Remote Operation in a Modern Context

Design of Graphical User Interfaces for Remote Operation of Autonomous Vehicles

Master's thesis in Interaction Design and Technologies

Isak Lindgren
Alexander Larsson Vahlberg

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2023

Master's thesis 2023

Remote Operation in a Modern Context

Design of Graphical User Interfaces for Remote Operation of Autonomous Vehicles

Isak Lindgren
Alexander Larsson Vahlberg

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Division of Interaction Design
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2023

Remote Operation in a Modern Context
Design of Graphical User Interfaces for Remote Operation of Autonomous Vehicles
Isak Lindgren
Alexander Larsson Vahlberg

Supervisor: Staffan Björk, Computer Science and Engineering
Mentors: Anders Lindgren Walter, Xueying Hai, Einride
Examiner: Mikael Wiberg, Computer Science and Engineering

Master's Thesis 2023
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Einride's autonomous electric truck driving on a rural road [1].

Typeset in LaTeX
Gothenburg, Sweden 2023

Abstract

The transition to autonomous vehicles on open roads might be critical for a sustainable future. One way of facilitating this transition is through the use of *Remote Operation*, the concept of maneuvering, assisting, and monitoring vehicles from a distance. This introduces many challenges in terms of for instance human-machine interaction, human-automation interaction, and situational awareness for the design of the remote operation interface. While solving these challenges may have critical importance for the successful application of remote operation and autonomous vehicles, there is a lack of established best practices for solving these issues in a modern context. The purpose of this master's thesis is therefore to answer the question **"What should be considered when designing graphical user interfaces for the remote operation of autonomous vehicles?"**. In order to do this, research was conducted through the process of *Research through Design*, utilizing an iterative design process in order to create and refine a set of guidelines. The process consisted of a pre-study and two design iterations, mainly focusing on the latter parts of the design process. This was done to create a design artifact in the form of a high-fidelity software prototype used as a vehicle for the design process, providing an example of how to apply these guidelines within a specific context. The final set of guidelines introduces the concepts of **Prioritization**, **Layout**, **Coding** as the primary aspects of presenting information in addition to the relevant aspects of feedback, interaction, and automation transparency. Together, these aim to provide a solid foundation for designing graphical user interfaces for remote operation of autonomous vehicles.

Keywords: Human-Machine Interaction, Human-Automation Interaction, Graphical User Interface, Interaction Design, Situation Awareness, Remote Operation, Teleoperation, Research through Design, Autonomous Vehicles

Acknowledgements

Our sincerest thanks to everyone who supported us in this project. Our mentor at Einride, Xueying Hai for proactively connecting us with the right tools and people. Anders Lindgren Walter for providing us with this opportunity, and for trusting our capabilities. Everyone at Einride who has taken time from their busy schedules to discuss, attended presentations, participate in usability testing, and provide us with help and support. Our supervisor at Chalmers, Staffan Björk for his valuable support and guidance through the project and for sharing his wisdom.

Isak Lindgren & Alexander Larsson Vahlberg, Gothenburg, 2023-06-19

Contents

List of Figures

1
Introduction

Transportation is responsible for 23% of the global energy-related carbon dioxide emissions [2] and road traffic is responsible for between 20-50 million injuries and 1.25 million deaths worldwide per year [3]. Nevertheless, transportation is essential for connecting society, making technological advancements in the field of transportation necessary in order to achieve a sustainable society and reach the UN's Sustainable Development Goals [4]. One such advancement is the development of autonomous vehicles (AV) which is predicted to decrease crash rates [4] while also reducing greenhouse gas emissions significantly [5]. However, in addition to the technological challenges of developing AVs, commercialization and widespread deployment introduces the challenge of gaining trust from the general public. Due to safety, legal, and technical reasons, the need for a methodical, holistic, and safe introduction of AVs in society is essential. Remote maneuvering, assistance, and monitoring of vehicles, also known as *remote operation* (RO) or *teleoperation* (TO), will therefore most likely become more prominent in society as it holds a key role in transitioning to the wide distribution of fully autonomous vehicles [6]. The Swedish company Einride [7] has developed an autonomous electric truck that can be teleoperated [8], with the goal of utilizing electrification and automation of freight transport to change road freight for the better [9].

RO will likely replace or at least serve as a complement to the manual or autonomous driving of vehicles to a great extent in the near future. However, removing the driver from the physical cockpit and into a remote station comes with many challenges in terms of situational awareness due to the inherent limitations of interacting with the world through a screen. This issue is further emphasized by the fact that the most important senses when driving are the visual and aural, followed by the haptic and vestibular [6]. Thus, poorly designed graphical user interfaces (GUI) that do not display the right information at the right time are an important issue of RO that is detrimental to situational awareness. Failing to deliver the required information in a satisfying manner may have dire consequences in terms of material damage and loss of life. This calls for urgent research into the design of RO software and GUIs.

1.1 Problem Description

The challenges of moving the driver from the physical cockpit to RO stations emphasize the importance of good GUI design. For example, a poorly designed GUI could cause the operator to miss highly critical information causing the vehicle to crash. Repeated accidents involving remotely operated vehicles could lead to increasingly restrictive legislation or investigations as seen in the case of AVs after numerous accidents involving Tesla's "Autopilot" [10]. This could potentially halt the commercialization of remotely operated vehicles in general. Not to mention the well-being of drivers and pedestrians that would be at risk. The success of remotely operated vehicles, and thus AVs, as a whole, may therefore somewhat rely on the user experience and ability of the GUI to correctly portray the environment and state of the vehicle. The GUI should thus be designed for safety by allowing for the highest possible situational awareness. Moreover, moving the vehicle operator from the physical cockpit to a remote location utilizing a GUI allows for the exploration of novel methods of visualizing information, not limited to metaphors for the traditional cockpit.

1.2 Research Question

While the remote operation of vehicles has been researched for decades [11], it has recently become more relevant due to advancements within autonomous technology. As such, there is a need for establishing best practices reflecting the new technological environment and new areas of application, something that is currently seemingly lacking. Guidelines such as those presented by Flising & Widerstrand [12], could help establish such highly necessary best practices, and the aim of this project is therefore to continue the effort of establishing these. For this reason, the general purpose of this thesis will be to answer the following research question:

What should be considered when designing graphical user interfaces for the remote operation of autonomous vehicles?

1.3 Aim

The aim of this thesis is to answer the research question by gathering knowledge from existing theory and insights and through an iterative process exploring the design of a GUI for RO and guidelines in tandem. The goal is thus to condense knowledge gained through this process into a set of guidelines for future design within this field. Moreover, this thesis will develop said guidelines within the specific context of autonomous electric trucks operating in a freight company.

1.4 Delimitations

While a driver primarily uses four senses when driving, visual, aural, haptic, and vestibular [6], the main focus of this thesis will be on the visual. While work on this project might yield insights into the lack of other sensory feedback and the workstation as a whole, this will not be the main concern due to the expertise of the authors lying within GUI design and software engineering. Moreover, due to time constraints, the work will focus on traditional GUIs, not utilizing virtual or augmented reality (XR).

Remote operation is at the moment generally performed with a 1:1 ratio between operators and vehicles but is expected to transition into a 1:N ratio in the future. Therefore, this thesis is mainly focused on a 1:1 ratio where the operator is utilizing a direct control interface (see section 2.1), with future 1:N scenarios taken into consideration. Moreover, the design will adhere to the current technical setup used by Einride as can be seen in figure 2.2b.

While the aim of this project is to evaluate a GUI in as close to a real-life setting as possible, actually conducting tests in real settings will not be possible due to technical, safety, and proprietary reasons. Thus, the final prototype will not be able to interact with any physical trucks, but will rather utilize other means to simulate the real experience of driving a truck. This means that the prototype will not be able to correctly respond to any interactions aimed at controlling the truck during evaluation. Instead, a dynamic visualization will be created, simulating the experience as realistically as possible under the circumstances.

2
Background

In this section, some background information is presented, providing context to the thesis. It covers stakeholders, some relevant actors in the AV industry, state-of-the-art research, and ethical considerations. As such, it provides a technical, legal, and ethical foundation for further research and discussion.

2.1 Autonomous Driving & Remote Operation

While automation to some extent has been present in a multitude of fields for decades, it has only recently become prominent in the application of AVs due to technical advancements. The most widely adopted definition for different levels of autonomy for AVs is defined by the Society of Automotive Engineers (SAE) [13]. They provide a taxonomy consisting of six levels, from zero to five, ranging from no driving automation to full driving automation, see figure 2.1.

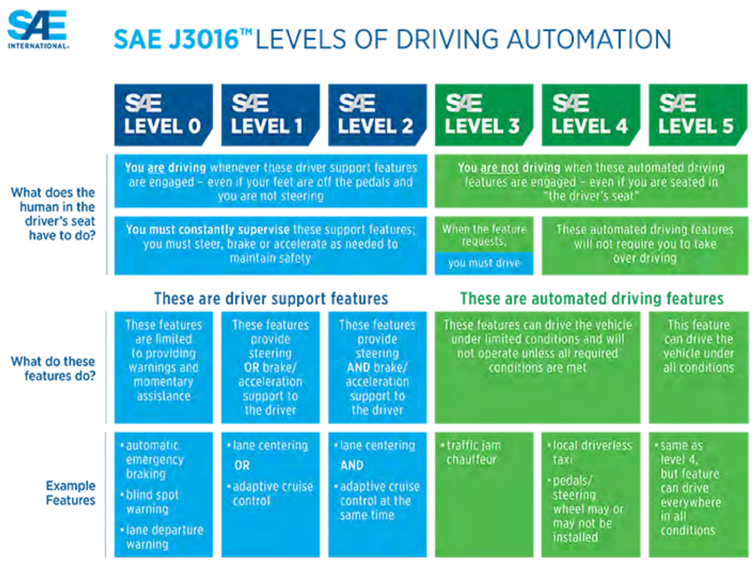


Figure 2.1: "Consumer-friendly" description of SAE's levels of automation [14]. Notice that there are six levels of automation, each increasingly technologically advanced than the previous. RO is relevant from level three and up.

According to SAE, the use of RO is only relevant for AVs that have reached level three or higher in the taxonomy of automation as it is only on these levels the driver does not physically have to be present in the driver's seat. RO of AVs currently often takes the form of what Fong and Thorpe [11] describe as a direct control interface. These are also called "inside-out" interfaces due to the operator feeling as if they are inside the vehicle. Utilizing this type of interface is appropriate when human decision-making is required and the environment can support high bandwidth and low latency [11]. This means that the operator acts as a driver in the traditional sense but from a distance when and if required.

As the level of vehicle automation increases, the frequency of human intervention by taking direct control of the vehicle will decrease and the need for another type of interface will be required. This is described by Fong and Thorpe as a supervisory control interface [11] which requires a certain level of automation, such that the vehicle is capable of safely achieving some goals independently. This type of interface is used to generate high-level commands as well as monitor and diagnose vehicles and can be compared to how a supervisor interacts with their subordinates. To perform supervisory control the operator breaks down tasks into subtasks, which the vehicle's automated systems then solve. Fong and Thorpe also mention that this type of interface utilizing more abstract commands is better suited for higher latency than direct control interfaces.

Endsley and Jones argue that a human will always have to be in the loop to correct automation and "plug the holes" [15, p. 170] where automation fails. Both supervisory and direct interfaces come with their respective challenges but still, serve a purpose in the RO of AVs in separate stages of automation or as a complement to each other. For instance, direct control might be used as a backup to supervisory control. This creates the need for GUIs that are flexible and can quickly be adapted to new legislation and technological advancements and thus requirements.

2.2 Legislation

Legislation regarding AVs and RO is a complicated matter. While there are a number of different international treaties regarding traffic regulations, such as the Geneva [16] and Vienna conventions [17], not all countries have signed them. Additionally, the EU has defined regulations common for its members [18], and the US has federal laws regulating some aspects of autonomous driving [19]. Moreover, occasionally, legislation around teleoperated AVs is exempt from legislation concerning traditional trucks or is regarded as an exception allowing manufacturers to diverge from current legislation. Additionally, Ilkovà and Ilka describe that testing of AVs is done using ad-hoc legal permits across the EU [19]. For example, to be able to test AVs on public roads in Sweden, an application for a trial permit for a temporary grant must be filled in. A form should be sent to Transportstyrelsen, containing among other things, a technical description of the vehicles included in the trial, proof that the operation is going to be conducted in a traffic-safe manner, and that the trial operation will not cause any significant inconvenience to the surroundings as well as an overall description of the operation [20]. Moreover, there is no requirement for any specific driver's license to take part in such an operation, but rather that the people responsible are required to ensure that anyone participating is competent enough for the task [21]. Due to the legal environment surrounding AVs, any research and development made in the field must consider legislation, its progress, and how it might affect the area of study.

2.3 Stakeholders

There are many different actors who are affected and might be interested in the results produced by this thesis. Some of these actors are affected directly, while some are affected indirectly and might simply be interested in any progress made in the field of AVs. Therefore, some of these are presented below.

2.3.1 Einride

This thesis is done in association with Einride [7], who therefore is one of the primary stakeholders. Einride is a transport company looking to revolutionize the transport sector through the development of electrical autonomous freight trucks. This is done with the aim of utilizing autonomy and electrification to achieve a more desirable future, aligned with human-centric values and ecological necessities [9]. Einride provides a number of services, all with the combined end goal of providing capacity-as-a-service (CaaS) [22], utilizing electric vehicles, AVs, fleet management, and other modern technologies to optimize routes. As part of reaching their goals, Einride has developed an electric freight truck that has reached level four according to SAE's taxonomy of automation [8], [23], which can be seen in figure 2.2a. Its corresponding RO station can be seen in figure 2.2b.

(a) Einride's autonomous electric truck driving on a road, modified from [7].

(b) Einride's Remote Interface, consisting of three screens, a stream deck, a lever, and a dial [24].

Figure 2.2: The Einride Autonomous and its RO station.

2.3.2 Academia

This thesis is also done in association with Chalmers University of Technology [25], who are stakeholders in any contributions to academic knowledge this thesis can provide. Moreover, as technology develops and new fields are encountered, it is only natural that academia as a whole has an interest in research in these areas. As such, any knowledge provided in terms of the development of GUIs and guidelines for RO is of high interest as this is a novel field.

2.3.3 End-users

The remote operators at Einride, who are the end-users, are some of the primary stakeholders as they are the people who are directly affected by any development within the field. These currently consist of both engineers and people specifically employed with the purpose of being remote operators. The remote operators are required to have licenses for class 'C' trucks as well as previous experience driving traditional trucks. Who the future end-users are is however still undetermined as the sought-after qualities of a remote operator are currently not clearly defined and are subject to change. It is however likely that they would require some technical knowledge, while also being familiar with maneuvering similar systems. Therefore, within the scope of this thesis, the current end-users, i.e., both engineers and remote operators, are a suitable target audience for this research.

2.3.4 Thesis Authors

The authors of this thesis are also stakeholders in the thesis. The specific approach taken and the angles considered during the thesis are directly related to the unique perspectives of the authors. For instance, the authors' educational backgrounds in software engineering and interaction design will contribute to the problem being formulated as a software and design problem. Furthermore, which tools and methods are applied during the thesis will largely be based on the tools and methods that were learned during these studies. In addition to educational background, the authors have interests and personal experiences that too can affect the thesis including for instance game development and professional driving experience. Also, the authors of this thesis have certain goals and ambitions, primarily that the work should provide value to Einride and academia as well as new expertise in line with the authors' interests and career goals.

2.3.5 General public

As this thesis is part of a process that aims to ease the transition from traditional vehicles to large-scale deployment of AVs on public roads, the general public is also a stakeholder. Although they are neither end-users, academia nor part of Einride, they will be affected by the wide societal change that comes with the inevitable arrival of AVs. Ethical considerations need to be taken into account regarding the effects on the general public in terms of physical safety, privacy, and consent of being exposed to any risks associated with AVs. Some of these aspects will be discussed in the following section.

2.4 Ethical Issues

AVs and RO aim to provide a more sustainable future in terms of social, ecological, and economic sustainability. Specifically, the technologies should provide a more secure means of transport, decreasing accidents [4] and emissions [5], while allowing the infrastructure to be more efficiently used. However, when incorporating more technology into vehicles, new challenges and vulnerabilities appear. One such vulnerability is the risk of hacking [26]. This applies to both the hacking of singular vehicles, as well as the added risk of attacks on vehicle manufacturers' servers, which contain sensitive data and possibly access to a whole fleet of vehicles. As claimed by Tesla's CEO Elon Musk at the NGA 2017 Summer Meeting: "one of biggest risks for autonomous vehicles is somebody achieving a fleet-wide hack" [27]. Moreover, with no drivers present, there might also be an increased risk of physical attacks on the vehicles, such as vandalism or theft.

One important aspect of the widespread deployment of AVs is gaining the trust and acceptance of the general public. This might put strong requirements on the transparency of the underlying algorithms, which should therefore be prioritized. Moreover, Menon [28], argues that even if the AV operator has full insight into the system's complexities and has consented to the use and risks involved, road users such as pedestrians and other drivers might not have. Being the primary actors at risk it is important that they too are offered the required information and a choice to consent or not. However, it is worth noting that due to varying levels of technical knowledge, the levels of understanding of such information would differ greatly. In addition to gaining the consent of other road users, there is also a legal issue in terms of deciding accountability in case of an accident or other situations in need of holding someone accountable. Is it the passenger, the remote driver, or the company that is accountable in the event of an accident? This is further complicated as it might differ legally between regions.

One phenomenon always present with the advent of new technology is the first-mover advantage, the fact that the first company successfully launching a product gains a market advantage. As discussed by Menon [28], this commercial interest incentivizes companies to find shortcuts, such as not revealing known issues to the public, thus resulting in a less safe product. On the other hand, instead of covering up their issues, some manufacturers boast that their technology is more advanced than it is in an attempt to "hype" their product and generate more interest. To some extent, this is already happening with discussions online regarding the true level of autonomy of Tesla's Autopilot [29].

There are some additional general considerations regarding both AVs and teleoperated vehicles where one is that they both commonly utilize cameras as part of their technology. This means that people present in traffic will be filmed without their consent, and possibly without knowing. While in the case of teleoperated vehicles, this video is live-streamed to an operator, it is also highly likely that the video will be stored for later use, either for educational purposes or as data logs.

In conclusion, it is clear that there are many ethical aspects to consider when dealing with AVs and teleoperated vehicles ranging from issues regarding safety and privacy to trust and consent. Some additional issues worth noting include the "Trolley problem" [30] and while this thesis focuses on the use of RO within a civil use case, its potential use in military operations is undeniable. As such, this aspect should also be considered when further developing this technology.

2.5 State-of-the-art

RO of vehicles has successfully been used for decades in various contexts ranging from reconnaissance, combat, exploration, rescuing, and sample collections [11]. One of the more famous recent examples of this is NASA's Curiosity rover [31] which in 2011 was sent to Mars to collect research data. However, most previous applications of RO are under very specific circumstances for very specific purposes, often in hazardous environments. In contrast, the launch of widespread RO of vehicles on public roads is a completely different issue simply in terms of scale. While this research focuses mainly on the design of GUIs for RO of on-road vehicles utilizing direct interfaces, there is not a lot of material on the topic. Nevertheless, the encountered state-of-the-art is presented as follows.

2.5.1 TuSimple

TuSimple [32] is an American autonomous trucking company [33]. They are developing a commercial-ready truck that is currently operating at SAE level four. As such, they can operate without any human intervention under defined driving conditions but are operated with a safety driver behind the wheel to monitor the system and a safety engineer in the passenger seat.

TuSimple was the first company to launch an autonomous semi-truck on public roads without a human on board, interacting with other motorists [32]. To enable this, they have designed a number of tools and technologies, such as TuSimple Connect which is a proprietary monitoring system that allows for monitoring of the health of all vehicles in use [34]. It is also designed to be able to put any vehicle into a "Minimum Risk Condition" (MRC), such as pulling it into an emergency lane, taking it to a complete stop, and then calling for assistance. Furthermore, it also allows for diagnostics, and vehicle scheduling and is used for inspection before missions.

2.5.2 Roboauto

Roboauto is a Czech company [35] focused on developing automation and RO of vehicles, among other innovative technology. One such scenario where they apply RO is for shunting trains, a task explicitly requiring you to move several sets of trains

[36], which streamlines the process without having to physically move from train to train. Another scenario is a remotely operated taxi that arrives at the customer's door at a chosen time, allowing the customer to drive to their destination. Thereafter, a remote operator connects to the vehicle and either drives it to the nearest parking or to the next customer. Their technology has also been applied to other types of vehicles and they claim that their technology and software are "ready for integration into any vehicle" [36].

