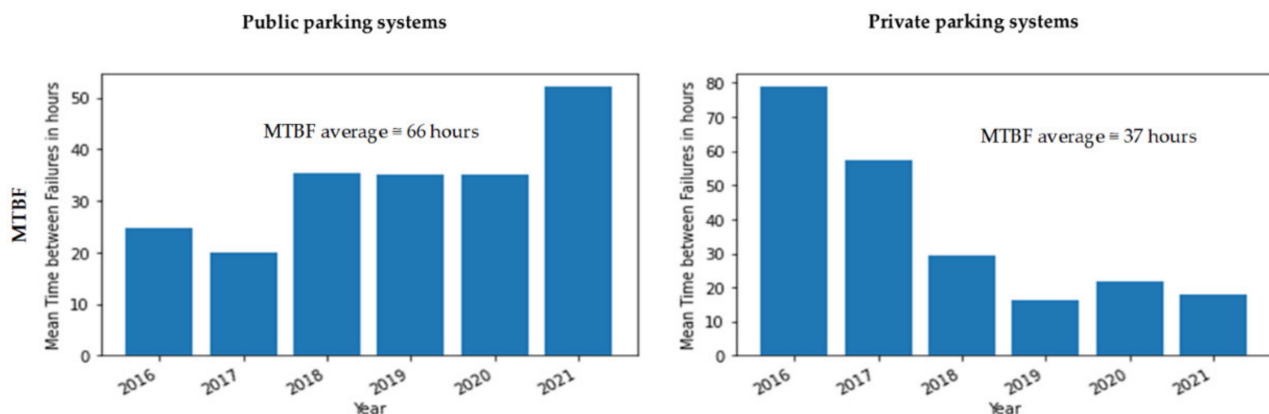

Figure 11. Mean Time between Failures (MTBF) in hours for all failures for public (**left**) and private (**right**) parking systems.

Figure 12 shows the month of the year for all failure events in public (left) and private (right) parking systems. Figure 13 portrays the time of day (hour) for all failure events in public (left) and private (right) parking systems. We note from Figure 12 that failure events decrease dramatically in July, which is a summer vacation month. In other words, in July, parking systems are not used as much as in other months of the year. We notice from Figure 13 that most failure events occur at 07:00 and 15 to 18:00 h, which are rush hours for both public and private parking systems. We conclude that the parking systems fail most when it is used more, which makes sense. This latter conclusion concurs with the state-of-the-art observations [5,6].
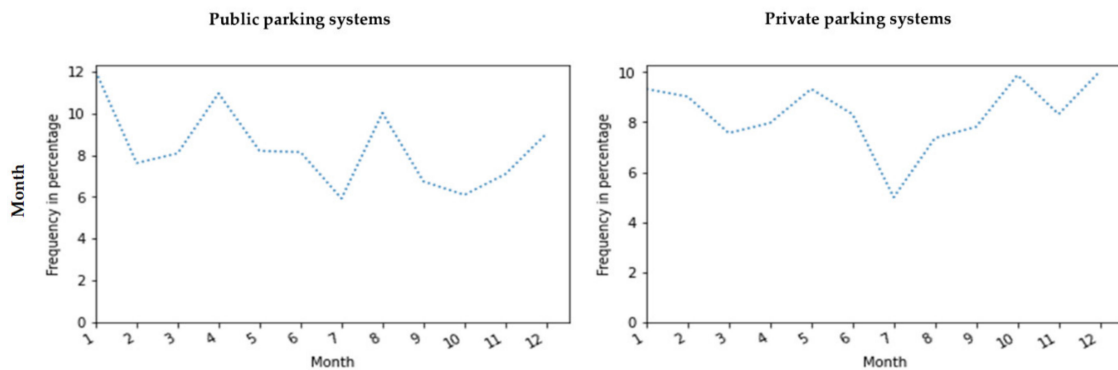

Figure 12. The month of the year for all failure events in public (**left**) and private (**right**) parking systems.
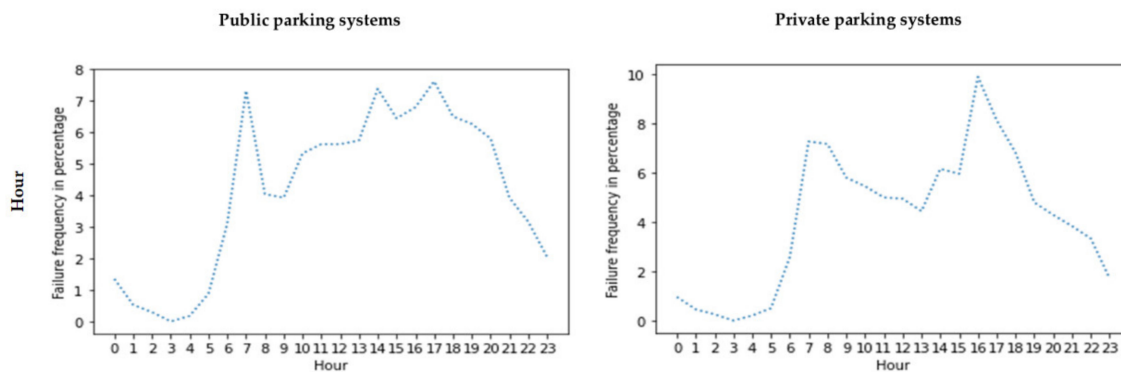

Figure 13. Time of day (hour) for all failure events in public (**left**) and private (**right**) parking systems.

Gate as a Use Case

We dug deeper into the most frequently mentioned part: the gate. Figure 14 depicts the average MTBF in hours for all failures gates for public (left) and private (right) parking systems. We observe that the MTBF for the gate for the private parking systems is more than double (153 h) that of the public parking systems (66 h). This over-double value may refer to the number of private parking systems included in the data analysis, which is 17 private parking systems versus one public parking system. Another reason could be the lack of documenting all failure events in private parking systems, as mentioned before.
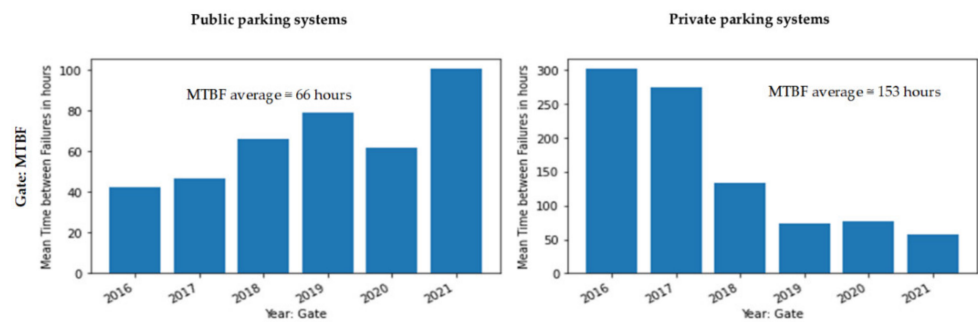
Figure 14. Mean time between failures (MTBF) in hours for all gate failures for public (**left**) and private (**right**) parking systems.

Moreover, we performed and determined the same analysis for the gate as the parking systems as a whole. Figure 15 visualizes the month of the year for all gate failure events in public (left) and private (right) parking systems. Figure 16 shows the time of day (hour) for all gate failure events in public (left) and private (right) parking systems. We have the same conclusions for the gate failure data analysis as for the parking systems failure. These conclusions also include that the gate failures occur mostly at 07:00 and 15 to 18:00 h, which are the rush hours. In addition, the gate failures decrease in July when it is vacation time. In other words, the gate fails the most at when there is high usage for the parking system.
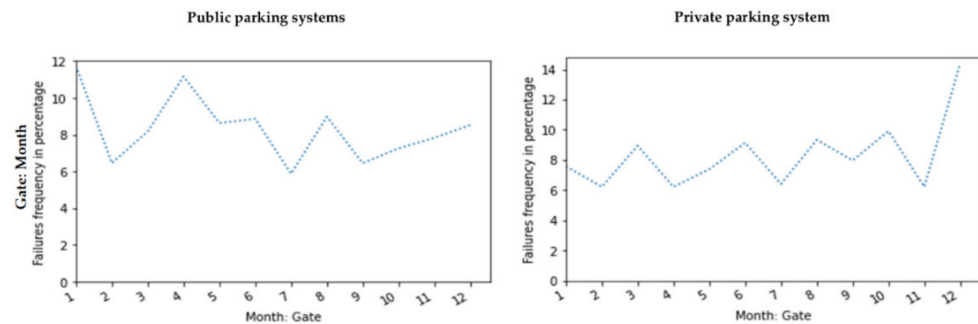

Figure 15. The month of the year for all gate failure events in public (**left**) and private (**right**) parking systems.
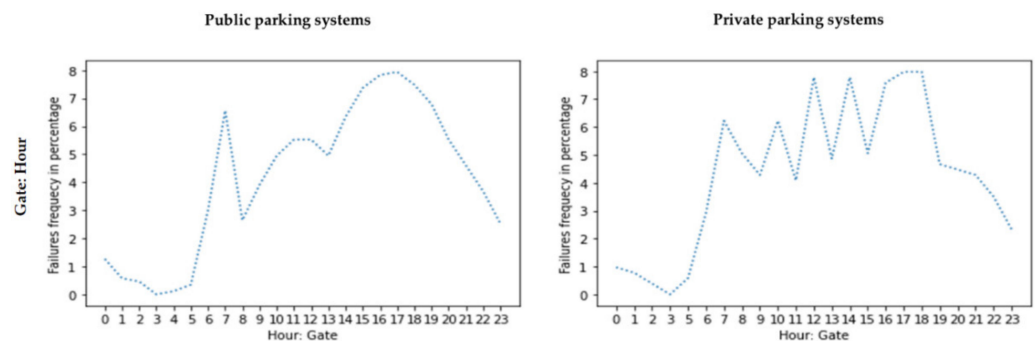

Figure 16. Time of day (hour) for all gate failure events in public (**left**) and private (**right**) parking systems.