2.5.3 Phantom Auto

Another company that is also at the forefront of teleoperated vehicles is Phantom Auto [37] which focuses on logistics, especially forklifts and yard trucks. They utilize RO through a direct control interface to allow one person to control multiple forklifts, thus increasing the effectiveness of each driver, while also allowing for remote driver training [38]. This is useful in terms of being able to more easily share the knowledge and experience of more senior drivers and therefore more quickly being able to produce more experienced people. Furthermore, ITS ConGlobal [39], North America's largest integrated intermodal services provider has partnered up with Phantom Auto. They are currently using Phantom Auto's technology to both operate vehicles as well as to train new drivers remotely [38].

2.5.4 Teleoperation of Highly Automated Vehicles in Public Transport

In 2021 Kettwich *et al.* [40] published research on how to design a human-machine interface (HMI) for RO. Their research is tailored for highly automated shuttles that have achieved SAE level four, with the target audience being "control center professionals" within public transport in German urban mobility networks. The RO interface presented is a supervisory control interface utilizing waypoints, meaning that the operator controls the vehicle by directing it to a certain waypoint. Moreover, the paper focuses explicitly on two major functions that supervisory control entails; monitoring the situation and intervening if needed. Regarding monitoring, they identified the importance of keeping the operator in the loop and attempting to increase their situational awareness. Furthermore, when incidents occur and human intervention is required, the transition must be seamless. They also mention an important concept in RO, which is that the operator should physically feel present at the remote site, known as *telepresence*.

To conduct this study Kettwich *et al.* designed a GUI shown in figure 2.3, used for monitoring and remotely operating public transport from a control center when necessary [40]. The GUI was produced through a user-centered design process and was evaluated using questionnaires in combination with usability testing and a concluding structured interview. The evaluation was conducted with users that had expertise from working in control centers. Some of the conclusions were that the level of information displayed was satisfactory, without being disturbing, displaying a map was appreciated, and using head-mounted displays did not improve the driver's performance, but in some cases rather increased their error rates. Moreover, some of their proposed changes for the next iteration of the prototype included but were not limited to; displaying location, such as street name or building numbers, adjustable camera angles, and documenting actions taken. They further state that in this case, the workload was rather low, risking low vigilance. One potential solution presented was to aim for moderate vigilance through a medium workload by increasing the number of vehicles the operator is responsible for.

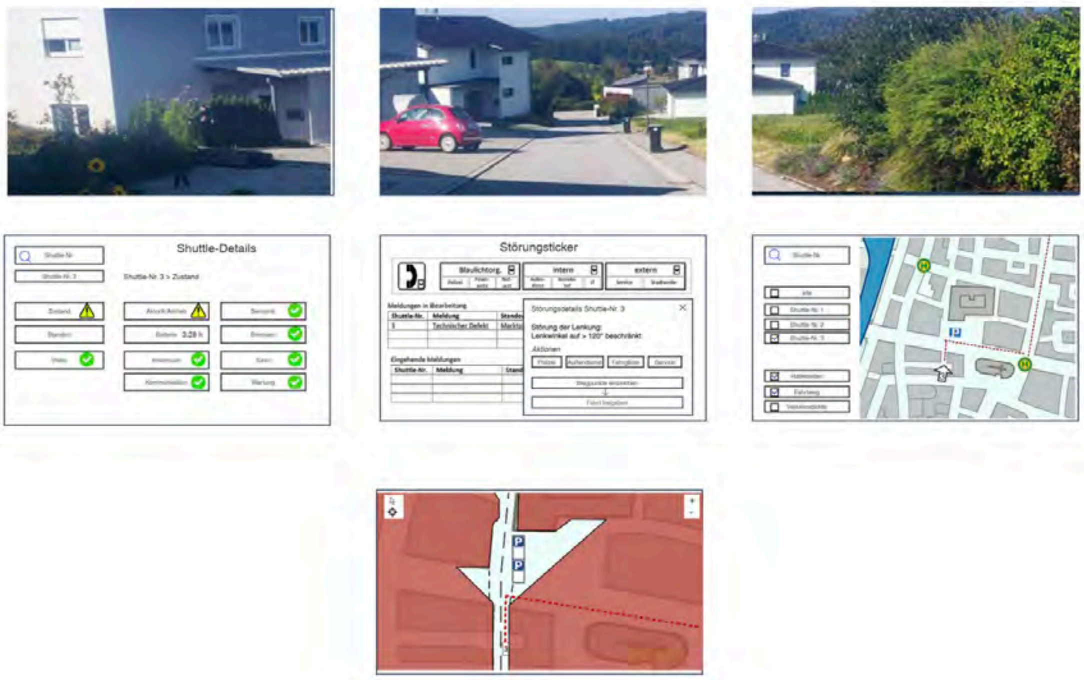


Figure 2.3: Initial prototype by Kettwich *et al.* using seven screens. The top three display video, the middle three represent the "details screen", "disturbances screen" and "map screen" respectively, and the bottom screen is a touchscreen and is used to set waypoints [40]

Kettwich *et al.* also propose some adjustments to their solution such that it can be utilized in other scenarios [40]. One such adjustment is that if the AV encounters an issue where the calculation of trajectories between waypoints is disabled, a remote operator is required to safely maneuver the AV through a direct control interface. Another scenario where adjustment is required is using the GUI for the transport sector where the proposed changes include: utilizing a separate screen that displays additional information about the customer, the route, and the goods transported. Additionally, the map could display logistic centers and customers instead of bus stops.

2.5.5 AstaZero

AstaZero [41] is a full-scale test environment used to research and develop various technologies for the future of road safety (see figure 2.4). The name is a combination of "Active Safety Test Area and Zero", referring to the Swedish parliament's vision for zero deaths or seriously injured in traffic [42]. AstaZero is owned by the Research Institutes of Sweden (RISE) [43] and Chalmers University of Technology. Furthermore, it is financed by a number of companies and organizations including but not limited to Volvo, Veoneer, Scania, and The Swedish Transport Administration [42]. The test site is specifically developed for research into active safety and provides opportunities to perform active tests in areas of for instance communication, automation, and road safety. Some examples of testing done at the facility are related to various types and levels of automation such as automatic brakes or skid-control, and how the introduction of electric vehicles changes accident situations [44]. AstaZero has five types of environments/test tracks as can be seen in figure 2.4, each specialized for a certain setting and set of tests.

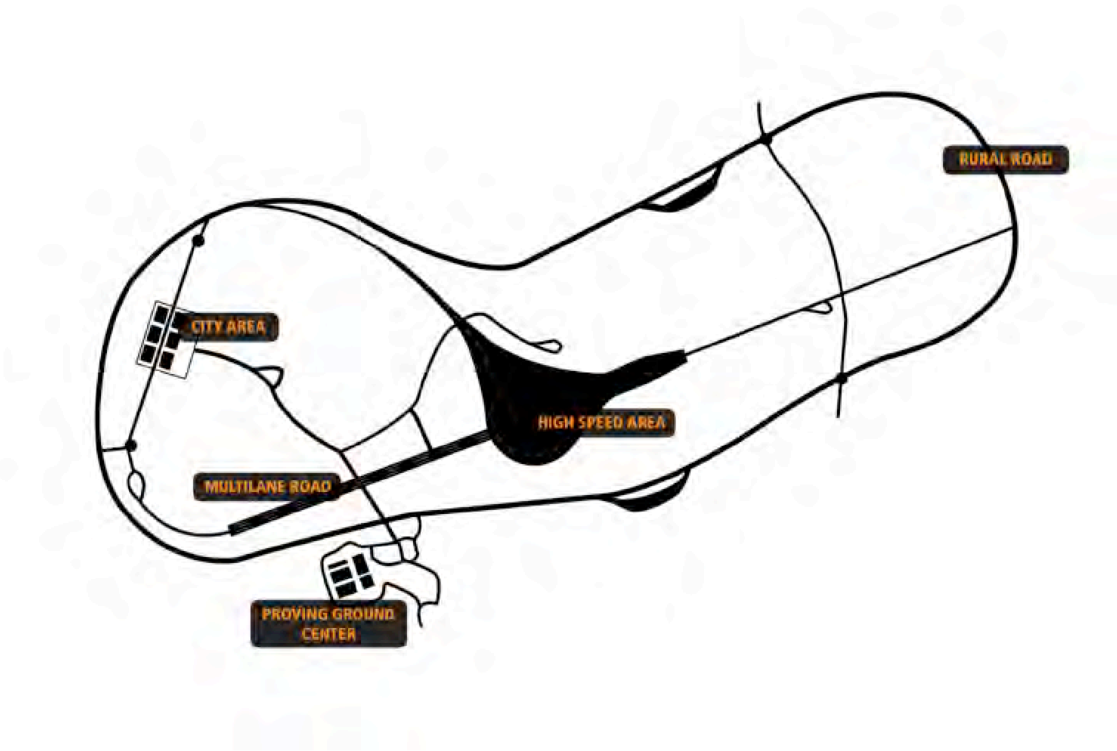


Figure 2.4: Map of AstaZero testing areas [45]. There are five test tracks.

3
Theory

In this chapter, relevant theoretical frameworks and concepts applied throughout the project are described, providing a solid foundation for the rest of the thesis. The first sections describe general concepts applicable to the design of digital products and information visualization, followed by how to apply user-centered design through designing for situation awareness and human-automation interaction.

3.1 Graphical User Interfaces

Graphical user interfaces have almost existed for as long as computers, providing the possibility of interacting with the computer through visual representation. Through the years interactions have existed in different forms, such as text commands through the command line interface (CLI), the “window, icon, menu, and pointing device” (WIMP), and lately, as gestures through the use of touch-screens [46]. With any form of GUI or interaction, there are some general concepts to consider including *Flow* and *Excise*, *Posture*, and accessibility which are presented below.

3.1.1 Flow & Excise

Flow and Excise are some of the most fundamental concepts to keep in mind when designing digital products. This is due to the fact that both concepts affect a number of different areas, such as efficiency, level of engagement, and user experience. Moreover, Cooper *et al.* [47] describe flow as being a state of mind in which a person can find themselves where they feel extremely productive and focused. They further argue that if the user could somehow reach their goals without the need for a product, they would. Furthermore, if a product is required, they would prefer to use it without having to interact with a user interface. As such, Cooper *et al.* argue that when a user interacts with a product, they should be face-to-face with their objectives, being unaware of the interface of the product. With this in mind, they remind designers that the best interface is no interface and a user interface is therefore inherently in the way of users reaching their goals. Thus, they suggest that the goal of designers is to minimize the work a user performs through the interface while moving them toward their goals. Consequently, the concept of excise is introduced, as any work that does not immediately move the user closer to their goal. Conclusively, the goal of the designer is to create a user interface that minimizes excise and keeps the user in a state of flow while avoiding interruptions that would break flow.

3.1.2 Posture

Cooper *et al.* [47] describe a product's "posture" as its behavioral stance, how it presents itself, and in which context it is to be used. A product's posture is determined by the amount of attention the user is expected to devote to the product. They further define three types of posture for desktop applications: *daemonic*, *transient*, and *sovereign* [47]. Daemonic applications, such as internet connection, do not normally interact with the user. Transient applications often present a single function and are utilized when needed, such as a calculator, often filling a supportive role to sovereign applications. Sovereign applications tend to monopolize the users' attention for long continuous periods of time.

When designing for applications with a sovereign posture there are a number of things to consider. Cooper *et al.* [47] specifically mention the fact that a user only spends a relatively short time as a novice, in comparison to the total time using a sovereign application. As such, its users will progress to become at least intermediate users making these the users to target. However, even if the target is intermediate users, powerful tools for expert users should still be provided, and making the application easy to use for new users increases the chance and speed of transforming them into intermediate users. Furthermore, as this type of application is usually used in its maximized state, such applications can and should utilize the full space required. Moreover, due to the fact that users often spend long periods of time using this application, it is recommended to be sparse in the application of color. This is done to give more emphasis to the areas where color is applied and to not contribute to visual load. Cooper *et al.* further suggest that you should utilize rich visual feedback, i.e., extra information about the state of the application that does not disturb the user but can enrich its experience once they have understood its meaning. Moreover, when using a sovereign posture application, the user is expected to be seated and fully focused on the application. This implies that higher demands can be put on the user and the designer can thus use more aggressive direct-manipulation idioms. Cooper *et al.* also bring up the aspect of customization, and how intermediate and expert users will want to configure the interface and optimize it for better performance. This is particularly important for sovereign applications targeted at users who spend the majority of their workday using a single interface.

3.1.3 Accessibility

WHO estimates that 1.3 billion people or 1 in 6 people experience serious disability globally [48], not accounting then for "non-serious" disability. The implication is that by considering the accessibility of a system, one can include the incredibly large section of people with disabilities that would otherwise not be able to use the system. Moreover, people with minor disabilities and people without disabilities will in turn likely see positive effects as well. On that note, the "Curb Cut Effect" is a metaphor that has been widely applied in the computer technology industry [49]. The term stems from a movement during the 1970s that had cuts installed in street curbs allowing wheelchairs to scale them. The term is used today to refer to the general willingness of people to alleviate issues of disability and the fact that accessibility features often bring improvements for everyone.

In addition to standing for the positive effects of accessibility, the term “a curb cut” is often used as a noun referring to any accessibility feature [49]. One could refer to the computer itself as a curb cut as it enables connection with the world without the need for much physical movement. Moreover, the application of RO for truck driving is in many ways a curb cut that situates the driver in a controlled environment. RO also opens up possibilities for working from home, further alleviating common physical disabilities. It is however important to note that any accessibility feature could spawn new unintentional issues. In the case of the curb cut, it had the unintentional effect of being a safety hazard for people with visual impairment [49]. This is an example of how accessibility issues can be complex and cannot be achieved by simply focusing on one kind of disability.

Cooper et. al. introduce the concept of “command modalities”, different modes through which a user can interact with an interface such as shortcuts, mnemonics, accelerators, or direct manipulation [47]. Some command modalities provide a pedagogical effect by way of descriptive text and the like. Through these “pedagogical modalities” users can learn about complicated commands before transitioning into more direct modalities (immediate and invisible commands). The path from the pedagogical modality to the more direct modalities is called a “memorization vector” and should be made clear for the user [47]. This implies that there should be a pedagogical modality with any immediate modality to aid learning. However, destructive commands are exempt from this rule, as they should be protected. By creating clear memorization vectors, users can create their “working set”, the set of commands that a user remembers and will use most often. The designer can thus try to anticipate the working set of most users by analyzing usage patterns or through scenario studies and subsequently provide easy-to-access immediate commands for these. In general, providing multiple command modalities for a critical command increases the chance for a person with a disability can access that command [47].

3.2 Information Visualization

Information visualization is a field of study that applies research from cognitive neuroscience to present data in ways that humans can quickly perceive [50]. As mentioned by Ware: "The ultimate goal of interactive visualization design is to optimize applications so that they help us perform cognitive work more efficiently." [50, p. 22]. Through learning from this field one can make better design decisions by considering the capabilities of human perception and cognition. This provides a theoretical base, grounded in the structures of visual perception, to the otherwise heuristically based field of interaction design [50]. In practice, this provides insights into details of interaction that would otherwise not be considered since humans are seldom fully aware of their processes of perception. Therefore, by utilizing information visualization, designers may understand their users on a deeper level, producing interfaces that facilitate more work in less time and with less cognitive load [50]. As such, important theories and considerations from information visualization are covered here.

3.2.1 General Considerations

One way of viewing an interactive information visualization, such as a GUI, is as a component of the cognitive system of human-computer interaction (HCI) [50]. In this cognitive system, the human brain acts as a prediction engine and decision-maker that largely relies on its visual system for input. By making use of the visual system, the user of an information visualization enters a visual thinking process where a query is formulated and a search initiated to recognize visual patterns. This puts emphasis on making the pattern recognition process as effortless as possible. Data can be represented in ways that aid or inhibit the recognition of patterns [50], and should thus be considered.

One aspect of visual search is the mechanical movements of the eye called "saccadic" eye movements, which according to Ware [50] is a "ballistic" movement that cannot be changed once it has begun. Each saccadic movement is a direct act of visual search. This implies that humans are less sensitive to input during visual search due to the saccadic movements required, something that is called "saccadic suppression" [50]. This results in important considerations for the design of for example alerts. To aid the visual system in its search some other relevant practical considerations include for instance, that content should be arranged so that subsequent saccadic movements are within five degrees of each other [50]. Furthermore, when presenting information through visuals, human sensory capabilities should be kept in mind to allow data elements and patterns to be more easily identified, i.e., be more salient. Additionally, the visual system, and the nervous system in general, is fundamentally designed to react to changes, spatial or temporal. Conversely, humans are bad at reading absolute light values which can lead to errors when reading information. Instead, information visualizations should display patterns of changes through space and/or time [50].

3.2.2 Glyphs & Symbols

Within information visualization, there is a general distinction between *symbols* and *glyphs*. Symbols are graphical objects that represent an entity [50], such as an icon on a map, or different shapes representing different types of data in a graph. Glyphs on the other hand represent both entity and quantitative value. A glyph should be easy to find, while also being able to in an accurate manner portray the resolution required for the quantities expressed. By representing data using glyphs one might also take into account a number of more specific considerations for effective search through glyph coding that manages the direction of attention.

3.2.3 Attention

Attention according to Ware is a set of processes that involve the entire visual system [50]. By moving the eyes and capturing a new image, information is processed through the visual system and ultimately sent to the brain. In addition to the flow of information to the brain, mental predictions are sent in the opposite direction, to the lower components of the visual system. This information tunes the visual system and provides direction for future eye fixations. This process is called attention and implies that eye movements are "literally acts of reallocation of attention" [50, p. 22]. *Preattentive processing* is the theoretical mechanism providing discernible points of fixation for subsequent redirection of attention [50]. Pattern and information search can be assisted through deliberately designing for preattentive processing, for example through careful coding. This is due to preattentive processing having constant speed regardless of the number of non-targets. Some of the strongest characteristics used for preattentive processing are color, motion, orientation, and size. When trying to utilize preattentive processing, it is often better to add than to remove asymmetry, such as <u>underlining</u> a specific word instead of having all text but one word underlined. Ware also suggests that motion can be substituted by the appearance of new objects to grab attention. Moreover, humans are in general bad at perceiving things that they do not expect, where they do not have a preconceived notion of what something looks like [50]. This means brief unexpected events are often missed, calling for the need to introduce elements to users before they might need to respond to them in a critical situation. Moreover, this is particularly important when designing user interrupts such as errors or warnings. Ware also presents four basic visual requirements for user interrupts, they should; be easily perceived even if outside the area of focus, keep signaling if ignored by the user, not be irritating enough that the device is unpleasant to use, and be able to convey various levels of urgency [50].

According to Ware, the human visual sense has a limited amount of *feature channels*, channels through which it can perceive information (see figure 3.1) [50]. These are form and texture, motion, and color differences, each of which can be further divided. To make information noticeable, it is effective to utilize channels that are distinct from the rest of the visualization. An example of this is using motion if a design is otherwise static, colorful, and employs many different shapes. In general, information coded using one channel will be easy to separate from information in another channel through selective focus [50]. This is also useful when trying to separate conceptually distinct information. Moreover, *redundant coding* is where more than one channel is used to code the same information and is encouraged to ensure a stronger reaction, as well as for accessibility issues such as color blindness.

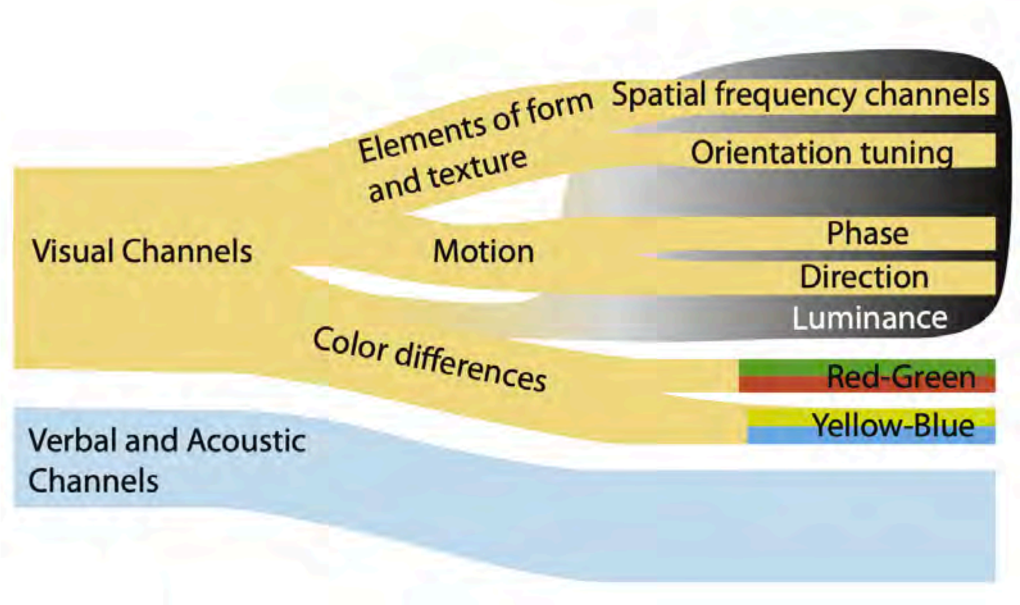


Figure 3.1: An overview of the visual channels [50], from left to right being further divided into elements of form and texture, motion and color.

3.2.4 Light & Color

Light is a highly central topic of information visualization since it is the basis for any visual information. *Luminance* is the physical measurement of the amount of light coming from a surface [50]. The study of luminance has a distinct role in information visualization which has a larger impact than color as the visual system is more sensitive to changes in the luminance channel. Correct use of luminance is therefore crucial for effective information visualization and as such comes with a couple of practical considerations. For instance, it is especially important to ensure contrast in luminance between a symbol and its background when it is small, finely detailed like text, or moving [50]. As a guideline, a minimum of 3:1 luminance contrast ratio should be applied for a clear distinction. One practical way to apply this is through a thin white or black border around a colored symbol. There are however limits to solely relying on luminance for coding information as it is too fundamental to perception to waste [50]. As such, Ware recommends that one avoids using grayscale to represent more than a few numerical values [50], and instead utilize color.

Figure 3.2: Chroma, Lightness and Hue [50].

The term *Chroma* is defined as the perceived vividness of color (see Figure 3.2) [50]. In addition to luminance, the "chrominance", i.e., the specific color of a symbol is useful in information visualizations. One application of color is as coding of information by providing visual distinction to symbols and glyphs. More important data or greater quantities should be made more visually distinct, such as by using greater chroma [50]. As a rule of thumb, Ware recommends lower chroma the larger the area it is applied on due to the fact that colors are more easily distinguished on larger areas. In contrast, small symbols should in general be coded using higher chroma. Furthermore, Ware recommends not using a sequence of chroma values to encode more than four values [50].

Color in general can be used for both numerical and nominal coding. Color is considered highly effective in nominal coding due to it inherently being perceived as categorical [50]. One can define nominal categories using colors by using hue as the denominator of a particular group membership while letting saturation and lightness represent secondary values [50]. Furthermore, the specific choice of color should also be considered. For instance, perceptually speaking, some colors are cross-culturally considered more valuable such as red, green, and blue, which therefore make for good nominal groups [50].

The fact that colors are associated with certain concepts and feelings is called *color semantics* and is culturally dependent. In Western society, color semantics usually include: green being associated with "go", safety, and vegetation; red being associated with heat, danger, and financial loss; as well as blue being associated with cold, melancholy, and calmness [50]. In China, however, red is associated with life and good fortune, while green can be associated with death [50]. Furthermore, certain combinations of colors into palettes have the potential to raise certain types of feelings in the user [51]. Color choices can therefore have a strong impact on the perceived seriousness of interface elements, and the interface as a whole. This in turn could affect the user's perceived urgency of the entire task which the interface facilitates.

There are some further pitfalls when it comes to designing with color. For one, peripheral vision is color blind [50] which encourages coding of peripheral objects to not solely rely on chroma. Furthermore, depending on an application's posture, color should be used in different ways [47]. For example for a sovereign application, strong colors should be used sparsely in the interface, while for a transient application strong use of colors is encouraged. When creating an interface centered around video, the video feed is often already rich in color. Accordingly, this puts importance on the deliberate and moderate use of color in the GUI to not induce visual noise and cognitive load. In addition, when designing for a company with an established brand this too must be taken into account when choosing colors.

3.2.5 Interaction

The concept of epistemic actions, is actions that are intended to uncover new information in interactive information visualizations [50]. Related to this is Shneiderman's mantra for how interfaces should present information and how interactions should support this. The mantra goes "overview first, zoom and filter, then details on demand." [52, p. 1]. Also related to interaction is the concept of feedback. According to Ware, a system should provide fast and clear feedback on the consequences of the user's actions in order to provide a better learning experience [50]. One way to provide feedback is through *cross view brushing* which is a technique where data is highlighted across many different views as a result of interaction [50]. Ware further mentions that if interfaces are designed to take advantage of previously learned interactions, the users will have an easier time executing their tasks. However, when designing for novel fields the possibility of utilizing completely new interactions needs to be weighed against using current best practices.

3.3 User Centered Design

In the current age of information, people are often more or less bombarded with information on a daily basis. Despite this, many feel less informed than ever [15]. This is true whether you are trying to navigate the subway, sitting behind a desk, or in the cockpit of an airplane. Thus, the problem is not lack of access to information, but rather access to the correct information at the correct time. User-centered design is, in contrast to technology-centered design, the design philosophy of molding the design around the capabilities and needs of the users [15]. Donald Norman describes user-centered design as design that "emphasizes that the purpose of the system is to serve the user, not to use a specific technology, not to be an elegant piece of programming. The needs of the users should dominate the design of the interface, and the needs of the interface should dominate the design of the rest of the system." [53, p. 61].

Endsley and Jones further describe a number of principles to consider when working with user-centered design. Firstly, the design should be organized around the user's goals, tasks, and abilities. As such, this type of design is suitable for complex systems, where the user is pursuing a number of varying goals over time, and where there are no clear sequences of actions available to reach these goals. Secondly, the technology should be organized in a way such that it supports the natural decision-making process. This decision-making process has been observed to consist of first assessing the situation given information about the current situation, whereafter an action is chosen [15]. Lastly, the technology must keep the user in control and aware of any changes to the state of the system. Specifically, this is important when some of the systems are automated, or semi-automated since if the user finds themselves out-of-the-loop, the risk for mismanagement increases.

3.4 Situation Awareness

To achieve user-centered design, one should strive for a high level of *Situation Awareness* (SA), as this is a practical way to fulfill the principles of user-centered design. As such, SA could be considered the key to user-centered design, and Endsley and Jones define SA as being “the engine that drives the train for decision making” [15, p. 11] (see figure 3.3). Endsley and Jones further define three levels of SA as: “*perception* of the elements in the environment, *comprehension* of the current situation, and *projection* of future status” [15, p. 14]. SA is studied in a wide variety of fields ranging from education to aviation and military operations [15]. Moreover, it is a central part of user-centered design and is used to define what information is relevant for a particular situation, job, or goal. They further mention that human errors are often related to a lack of SA. As such the error might not be caused by the human making an incorrect decision or having performed their tasks poorly, but rather that they have misunderstood the situation.

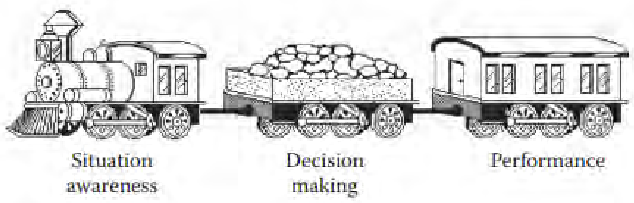


Figure 3.3: A visual illustration [15] of how situation awareness acts as the engine that drives the train for decision-making and performance.

Designing for good SA is difficult, and Endsley and Jones [15] therefore present principles to consider when designing for high SA as well as eight factors that might undermine SA. These range from attentional tunneling, requisite memory traps; workload, anxiety, fatigue, and other stressors (WAFOS); data overload, misplaced salience, complexity creep, errant mental models, and out-of-the-loop syndrome [15]. Some of the more important of these are out-of-the-loop syndrome and attentional tunneling. Out-of-the-loop syndrome happens when the user is uncertain of what the automation is doing, what it has done, or what the automation is intent on doing. Furthermore, attentional tunneling occurs when a user is overly focused on a single aspect of a system. Due to the narrowed frame of reference caused by this behavior, users often fail to catch themselves in this behavior and fail to see how detrimental it is to their overall performance [15]. On a similar topic, Ware [50] describes a concept called "useful field of view" (UFOV), as the size of the region from which a human can quickly take in information. Ware further describes tunnel vision, a comparable concept to attentional tunneling, where a person narrows their UFOV during stressful situations, which depending on the situation can be between 1-15 degrees. As such, critical information should stay within such a range from the user's central area of attention. However, a greater visual load has the effect of decreasing the UFOV which puts great emphasis on making sure that only critical information is within range and salient. Similar to these concepts is *Change Blindness* which according to Ware [50] is a phenomenon where big changes can go unnoticed unless the viewer has recently paid attention to that area. This phenomenon can also occur if changes coincide with a saccadic movement or blinking. Lastly, Endsley and Jones describe

3.4.1 Situation Awareness in Automation and Teleoperation

When working with automated systems, Endsley and Jones [15] note a number of things to keep in consideration for high SA. One of these is that greater autonomy leads to less need for the operator to act, which usually reduces vigilance and in turn the operators' SA. Conversely, lesser autonomy, or less reliable autonomy leads to the operator having to remain active and thus increases SA. Another phenomenon to keep in mind is the potentially rapidly varying pace of work, described by Sheridan and Parasuraman as "hours of boredom punctuated by moments of terror" [54]. Another issue when monitoring automation and keeping aware of sporadic events is that passive processing results in a worse performance than actively performing actions. Endsley and Jones compare this to riding a car from A to B and driving a car the same distance and then being asked to reproduce the route between the two locations [15]. In general, Endsley and Jones [15] emphasize that while the operator does not have to perform all tasks, they should still be in control of managing the system in order to maintain a high level of SA.

When dealing with teleoperated autonomous systems the complexity of achieving SA is further increased. As RO is performed at a distance, the issue of latency is constantly present. Gnatzig *et al.* argues that higher latency and the variability of latency affect the possibility of performing dynamic maneuvers at higher speeds [6]. Moreover, they also mention that a lack of SA produces a non-realistic driving feeling, i.e., a lack of telepresence. This could also lead to a decreased feeling of urgency, immediately affecting the performance of the operator while also making it harder for them to empathize with their fellow road users [55]. Furthermore, the operator is completely reliant on sensor data and the quality of said data when teleoperating, and is therefore exposed to environmental effects such as poor lighting, dust, etc. Moreover, Endsley and Jones [15] suggests that at least three cameras should be used to provide a good enough field of view. Lastly, they also recommend that the camera's directions and positions are made clear so that the operator does not confuse the effects of their actions based on the camera angle. Gnatzig *et al.* [6] also mention that designing a system for RO can be compared to designing a driving simulator and inspiration could therefore be found in similar environments such as games or driving simulators.

3.5 Human-Automation Interaction

Sheridan and Parasuraman introduce the concept of *Human-Automation Interaction* (HAI) as circumstances where a human is not simply a passive benefactor of automation, but where they: specify the automation's task goals and constraints as well as trade-offs between these, control the stopping and starting of the automation or otherwise modify its execution, and receive information, energy, physical objects or substances from the automation [54]. Designing for HAI is a complex matter and accident investigators have for decades been trying to gain a better understanding of the interplay between humans and automation through researching automation-related incidents, mainly from aviation [54].

Part of sub-par operator performance and a common cause of accidents is unsuitable levels of trust. The operator trusts automation in proportion to its reliability in relation to their self-perceived capabilities, choosing to rely on what seems best [56]. According to Sheridan and Parasuraman, over-trust and over-reliance, a.k.a. complacency, can lead to poor vigilance and subsequently poor SA which might cause the operator to miss when the system malfunctions [54]. Conversely, under-trust in the system could lead to excessive vigilance, putting an unnecessary strain on the operator's cognitive load which too can lead to reduced SA. This implies that trust levels have to be balanced depending on the situation [56]. Accordingly, one way of managing trust is by portraying uncertainty, which is difficult since data portrayed on a screen inherently tends to persuade people into thinking it is accurate [50].

In order to design an HMI that makes full use of the strengths of the human operator and automation system, one can approach the system as a collaboration. The concept of SA is usually centered around the human operator but could also refer to the system's SA of the operator [57]. This way, not only should the design be created with the intent of improving the operator's understanding of the automation but also by making sure that the automation understands the operator's goals and needs, behaving accordingly. This could be argued for as an accessibility perspective [57] as well as in general to achieve good collaboration. The system should then consider the operator's goals, in addition to the higher goals of the operation, when executing commands.

Taking the collaboration perspective further, by viewing the system as a social agent, one can improve performance and experience further through good social etiquette for computers, i.e., *computer etiquette*. Parasuraman and Miller describe good computer etiquette as being "non-interruptive" and "patient" [56]. They further posit that computer etiquette is likely to have a positive impact on both the operator's diagnostic performance and their trust in the system. An important factor in this is how etiquette that emulates a trustworthy person or an enjoyable interaction might increase trust. Furthermore, the effect of good computer etiquette is described as positive regardless of whether the advice provided by automation is reliable or not. Parasuraman and Miller also mention that good computer etiquette seems to compensate for advice with low reliability to the point of almost achieving the performance of a system with high reliability and poor etiquette [56]. However, it is noted that the highest performance and trust are achieved as a combination of high reliability of advice and good computer etiquette.

One key area of human-automation collaboration is found in the handover process when control is transferred from the human operator to the machine or vice-versa. This transaction needs to be clearly signaled, be frictionless, and occur in moments that support the goals and needs of the operator, in order to compensate for the flaws of automation or of the operator. Sheridan and Parasuraman mention that unpredictable automation behavior can severely hinder an operator's SA by taking them out of the loop [54]. Moreover, by involving the operator in decision-making regarding when to use automation, unpredictability might be lowered at the cost of raising the cognitive load.

4

Methodology

This chapter describes potential frameworks, methodologies, and methods for this thesis. The first section introduces how design can be utilized in research, followed by a description of the problem space, and ends with two sections describing methodologies for design and software development. In the chapter following this, some of the presented methodologies are chosen and argued for.

4.1 Research through Design

Research Through Design (RtD) is a model for conducting design research through the processes and skill sets normally applied in practical design. RtD is situated in HCI research and is by Zimmerman *et al.* described as a method for uncovering design knowledge for both design practitioners and researchers [58]. While design practitioners traditionally focus on making commercially viable products, through RtD they may instead apply their design knowledge in order to produce the "right" thing, i.e. an artifact that transforms the world from its current state into a preferred state. Moreover, this allows design research to utilize the designers' strengths to solve under-constrained problems, i.e., wicked problems [58]. Through this process, a design artifact is created that can be analyzed by the research and practice community as one solution to the undertaken problem. The artifact represents a possible manifestation of the combination of technical possibilities with theoretical knowledge. The idea is that many such artifacts produced within a problem area will start to reveal problem-solution patterns [58]. Furthermore, the process has the possibility of rendering insights into potential technical developments as well as feedback on the behavioral theories and models behind the design such as guidelines.

Zimmerman *et al.* provide four criteria or lenses for evaluating the research contribution produced by an RtD project [58]. Firstly, how clearly the process has been documented is critiqued where it should be possible to reproduce the process. There is however no expectation that two projects following the same design should reach the same conclusion. Furthermore, the selected methodology applied through the process needs to be motivated. Secondly, the contribution needs to display a level of invention, and as such requires that an extensive literature review has been performed. Thirdly, the work needs relevance by motivating why the proposed solution brings about a preferred change. Lastly, the contribution requires extensibility, allowing for future research and practice to make use of and improve upon the insights produced. RtD is an overarching model to contribute knowledge for design researchers and practitioners by utilizing a design process.

4.2 Wicked Problems

Wicked problems was first described by Rittel and Webber [59] when dealing with problems encountered in the field of social policy. These problems were described as "wicked", in contrast to traditional "tame" science problems, as they do not have any definite, objective solutions. Due to wicked problems as a concept often being difficult to grasp, another definition of wicked problems is provided by Zimmerman et al [58], who describes them as "a problem that because of the conflicting perspectives of the stakeholders cannot be accurately modeled and cannot be addressed using the reductionist approaches of science and engineering". Rittel and Webber have further defined ten characteristics of wicked problems, such as there being no definitive formulations of wicked problems, they have no stopping rule; they are not true-or-false, but good-or-bad and there is not a set amount of possible solutions [59]. Moreover, they are exceptionally hard to define and therefore often required to be redefined due to input from different stakeholders. Thus, such problems are often approached through an iterative process. Problems found in design practice are often inherently wicked, calling for an iterative process that tackles these challenges.

4.3 Design Thinking

Following a design methodology helps bring structure to design work, providing order and emphasizing the use of processes that aid in tackling wicked design problems. The term design thinking is inconsistently used in literature to refer to either a specific methodology or a philosophy to generally approach wicked problems in a "designerly" manner using creativity [60]. Considering the latter definition, there exists a multitude of concrete design methodologies that follow design thinking, including *Design Council's Double Diamond* [61], *Google's Design Sprint* [62], and the Institute of Design at Stanford (d.school) methodology for design thinking [63]. The methodologies are highly similar as they incorporate central aspects of design thinking such as being iterative, optimistic (solution-oriented) and human-centered [60].

The double diamond methodology is formulated around the concept of divergent and convergent thinking. The problem space is first explored, diverging the thinking, whereafter the problem is defined and the problem space is converged. The process then enters divergent thinking again by developing a breadth of ideas that are evaluated and selected by once again applying convergent thinking. The process thus forms a focus of thinking symbolizing a double diamond, hence the name, which puts emphasis on when to think in divergent or convergent terms. Google's design sprint is a methodology that is highly similar to the double diamond, focusing on divergent and convergent processes [62]. However, instead of two "diamonds", the methodology is split into three, where specifically the last diamond: develop and deliver has been split into two: sketch, decide; and prototype, validate. This puts emphasis on an evolution of prototype fidelity, where sketches must be included as a first prototype. Henceforth, the specific methodology popularized by d.school will be referred to as just "Design Thinking", although it is in actuality a methodological manifestation of design thinking.

Similarly to the previously mentioned methodologies, *Design Thinking* [63] is defined as a methodology consisting of a non-linear iterative design process that structures design work in order to solve wicked design problems. The approach applied through Design Thinking is user-centric, creating results that cater to the needs and wants of the end user. The iterative nature of design thinking is behind much of its power in solving complex problems. Dam [63] describes Design Thinking as consisting of five phases: *Empathize*, *Define*, *Ideate*, *Prototype*, and *Test* (see figure 4.1). For each phase, there exist design methods that are specifically applicable. Correspondingly, each phase is presented with some examples of appropriate design methods.

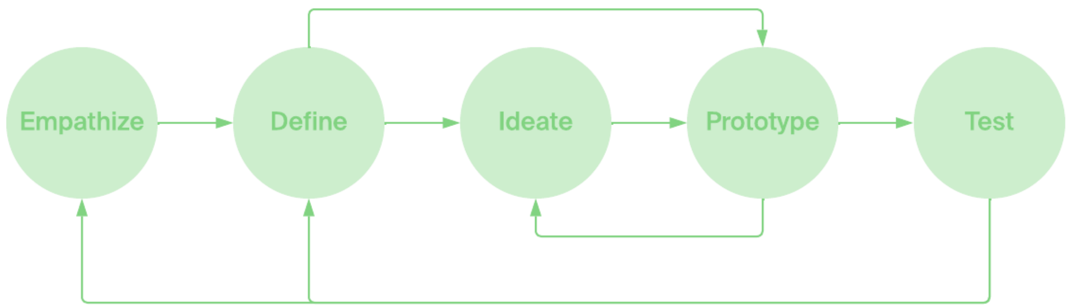


Figure 4.1: The iterative process of Design Thinking [63]. The figure shows examples of how each phase can loop back to any previous phase.