We used Python to analyze the failure data. We started by finding the most repetitive word in the description's text entry field. We calculated the MTBF for public and private parking systems to compare the results. Moreover, we analyzed the failure events that occurred relative to the month and the time of the day to explore patterns and trends among the failure events. Furthermore, we compared and linked the "most failed" part of the SOI with the other parameters (columns) and conducted the same analysis and calcula-

tions, such as failure rate and MTBF. Then, we discussed the results with the company to determine whether the data analysis and results made sense.

Failure Data Classification

We classified the public parking systems data using semi-supervised machine learning (since some of the data are labeled and some are not labeled data). We used the Markov model for this machine learning process [59]. We manually labeled (classified) a portion of the dataset. This classification includes User Failure, Software Product Failure, and Mechanical Product Failure. The dataset has 3910 samples in total, and 223 are labeled manually; the rest of the samples, i.e., 3687, are unlabeled and labeled by trained models. Throughout the classification, 80% of the labeled data are used for training the machine learning model, and 20% of the labeled data are used to test the trained models. Moreover, the dataset is utilized as one feature (a sentence: Sentence Model) and many features (word array for every sample: Word Model). Thus, two separate models are developed. Dividing samples into words reduces the performance. Thus, the Sentence Model provided better accuracy (0.666) than the Word Model (0.644). Figure 17 visualizes the failure data classification's results using supervised machine learning. We refer to Mechanical Product Failure as hardware, Software Product Failure as software, and User Failure as Human–Machine Interface issues (HMI). We notice that the hardware constitutes approximately 41% of the failures, software at 29%, and HMI at 30%.

Hardware Count: 1502 (41%)	Software Count: 1084 (29%)	Human-machine interface (HMI) Count: 1101 (30%)

Figure 17. Failure data classification using machine learning.

4.1.4. Weather Data Description

We collected the weather data for the public and private parking systems. We determined the private parking systems allocated within 30 km of Oslo to increase the weather's data analysis accuracy and validity. We collected the weather data for the same failure data period from 2016 to 2021. The weather data included many parameters: location, date, time, temperature, precipitation, snow, windspeed, visibility, sunrise, sunset, humidity, condition, and description. After pre-processing the failure data, we also conducted the same process for the weather data. We added the weather data's parameters to the same template that included failure data. The weather data pre-processing consumes less effort since it is more structured than failure data. However, this pre-processing does not cover the fact that the pre-processing still consumes approximately 80% of the data analysis period.

4.1.5. Weather Data Analysis

The private parking systems that are allocated within the 30 km radius of Oslo city have the same weather parameters as Oslo city, while the one public parking system is allocated in another city. Thus, we collected the weather data for this city. We refer to these two cities as city for private and city for public parking systems further in the data analysis.

Figure 18 shows the temperature on the y-axis and the date for the two cities where the private (left) and public (right) parking system is allocated. At the same time, the figure visualizes when it is snowing at which temperature in these two cities. We observe that there is mostly snow when the temperature is within −10 to 5 Celsius. We also noticed that it is snowing more in the city for private parking systems (ref. more blue dots left than right in Figure 18).
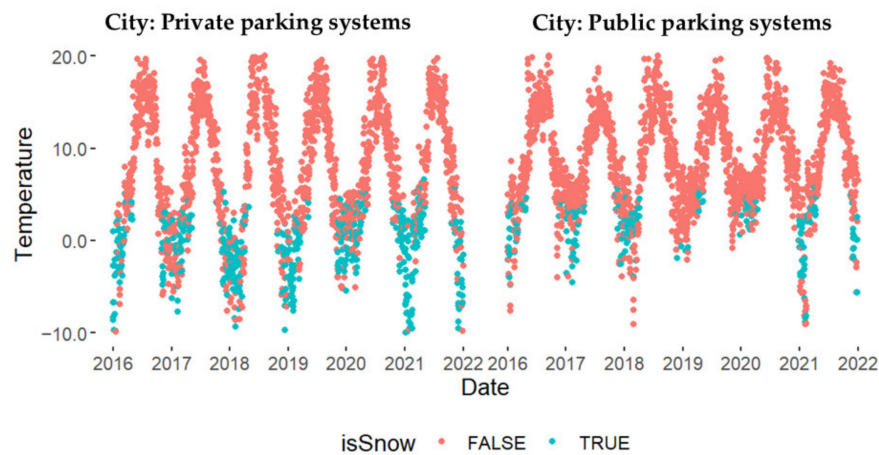
Figure 18. Weather data visualization (temperature and snow) for city private parking systems (**left**) and public parking systems (**right**).

The local government salts the roads when it is snowing and when the temperature is within 0 to 5 Celsius or when the temperature is between 5 and 10 Celsius. In addition to the snow and temperature, the local government salts the roads more when the humidity values are high in addition to the snow and temperature requirements [60]. An updated salt table can be found in [61]. The local government updated the requirements to salt the roads. This update mainly includes updating the temperature range to include minus degree ranges. The updated temperature ranges are above -3 Celsius, -3 to -6 Celsius, -6 to -12 Celsius, and under -12 Celsius

Figure 19 depicts the weather conditions for the two cities, i.e., city for private and city for public parking systems. The y-axis represents the number of days, while the x-axis shows the weather condition. We notice that the number of days it rains in the city for the public parking systems (above 1000 days) is higher than the days it rains for the city for the private parking systems (around 900). These days are determined from the weather data for the two cities from 2016 to 2021.
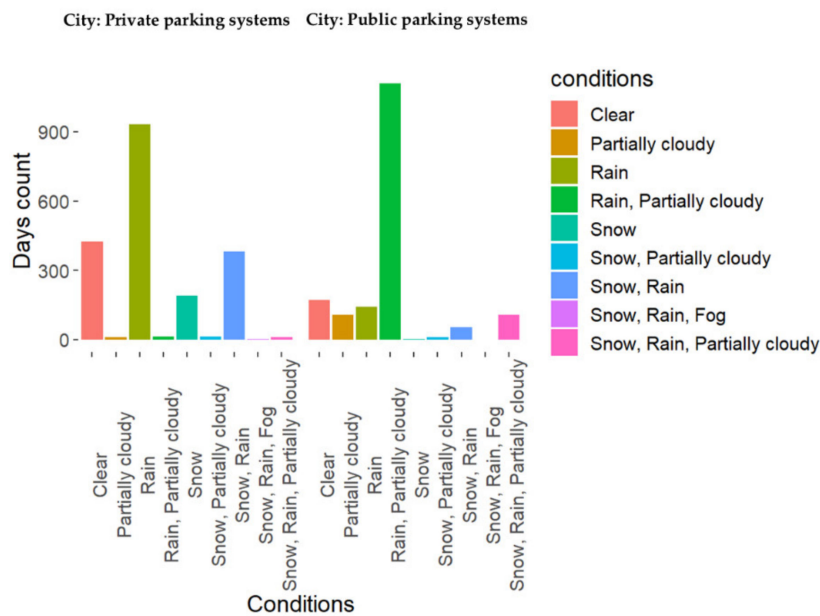

Figure 19. Weather data conditions for city for private parking systems (**left**) and for city for public parking systems (**right**).

Figure 20 portrays the temperature (x-axis) for the count of failure events (y-axis). We note that the failure event's peak is at around 0 Celsius for the city for the private (left), whereas the peak is at 5 Celsius for the city for the public (right).
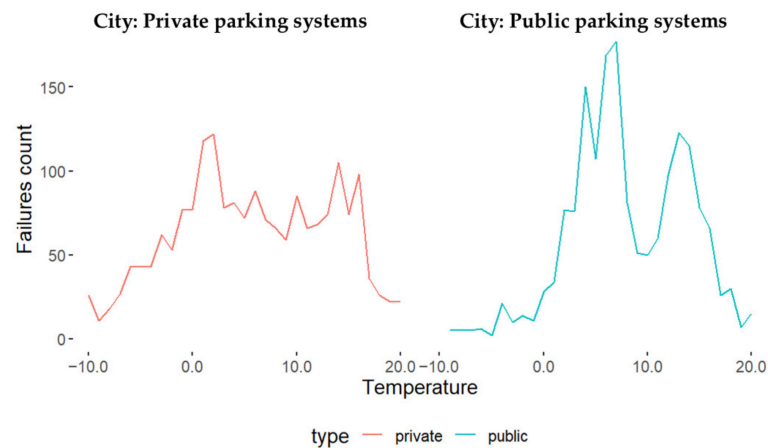

Figure 20. Temperature for failure events for private (**left**) and public (**right**) parking systems.