4.3.1 Empathize

The Empathize phase is focused on generating an understanding of the problem space by close observation and empathizing with the user [63]. It is also an opportunity for the designer to set aside their own biases. Consequently, this phase is suitable for methods either directly interacting with the target audience, and as such gaining insights into their needs and wants, or utilizing existing knowledge of the target audience and problem area to jump-start the design process.

Some suitable methods for empathizing with the user are literature reviews, interviews, and questionnaires [64]. Literature reviews provide relevant theoretical insights into a design problem. The drawback is that they do not allow for interaction with the target audience and a certain amount of caution should be exhibited in regard to the gathering of sources. Interviews on the other hand allow for direct interaction with the target audience, experts, or other stakeholders, generating qualitative data. However, interviews risk generating biased data due to the interviewer or, in the context of group interviews, due to participants influencing others' opinions. Furthermore, interviews are often time-consuming [64]. Questionnaires are a method capable of generating both quantitative and qualitative data, simple to produce and administer, and effective at gathering information [64]. Designing them is however often complex in terms of choice of questions, wording and response options to avoid biases and detailed results that need time-consuming interpretation. Depending on the target audience, getting a big enough sample size might also prove difficult.

An additional way to perform primary research is by utilizing the ethnographic method of field trips [65]. They allow for direct interaction with stakeholders and observation of how they interact with a product in-situ. Furthermore, they allow for the exploration of areas that are otherwise unavailable or unexplored through for example literature and also provide an opportunity to conduct other types of primary research, such as interviews. Consequently, they are especially effective at allowing the designer to empathize with the stakeholder in a realistic context. However, field trips are often time-consuming, require planning, and might be costly. They are also to be considered supplemental to other types of research, such as literature reviews, or other types of user evaluation [65]. As such, special care should be taken when considering utilizing field trips.

Secondary Research [66] is a concept that can be applied to strengthen and introduce new considerations to first-hand user-centric research, leveraging existing theories and research findings from secondary sources. This can take the form of literature reviews of contemporary research, statistical analysis, or any other method that produces knowledge of the specific design challenge or broader context. The emphasis of secondary research is on understanding the state-of-the-art theories, backgrounds, or solutions behind similar problems. Secondary research is no substitute for first-hand research but is important in order to strengthen both the understanding of the problem and the argumentation of solutions.

4.3.2 Define

The Define phase is focused on processing the insights gained from empathizing to find patterns, distilling insights from the previous phase to find the core problems for the user [63]. As such methods applied in this phase focus on processing information. The goal of the define phase is to define the problem space to act as direction for future ideation [67].

In order to find patterns and define the problem space there exist a multitude of helpful methods. For instance: clustering of information through *Affinity Diagramming* [64], creation of *Personas* [64] through *Empathy Mapping* [64] or IDEO's method for defining the target audience [68], and finally defining a set of *Impact Goals* [69] in order to frame the design problem [70]. Affinity diagramming is a common method for externalizing thoughts and insights and clustering them based on affinity. The method assumes that there is a set of highly contextual research data from observations or requirements that need to be organized. Both IDEO's method for defining the target audience and empathy mapping are methods for getting the design team to define the target audience. The methods ask the design team questions about the target user, which are then answered based on results from user research and literature review. IDEO's method focuses on shifts in dynamics surrounding a user such as their behavior and the community, services, and institutions around them to achieve their goals. In contrast, the questions posed when creating empathy maps focus more specifically on the user's actions and feelings. Behavioral patterns discovered in these methods can be used for the creation of personas. Personas are profiles of behavior defined from observations and research data, providing a human reference for communication with stakeholders and empathizing during design [64]. Furthermore, a structured method for defining a specific problem statement or "design challenge" is useful. In order to define the problem statement there needs to be a consensus among stakeholders regarding what impact should be achieved. Impact goals can be defined by organizing the design team's wanted impact from immediate and easy to long-term. From this, a long-term impact and a near-term key outcome can be defined by selecting from the top and bottom of the list respectively. By taking into account the long-term impact and near-term key outcome as well as the identified target group, the problem statement can be framed, serving as the basis for subsequent ideation.

4.3.3 Ideate

In the Ideate phase ideas are generated that potentially solve the problems defined in the previous phase [63]. Some methods useful for generating ideas are; *Brainstorming* [71], *How Might We* [72], and *Crazy 8's* [73]. The most recognizable of which is perhaps brainstorming. There are multiple variations of this method, each conducted in different ways. However, the core of the method is to produce a big quantity of ideas in an organized way. One of the keys for this to work is to involve a handful of people, such that it is possible to continue building on others' ideas. Additionally, it is important to make sure that this is done in a positive environment such that ideas keep flowing instead of "bad" ideas immediately being shut down, thus creating a less creative environment [71]. Instead, wild ideas are often encouraged in an attempt to produce a more creative environment.

Another rather similar idea is How Might We [72] where you utilize any insights of themes from previous phases. These are then re-formulated in a way such that they start with "How might we", followed by a problem statement, such as "How might we create a sense of safety in public toilets?" [72]. This act proposes that there is a possible solution, and the questions are used as frames for further ideation. This method can either be used by itself or as a base for starting a brainstorming session.

Crazy 8's [73] is a design method utilizing fast sketches to generate a multitude of ideas. It forces all participants to sketch eight ideas in eight minutes. This method emphasizes creative and "crazy" ideas and puts less focus on sketch quality. The main idea is that each participant has to create distinct ideas, pushing the participant's creativity more for each sketch. Doing this lowers the barrier to creativity and allows for all types of ideas to be put forward. This idea does however create an abundance of ideas, and some kind of quality screening should be made after this method has been used.

4.3.4 Prototype

Once ideas have been generated these are then conceptualized in the Prototype phase [63]. During this phase prototypes of varying degrees of detail, depending on the state of the project, are created and tested internally or externally. While there are various ways to produce prototypes, the method used is often based on the quality of the prototype, i.e., how much time is spent on creating it. They are further often defined as low, medium, or high-fidelity prototypes [74].

Low-fidelity prototypes are often made of paper and are a cheap, efficient way of creating a prototype that can quickly be iterated upon and disposed of. Paper prototypes also allow for creative thinking as the design is visibly unpolished. However, due to their simplicity and lack of detail, the user must often imagine how they might interact with the product [74]. One way to combat this is to utilize the *Wizard of Oz* [64] method to test the prototype and its interactions. Wizard of Oz is done by creating a low-fidelity prototype such as a paper prototype or a digital prototype whose functionality is simulated by the "wizard", who acts as a proxy between the tester and the system. This puts emphasis on the "wizard" to be consistent with respect to timing and system logic [64]. Done right, a user can experience the product before it is fully developed, thus producing interaction insights and more effectively spending valuable development time.

Wireframing [75] is a process that can be used to experiment with the flow and layout of a product early in the design process. Wireframes allow for experimentation of informational hierarchy, by prioritizing the display of functionality, as well as the general behavior of the product. Although the primary purpose of a wireframe is to quickly test design solutions with respect to functionality, layout, and navigation, they can be created with varying fidelity. By refining a wireframe with higher fidelity visuals and realistic interactions, wireframes can become engaging and allow for testing the intended interactions of the user. A higher-fidelity wireframe can therefore also be evaluated for its usability as it starts to resemble the finished product.

The fidelity of a prototype is not binary, but rather a spectrum. A high-fidelity prototype is defined as being high fidelity in relation to a lower-fidelity prototype, and can endlessly be further developed, becoming an even higher-fidelity prototype. As such, it is essential to note that the higher fidelity, the higher the cost, and the more it looks like a "real" product, the more superficial the feedback is likely to be. Furthermore, receiving feedback on a high-fidelity prototype on issues regarding core aspects of a product will require much more work to fix than had the prototype been of lower fidelity.

Another example of a prototyping method is the creation of *Storyboards* [64]. They can provide a visual narrative in a realistic context that allows the designer to empathize with the end users. A storyboard is a simple visual story, often presented in panels with supporting text that display a user in a predetermined scenario using the designed product to solve a problem. The level of detail required for a storyboard is very low and is only required to effectively point out the relevant details for stakeholders. As such, they are often used as a type of low-fidelity prototype. Providing too much detail risks shifting focus to unimportant details, rather than the story that is being portrayed. By providing a suitable level of detail, storyboards can quickly be produced to communicate and evaluate a use case for the design. This method also requires the team to first phrase a possible idea, and then express it as a storyboard, and as such might be better used in combination with other methods.

In the field of software development, there is a concept called *Minimum Viable Product* (MVP) [76]. The MVP is a software product that contains the minimum amount of features, while still being a working product that can be put in the hands of users. As such, it can be compared to a very high-fidelity prototype in the design process that is used for testing purposes. These are however costly to develop, and should only be implemented once other types of prototypes have been created and tested. Moreover, these should be designed with extreme care, such that work put into their development can be extended to avoid re-work at a later stage.

4.3.5 Test

Finally, in the Test phase prototypes are evaluated with target users to give feedback on improvements of the solution [63]. Feedback received during testing can direct the process back to any of the previous phases. In this phase, both qualitative and quantitative tests might be appropriate depending on the product that is developed. *Heuristic Evaluations* [64] is an example of a qualitative test method and *A/B Testing* [64] is an example of a quantitative test method. *Usability Tests* [64] can be classified as either, depending on how their execution.

Usability tests focus on finding empirical evidence of how to improve an interface. As such, a user is observed using a product and walking through a set of tasks or a scenario. The tasks and scenarios are designed with the typical end-user's goals in mind. During the tests, users are often encouraged to use the *Think-aloud* protocol [64], implying they should think out loud to provide more insights. This method is good for gaining insights from users while they are performing more or less realistic tasks. However, the results are in addition to being affected by the test validity and avoiding bias directly impacted by both the number of participants, as well as the number of evaluators [64].

A/B testing allows designers to compare two versions of a design. This is done by assigning test participants to either use design "A" or design "B", and then evaluate the results based on their performance in terms of reaching an objective [64]. This allows you to better understand which design is quantitatively performing better. However, it does not necessarily allow you to understand why it is so. Moreover, the method has to be performed with enough people to get a statistically relevant sample size, and as such requires a lot of time.

Heuristic evaluation, in contrast to usability tests, requires no actual users, but can instead utilize members of the design team as participants. During a heuristic evaluation, the product's interface is evaluated using best practices or principles such as Nielsen Norman Group's Ten Heuristics [77]. This allows for testing of the interface at an early stage, without involving users, and is thus quite flexible. This method allows for finding critical usability issues early in the design process to avoid the design being created based solely on personal preferences. However, it will most likely not provide any breakthrough insights but will rather make sure that the design is following best practices.

4.4 Agile Software Development

In addition to design methodology, the software development community has its own share of methodologies including *Waterfall* [78], *Spiral* [79], and *Agile* [80]. Agile is arguably the most popular approach in modern software development having widely replaced the other methodologies as it has proven highly effective due to its flexibility and ability to produce high-quality products [81]. The Agile concept is defined by a manifesto that dictates values and principles [80]. Agile aims to improve software development by producing higher quality products, lowering time spent on unnecessary tasks as well as managing complexity and change. There are many frameworks that in more detail specify the project proceedings while following agile principles. These include: *Kanban*, *Crystal*, and *Scrum* with Scrum being one of the most popular, largely due to its simplicity [82]. In Scrum, work is divided into bite-size tasks called “User-Stories” which are sorted by relevancy by a product owner to form a “Product Backlog” [83], often organized through a “Scrum board”. The product is continuously delivered through multiple incremental and iterative deliveries called “Sprints” where a number of user-stories are chosen and completed. A sprint can be seen as a small project that takes place during a fixed amount of time, often two weeks. At the end of each sprint, the team and stakeholders evaluate the process and work produced during the sprint. Each sprint is also associated with software testing and preferably continuous involvement of stakeholders in decision-making. At the beginning of each sprint, a sprint planning meeting is held that solidifies the why, what, and how of the upcoming sprint. In order to keep track of and adapt progress a “Daily Scrum” meeting is held where each developer shortly describes their progress. Finally, at the end of each sprint, a “Sprint Review” meeting is held where the outcome of the sprint is reviewed as well as a “Sprint Retrospective” where the process and tools are reviewed in terms of quality and effectiveness. Traditionally, a Scrum team consists of a number of roles: a *Product Owner*, a *Scrum Master*, and developers.

5

Initial Plan

This thesis was planned as another iteration of the already initiated design process started by Flising & Widerstrand [12], who recently conducted a highly recent and relevant master's thesis at Chalmers University of Technology. In an attempt to extend their work, this thesis intended to focus on the latter parts of the design process. The work was planned to be conducted through the development of a software prototype, partly due to the authors' interest and background in software engineering, and partly due to the process demanding prototyping of higher fidelity.

As the experience and effectiveness of a GUI changes depending on the context, it is impossible to arrive at a single objectively "best" design. Therefore, in order to provide one solution to this wicked problem, the plan was to conduct a pre-study that largely builds on the work of Flising & Widerstrand. In their thesis, multiple primary research methods were conducted with the same target audience as this thesis, i.e., the remote operators of trucks. This allowed for secondary research to be the planned primary method for empathizing with the user in this project. This research was primarily going to be that of literature reviews, focused on related state-of-the-art theory and design practice; as well as the material provided through the work of Flising & Widerstrand and video material of remote operations conducted by Einride. While not strictly planned, the authors hoped to be able to conduct some kind of primary research along the way, such as a study visit to AstaZero where Einride conducts regular testing of their remote operations to gain some first-hand insights. Moreover, some form of informal interviews and discussions with the remote operators and other relevant individuals within the field would also possibly be conducted. Furthermore, Einride's remote operation systems and general software requirements were planned to be researched early as these were intended to act as the basis for implementation. After this, a design was to be created in Figma [84] or a similar program by adhering to the knowledge gained through the research conducted. The design was then planned to be implemented into a software prototype on which formative evaluation was going to be conducted, either with end-users in as close to real circumstances as possible or, if not possible, through heuristic evaluation and expert analysis.

The prototype was planned to be implemented using pre-recorded video and data logs that simulate the driving experience in an attempt to create a dynamic GUI displaying all relevant information and interactions conducted without actually requiring a real truck. The knowledge gained through research, designing, and evaluating a GUI for the remote operations of trucks was then to be condensed into a set of guidelines. This initial planning resulted in the time plan seen in figure 5.1, which was later revised by the end of the initial pre-study or research phase taking newly gained information into account.

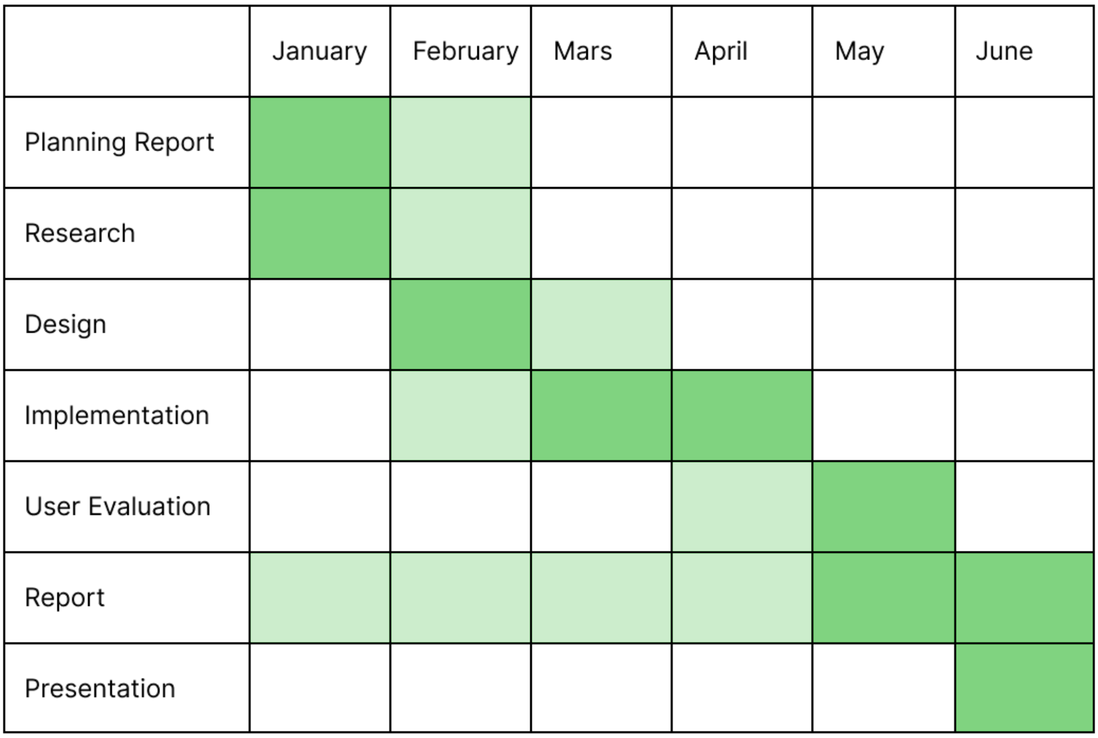


Figure 5.1: The initial time plan. The darker color displays the main focus of the month, and the lighter color indicates possible focus. Note that only the first week of June is part of the planning.

6

Process & Execution

This chapter describes the process, following the *Design Thinking* and *Research Through Design* methodologies. The chapter starts with describing the early phase of empathizing with the users through a pre-study which resulted in a revised plan for the rest of the project. The chapter continues to describe in detail the two major iterations, each part of the process is followed by some intermediate results and reflections.

6.1 Pre-study

In line with the initial plan for this pre-study, most of the time was spent conducting a literature review. To empathize with the target audience, significant effort was put into studying the primary research conducted by Flising & Widerstrand [12]. In addition to empathizing with the target audience, research into relevant theoretical fields was conducted to form a better understanding of the problem space. This included broad fields of study relevant to the design of GUIs such as information visualization, and accessibility as well as established best practices of GUI design. In addition to these general topics, further research was conducted into fields that are more specifically relevant when designing for remote operations. Some inspiration for relevant topics was found in the thesis from Flising & Widerstrand, which included the fields of SA and HAI. Subsequently, the main keywords used for conducting this part of the secondary research were; Situation Awareness, Automation, Human-Automation Interaction, Teleoperations, Remote Operations, Information Visualization, Accessibility, Graphical User Interface Design, or any combination of these. In addition to this, the results were filtered for the age of the article, as well as their peer-reviewed status and publisher. For more established fields such as GUI design, authors such as Alan Cooper [47] and Donald Norman [53] were used.

To gain some first-hand insights into the field on which this thesis is focused, a study visit to the test environment AstaZero was conducted. This was done as a one-day study visit during a day when new remote operators were being trained, and the authors thus gained insights into how the system was designed. This allowed the authors to gain a better understanding of the technical requirements and specifications of the design as well as the requirements and demands of the target audience. At the end of the pre-study, a more detailed plan was created for the upcoming phases of the design process and the project as a whole.

6.1.1 Intermediate Result: New Plan

During the Pre-study, an increased understanding was gained of the technologies and methodologies suitable for this project, and as a consequence of this, the initial project plan was updated with a more specific methodology in mind to include greater detail (see figure 6.1). The overarching design methodology chosen at this stage was *Research through Design*, which set a general framework for the project that incorporated *Design Thinking* [63]. This methodology assists in both moving toward a solution while also iteratively defining the goal of the solution. As such, it is well equipped to tackle under-defined and rapidly changing problem spaces, such as in the emerging markets of remote operation and AVs. As described in section 4.3, the *Design Thinking* methodology consists of five phases. At this time of the project, the plan was to conduct two iterations of the design process, with the first iteration consisting of all five phases where this pre-study was considered the first, i.e. the *Empathize* phase. The second iteration would on the other hand only consist of the final two or possibly three phases, as can be seen in figure 6.1. Both the guidelines and the prototype were intended to focus on providing an appropriate degree of telepresence and SA, such that operators make informed decisions that portray a suitable weight in comparison to the consequence of an interaction. This focus was chosen as this would protect fellow road users and pedestrians and would contribute to a more safe launch of AVs. Moreover, the report was planned to be worked on in parallel with all other phases and iterations while being the primary focus of the final few weeks.