Figure 21 shows the humidity values (x-axis) versus the number of failure events (y-axis) for the city for the private parking systems (left) and public parking systems (right). We note that most failure events for private and public parking systems occurred when the humidity values were between 70 and 90.
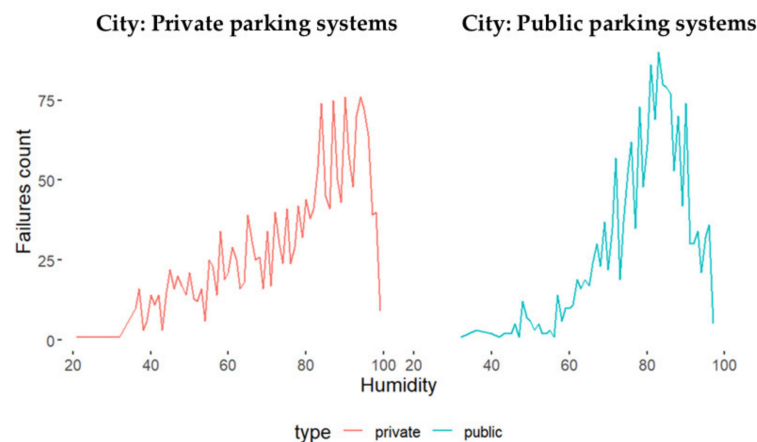

Figure 21. Humidity values for failure events for city for private parking systems (**left**) and city for public parking systems (**right**).

From Figure 21, we assumed a correlation between failure events and humidity. Thus, we calculated Pearson's correlation to figure out the correlation by assuming a normal data distribution. We found a positive correlation between humidity and failure events. This correlation is higher for private parking systems (80%) than the public ones (71%). Furthermore, we also ran a Pearson's correlation test for the precipitation weather data parameter inspired by Figure 19. We found a negative correlation between the precipitation and the failure events for private and public parking systems. The value for the negative correlation is −66% for the private and −62% for the public. This negative correlation is higher with its value for the public than the private one. This latter result makes sense as it rains more in the city for public parking systems. This rain washed the salt in the car's wheels and exposed less salt to the metal parts in the parking system. This salt causes the metal parts to rust. This rust increases failure events.

The weather data and failure data analyses form the following explanation for the failure events: (1) Humidity affects the failure events positively. In other words, increased

humidity leads to increased failure events, and (2) rain negatively affects the failure events: more rain, fewer failures. We also know that the local government salted roads. The cars' wheels take this salt to the SOI. This salt also falls into the tracks for the gates. These tracks and their relays are made (manufactured) of metal. The salt leads to rust in these parts. This rust increases the failure events as it affects the gate's functionality when opening or closing as it should. However, rain washes the salt away from the car and its wheels. Thus, we noticed that failure events have a higher value for the negative correlation for the public one than the private one. In addition, there is less cold and less snow in the city for the public parking system and, therefore, less salt on its roads.

However, we also noticed less values for the MTBF for the gate's failures for the public than the MTBF for the private. On the other contrary, we observe a higher MTBF for all failure events for the public than for the private ones. This contradiction may point to the lack of proper documentation of the failure events for the private parking systems versus the public ones. This lack of proper documentation may refer to the fact that the public has a local contractor that conducts maintenance and not the company itself. This contractor must document the failure events that are fixed in order be paid. Another reason to have less MTBF for gate failure for the public could be that the users are obtaining different platforms each time, rendering them less familiar with the parking systems. In contrast, the private ones always have a fast platform and a fast-trained user. Furthermore, the positive and negative correlation values for humidity and precipitation (rain), respectively, are around each other for public and private parking systems. This conclusion indicates that the results make sense mostly for all failure events for the two types of parking systems.

4.1.6. Conceptual Modeling
Principles, Objectives, and Recommendations in Applying Conceptual Modeling

We used core principles, objectives, and recommendations in applying conceptual modeling. The main principles applied here include using feedback and being explicit. Using feedback indicated whether we moved in the correct direction by moving back and forth from the problem domain and solution domain and indicated whether our solution solved the problem. These principles facilitated reaching our objectives. In turn, the objectives were to establish understanding, insight, and overview, support communication and decision making, and facilitate reasoning. The main principles and objectives translated into ten main recommendations [50]. We applied the following ten recommendations as follows:

1. Timeboxing: We used timeboxing for developing several models. The timeboxing varies from minutes to days to weeks.
2. Iteration: We iterated using feedback from subject-matter experts' input. The experts included domain scholars and industry practitioners, focusing on the company's key persons.
3. Early quantification: We translated the principle of being explicit in quantifications early. We used failure data analyses for this early quantification. Early quantification aids in being explicit and sharpening discussions by using numbers. However, these numbers can evolve as we conduct quantifications at an early phase, and more confidence using validations may be needed.
4. Measurement and validation: We calculated and measured numbers to have an indication for some essential measurements. For instance, we calculated the failure rates and MTBF to indicate the SOI's reliability. We validated these numbers by using evidence and arguments from the literature and the company.
5. Applying multiple levels of abstraction: We considered the size and complexity of these levels of abstraction. We aimed at connecting high-level abstractions to a lower level to achieve concrete guidance. For instance, we considered conceptually modeling all APS as an SOI. Furthermore, we also generated conceptual models by considering the gate as an SOI and its relation to the APS as a whole.

6. Using simple mathematical models: We used simple models to be explicit and to understand the problem and solution domain. These models aimed at capturing the relation between the parts and components of the company's SOI to be able to reason these relations.

7. Analysis of credibility and accuracy: We made ourselves and the company aware of the numbers. These numbers comprised an early quantification that needed further verification and validation within more extended iterations.

8. Conducting multi-view: We applied six main (different) views. These views are customer objectives ("what" for the customer), application ("how" for the customer), functional view ("what" of the company's case study request), conceptual view, and realization view. The conceptual and realization views describe the "how" of the company's case study request. These six views include more relevant views. Muller [49] describes these views accompanied by a collection of sub-methods. We iterated over these views by using different abstraction levels.

9. Understanding the system's context: We conducted several research data collection methods, including workshops, participant observation, and interviews to understand the SOI's context. The workshops and interviews were conducted mainly with the company's key persons. However, using participant observations, we also conducted informal interviews with the end-users. We also conducted a literature review to understand the company's case study context. We needed to understand the SOI and case study's context for reasoning.

10. Visualizing: We used visualization to develop all figures (models) conducting the multi-view. The visualization facilitated communication, common understanding, reasoning, and decision making, stimulating discussions among the domain experts: scholars and the company [62–65].

We applied conceptual modeling using these recommendations to understand, reason, make decisions, and communicate the system's specification and design [66]. This application ensures the company's need fulfillment by eliciting customer value and business value propositions. These propositions drive the system's requirements, further driving the system's design. On the other hand, design and system requirements enable customer and business value propositions [67]. However, we observe that the most significant three recommendations to apply conceptual modeling are as follows: early quantification using data analysis, time boxing, and iterations. The other recommendations are mentioned for the sake for a comprehensive implementation of the ten recommendations.

We started conceptual modeling by developing a value network. Furthermore, we developed a workflow to understand how the SOI is working, focusing on the gate. We continued conceptual modeling by creating a functional flow for the maintenance process. The maintenance functional flow was an input to estimate the maintenance task's cost. The functional and workflow analyses enhanced a shared understanding among various stakeholders from industries and academia.

Value Network

A value network is a model showing interactions between stakeholders in the product's life cycle. Figure 22 shows the value network for understanding the customer objectives: what of the customer. The figure categorized the product life cycle into two phases: the creation phase, also called product development, and the use phase, also called the operation phase. The value network explains the critical stakeholder within the development (creation) and operation (use) phases. This understanding facilitates identifying available data sources. These data sources include feedback data from the operation (use) phase that aid decision making during the product development (creation) phase. For instance, we can use failure data that are manually logged from maintenance personal in the operation phase to support decision making for the development team in the creation phase. In other words, we believe the value network identifies the most appropriate data based on understanding the SOI's most critical or active stakeholders.
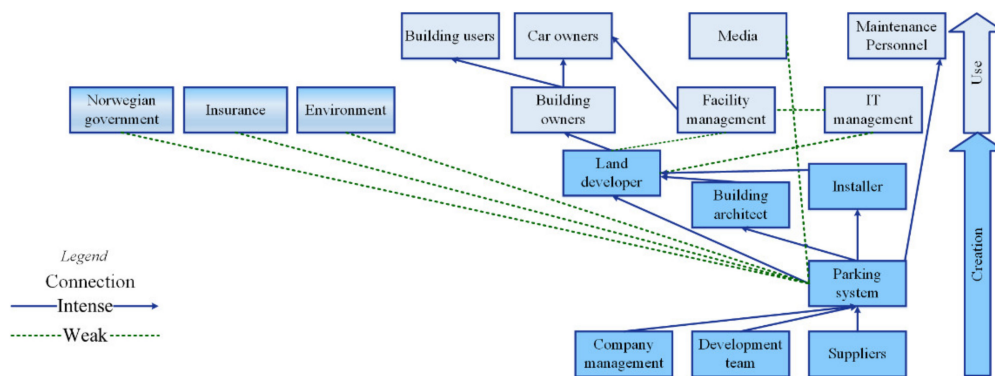
Figure 22. Value network.

The figure visualizes the stakeholders with two colors. Blue colors indicate stakeholders that are involved in the creation phase. On the other hand, light blue shows that the stakeholders are engaged in the use phase. However, the figure also portrays three other stakeholders that have a mix of blue and light blue: insurance, environment (environmental factors), and government. These stakeholders are involved in two phases: the creation and use phase.

Workflow

We developed a workflow to understand how the system is operating. This workflow focuses on the gate. This focus is a result of the data analysis's results. The data analysis showed that the most repetitive word is the gate. Thus, we developed a workflow showing how the gate operates as a part of the SOI and its relation to the end-users and SOI. In addition, from the failure data classification, we observed that the failures are mainly divided into two main categories, use failures (HMI issues) and product failures, where product failures include mechanical and software product failures. Therefore, we indicated the performance of the SOI in the workflow and the performance of the user using two different colors and using the legend. Figures 23 and 24 visualize the workflow for car entry and retrieval for the SOI, respectively. We visualized a red circle for the gate in both workflows as we focused on the gate as a use case.

Functional time flow for the maintenance process illustrates the steps that the maintenance personnel conducted for a typical maintenance task accompanied by its time duration. This model focuses on the gate as a use case. However, the model applies also to any failure events. In this model, we assumed that the parking system allocation is 50 km away from the company's headquarters, where it is allocated around the location of Oslo city. This assumption increases the maintenance task's duration where the parking system is closer than 50 km from the company's headquarter. Another assumption we made is that the maintenance personnel have some experience. Thus, the model's time duration accuracy is within $\pm$ 30%. In other words, the maintenance task duration can increase to 6.5 h or decrease to 3.5 h. This duration assumes that the failures cannot be fixed remotely. If the failure can be fixed remotely, the duration decreases to 15–20 min.
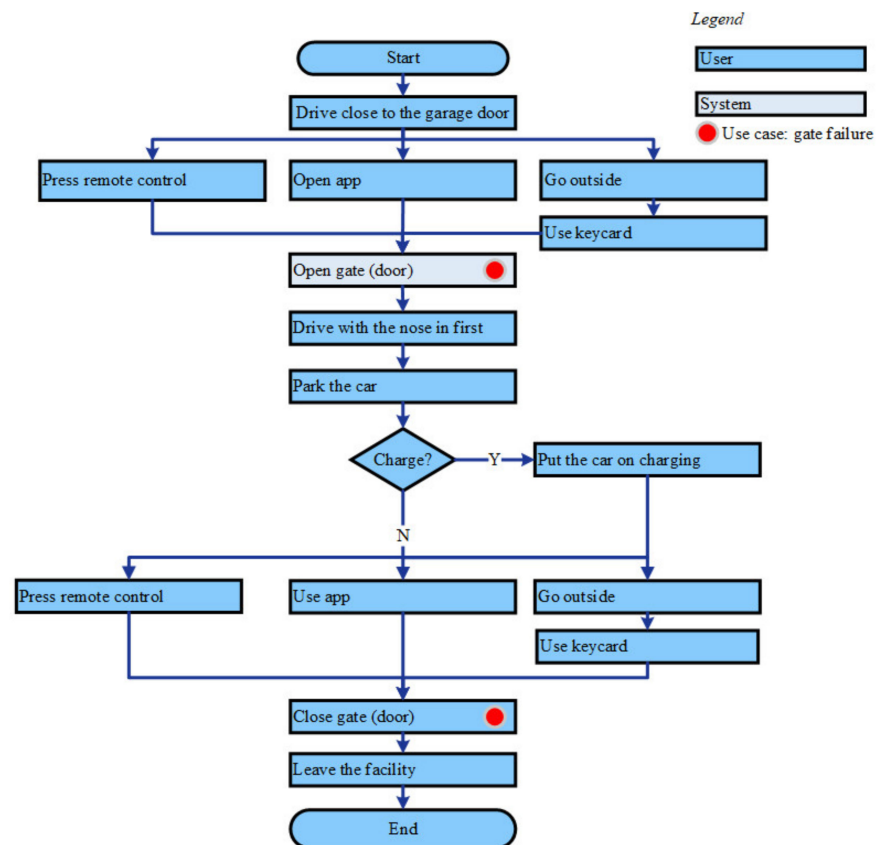
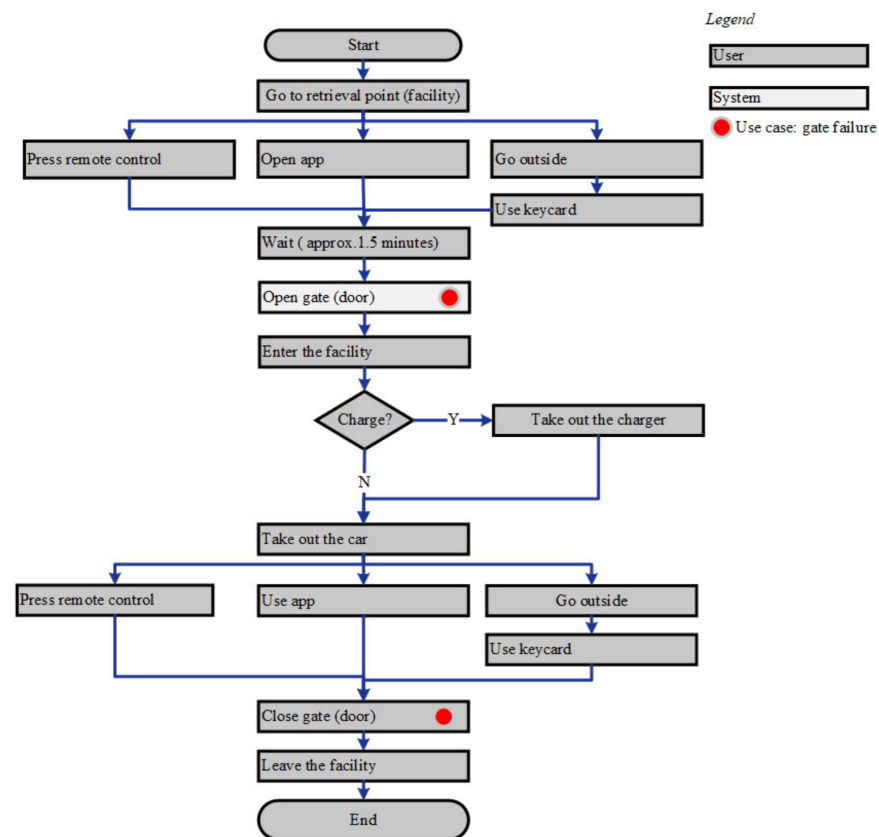**Figure 23.** Workflow for car entry to the SOI.

Figure 24. Workflow for car retrieval from the SOI.

Functional Flow for the Maintenance Process

Figure 25 visualizes the functional flow of the maintenance process.
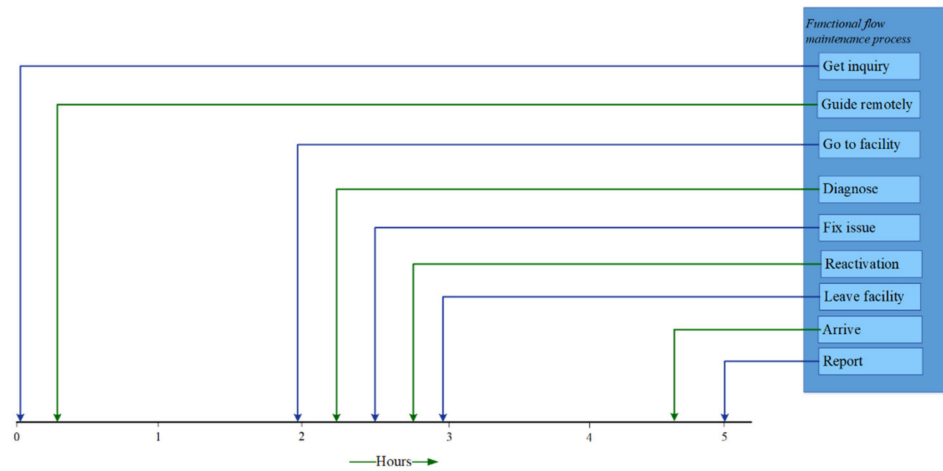

Figure 25. Functional flow for the maintenance process in the company, focusing on the gate as a use case.

Cost Estimation Model

Figure 26 visualizes a cost estimation model per maintenance task. This model includes direct and indirect costs. These costs include several factors that we visualized in the model. We made several assumptions in this model, which includes maintenance personnel traveling to the facility (parking system), and the indirect cost is only one hour as it includes factors outside the actual maintenance task. These factors include spare shipping, storage, invoicing, and accounting costs. We estimated the indirect costs to be not more than one hour, even though it can be more than that. However, these assumptions affect the cost model's accuracy. Thus, we assume that the model's accuracy is located within a range of ± 30%. In other words, the estimated cost is approximately 3.7 kNOk ± 30%. Thus, the estimated cost per maintenance task can be increased to approximately 4.81 kNOK and decreased to approximately 2.59 kNOk.

The estimated cost model per maintenance Task ≈ 3.7 kNOK

Direct cost = 5 h × 620 NOK per h = 3100 NOK				Indirect cost = 1 h × 595NOK per h = 595 NOK		
Direct cost 620 NOK per hour (h)				**Indirect cost** 595NOK per hour (h)		
Labor 350 NOK per hour (h)	**Transport** 200 NOK per hour (h)	**Parts and materials** 50 NOK per hour (h)	**Tools & equipment** 20 NOK per hour (h)	**Spare's shipment cost** 300 NOK per hour (h)	**Invoicing and accounting** 175 NOK per hour (h)	**Storage** 120 NOK per hour (h)

Figure 26. The estimated cost model for each maintenance task.