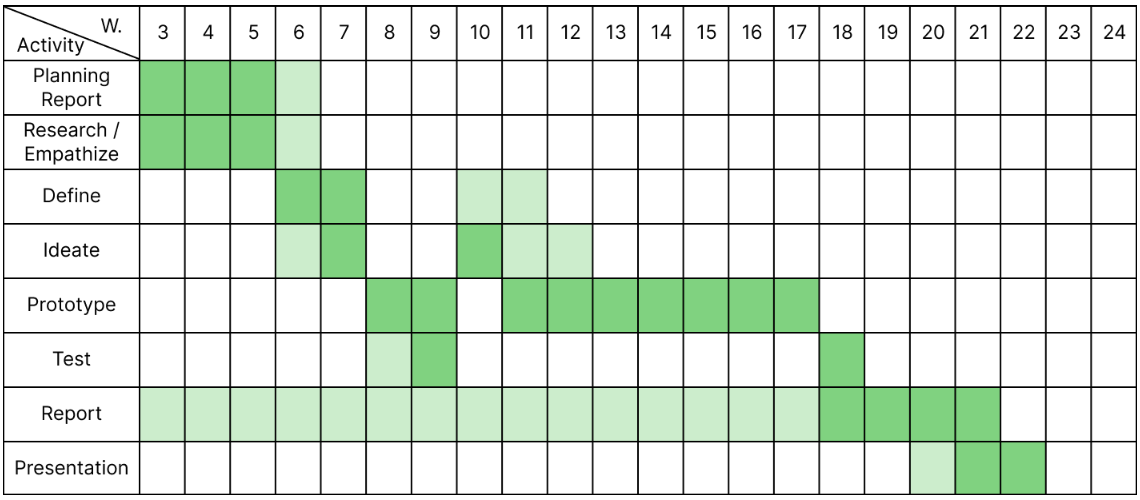


Figure 6.1: Updated planning. The darker color is displaying the main focus of the week, and the lighter color indicates possible focus.

Plan for the First Iteration

The first iteration was intended to consist of all five phases, where the conducted pre-study was considered the initial *Empathize* phase of the first iteration. Therefore, for the planning of the Empathize phase, the reader is instead referred to the planning of the pre-study in chapter 5. The *Define* phase was planned to be conducted using *Affinity Diagramming*, in order to process the information from the previous phase.

This particular method was chosen since a target audience was already defined, and it allows for the clustering of information based on affinity. It could therefore be utilized for comparing and combining insights from many different sources. As such, this phase was planned to result in an initial set of guidelines that could be evaluated and fine-tuned throughout the project.

The *Ideate* phase was intended to utilize *Crazy-8's* in order to generate sketches for use as a base in the upcoming prototype phase. This method was deemed appropriate since it generates a multitude of ideas in a short period of time, and allows for input from both designers while simultaneously generating more "crazy" and creative ideas. This property might be particularly desirable when ideating in a novel field such as this. Subsequently, depending on the quality of the output of the previous phase, the *Prototype* phase was planned to start off either as a paper prototype or a low-fidelity digital prototype, which was then to be expanded upon. At some point during this phase, the plan was to move to a digital prototype. The digital prototype was to be created in the digital interface design tool Figma [84] due to the authors' prior experience with the tool.

The *Test* phase was the planned final phase of this iteration and was to be conducted through formative evaluation using either a *Heuristic Evaluation* based on Nielsen Norman Group's Ten Heuristics [77] and the guidelines generated from the define phase, or *Usability Testing* depending on the availability of stakeholders and users from Einride. These particular methods were chosen based on the availability of end-users, time constraints, and the ability to capture holistic high-quality insights. These tests were intended to be used to improve both the design itself, as well as to provide an improved iteration of the created guidelines.

Plan for the Second Iteration

The second iteration was planned to only consist of the two or three last phases of Design Thinking. Depending on the results of the tests from the first phase, some ideation was planned to address any potential improvements or issues discovered in the conducted tests. Moreover, this was to be conducted using either another iteration of Crazy-8's or simply ad-hoc ideation depending on the situation. The *Prototype* phase of this iteration was planned to generate a higher fidelity prototype, implemented as a desktop application using a front-end framework such as React [85]. This prototype was also planned to serve as a design artifact, providing a contribution to the design knowledge base in itself. Moreover, this process was planned to follow an agile software development practice such as Scrum but adapted to a team size of two. As such, it was likely that some, but not all agile features and processes would be utilized, such as user stories, tasks, an informal daily stand-up, and the continuous involvement of stakeholders. This iteration was planned to be initiated by more detailed research into the technical tools available or preferred by Einride for both raising understanding of usage as well as asserting viability.

The final *Test* phase was planned to be conducted through *Usability Testing* where GUI usability would be tested and observed to the highest degree possible. The tests would also involve questions to the users about general aspects of their user experience as well as specific questions targeted toward evaluating the GUI with respect to the developed guidelines. The plan was to perform these tests under as realistic circumstances as possible, and as such make use of the real work-stations utilized by Einride, as well as video and data from real trucks. Any findings were furthermore planned to be used to evaluate the viability of the created guidelines and feed back into the creation of a final iteration of both GUI design and guidelines. After this iteration, the focus would shift to finalizing the report and preparing a presentation.

6.1.2 Reflection on the Pre-Study

The pre-study was executed mainly according to the initial plan, consisting of secondary research, mainly literature reviews as well as a study visit to AstaZero. The study visit was luckily conducted during a day when new remote operators were being trained and as such yielded better results than expected. In particular, it yielded insights into how the current systems are used, and under what conditions they are used. This was of high importance as the only other material available was video material of relatively poor quality. Moreover, this allowed the authors to gain insights into the perspective of both experienced and new remote operators.

Mainly using secondary research methods means that most of this phase was conducted without involving the target audience in any major fashion, something which is inherently contradictory in a phase whose main focus is on interacting and empathizing with exactly them. This means that much of the initial design and insights would be based on second or third-hand information, and as such would not incorporate user feedback until later in the process. This disadvantage was however known beforehand, and the method used was knowingly chosen anyhow to be able to focus more on the latter parts of the design process.

6.2 First Iteration

The first iteration consisted of five phases: an initial pre-study which is described in section 6.1, a define, ideation, prototype, and test phase. This resulted in a project plan, a zeroth iteration of guidelines, paper sketches, and a medium-fidelity Figma prototype, which was the subject of two heuristic evaluations and an expert review. The results were then fed back into updating both design and guidelines over the next iteration.

6.2.1 Define

Affinity diagramming was chosen as the method for processing the information generated by the first phase. In order to utilize this method, the information first had to be cleaned and put into a more manageable format, which can be seen in figure 6.2.

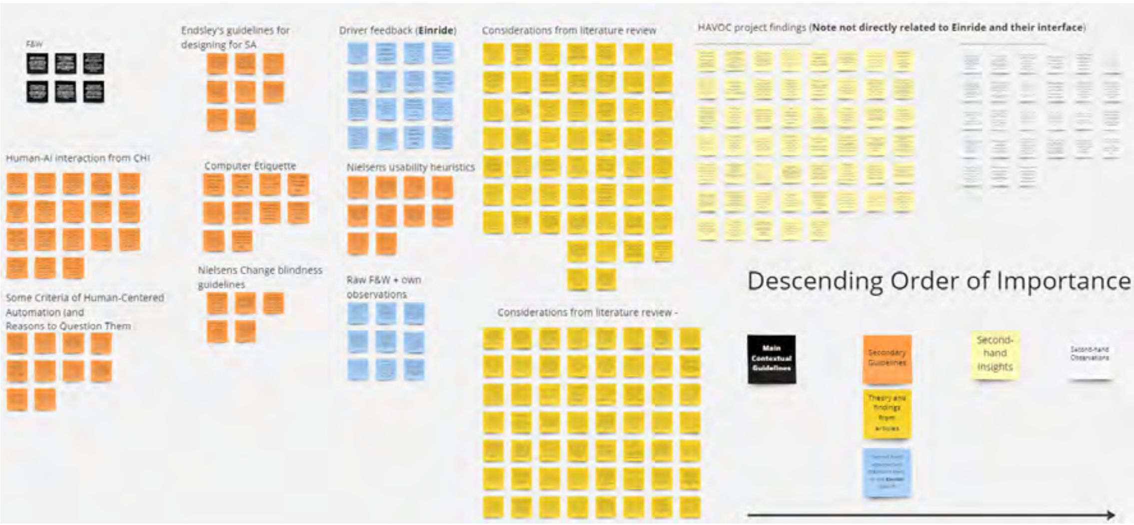


Figure 6.2: Cleaned data which was used for affinity diagramming. Note the color-based coding of importance and source.

The data included design guidelines and theory on Human-AI interaction [86], SA [15], usability [77], Human-Centered Automation [54], mitigating change blindness [87], and Computer Etiquette [56] as well as other considerations from theory such as findings from the HAVOC project [88], system-transparency [15], [54], facts of human perception such as Ware [50], internal company discussion, observations of the currently used GUI and more. These were then put into digital sticky notes and loosely coded based on their perceived importance and source. In general, information was deemed increasingly important based on the proximity of its source to Einride's RO solution. Additionally, information that had been distilled into insights and guidelines was considered more important, as such are generally less prone to sampling bias. Notes were coded as follows, in decreasing order of importance: black signified the guidelines from Flising & Widerstrand and were given special importance; deep orange signified secondary guidelines; orange signified theory from the research literature; blue signified observations of Einride's remote operations system from both the authors and others; secondary insights from the literature were marked with light yellow; deemed least important was secondary observations, marked with white. This coding made it easier to prioritize how strongly to consider a concept which aided the formulation of guidelines.

Once the data had been cleaned, it was clustered into six affinity clusters; **Interruptions**, **Information**, **Feedback**, **Interaction** and **Human-Automation Collaboration** plus an additional general considerations affinity (see figure 6.3). Each major affinity was further clustered into many related sub-affinities. Concepts in each sub-affinity cluster were loosely ordered from top to bottom based on importance.

Figure 6.3: Affinity diagram consisting of six types of insights.

It was clear that solely using affinity diagramming did not directly allow the data to be condensed into an initial set of guidelines, as it did not portray the nonlinear complexities inherent in the problem. The data therefore had to be further processed, to reveal the dynamics of the problem. Subsequently, an ad-hoc system diagram [89] was created, see figure 6.4, that expressed many of these complex relations between insights and concepts. Cause and effect between ideas and concepts were expressed through the use of arrows, which caused groups to dynamically emerge. By looking at the diagram one can get a visual sense of how require information is collected, presented through the UI, as well as perceived and processed in cognitive processes.

One can also see how the user's collaboration with the automated system and the agency afforded to them affects their perception of information. The resulting diagram was used as a visual tool for recalling and understanding what affects the performance of a remote operator.

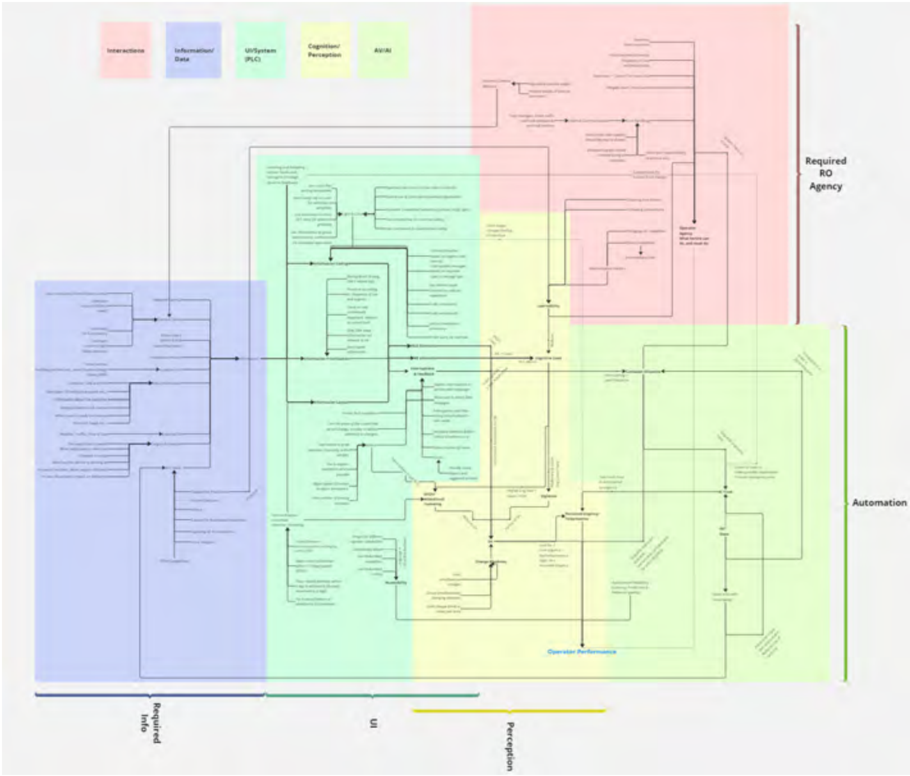


Figure 6.4: A system diagram based on the insights from affinity diagramming, displaying relations between said insights. Color-coded into Required Information, UI, Required Agency, Cognition/Perception, and Automation.

6.2.2 Intermediate Result: Zeroth Iteration of Guidelines

Using the results from the affinity and system diagrams from the define phase, an initial set of guidelines could be produced and are presented here:

G1. *Design for inherent human characteristics by presenting information in a way that produces moderate **Cognitive Load**, appropriate **Vigilance**, and high **Situation Awareness (SA)**.*

   a. *Design to mitigate **Change Blindness** by grouping elements and components while limiting the number of simultaneous changes, as well as making changes visible more than once by for example blinking.*

   b. *Design for an appropriate degree of **Perceived Urgency** and **Telepresence** by managing trust and maintaining high SA, through deliberate visual presentation.*

   c. *Design for high **Learnability** by providing memorization vectors, supportive predictions, and a balance of conventional and new idioms.*

   d. *Design for **Accessibility***

      i. *by using redundant coding and modalities*

      ii. *and adapting to the user context and different cognitive capabilities.*

G2. *Utilize **Prioritization** to filter information and interactions based on the frequency of use and importance to the user's goals.*

   a. *Only present information that is contextually relevant to the current task.*

   b. *Prioritize according to the frequency of use and urgency.*

   c. *Do not repeat information.*

G3. *Design for a **Layout** that battles **Attentional Tunneling** and decreased **Useful Field of View (UFOV).***

   a. *Consider the trade-off between obstructing critical video information and staying within **UFOV***

      i. *Keep critical information within 1-15 deg (typical **UFOV**)*

      ii. *Place related elements within 5 deg.*

      iii. *Avoid attention fixation to GUI elements.*

G4. *Utilize information **Coding** to make frequently used, urgent, and critical information more salient.*

   a. *Color*

      i. *Consider color convention/semantics (cultural, traffic light...).*

      ii. *Use colors that portray seriousness as this affects the **Perceived Urgency** of the driving situation.*

iii. *Use color sparsely and deliberately.*

iv. *Do not rely solely on color for grabbing attention in the periphery.*

v. *Use luminance contrast (3:1 ratio) for attention-grabbing.*

vi. *Use chroma/hue for nominal coding.*

vii. *Use luminance for quantitative coding.*

viii. *Use chroma/hue as the group denominator and luminance for individual separation.*

b. *Code system messages according to response/message type and severity.*

c. *Prefer icons (or glyphs) for an overview of exact values and names.*

d. *Utilize color, motion, orientation, and size to allow for the use of **Preattentive Processing**.*

e. *Use semantic and consistent coding.*

G5. *Prefer **Rich Modeless Feedback**, display potential future consequences of interactions as well as useful interruptions when necessary.*

a. *Utilize motion to grab attention, especially in the periphery, or to express operations.*

i. *Adapt magnitude of motion according to the desired level of annoyance and amount of other movements present.*

b. *Adapt interruptions depending on the context in terms of timing, and by allowing the users to detect false messages.*

c. *Present interruptions using precise, plain language and by indicating the location of cause (physical or otherwise) as well as impact and suggested solution(s).*

d. *Introduce elements before their use in critical situations.*

G6. *Provide operator agency through modes of **interaction** that work towards users' needs and goals.*

a. *Utilize the strengths of automation and human characteristics depending on the task.*

b. *Design for the system to be patient with the user to the greatest extent possible.*

c. *Provide error handling when the automated system fails.*

d. *Accommodate seamless state transition between autonomous and remote operation.*

e. *Support the operator's meta tasks.*

f. *Allow for dynamic camera behavior such as adjustable angles and allowing for contextual details on demand, indicate physical locations of camera views, and keep consistent location.*

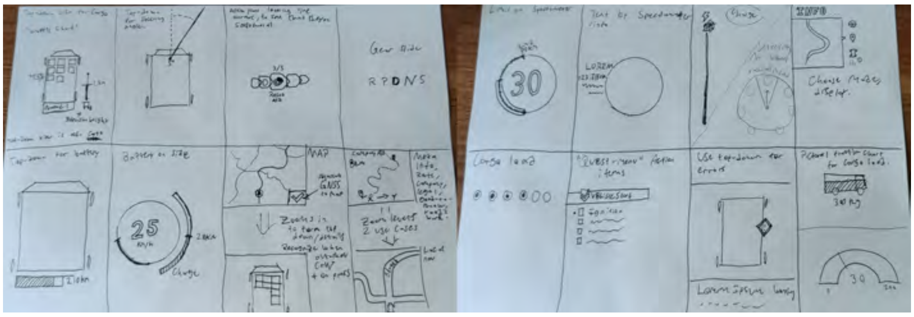
G7. *Design for suitable levels of **trust** in the autonomous system.*

a. *Increase trust through the use of good computer etiquette such as avoiding interruptions, displaying autonomous system state and confidence levels in decisions, allowing for safe exploration, and adapting to the users' goals and strengths.*

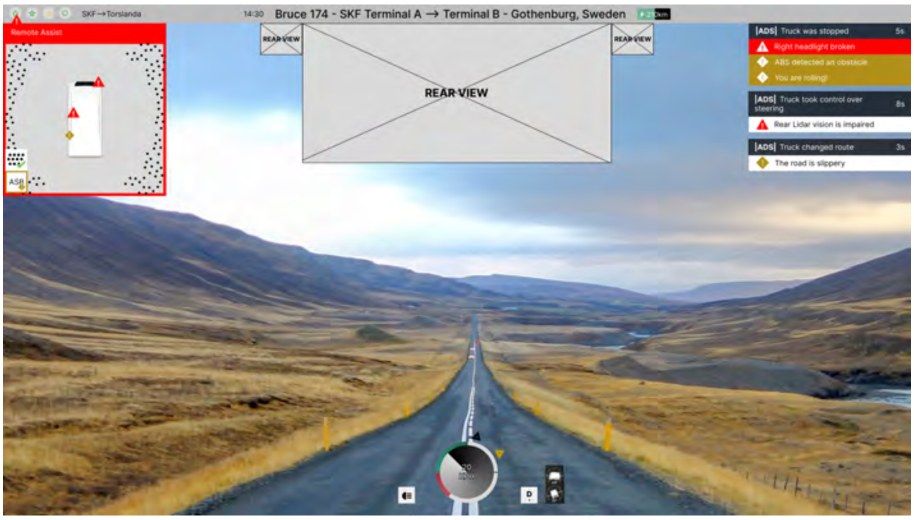
b. *Avoid over-trust by presenting users with the capabilities of the autonomous system and by providing error margins with uncertain information.*

6.2.3 Ideate & Prototype

To quickly generate a number of ideas as well as to utilize the strengths of sketching, Crazy 8's was conducted. The ideation was primarily based on the design produced by Flising & Widerstrand [12], the existing Einride GUI, and newly gathered theoretical knowledge. Furthermore, inspiration was also found in other state-of-the-art RO GUIs and to a smaller extent games and simulators. This method generated a multitude of ideas for different types of components, idioms, and layouts, some of which can be seen in figure 6.5a. These ideas were then brought into Figma to be further processed and ideated upon. This in itself acted as a design process of minor pseudo-iterations, alternating between ideating, prototyping, and in an unofficial capacity evaluation of the results, but with a focus on the exploration of new ideas. Concepts that started taking shape during this phase included: the use of a top-down view for more information, system transparency through an event log, checklists for sequential tasks, and more. As a result, it was also during this phase many of the key concepts of the final GUI were conceived as can be seen in figure 6.5b.



(a) Ideas generated through the use of Crazy-8's. Some of which were later utilized in higher-fidelity prototypes.



(b) A low fidelity GUI that was designed as a part of the ideation phase with some of the key components of the end result already starting to take form.

Figure 6.5: Some initial ideas generated through Crazy 8's, and an early digital prototype.

During this time, one highly important subject of ideation and prototyping was choosing the information layout. The initial video layout was chosen according to figure 6.6, in order to provide continuous vision around the front of the vehicle. This meant that the view was uninterrupted by the placement of side mirrors which in the current GUI were placed closer to the center.