We used the cost estimation model to estimate the cost of the maintenance task per year based on the MTBF calculation for all failure events and the gate's failure events. We have several MTBFs as we calculated them for all failures and gate failures for public and private systems. The average MTBF values for all failures are 66 h for the public and 37 h for the private (ref. Figure 11). On the other hand, the MTBF for the gate's failures is 66 h for the public and 153 h for the private (ref. Figure 14). We use the following mathematical model to estimate the cost based on the MTBF values.

The number of maintenance tasks per year ≈ 24 h/day ÷ MTBF value in hours = number of tasks/days × 365 days/year ≅ amount of tasks/year.

Furthermore, we multiplied the number of tasks with the estimated cost per maintenance task (2.59 kNOk, 4.18 kNOK). Using the average MTBF for the public parking

system, i.e., 66 h, as an example, we obtained the following: the number of maintenance tasks per year $\cong$ 24 h/day $\div$ 66 h = 0.36/day $\times$ 365 days/year $\cong$ 133 tasks/year. Using the estimated cost model that is mentioned above (2.59 kNok, 4.18 kNOK), the estimated cost for all failures of the public parking system in a year $\cong$ 133 $\times$ (2.59 kNok, 4.18 kNOK) $\cong$ (344 kNOK, 640 kNOK). In other words, the cost for all failures of the public parking systems per year lies between approximately 0.3 MNOk and 0.6 MNOK. Table 1 shows the estimated cost per year for different MTBFs mentioned above:

Table 1. Estimated cost per year for the maintenance based on average MTBF values.

MTBF Values	Estimated Cost Per Year in MNOK
66	0.3–0.6
37	0.6–0.9
153	0.1–0.2

It is significant for the company to be aware of such costs as they rarely invoice customers with such maintenance tasks. However, the company started a new invoicing strategy by invoicing more as a result of this cost estimation model.

4.2. Case Study: Conclusion and Recommendation

Based on the conceptual modeling and data analysis in the previous subsection, we created conceptual models to communicate the conclusion and recommendation for the case study. These recommendations and conclusions can assist the company and practitioners in the same domain in terms of implementing the PD process. In this context, we developed a casual loop for the system's reliability and created a generic representation of the cause and effect of the failure events using the Fishbone diagram in Appendix A. Ultimately, we evolved an elevator pitch model that includes the conclusion and recommendation for the case study, i.e., increasing the system's reliability based on data-driven methodology.

4.2.1. Cause and Effect of the Failure Events

We believe that the cause and effect of the failure events aid in facilitating the problem's understanding based on conceptual modeling and data analysis results. Appendix A includes Figure A1, which portrays a generic representation of the cause and effect of the failure events using the Ishikawa diagram, which is also known as the Fishbone diagram. The effect (problem) is an inferior system's reliability. The cause is categorized into main and secondary causes. We categorized the secondary causes into eight main causes: (1) use failures (HMI issues), (2) product failures, (3) maintenance process (method), (4) maintenance personnel, (5) environmental impact, (6) measurement of system's reliability through MTBF, (7) the lack of design improvement, and ultimately (8) a lack of systems engineering overview, including reliability engineering activities such as reliability programs. Each category has a primary and secondary cause(s). For instance, the use failures category has "User unfamiliar with the system" as the primary cause. At the same time, "push the wrong button," "pass over gate closing," and "forget close gate (door)" are secondary causes. The two main categories are use failures and product (mechanical and software) failures and they are obtained by classifying failure data using semi-supervised machine learning. However, the other six categories lead to product failures where measurement of the system's reliability is a measurement of the product failures rate, which leads to the MTBF measurements. We also included the environmental factors as a result of the weather data analysis in the figure (Figure A1), where we have humidity, salt to icy roads, and precipitation, as we observed a correlation between those parameters and the failure events.

4.2.2. Casual Loop System's Reliability

Causal loops provide a visualization of our understanding of a particular problem, its variables, their dynamics, and their connections to one another. Causal loops aid in building a rational story by linking several loops together [68]. Figure 27 shows a casual loop for the

system's reliability. Figure 27 consists of three reinforcing loops (positive feedback loops) with three different colors, i.e., red, green, and blue. The blue loop illustrates that increasing the system's reliability decreases failure events. On the other hand, more failure events (data) reduce the system's reliability. The red loop shows that increasing the system's reliability increases the profit in the long term. With respect to the short-term increase in a system's reliability, including increasing the design's robustness, it affects profits negatively as its costs include reliability engineering activities in the starting phase. The green loop illustrates the use of conceptual modeling and data analysis to enhance the design robustness and decrease the failure events. Increasing design robustness improves the system's reliability. This causal loop focuses on future solutions based on conceptual modeling and data analysis results of this case study.
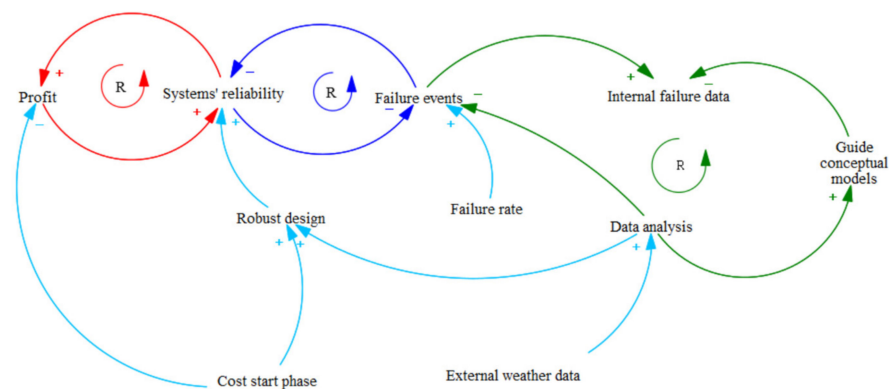

Figure 27. Casual loop: system's reliability.

4.2.3. Elevator Pitch: Conclusion and Recommendation

Figure 28 visualizes the conclusion and recommendation in the form of elevator pitch for the company, focusing on the company's management. Figure 28 comprises four boxes:

- The left upper box lists the customer value proposition. This value includes increasing the system's reliability, which decreases the system's downtime and increases its availability.
- The upper right box lists the business propositions. These propositions are field-proven designs that allow 24/7 operations in order to increase the market share and thus maximize profits.
- The middle box lists the system's requirements. The most vital system requirement is that the system shall operate with minimum failures under its European condition. These European conditions include mainly the western Europe market. This requirement is triggered by the customer and business value propositions. The system's requirement focuses on increasing the system's reliability.
- The lower box in the middle lists the system's design and technology. The design and technology include several aspects, focusing on integrating data into the company's development and maintenance processes by adopting more data-driven methodologies. This methodology can use feedback data such as failure data in the early phase of the PD process and the maintenance process. The data-driven methodology should include external data such as weather data to investigate the environmental factors and to form an explanation and correlation between failure events and other environmental factors. These factors include humidity, snow, salt, temperature, etc. The design and technology suggest improving the HMI within the PD and, in addition, improving the software issues (bugs) and hardware issues. Improving these issues is more urgent than the plan for the company, which includes introducing a sensor strategy toward CBM to build further digital twins. We conducted a case study exploration of this actual need in [15]. However, these envisions regarding the CBM can start by adapting the failure data analysis. For instance, the sensors can be in-

stalled in the most critical subsystem for remote health monitoring, further building up the CBM by using sensor strategies and by installing sensors for the next critical subsystem, i.e., the platform (ref. Figure 10).
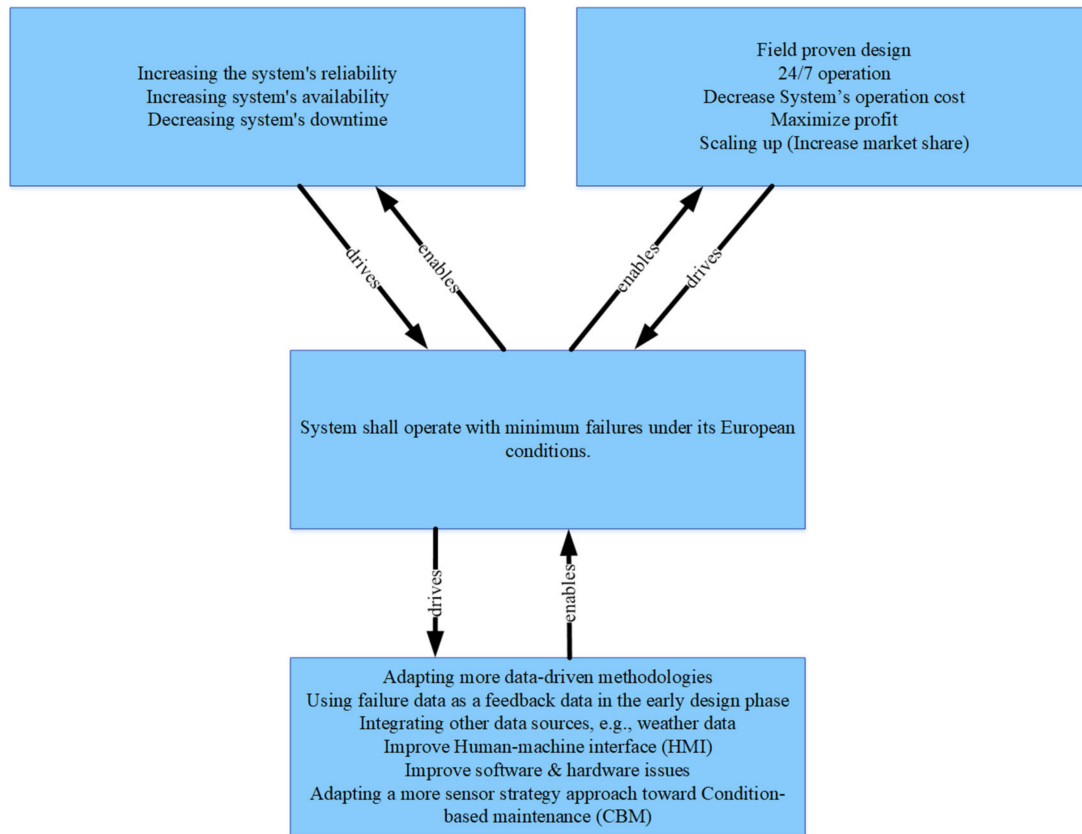

Figure 28. Elevator pitch: conclusion and recommendation (fitness-for-purpose).