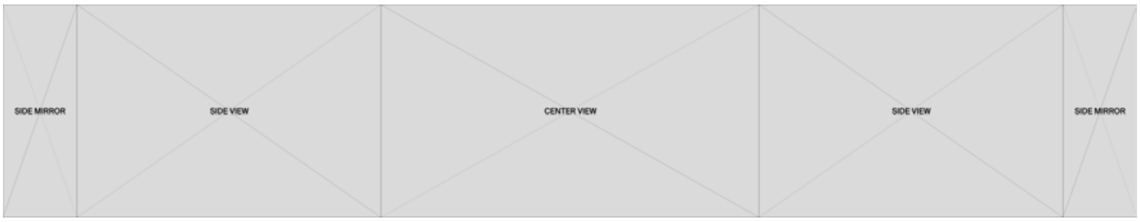


Figure 6.6: Layout and placeholder elements of the three screens. As can be seen, the side screens each consist of video information of the side view mirror, and the side window, with the side mirror being placed on the outer side of each screen.

In order to adhere to **G3**, all GUI elements were placed on the center screen, as can be seen in figure 6.7. Moreover, through an analysis of the video material available, a generalization and assumption were made that different areas of the front view would likely contain important information more or less often. As a result of this assumption, the layout was split into three sections as can be seen in figure 6.7. Of these, the green area was deemed to mostly consist of less important information such as sky or roofs, with a low risk of incoming objects. The red area on the other hand was deemed to be both where the most important information was, consisting of road and traffic, and where the risk of new objects appearing is highest, most likely coming from the sides or horizon. The yellow area is an area that often contains the same type of information as the red, but while being closer positioned to the body of the vehicle has a lower risk of new objects appearing within it. Therefore it was decided that most GUI elements should be placed in the green area if possible, as can also be seen in figure 6.7. Aside from keeping elements from occluding important video information, another motivation was to make contextually relevant information readily available. For this another assumption was made that the driver's eye gaze would generally be fixated on the center of the road. This meant that the final aim was to place elements as close to the center of the road as possible while minimizing visual load and avoiding the occlusion of important video information.

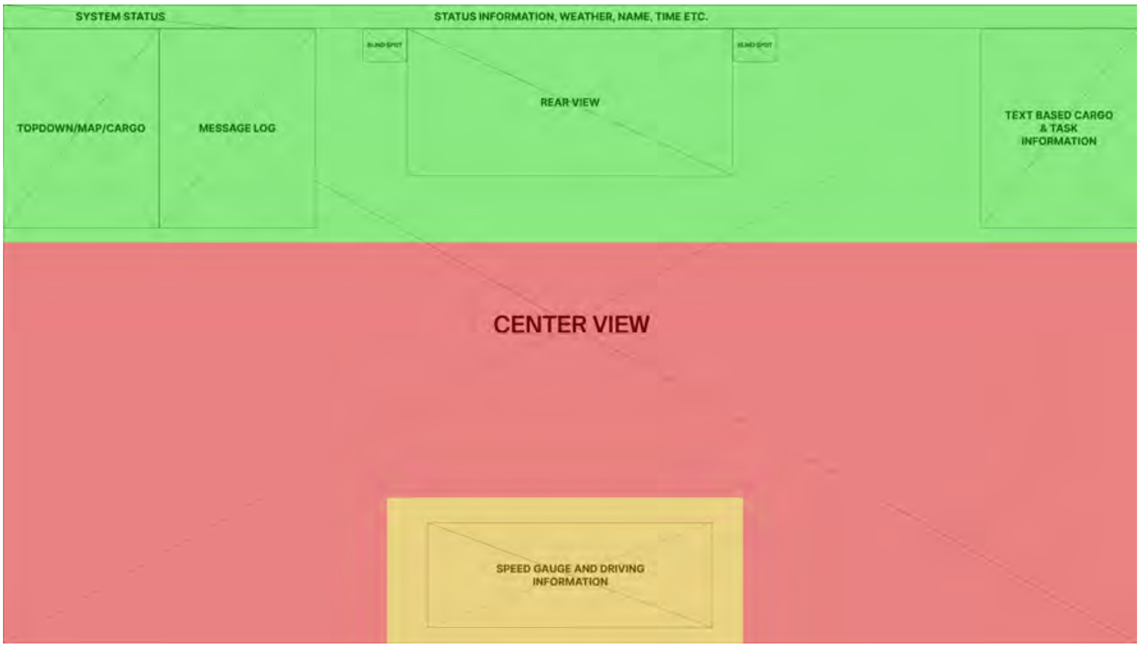


Figure 6.7: Layout split into three sections based on approximated importance of video information. Red: important, yellow: less important, green least important. Placeholder GUI elements can also be seen.

Moving into the prototyping phase, the focus was put on improving the concepts that were generated in the ideation phase. This process also involved pseudo-iterations that continuously increased the level of fidelity, soon resulting in a medium-fidelity prototype containing a number of components that together formed a complete first iteration of the GUI. The prototype contained most of the information that was discovered as being required during the define phase. This consisted partly of much of the information present in the current operational GUI, with the addition of information that according to literature would be useful for the operator. Considering this information and the layout presented previously, the GUI of figure 6.8 was created. In order to minimize the presence of irrelevant information, all components in the GUI were minimized until the user requested them to be maximized. Furthermore, the GUI was created using a dark theme due to the need for reduced use of color since the application was of sovereign posture and focused heavily on video information. Each component in its maximized form is further described below, with their placement being shown in figure 6.7.

Figure 6.8: An example of how the center screen containing all GUI elements could look with all views minimized.

6.2.4 Intermediate Result: The Design

A top bar was added for a few reasons, as can be seen in figure 6.9. Firstly to provide a persistent area of the GUI while not covering any important video. The top bar would therefore provide information that was relevant at all times. In accordance with **G3**, this type of information was split into two sections based on affinity. The left side of the bar consisted of more "technical" information regarding the truck's state such as autonomous state, internet connection, automatic safety brake (ASB), GPS (GNSS), and lidar status. The status information displayed in these components was designed to adhere to **G4**. Consequently, these would change color depending on their status, from green to yellow to red, while also possibly blinking as their status becomes more urgent (as stated in **G5a**). This follows the established color convention of traffic lights and is used throughout the design. In addition to color, symbols are used to double code status information and indicate that more information might be found in the event log through cross-view-brushing. Although these symbols are small, they are highly salient due to the strong luminance contrast between them and the background. The rest of the top bar displayed information such as time, vehicle name, weather, and battery status. Secondly, the information contained in the bar could be particularly important in a future scenario where an operator is expected to switch between vehicles and contexts more often and can therefore be used to quickly gain SA. Thirdly, the bar is a common idiom from more traditional sovereign applications, and as such provides familiarity and some level of seriousness that is commonly present in applications with such a bar.

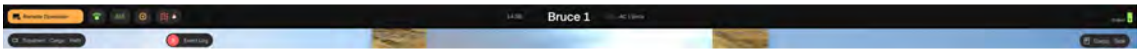


Figure 6.9: The top bar, with the cluster on the left side displaying the technical state of the vehicle, and other relevant data spread throughout the bar.

A group of video displays showing the blind spots on both sides of the vehicle, as well as a rear-view were also attached to the top bar, as can be seen in 6.10. This cluster was intended to act as a metaphor for the similarly positioned rear-view mirror of a regular car, familiar to users with a regular driver's license. Following this metaphor also put this cluster in the top center, adhering to the layout partitioning presented earlier. As a result, this cluster aimed to provide familiarity and thus an immediately understandable idiom for looking behind the vehicle.

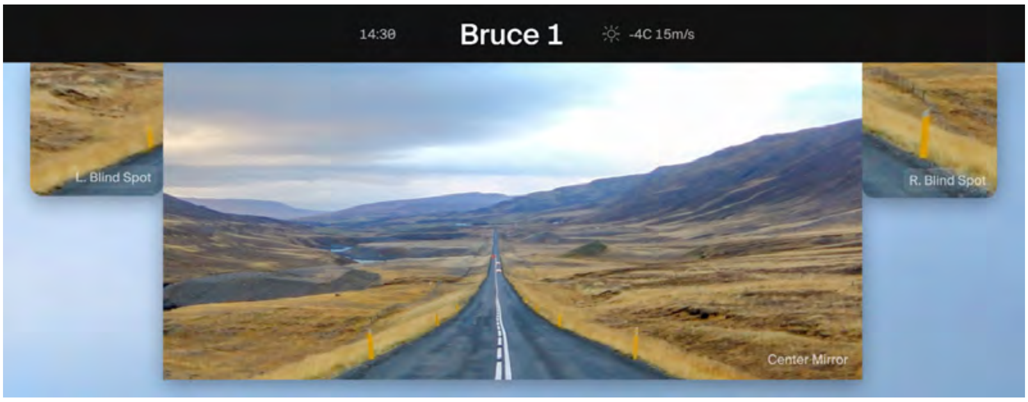


Figure 6.10: A block containing three camera views, the blind spots on each side, and a rear-view camera. Their placement bears resemblance to rear-view mirrors in regular cars.

Three other types of components were also attached to the top bar. These were the *Overview*, *Event Log*, and *Details* components. The three components follow the same conventions as each other where they can all be expanded or minimized at will. Additionally, which components are expanded at any one time can be managed according to the user's needs by pressing their respective text buttons. In the overview and details components, information is presented in separate views. It was therefore assumed that the information residing within the same component had a low risk of being requested at the same time. Moreover, the information in the overview and details components were intended to complement each other and were intended as a way for the user to gain the level of understanding currently needed for adequate SA.

The overview component was designed as a visual-based component intended to provide the user with an overview of the situation, see figure 6.11. Firstly, this contained a top-down camera view, with a few optional overlays displaying lidar data (white dots), the truck's alignment with its intended path (green), the intended path (blue), and the physical location of errors/warnings on the truck. These overlays were intended to be togglable at will, such that any combination of them could be used at any given moment. Furthermore, through the display of many types of top-down information in one location the user's search is facilitated. The combination of lidar and camera data also serves as information redundancy should one or the other fail to deliver data. Secondly, the overview component also contained a camera view of the truck's cargo hold. Finally, the component also contained a map view showing the truck's progress along the path toward its current goal in addition to potential events along the way.

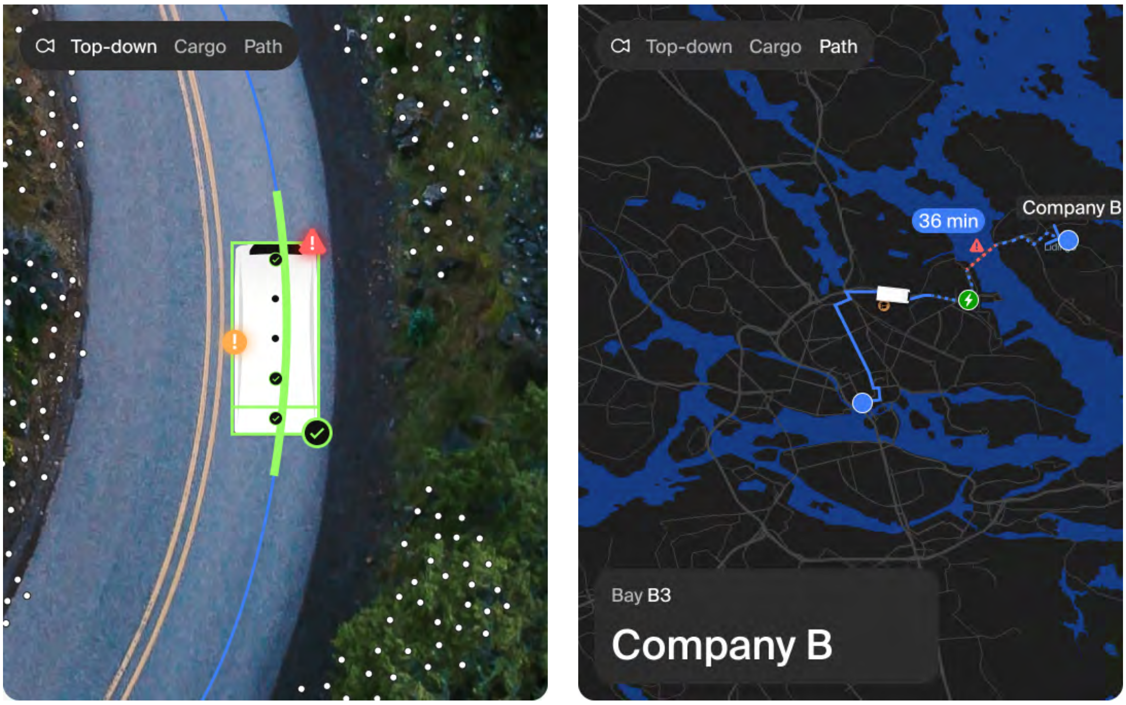


Figure 6.11: The overview component consisting of the top-down, cargo, and path views. Shown here are; the top-down and path views. The cargo view is not shown due to it only consisting of a camera feed.

The details component aimed to increase the operator's SA through detailed information about the current situation and the future (see figure 6.12). Specifically, this includes details about the type of cargo that is currently loaded, as well as the vehicle's current task and progress. An example of how the details and overview components complement each other can be found in the task view which was designed to describe meta-tasks and specific requirements for completing certain tasks, while the path view was intended to give a quick overview of the truck's location and where it is headed geographically. Through their similar visual structure, the operator should be able to associate the chronological task information with the geographical path information. Specifically in this case, the association is portrayed through the similarity of the blue lines, and how they transition from dashed to filled depending on the progress of a task, in both the top-down and path views.

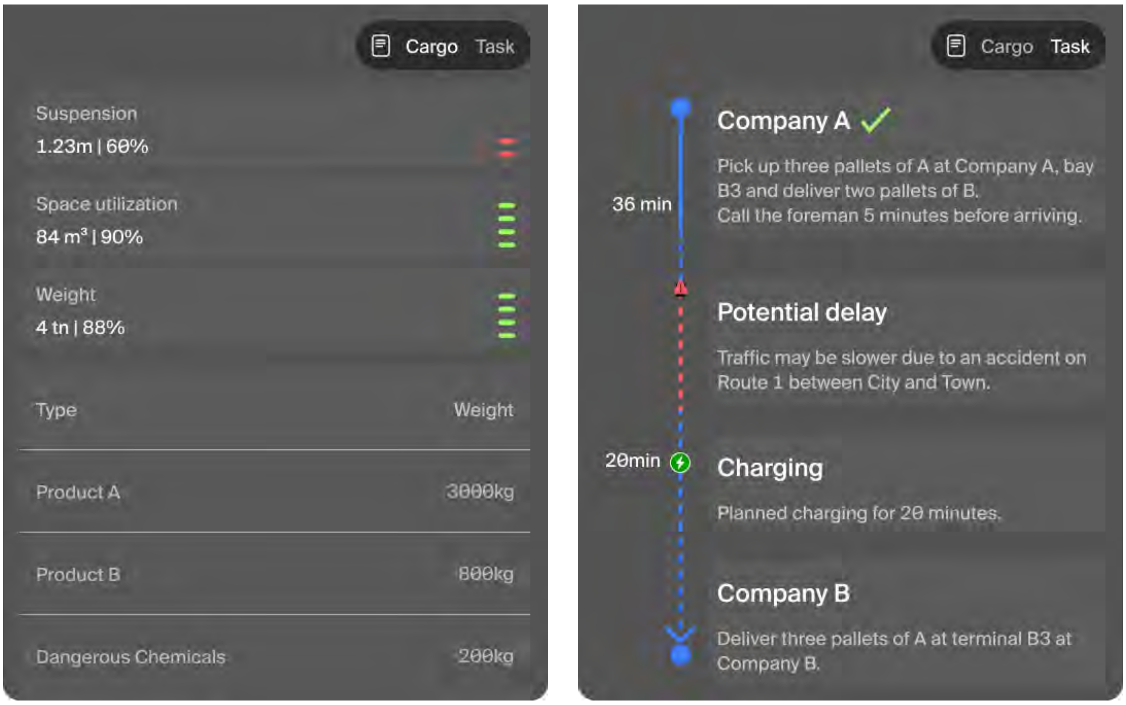


Figure 6.12: The details component displaying detailed text-based descriptions of the current cargo and task.

The event log contains a chronologically ordered list of any previous events to provide the operator with detailed SA and system transparency. Events include system status and behavior, detection of obstacles, malfunctioning subsystems, as well as external events such as extraordinarily slippery roads. Events were categorized into one of four types; behavior and general messages from the system, positive messages such as completion of tasks, warnings, and errors, all of which are exemplified in figure 6.13. These were color-coded depending on their severity, according to the traffic light convention previously mentioned. In its minimized form, the event log only displays recent messages for a short time and also displays how many unhandled events there are through a number indicator. Messages appearing in the event log, when closed, would grab attention to a moderate degree due to their sudden appearance. In its maximized form it is possible to scroll through the entire log, and as such provides some transparency into the behavior of the autonomous system. It could also be used to both track the progress of the truck's task and act as a way for the operator to understand how they ended up in their current situation. This would be particularly useful in a 1:N operation situation where operators would need to diagnose new situations frequently.

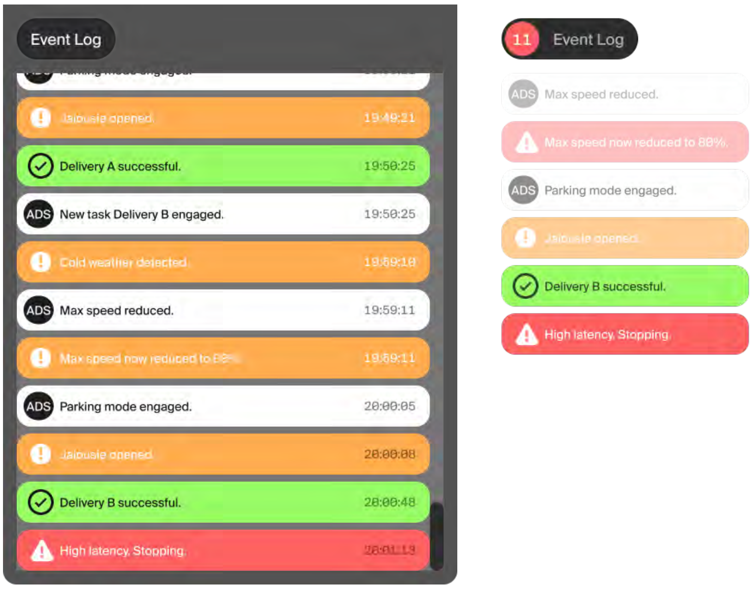


Figure 6.13: The event log in opened and closed versions.

The dashboard component was placed in the yellow area of the layout, (see figure 6.14), which kept it near the operator's eye gaze while occluding as little important information as possible. It was also placed in this area as a metaphor for traditional dashboards. This component further adhered to **G4** by containing only the most important information directly related to actual driving such as current speed, throttle, brake, gear, bearing, and legal speed limit. The component is modular, allowing for displaying a curated selection of information through the addition or removal of minor status blocks. Colors were kept as monochrome as possible to reduce cognitive load and retain the semantics of the system status colors. However, red and green for brake and throttle respectively were used due to their strong semantic connection to stop and go.

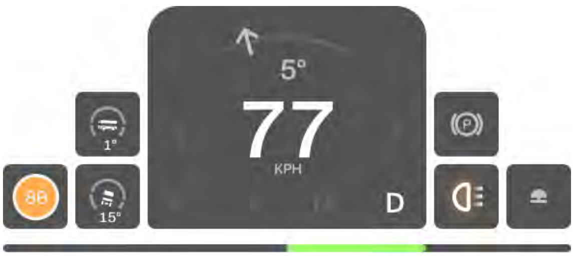


Figure 6.14: The dashboard component displaying relevant driving information.

6.2.5 Test

The Figma prototype was tested using two methods. First, two heuristic evaluations were conducted by the authors using Nielsen Normans's Ten Heuristics [77] and the zeroth iteration of guidelines, see section 6.2.2. This was followed by an expert review in the form of a presentation for some of the remote operators, engineers, and designers of the RO software which was initially not planned. After the presentation, participants had the opportunity to provide both feedback on the existing GUI as well as provide ideas for how to further develop or re-design the GUI.