There are enable and drive relationships between the boxes. Similarly to Figures 27 and 28 also simultaneously shows the result of conceptual modeling and the data analysis's results. Conceptual modeling supported the data analysis. At the same time, the failure data analysis generally supported conceptual modeling and the elevator pitch, which includes the conclusion and recommendation model specifically.

5. Discussion

We divided this section into four subsections. The first three subsections discuss each research question for this case study concerning the research's results. This section ends by listing the study's limitations.

5.1. Conceptual Modeling and Data Analysis Improve System Developers' Understanding

The first research question, RQ1, is: "How can conceptual modeling and data analysis enhance product developers' understanding of the system?". This study exemplified conceptual modeling and data analysis implementation for a small–medium-sized company that delivers APS, including maintenance and installation. The company is in the transition of developing its own systems. We applied conceptual models and data analysis iteratively and recursively in this study. We conducted several iterations by conducting the data analysis and developing conceptual models. These iterations were also used as inputs for one another. In addition, we used conceptual models as inputs to guide the data analysis, and we used the data analysis as an input to support conceptual models. The feedback from the company indicates that this combination of data analysis and conceptual models is valuable for increasing the understanding of the systems' reliability. The same feedback was

also obtained from other practitioners from the industry and academia when we presented the results to them.

Conceptual models aid in explaining the business and customer value proposition. This value is summarized in the elevator pitch (Figure 28). Several conceptual models can be applied using multi-views and different levels of abstraction. We developed a functional flow for conducting the maintenance tasks for all failure events, focusing on the gate (ref. Figure 25). This focus is a result of the data analysis, as we figured out that the gate is the most critical subsystem. Furthermore, we developed two workflows for car entry and retrieving (ref. Figures 23 and 24), focusing on the gate, i.e., to understand the user and the system part of opening and closing the gate. The functional and workflow analyses aid in providing a shared understanding among various stakeholders within the company and researchers.

The challenge is to develop the optimal number of conceptual models that provides a complete and sufficiently realistic picture to enhance the product developer's understanding but that does not overwhelm the researchers or practitioners when implementing the conceptual models. In this context, we found that the principles, objectives, and recommendations mentioned in Section 2.2 aid early validations with the needed number of conceptual models [50]. In addition, these recommendations assist in developing several conceptual models at the first iteration, which enhances the product developer's understanding via visualization and communicating shared understanding. The second iteration emphasizes the most significant conceptual models. The third iteration focuses on improving the understating and improving these effective models' quality. In this study, we showed only the most significant conceptual models from the third iteration.

Failure data analysis aided conceptual modeling in order to explore an inferior system's reliability by calculating the failure rates and MTBF. In addition, failure data classification assisted in classifying the failure events into product (mechanical or software) failures and HMI issues (user failures). We observed a need to fix failures that decreases the system's reliability. These failures include, among others, software issues such as software bugs and mechanical issues such as over friction. This need is more urgent in the short-term period than building up a sensor strategy toward CBM that can be developed into a digital twin. The CBM can be the long-term vision.

On the other hand, weather data helped in discovering the impact of environmental factors on failure events. For instance, we discovered that salting roads in Norway negatively affects failure events, especially in a colder city that rains less. These data analyses supported conceptual modeling developments and discoveries. However, we conducted and analyzed the data analysis in an iterative manner. This iterative process included preprocessing the data in a template (or frame) in a manner that makes sense when analyzing it. We consumed 80% of the data analysis period (almost one year) in pre-processing, which concurred with the literature [69,70]. Pre-processing operations include understanding the data and its metadata. We conducted several interviews and observations to understand the data. In parallel, we iterated the conceptual modeling process and iteratively supported the data analysis until conceptual modeling and data analyses came into unity. We generated one template at the end for failure and weather data.

The failure data and weather data analysis aided in forming an explanation by examining the relations and correlations among the failure's events and the environmental factors. In addition, this data analysis permitted the identification of problems and anticipatory thinking regarding the differentiations in failures between two types of parking systems, i.e., public, and private. Implementing conceptual modeling and data analysis iteratively and recursively aid in enhancing the understanding of the product's developers as well as communication and decision making to increase the systems' reliability.

5.2. Conceptual Modeling Facilitates Data Identification to Increase A System's Reliability

The second research question, RQ2, is: "How conceptual modeling facilitate data identification to increase a system's reliability?". We identified, collected, and analyzed

data in order to increase the system's reliability. Identifying data that can increase a system's reliability is a result of mainly examining conceptual models together with research data collection. The study showed that the value network is one of the main conceptual models that aid in this identification by identifying, visualizing, and understanding the critical stakeholders (data sources) in the development and operation phases. We identified and collected two sources of data: internal and external data. Internal data in terms of failure data and external data consisted of weather data for investigating the environmental factors that affect the failure events.

In addition, research data supported the study's data sources identification. The research data include interviews, observation, and workshops. Furthermore, we had an office in the company. This allocation also aided in enhancing the value network and facilitated the data identification for the case study. For instance, identifying the external data, i.e., weather data to investigate environmental factors that affect failure events, is supported by the research data collection alongside the subject-matter expert from the industry and academia, mainly the practitioners in those sectors.

5.3. Eeffective Conceptual Models That Facilitate the Product Developers' Understanding

The third research question, RQ3, is:" What are effective conceptual models for facilitating product developers' understanding of an APS?". For this study, we developed approximately twenty models at the first iteration. In the second iteration, we proceeded with fifteen models. In the third iteration, we concluded with five models that are the most significant ones that provide the product developers with the overview and the needed understanding regarding the system's reliability. These models that we found to be the most effective include, among others, the value network, workflow, functional flow, fishbone diagram, casual loop, and fitness-for-purpose (elevator pitch).

5.4. Research Limitations

Limitations are connected to including in-system (sensor) data as it was not technically possible to collect sensor data for the same period of failure data. In addition, there is a lack of longitudinal research over multiple case studies. Research in such environments is complex and has several context-dependent factors that can influence the outcome. However, one of the industry partners conducted the first iterations using this study's methodology, and the results are promising.

The number of conceptual models that we need to enhance the product developer's understanding of the systems' reliability is highly context-dependent. The numbers are also dependent on the expertise and experience of the people involved in developing those models. Furthermore, we need to measure the effectiveness of the conceptual models we use in this study by using different means, such as interviews, workshops, observations, and surveys with other partners in the research project.

On the other hand, sustainability as an aspect that had not been included in this case study. We could consider analytics related to the sustainability aspect. Sustainability analytics and digitalization emphasize data identification, collection, and analysis to support early decision making in the early phase to ensure a sustainable product development process, focusing on humans in the center [71]. This aspect is not included due to the limited data available for the analysis in this context. The sustainability analytics covers goals 8 and 9 of the United Nations 17 sustainability Development Goals [72]. These goals, i.e., goal 8, focus on mobilization, automation, and artificial intelligence areas.

6. Conclusions

This study exemplifies the use of conceptual modeling and data analysis to increase a system's reliability by enhancing the system's developer understanding. The case study was carried out for a medium-sized company. The company delivers APS. The combination of conceptual modeling and data analysis facilitates the developer's understanding of the system's reliability. We conducted this combination iteratively and recursively. Data

analyses and conceptual modeling guide and support each other. Data analyses support conceptual modeling, where conceptual modeling facilitates understanding and communication regarding data and its analysis. This understanding includes identifying the needed data to analyze to explore a system's reliability. In this context, we collected internal and external data. Failure and weather data represent internal and external data, respectively. We analyzed the collected data to explore patterns, correlations and form an explanation to support conceptual modeling.

We found that conceptual modeling and data analysis complement each other. We also found that this combination increases the understanding of the SOI in general and its most critical parts. This understanding aids in forming an explanation and observing a relation or correlation for failure events. Furthermore, this understanding enhances the early phase of the NPD process by closing the loop using feedback data. In this study, we use failure data as feedback data. We use weather data to understand the impact of environmental factors on failure events.

On the other hand, we complement the data analysis results using conceptual modeling to facilitate communication and providing a shared understanding of the problem and solution domain regarding the system's reliability among various stakeholders in the company and academia. Conceptual modeling connects the overall understanding with a more in-depth data analysis. Conceptual modeling supports the decision-making process via a data-driven methodology. This decision process includes the maintenance process and product developers' decisions in the early design phase of the new product development process.

Further research can include a more comprehensive data analysis, including normalization and regression analysis. In addition, further research can include the sustainability aspect, such as using AI to adapt the generative design and reduce and optimize materials. Furthermore, AI can be combined to develop a digital twin to secure optimal designs and to provide an overview of real-time emissions. In addition, further research can also include using blockchain to trace the supply chain's materials and workforces.

Appendix A

This appendix shows Figure A1, which was discussed earlier in the Results section (ref. Section 4.2.1).
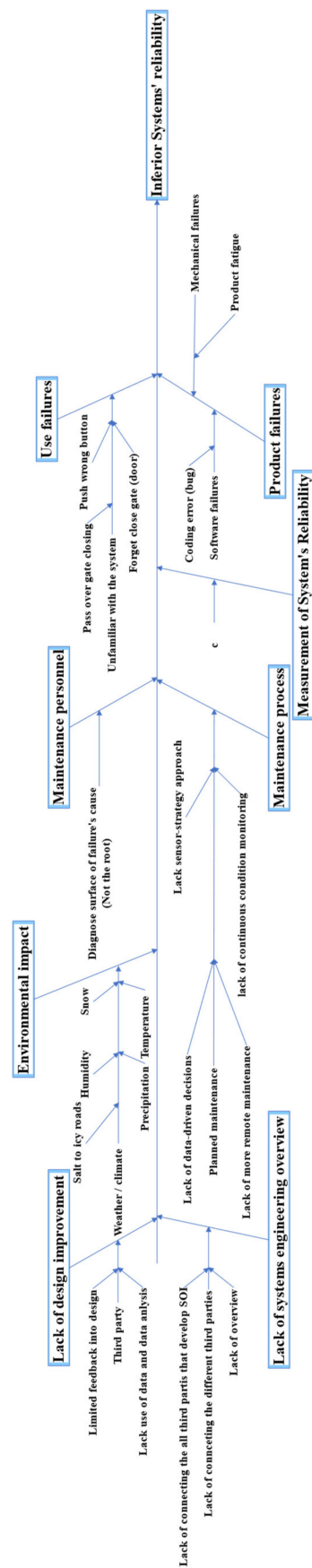
Figure A1. A generic representation of the cause and effect of the failure events using an Ishikawa diagram, which is also known as a fishbone diagram.

References

1. Research, H.A. What Is an Automatic Parking System? Available online: https://www.caranddriver.com/research/a31995865/automatic-parking-systems/ (accessed on 15 May 2022).
2. Nourinejad, M.; Bahrami, S.; Roorda, M.J. Designing Parking Facilities for Autonomous Vehicles. *Trans. Res. Part B Methodol.* **2018**, *109*, 110–127. [CrossRef]
3. Zhu, Y.; Ye, X.; Chen, J.; Yan, X.; Wang, T. Impact of Cruising for Parking on Travel Time of Traffic Flow. *Sustainability* **2020**, *12*, 3079. [CrossRef]
4. Grand View Research Automated Parking Systems Market Size Report, 2020–2027. Available online: https://www.grandviewresearch.com/industry-analysis/automated-parking-systems-market (accessed on 21 January 2022).
5. Cudney, G. Parking Today | Articles—Automated Parking: Is It Right for You? Available online: https://www.parkingtoday.com/articledetails.php?id=181&t=automated-parking-is-it-right-for-you (accessed on 21 January 2022).
6. Robles, F. Road to Robotic Parking Is Littered with Faulty Projects. *The New York Times*, 27 November 2015.
7. Barnard, R.W.A. 3.2.2 What is wrong with reliability engineering? In Proceedings of the INCOSE International Symposium, Utrecht, The Netherlands, 15–19 June 2008; Wiley Online Library: Hoboken, NJ, USA, 2008; Volume 18, pp. 357–365. [CrossRef]
8. Hosseini, H.N.; Welo, T. A framework for integrating reliability and systems engineering: Proof-of-concept experiences. In Proceedings of the INCOSE International Symposium, Edinburgh, UK, 18–21 July 2016; Wiley Online Library: Hoboken, NJ, USA, 2016; Volume 26, pp. 1059–1073. [CrossRef]
9. Dhillon, B.S. *Robot System Reliability and Safety: A Modern Approach*; CRC Press: Boca Raton, FL, USA, 2015.
10. Kozak, K.H.; Graham, C.H.; Wiens, J.J. Integrating GIS-Based Environmental Data into Evolutionary Biology. *Trends Ecol. Evol.* **2008**, *23*, 141–148. [CrossRef] [PubMed]
11. Jørgensen, S.E. *Handbook of Environmental Data and Ecological Parameters: Environmental Sciences and Applications*; Elsevier: Amsterdam, The Netherlands, 2013; Volume 6.
12. Klein, G.; Phillips, J.K.; Rall, E.L.; Peluso, D.A. A Data–Frame Theory of Sensemaking. In *Expertise Out of Context*; Psychology Press: London, UK, 2007; pp. 118–160.
13. Weick, K.E. *Sensemaking in Organizations*; Sage: Newcastle upon Tyne, UK, 1995; Volume 3.
14. Muller, G. Applying Roadmapping and Conceptual Modeling to the Energy Transition: A Local Case Study. *Sustainability* **2021**, *13*, 3683. [CrossRef]
15. Ali, H.B.; Muller, G.; Salim, F.A. Applying conceptual modeling and failure data analysis for "Actual Need" exploration. In Proceedings of the MODERN SYSTEMS 2022: International Conference of Modern Systems Engineering Solutions, Saint-Laurent-du-Var, France, 24–28 July 2022.
16. O'Connor, P.; Kleyner, A. *Practical Reliability Engineering*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
17. Walden, D.D.; Roedler, G.J.; Forsberg, K.; Hamelin, R.D.; Shortell, T.M. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
18. Long, E.A.; Tananko, D. Best practices methods for robust design for reliability with parametric cost estimates. In Proceedings of the 2011 Proceedings—Annual Reliability and Maintainability Symposium, Lake Buena Vista, FL, USA, 24–27 January 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1–6.
19. Barnard, A. Reliability engineering: Value, waste, and costs. In Proceedings of the INCOSE International Symposium, Edinburgh, UK, 18–21 July 2016; Wiley Online Library: Hoboken, NJ, USA, 2016; Volume 26, pp. 2041–2054. [CrossRef]
20. Hollis, R. Put Engineering Efforts Back in Reliability Techniques. *IEEE Trans. Parts Mater. Packag.* **1965**, *1*, 297–302. [CrossRef]
21. Coppola, A. Reliability Engineering of Electronic Equipment a Historical Perspective. *IEEE Trans. Rel.* **1984**, *33*, 29–35. [CrossRef]
22. Krasich, M. How to Estimate and Use MTTF/MTBF Would the Real MTBF Please Stand Up? In Proceedings of the 2009 Annual Reliability and Maintainability Symposium, Fort Worth, TX, USA, 26–29 January 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 353–359.
23. Ridgway, M.; Baretich, M.F.; Clark, M.; Grimes, S.; Iduri, B.; Lane, M.W.; Lipschultz, A.; Lum, N. A Rational Approach to Efficient Equipment Maintenance, Part 2: A Comprehensive AEM Program. *Biomed. Instrum. Technol.* **2018**, *52*, 350–356. [CrossRef]
24. Unhelkar, B. *Big Data Strategies for Agile Business*; CRC Press: Boca Raton, FL, USA, 2017.
25. Jifa, G. Data, Information, Knowledge, Wisdom and Meta-Synthesis of Wisdom-Comment on Wisdom Global and Wisdom Cities. *Procedia Comput. Sci.* **2013**, *17*, 713–719. [CrossRef]
26. Bellinger, G.; Castro, D.; Mills, A. Data, Information, Knowledge, and Wisdom. 2004. Available online: https://www.systems-thinking.org/dikw/dikw.htm (accessed on 21 December 2022).
27. Ali, H.B.; Salim, F.A. Transferring Tacit Knowledge into Explicit: A Case Study in a Fully (Semi) Automated Parking Garage. In Proceedings of the Society for Design and Process Science, Online, Kongsberg, Norway, 15 December 2021.
28. Laun, A.; Mazzuchi, T.A.; Sarkani, S. Conceptual Data Model for System Resilience Characterization. *Syst. Eng.* **2022**, *25*, 115–132. [CrossRef]
29. Henderson, D.; Earley, S. *DAMA-DMBOK: Data Management Body of Knowledge*; Technics Publications: Basking Ridge, NJ, USA, 2017.
30. Birkland, T.A. *Lessons of Disaster: Policy Change after Catastrophic Events*; Georgetown University Press: Washington, DC, USA, 2006.
31. Kunreuther, H.; Useem, M. *Learning from Catastrophes: Strategies for Reaction and Response*; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 2009.