Due to time restraints, the heuristic evaluations were conducted by only two people, the authors. These were done individually and independently of each other to utilize the two available perspectives, in an attempt to gain more and better insights. The individual insights gained were then combined and discussed. Moreover, the evaluations were conducted by considering the GUI through the lenses of first Nielsen Norman's Ten Heuristics and later, the initial iteration of guidelines presented in section 6.2.2. This was done while keeping both the GUI as a whole as well as individual components in mind. Furthermore, the authors tried to do this from the perspective of the intended target audience with a certain level of understanding of the system, as well as at the very least having experience from driving a car if not a truck.

6.2.6 Intermediate Result: Heuristic Evaluation

Two heuristic evaluations were conducted yielding results that served to provide insights into the design of both GUI and guidelines. As such, the results of these evaluations are presented below.

Nielsen's Usability Heuristics

Despite some guidelines not being entirely applicable, the evaluation using Nielsen's usability heuristics uncovered some useful insights and potential changes. For instance, it was identified that placing the autonomous state closer to the driving dashboard makes it easier to know whether driving and interactions from the operator are possible/required or not. Furthermore, since most interactions with the system were actually changing the physical state of the truck, it was considered difficult to provide undo/redo actions for more safe exploration. However, it was speculated that it could be possible to allow for an interaction that instantly minimizes all views and returns the GUI to its default state or possibly even hides the GUI as a whole, serving as a form of escape hatch.

Another aspect discovered during the evaluation was the potential introduction of dynamic/automatic behavior of the UI elements, such as opening the top-down view if an error occurs physically on the truck. Moreover, there were some considerations about some of the more abstract concepts regarding the truck such as the lidar. One proposed solution at this stage was, to use some kind of text or tooltip overlays to further describe abstract concepts, in addition to their symbols. However, any training performed by new operators could cover such concepts and might thus not be needed. Outside of these proposed changes, the interface was deemed to follow the usability heuristics quite well.

Zeroth Iteration of Guidelines

Using the initial version of the guidelines presented in section 6.2.2 to conduct a heuristic evaluation yielded similar results to using Nielsen's heuristics. However, some new insights, more specific to the domain of RO were also gained. For example, the dashboard was identified as a potential target for attentional tunneling. It was argued that should the dashboard become the focus of attentional tunneling, the user's gaze would still be nearby the most important video information, and was therefore considered acceptable. Additionally, in this iteration of the design the interface provided meta-task information to some degree. However, it was proposed that it might be interesting to explore if a separate screen would better portray such information. Specifically considering future scenarios where the operator might switch between trucks more often, and therefore more frequently be required to quickly gain a high level of SA. This was however deemed to be outside the scope of this thesis. Furthermore, the GUI was found to partly display potential consequences of actions by providing the bearing/heading of the truck, thus allowing the user to infer the consequences of their actions. Moreover, the intended path displayed in the top-down view was also found to serve a similar purpose.

In terms of accessibility, color blindness was somewhat managed through redundant coding, and dyslexia through a focus on icons and symbols. Furthermore, handling trust in the automated system was deemed a possible area of improvement by exploring how to provide more system transparency. One proposed way to do this was by presenting confidence levels in any actions taken by the autonomous system. This would, however, require access to details from the implementation of the autonomous system, which was not possible.

Regarding interaction, dynamic camera behavior was considered to be an area that could be further explored in terms of size or visibility, both through contextually changing the sizes of camera views, as well as allowing the user to change their size and positioning in the GUI. Camera movement, although likely a wanted feature, was not planned to be explored due to technical limitations. Furthermore, no direct memorization vectors were identified, it was however proposed that this could be introduced through tooltips and further explored later in the design process. Redundant modalities were deemed difficult to incorporate due to the limitations set by the physical modes of interaction and no major focus had therefore been put into exploring this. Moreover, despite displaying errors, the GUI did not offer ways for the operator to solve all types of errors. This was in part due to the vehicle being remotely operated and things such as fixing a flat tire might not be possible from a distance. Solving this type of situation could however be done by providing direct access to external partners or people on-site through the GUI.

Motion was planned to be used to grab attention in the form of blinking although the specifics of interruption and attention-grabbing needed to be explored through a dynamic prototype. This also included the timing, frequency, and quantity of signals in order to grab attention at the right moments while staying patient with the user. This was deemed especially important as overly stressing the user even after they have perceived the full situation and are in the process of problem-solving would likely be detrimental to their performance. Wiggling or other forms of movement were therefore planned to be explored for less urgent signals. Furthermore, critical components were designed to signal their existence before signaling an error or warning. On this topic, it was identified that there was a trade-off between contributing to cognitive load and aiding system clarity. Which potential error or warning state to include as a permanent UI element should therefore be prioritized based on their frequency and potential severity. The impact of errors was described in the event log as messages describing the error as well as indicated through the other messages from the autonomous driving system. The cause of errors was indicated using symbols across the interface with matching messages in the event log. How well the user can match an error symbol to its corresponding message was deemed to require further exploration. Only presenting the messages in place was considered to remove any benefits that come from displaying them in chronological order and it might therefore be beneficial to have this information in multiple places.

6.2.7 Intermediate Result: First Iteration of Guidelines

All forms of testing done during this iteration provided insights both regarding the GUI design as well as the initial iteration of guidelines and how they could be improved. As such, some changes were made to this iteration of guidelines. Guidelines that are not mentioned here were kept as they were, and only a few guidelines with greater changes are covered below in their entirety. Changes in wording and sentence structures were also made for clarity.

In **G2**, "Do not repeat information was removed", as it was realized that some repetition of information can be needed, and useful. Specifically in terms of cross-view brushing, and allowing for information to be displayed in varied detail depending on the use case. It was realized that **G4** was too detailed, and was therefore shortened for brevity, instead referring to already established guidelines. **G7** was mainly changed in terms of wording to make it more understandable. These updated guidelines were thus changed to the following:

G2. *Utilize **Prioritization** to filter information and interactions based on the frequency of use and importance to the user's goals.*

   a. *Only present information that is contextually relevant to the current task.*

   b. *Prioritize according to the frequency of use and urgency.*

G4. *Utilize information **Coding** to make frequently used, urgent, and critical information more salient.*

   a. *Utilize color, motion, orientation, and size to allow for the use of **Preattentive Processing**, for more information see Ware guidelines 5.2,5.4,5.5 [50].*

   b. *Prefer icons (or glyphs) for an overview of exact values and names.*

   c. *Use semantic and consistent coding.*

   d. *Color*

      i. *Consider color convention/semantics such as cultural conventions or the traffic light metaphor.*

      ii. *Use colors that portray seriousness as this affects the **Perceived Urgency** of the driving situation.*

      iii. *Use color sparsely and deliberately.*

      iv. *Use the different aspects of color theory such as chroma, hue, and luminance, and different feature channels for coding, see Ware guidelines such as 3.4 and 4.6 [50].*

      v. *Do not rely solely on color to grab attention in the periphery.*

   e. *Code system messages according to response/message type and severity.*

G7. *Design for suitable levels of* ***trust*** *in the automated system to achieve appropriate vigilance for each situation.*

   a. *Increase trust through the use of good computer etiquette.*

      i. *by avoiding unnecessary interruptions.*

      ii. *by allowing for safe exploration.*

      iii. *adapting to the users' goals and strengths.*

      iv. *by being patient.*

   b. *Avoid over-trust by designing an automated system that is transparent.*

      i. *that explains what decisions have been made and why.*

      ii. *by presenting users with the capabilities of the autonomous system.*

      iii. *regarding its level of confidence in decisions made.*

6.2.8 Intermediate Result: Expert Review

The expert review yielded a lot of positive feedback where the audience in general expressed that the proposed GUI was a clear improvement to the currently used GUI. However, some notable things that were brought up include the placement and size of the rear-view cameras, as well as some minor concerns about clustering most of the GUI elements in the top part of the screen. The main issue brought up was that while the top area of the video information might not be critical under most conditions such as driving on flat ground, it might contain more important information under other circumstances, such as when moving uphill, downhill, inside, or in construction areas where objects mighty move above the truck. It was however noted that this could be countered through dynamic behavior, or through customizability. Moreover, it was mentioned that center rear-view cameras are seldom present in trucks, and people with experience driving traditional trucks mentioned that they believe it to be redundant at least until they are moving in reverse. They mentioned that they instead tend to focus on the side mirror views.

6.2.9 Reflection on the First Iteration

This iteration was conducted according to the initial plan with some minor adjustments. For the define phase, affinity diagramming was conducted as planned but had to be accompanied by an ad-hoc system diagram to provide enough clarity to the problem to yield a zeroth iteration of guidelines. This was done since, as mentioned in section 6.2.1, an affinity diagram allows for stating the existence of concepts and in only a limited capacity their relation. Affinity diagrams do not make apparent more intricate and complex relations between the concepts. For example, one concept might relate to another for one reason, and to a concept placed in an entirely different cluster for some other reason. As such there was a need to further process the information yielded by the affinity diagram to allow for complex nonlinear relations between concepts to surface.

One of the major drawbacks of the combination of methods used in the empathize and define phases is the fact that they once again process secondhand information, something that increases the distance between the insights and the information source even further. Similar to the "telephone game", one could argue that such a process risks losing important information due to the down-sampling of insights with each step separating the researcher and the first-hand study. Thus the process would clearly be improved by validating insights with more nascent first-hand data, although this was partly done through the field trip.

Using Crazy-8's as a method for ideation turned out well, as it allowed for a natural transition from ideation to prototyping, and many of the ideas generated were further developed throughout the iteration. Moreover, moving to a digital prototype relatively quickly allowed for its use in an expert review, something that was deeply needed due to the previously very limited involvement of users or experts in the field. Moreover, two heuristic evaluations were conducted, something which was more than planned initially. While no major conclusions could be drawn from any of these, they worked as a form of validation, ensuring that the design and process were moving in the right direction. Furthermore, evaluating the GUI using two different heuristics/guidelines proved fruitful. By initiating the evaluation with Nielsen's heuristics, these could be seen as a form of benchmark as they allowed for some insight into how much of the issues found in the second heuristic evaluation was due to the GUI being poorly designed and how much was due to ill-formed guidelines. This evaluation, therefore, served as an evaluation of both GUI and guidelines. The guidelines were updated after this, something that lead to some overlap between guidelines. In particular, **G7** overlaps with **G5** and **G6**. This was however regarded as acceptable due to the complex relations underlying the topic. The hope was however to continue to iterate and minimize repetition further.

6.3 Second Iteration

The second and final iteration of this thesis was planned to only consist of an ideate, prototype, and test phase. These phases continued the process based on the results yielded from the first iteration. This first resulted in an updated design which was used to conduct usability tests, which in turn yielded some final changes to the design and a final iteration of guidelines.

6.3.1 Ideate & Prototype

During this prototyping phase, the implementation of the design as a software prototype consisted of many pseudo-iterations containing unofficial ideation, prototyping, and testing phases, similar to the first. However, before prototyping was initiated, a final discussion was held with Einride regarding possible choices of technology. The goal was to find a suitable technology existing within their ecosystem to make the development process as smooth as possible. The chosen technology stack consisted of TypeScript [90], React [85], and Electron [91] which allowed for the building of a cross-platform desktop app while being able to use existing Einride React components where possible.

In order to develop the prototype, access to data logs and their corresponding video material was required. Thus a recording session with one of the remote operators was conducted where they were asked to perform three different scenarios that could then also be used for testing purposes later on. These consisted of recording the required startup process of the vehicle, reversing into a loading bay and opening the jalousie of the cargo space, and some general “city driving” passing a crossroads as well as going through a roundabout. These were chosen as they allowed for covering a large part of the different possible states of the vehicle, while also displaying different types of driving and scenarios that would be frequent in a real scenario. Moreover, reversing into a loading dock was considered especially challenging, and therefore chosen as one of the scenarios.

Once prototyping and ideating of the software prototype were started, different types of dynamic behavior were explored. For instance, different types of pulsation and movement in combination with color were explored to grab attention. Additionally, expanding the center rear and blind spot camera views while also opening the top-down display when changing gear to reverse. Some general changes that were made during this ideation and prototype phase include reduced use of colors, making elements more transparent and generally decreasing the visual load, adding drop-shadow to elements, and making the background of most components blurred for increased salience against the background. No major changes regarding the positioning of components were made in this iteration. However, most components were re-designed on a more detailed level, and these changes are as such presented below.

6.3.2 Intermediate Result: The Design

Starting with the dashboard, a set of temporary status symbols was put at the top to indicate statuses such as the activation of the parking brake, emergency brake, horn, hazard lights, and more (see figure 6.15). These indicators were only shown when something was out of the ordinary and therefore affects the behavior or performance of the vehicle in some capacity. Further changes include the redesign of the gear indicator to save vertical space as well as a new banner that provides the operator with an overview of the system status. This banner was added so that once something unexpected would happen, the driver would have easy access to information on the current behavior of the vehicle, to reduce the need to dart their gaze between the upper left corner and the dashboard to gain quick SA.

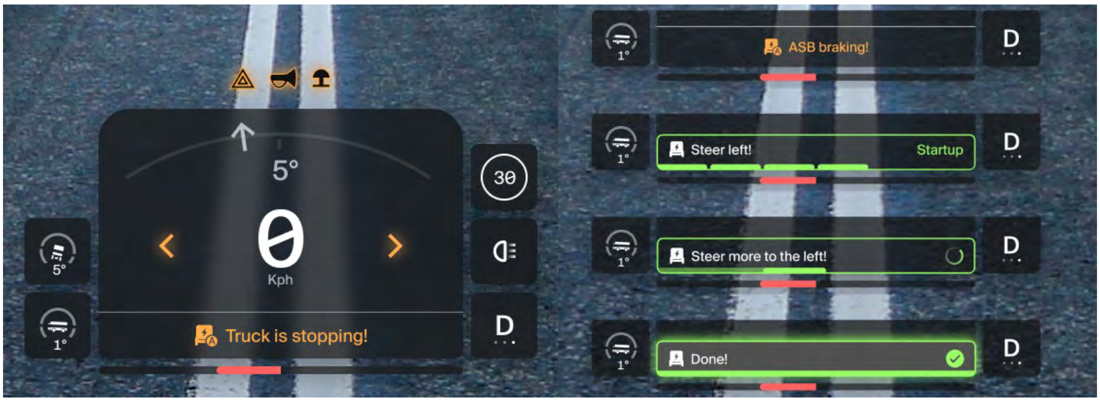


Figure 6.15: The dashboard used during usability testing showing temporary status indicators, and the dashboard status banner asking the operator to perform certain tasks.

This banner served multiple purposes, where it provided a more expansive explanation of the autonomous state while also providing more details regarding what actions were currently being performed, such as braking due to detecting an object. This was also used to provide the user with hints on how to solve certain situations and to ask the user to perform certain actions. One example of this is where this is utilized to perform a safety checklist, or startup sequence for the truck. In figure 6.15, the banner is shown in multiple states.

A contracted, smaller version of the dashboard, as seen in figure 6.16, was designed as a potential alternative for the operator. This version reduced some of the visual flare and removed some information that could potentially be considered secondary, such as pitch, roll, speed limit, and headlight status. Furthermore, as the vehicle would potentially lose capability during operation, ways to visualize such degradation were also explored, see figure 6.16. In general, degradation was visualized in proximity to the value it concerned.

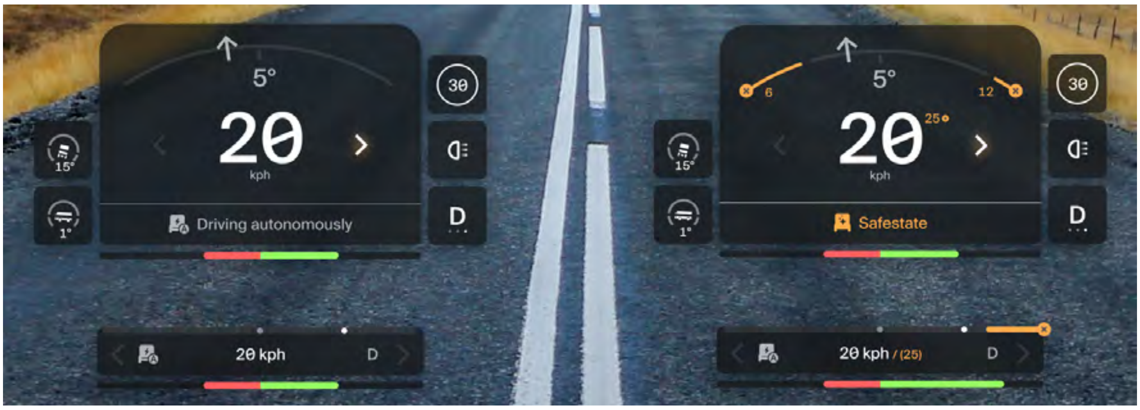


Figure 6.16: To the left the dashboard is shown in its maximized and minimized state. To the right, the dashboard in its maximized and minimized state shows degradation in terms of steering angle and speed.

Some changes were also made to the top-left section of the GUI as seen in figure 6.17. For instance, the lidar overlay for the top-down view was changed from displaying lidar point clouds where they were detected and now instead showing any detected objects within a certain distance of the vehicle while also displaying distance measures to aid the operator. Additionally, the closed version of the event log was redesigned to minimize visual load through reduced use of color and decreased opacity of event log messages. Event log status was instead expressed using a status symbol displaying the most urgent event type that had not been handled.

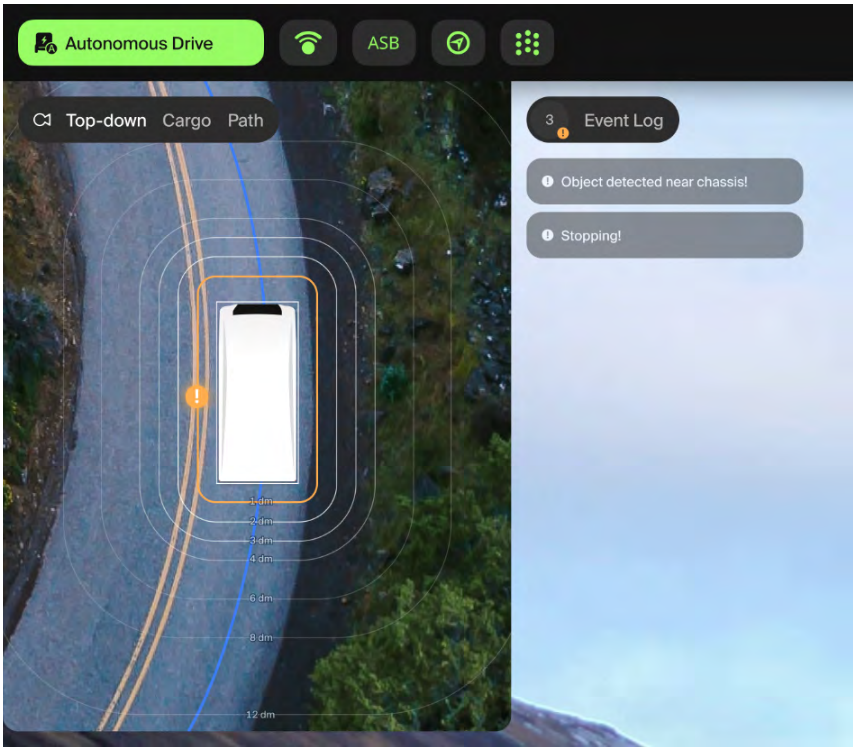


Figure 6.17: Top left area of the GUI showing the top-down view with lidar path overlay and the minimized event log.

Some changes were also made to the open version of the event log, where color usage was reduced as seen in figure 6.18. This had the added effect of increasing the clarity of cross-view-brushing that made it easier to perceive how a status symbol in for instance the top-down view corresponded to a specific event-log message. Furthermore, logs that had already been handled were decreased in opacity and an indicator for new events was added. Another change also seen in figure 6.18 is the redesign of the error state of status bars in the top bar to reduce color use.