32. Ramos, A.L.; Ferreira, J.V.; Barceló, J. Model-Based Systems Engineering: An Emerging Approach for Modern Systems. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2011**, *42*, 101–111. [CrossRef]
33. Engen, S.; Falk, K.; Muller, G. Conceptual Models to Support Reasoning in Early Phase Concept Evaluation-A Subsea Case Study. In Proceedings of the 2021 16th International Conference of System of Systems Engineering (SoSE), Västerås, Sweden, 14–18 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 95–101.
34. Plattner, H.; Meinel, C.; Leifer, L. *Design Thinking Research: Making Design Thinking Foundational*; Springer: Berlin/Heidelberg, Germany, 2015.
35. Wettre, A.; Sevaldson, B.; Dudani, P. Bridging Silos: A New Workshop Method for Bridging Silos. In Proceedings of the RSD8, Chicago, IL, USA, 13–15 October 2019.
36. Neely, K.; Bortz, M.; Bice, S. Using Collaborative Conceptual Modelling as a Tool for Transdisciplinarity. *Evid. Policy* **2021**, *17*, 161–172. [CrossRef]
37. Sargent, R.G. Verification and Validation of Simulation Models. *J. Simul. JOS* **2013**, *7*, 12–24. [CrossRef]
38. Balci, O.; Arthur, J.D.; Nance, R.E. Accomplishing Reuse with a Simulation Conceptual Model. In Proceedings of the 2008 Winter Simulation Conference, Miami, FL, USA, 7–10 December 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 959–965.
39. Montevechi, J.A.B.; Friend, J.D. Using a Soft Systems Methodology Framework to Guide the Conceptual Modeling Process in Discrete Event Simulation. In Proceedings of the 2012 Winter Simulation Conference (WSC), Berlin, Germany, 9–12 December 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 1–12.
40. Robinson, S. Conceptual Modelling for Simulation Part I: Definition and Requirements. *J. Oper. Res. Soc.* **2008**, *59*, 278–290. [CrossRef]
41. Robinson, S. Conceptual Modelling for Simulation Part II: A Framework for Conceptual Modelling. *J. Oper. Res. Soc.* **2008**, *59*, 291–304. [CrossRef]
42. Gorod, A.; Hallo, L.; Ireland, V.; Gunawan, I. *Evolving Toolbox for Complex Project Management*; CRC Press: Boca Raton, FL, USA, 2019.
43. Lavi, R.; Dori, Y.J.; Dori, D. Assessing Novelty and Systems Thinking in Conceptual Models of Technological Systems. *IEEE Trans. Educ.* **2020**, *64*, 155–162. [CrossRef]
44. Muller, G. Tutorial Architectural Reasoning Using Conceptual Modeling. In Proceedings of the at INCOSE International Symposium, Seattle, WA, USA, 13–16 July 2015.
45. Blanchard, B.S. *Systems Engineering and Analysis*, 5th ed.; Prentice Hall International Series in Industrial and Systems Engineering; Pearson: Boston, MA, USA, 2011; ISBN 978-0-13-714843-1.
46. Tomita, Y.; Watanabe, K.; Shirasaka, S.; Maeno, T. Applying Design Thinking in Systems Engineering Process as an Extended Version of DIKW Model. In Proceedings of the INCOSE International Symposium, Adelaide, SA, Australia, 15–20 July 2017; Wiley Online Library: Hoboken, NJ, USA, 2017; Volume 27, pp. 858–870. [CrossRef]
47. Jackson, M.C. *Critical Systems Thinking and the Management of Complexity: Responsible Leadership for a Complex World*, 1st ed.; Wiley: Hoboken, NJ, USA, 2019; ISBN 978-1-119-11838-1.
48. Sauser, B.; Mansouri, M.; Omer, M. Using Systemigrams in Problem Definition: A Case Study in Maritime Resilience for Homeland Security. *J. Homel. Secur. Emerg. Mang.* **2011**, *8*, 102202154773551773. [CrossRef]
49. Muller, G. CAFCR: A Multi-View Method for Embedded Systems Architecting; Balancing Genericity and Specificity. Ph.D. Thesis, Technical University of Delft, Delft, The Netherlands, 2004.
50. Muller, G. Challenges in Teaching Conceptual Modeling for Systems Architecting. In *Advances in Conceptual Modeling, Proceedings of the International Conference on Conceptual Modeling, Stockholm, Sweden, 19–22 October 2015*; Springer: Cham, Switzerland, 2015; pp. 317–326.
51. Ali, H.B.; Langen, T.; Falk, K. Research Methodology for Industry-academic Collaboration—A Case Study. In Proceedings of the INCOSE International Symposium CSER, Trondheim, Norway, 24–26 March 2022; Wiley Online Library: Hoboken, NJ, USA, 2016; Volume 32 (Suppl. S2), pp. 187–201. [CrossRef]
52. Yin, R.K. *Applications of Case Study Research*, 3rd ed.; SAGE: Los Angeles, CA, USA, 2012; ISBN 978-1-4129-8916-9.
53. Potts, C. Software-Engineering Research Revisited. *IEEE Softw.* **1993**, *10*, 19–28. [CrossRef]
54. Altrichter, H.; Kemmis, S.; McTaggart, R.; Zuber-Skerritt, O. The Concept of Action Research. *Learn. Organ.* **2002**, *9*, 125–131. [CrossRef]
55. Stenström, C.; Al-Jumaili, M.; Parida, A. Natural Language Processing of Maintenance Records Data. *Int. J. COMADEM* **2015**, *18*, 33–37.
56. Removing Stop Words from Strings in Python. Available online: https://stackabuse.com/removing-stop-words-from-strings-in-python/ (accessed on 7 September 2022).
57. GeeksforGeeks. Snowball Stemmer—NLP. Available online: https://www.geeksforgeeks.org/snowball-stemmer-nlp/ (accessed on 21 December 2022).
58. Han, S. Googletrans: Free Google Translate API for Python. Translates Totally Free of Charge. Available online: https://py-googletrans.readthedocs.io/en/latest/ (accessed on 21 December 2022).
59. Gao, X.; Zhu, N. Hidden Markov Model and Its Application in Natural Language Processing. *J. Theor. Appl. Inf. Technol.* **2013**, *12*, 4256–4261. [CrossRef]
60. Værmeldingen Forteller når Veien Saltes. Available online: https://www.vg.no/i/0m8x0 (accessed on 21 August 2022).

61. Statens Vegvesen. Statens vegvesen Fellesdokument Driftskontrakt Veg D2: Tegninger Og Supplerende Dokumenter D2-ID9300a Bruk Av Salt. Available online: https://www.mercell.com/m/file/GetFile.ashx?id=151465704&version=0 (accessed on 21 December 2022).
62. Engebakken, E.; Muller, G.; Pennotti, M. Supporting the System Architect: Model-Assisted Communication. In Proceedings of the Systems Research Forum, Kauai, HI, USA, 5–8 January 2010; World Scientific: Singapore, 2010; Volume 4, pp. 173–188. [CrossRef]
63. Muller, G.; Pennotti, M. Developing the Modeling Recommendation Matrix: Model-Assisted Communication at Volvo Aero. In Proceedings of the INCOSE International Symposium, Rome, Italy, 9–12 July 2012; Wiley Online Library: Hoboken, NJ, USA, 2012; Volume 22, pp. 1870–1883. [CrossRef]
64. Polanscak, E.; Muller, G. Supporting Product Development: A3-Assisted Communication and Documentation. Unpublished Master Project Paper at Buskerud Uni. Col. 2011. Available online: https://gaudisite.nl/SEMP_Polanscak_A3.pdf (accessed on 21 December 2022).
65. Stalsberg, B.; Muller, G. Increasing the Value of Model-Assisted Communication: Modeling for Understanding, Exploration and Verification in Production Line Design Projects. In Proceedings of the INCOSE International Symposium, Las Vegas, NV, USA, 30 June–3 July 2014; Wiley Online Library: Hoboken, NJ, USA, 2014; Volume 24, pp. 827–842.
66. Muller, G. System and Context Modeling—The Role of Time-Boxing and Multi-View Iteration. In Proceedings of the Systems Research Forum, Orlando, FL, USA, 13–16 July 2009; World Scientific: Singapore, 2009; Volume 3, pp. 139–152.
67. Muller, G. Teaching Conceptual Modeling at Multiple System Levels Using Multiple Views. *Procedia CIRP* **2014**, *21*, 58–63. [CrossRef]
68. Tip, T. Guidelines for Drawing Causal Loop Diagrams. *Syst. Think.* **2011**, *22*, 5–7.
69. Hoyt, R.E.; Snider, D.H.; Thompson, C.J.; Mantravadi, S. IBM Watson Analytics: Automating Visualization, Descriptive, and Predictive Statistics. *JMIR Public Health Surveill.* **2016**, *2*, e5810. [CrossRef] [PubMed]
70. Ali, H.B.; Helgesen, F.H.; Falk, K. Unlocking the Power of Big Data within the Early Design Phase of the New Product Development Process. In Proceedings of the INCOSE International Symposium, Virtual Event, 17–22 July 2021; Wiley Online Library: Hoboken, NJ, USA, 2021; Volume 31, pp. 434–452. [CrossRef]
71. Deloitte Development LLC. Deloitte Sustainability Analytics the Three-Minute Guide. 2012. Available online: https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Deloitte-Analytics/dttl-analytics-us-ba-sustainability3minguide.pdf (accessed on 21 December 2022).
72. United Nations. The Sustainable Development Goals Report. 2017. Available online: https://unstats.un.org/sdgs/files/report/2017/thesustainabledevelopmentgoalsreport2017.pdf (accessed on 21 December 2022).