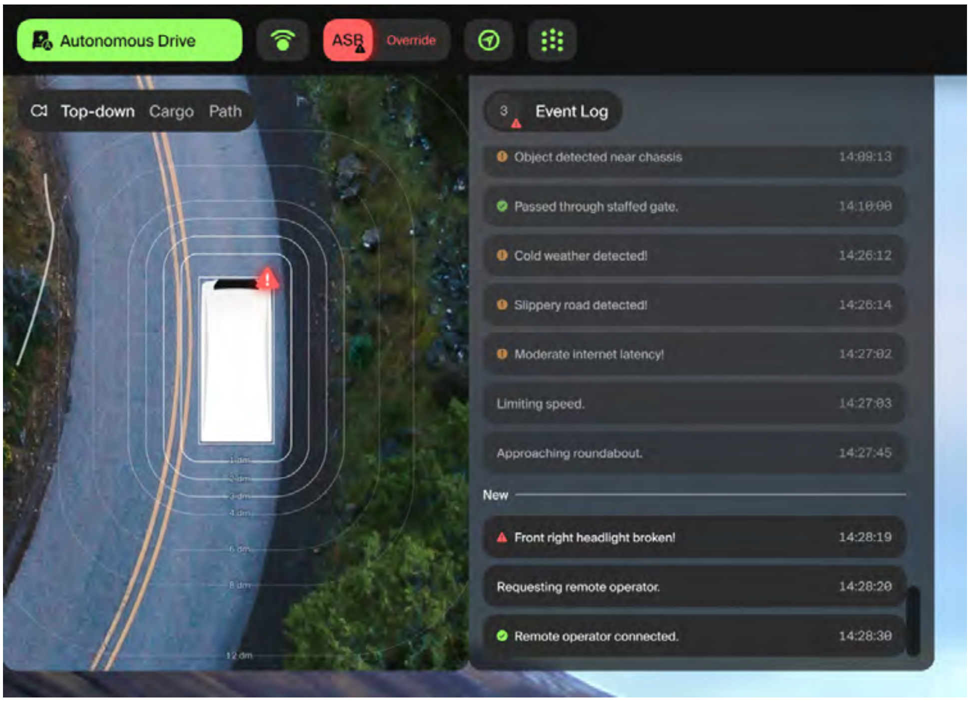


Figure 6.18: Top left area of the GUI showing errors and an open event log. Changes were in terms of displaying error status, and the salience of new versus old logs.

The biggest change to the top bar was in terms of the group of video views attached to it. In this version, only the rear-view camera view was persistent and was made smaller than in the previous iteration, as can be seen in figure 6.19. Once the vehicle was put into reverse, it was expanded and the blind spot cameras were also made visible, as can also be seen in figure 6.19. This was specifically added as a response to the feedback from the first iteration.

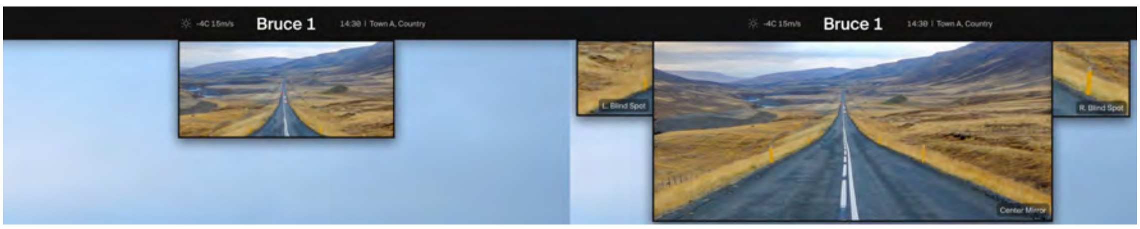


Figure 6.19: The center area of the top bar and the elements attached to it as used for usability testing, to the left in its minimized form and to the right in its maximized form.

Some minor changes were also made to the details component in terms of layout and visual fidelity as seen in figure 6.20. Furthermore, both views of the component received the addition of a title and subtitle hosting relevant information. Also, task cards were complemented with information on the predicted time or step of a task. Subtasks were now hidden by default and could be shown by expanding the card and the progress of sub-tasks was made clear through checkmarks.

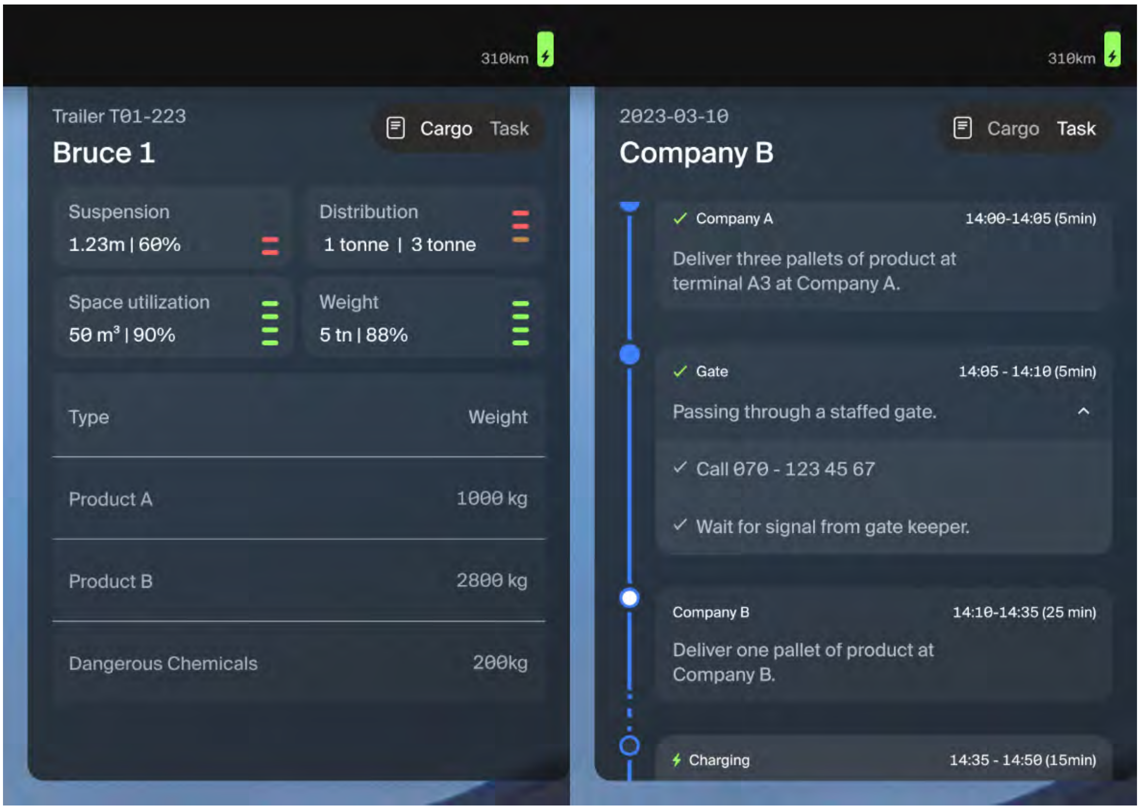


Figure 6.20: The design of the details view used during usability testing. Changes from the previous iteration include an increase in visual fidelity, more information, and interactivity of task cards.

6.3.3 Test

The second iteration of the project was concluded with usability testing of the software prototype. The plan was to conduct this round of testing in an environment as close to real as possible. This was however not possible due to unfortunate logistical circumstances which meant that the tests were conducted using the setup seen in figure 6.21, instead of something similar to what can be seen in figure 2.2b. Also, as previously mentioned the prototype used for testing was not in control of a physical truck. Moreover, the only device required for the testing was a mouse, as all GUI interactions were designed to be conducted using a pointing device for ease of testing.

Figure 6.21: The station used for usability testing.

Testing was conducted with nine participants who were selected for their range of experience with technology and RO and included engineers and professional remote operators at Einride. Participants were sent a form beforehand that both described the test and asked some questions about their background. This form also contained information describing how any information gathered during the usability testing would be handled, that the intention of the test was to test the usability of the design, not their performance, and informing them that their contribution would be anonymous. Moreover, only people who actively wanted to participate were asked to participate, and it was made clear that they were not required to participate.

All participants were found to have a driver's license where around half had at least a 'C' license for heavy trucks, around half had also worked professionally with driving and a majority worked with the current RO interface at least a couple of times per week. At the testing site, participants were briefed further about the manner of the test, asked to think out loud, as well as to imagine themselves driving the vehicle while being encouraged to interact with the GUI as much as they liked during the test. The participants were then walked through three scenarios, and before starting the playback of each scenario participants were encouraged to familiarize themselves with the GUI. At the end of the first scenario participants were specifically tasked with finding the task details view in an attempt to teach them its use, as in the second scenario, participants were tasked with finding out on their own what driving task was to be performed before starting playback. The final scenario was slightly longer and less intense, allowing the participants to fully immerse themselves, testing out different configurations of the GUI while exploring its functionality and features. After the final scenario had been completed, the participants were asked to fill in another questionnaire which can be found in appendix A. The results from this were then used as a basis for a short unstructured interview in order to gain further insights into their experience of the design.

6.3.4 Intermediate Result: Usability Testing

Feedback uncovered during the usability testing was mostly positive. Generally speaking, color seems to have been successfully applied, and some participants expressed how they felt primed to look for red error-state signals. It was also found that participants appreciated the dynamic resizing of the rear-view mirror and the visibility-toggling of the top-down component when the truck was put into reverse.

Some of the more popular GUI elements were the event log and task view and participants expressed their appreciation of these components as they allowed the users to better understand what was happening and why, proving their ability to contribute to SA. Additionally, the participants were observed using the task view as intended to increase their understanding of the tasks to be conducted in the upcoming scenario, often even before being prompted to do so. Additionally, the top-down view was greatly appreciated, in particular, its ability to portray the physical location of issues, driving path alignment, and distances to objects. Distance measures were furthermore expressed as potentially useful in more than just the top-down view.

Despite the generally positive feedback, some issues of varying severity were raised during testing. For instance, participants had issues with understanding certain symbols, such as the lidar symbol, which supported the evidence from the heuristic evaluation in section 6.2.6 that tooltips or more information might be needed. Related to this, was that the design used for displaying the technical information regarding autonomous state, internet connection, GNSS, and lidar was slightly confusing for many of the participants. They expressed that it felt as if the area lacked information, and that they expected the status indicators to be interactive. This was either due to their positioning and the resemblance to more traditional sovereign applications with file menus, or simply because they expected there to be more details available. Furthermore, they expressed a belief that the ASB-override indicator was a toggle button due to its appearance.

The more experienced drivers also expressed that when reversing, they tend to heavily rely on the use the side view mirrors. Therefore, a want was expressed for having them closer to the center screen to avoid having to continuously turn their heads between the outer edges of the side screens. It was also mentioned that the relevancy of blind spot vision is independent of which direction the vehicle is moving.

The design of the throttle/brake bars was initially confusing for many but proved to be effective once they had learned their meaning and as such proved to be an effective idiom. The red color used in the bar however confused some as it semantically clashed with the error status it symbolized throughout the rest of the GUI. Additional points include that it was sometimes difficult to gauge the relevancy of an event log due to the displayed time being absolute. Some also expressed a want for more interaction with the event log in the form of handling individual logs or writing their own logs. Participants also specifically asked for the ability to customize the GUI, for example by choosing how to display particular elements such as the throttle/brake, and that elements non-critical to safety should in general be customizable. Other minor issues concerned: the sizing of elements, specific coding of certain signals, and difficulty learning new symbols.

Due to time constraints, only some of the more approachable issues were implemented as potential solutions. These were making the blind spot cameras always visible; rearranging the general layout to minimize the distance between side view mirrors; expandable internet, ASB, GNSS, and lidar elements, which were also redesigned; relative time stamps for event logs messages as well as the removal of the legal speed limit block from the dashboard. Some additional general changes were also made in regards to the sizing of certain elements such as the dashboard and rear-view, which were made larger. All changes can be seen in the final results in section 7.1.

7

Results

As the process followed RtD, it is subject to the four lenses of evaluation described by Zimmerman *et al.*[58]. In short, these require the research to have a clearly documented process, a motivation for the selected methodology, to display a level of invention in regards to its contribution, a motivation of why the proposed solution brings about the wanted change, and for the result to be extensible. Applying these lenses to the research presented in this thesis it can be concluded that said requirements have been fulfilled. Firstly, the thesis provides a comprehensive description of the conducted process, where all considered methods are listed and selected methods are motivated. Moreover, an extensive literature study was conducted, the results of which can be seen in chapter 3. Furthermore, the produced changes are inventive as they provide guidelines for design in a field that lacks such established guidelines. The motivation behind the guidelines can be found in the guidelines section of this chapter, asserting that the preferred change has been made, by clearly providing an answer to what to consider when designing GUIs for this context. Both the guidelines and the software prototype can easily be extended, either by simply utilizing the guidelines for improved design or by continuing the iterative design process of both the prototype and guidelines in tandem. Subsequently, the resulting design artifact and academic contribution in the form of guidelines are presented in this chapter.

7.1 Software Prototype

The final design artifact was implemented as a software prototype in the form of a video player that synchronizes data and video logs. This was done in order to as realistically as possible reproduce the feeling of remotely operating a vehicle during testing, without having direct access to a vehicle. This means that the software prototype could fully reproduce the state of the vehicle as it was at the time of recording while representing said state through a completely new GUI. It does however also mean that while the GUI was interactive in terms of manipulating the GUI, test participants had no control over the state of the vehicle. Despite this fact, the produced prototype was useful in evaluating the dynamics of the design. The resulting design furthermore has value as an example of how one could utilize the guidelines for the specific context of this thesis, thus providing some insight into the usefulness of the guidelines. The design can therefore be used as a visual example of the text-based guidelines.

Striving for high SA has been a constant focus throughout the project, and as a consequence, the structure of the final design somewhat reflects the three levels of SA as described by Endsley and Jones [15]. The operator achieves the first level, perception of the situation, by looking at the top-down view, camera views, top bar, and dashboard. Moreover, the operator achieves the second level, comprehension, through detailed explanations of the past and current states in the task and cargo detail views as well as the event log. The operator finally achieves the third level, projection, and thus full situational awareness through predictions and descriptions of future states in the path view, task view, top-down path overlay, and deduction based on event logs.

The final design is presented in this section, providing an overview as well as going into some detail for each component. However, for further details regarding exact design decisions, see section 6.2.3, 6.3.1, and 7.2 for examples in regards to guidelines.

7.1.1 Final Design

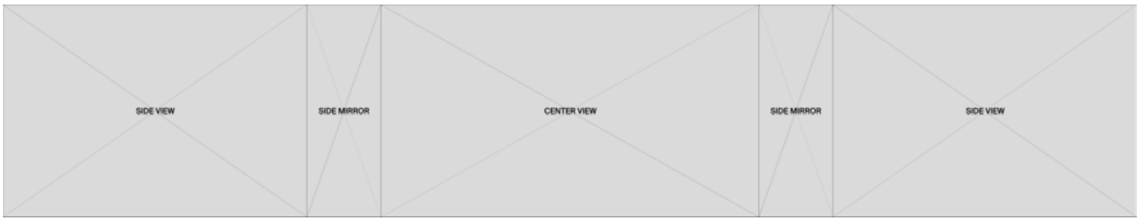
The design was intended to as a whole follow Shneiderman's mantra of "overview first, zoom and filter, then details on demand." [52, p. 1], in addition to the guidelines later presented in section 7.2. Appropriately, figure 7.1 provides an overview of the final prototype's center screen with all components in their default, "positive" states. In other words, shown components are minimized and without errors or warnings as indicated by their green colors.

Figure 7.1: The final design. The system is fully operational and optional components have been minimized.

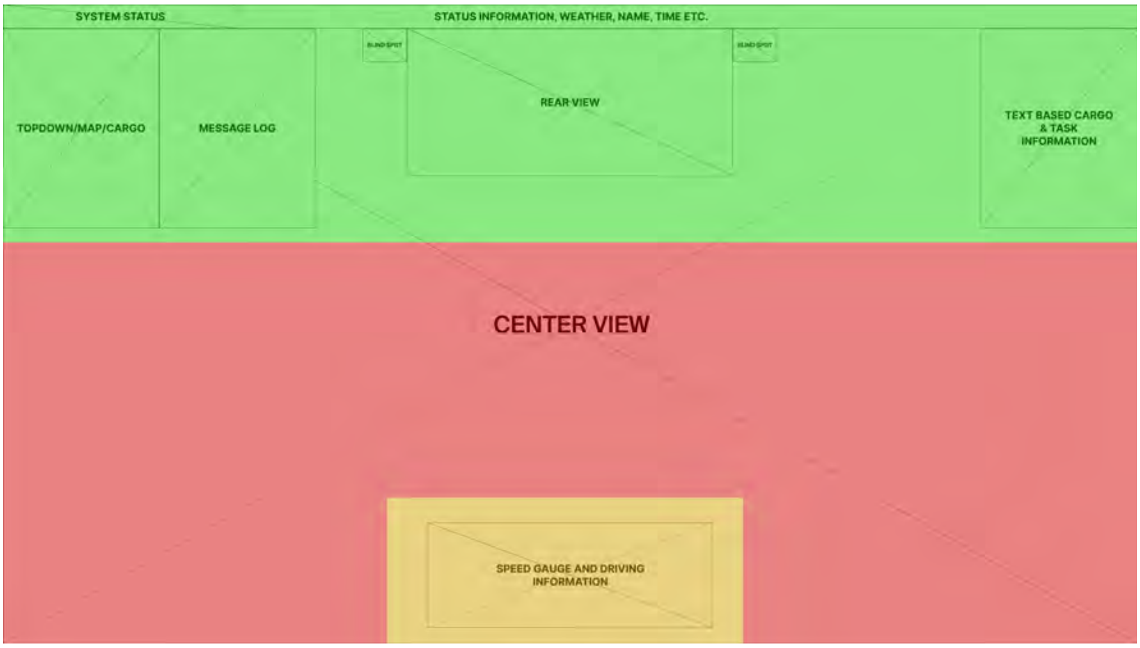
In figure 7.2, the GUI is presented with many components in their maximized state. In this version, the vehicle is in reverse, as can be seen in the dashboard, and therefore shows an expanded rear-view camera. Additionally, the top-down view is open, indicating a warning which can be seen in place, in the event log as well as in the top bar where the ASB indicator is notifying the operator that it has detected an object.

Figure 7.2: The design in a fully operational mode where some optional components have been maximized.

The general layout of the GUI in terms of video displays can be seen in figure 7.3. Furthermore, the positioning of elements in broad terms can be seen in figure 7.3b, for more details about the partitioning shown, and the chosen layout the reader is referred to section 6.2.3.



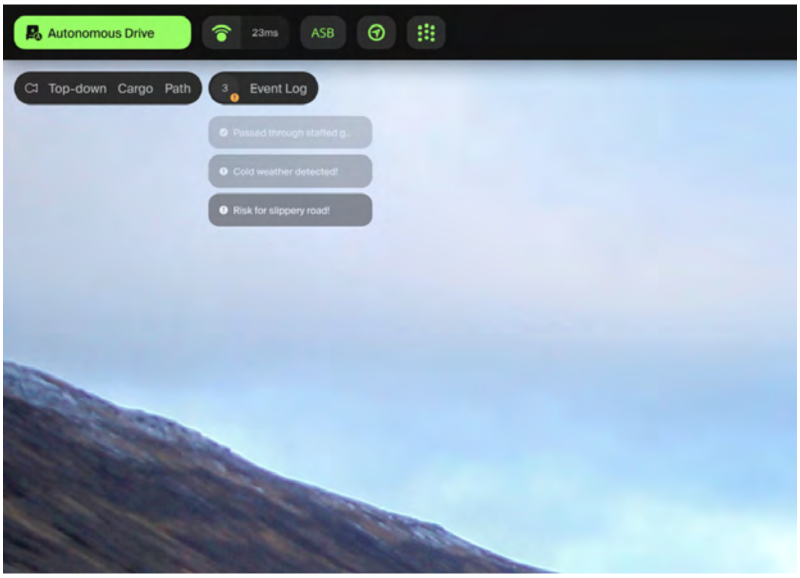
(a) The final layout of the design with the side mirrors placed on either side of the center screen, with the side views on either side of them.



(b) Basic element placement and video importance hierarchy, where green signifies the most safe area for element placement.

Figure 7.3: The layout used for the final design

In the top left corner, the overview and event log components are attached to the top bar. These can both be minimized, as seen in figure 7.4a and maximized as seen in figure 7.4b. As such, the information density of the GUI can vary heavily depending on its current state. This area also displays the general state of the vehicle through banners displaying autonomous state, internet, ASB, GNSS, and lidar status. The messages in the event log show urgency through an icon and color, a text-based message, and time both as an absolute time stamp as well as how much time has passed since their onset, as can be seen in figure 7.4b.

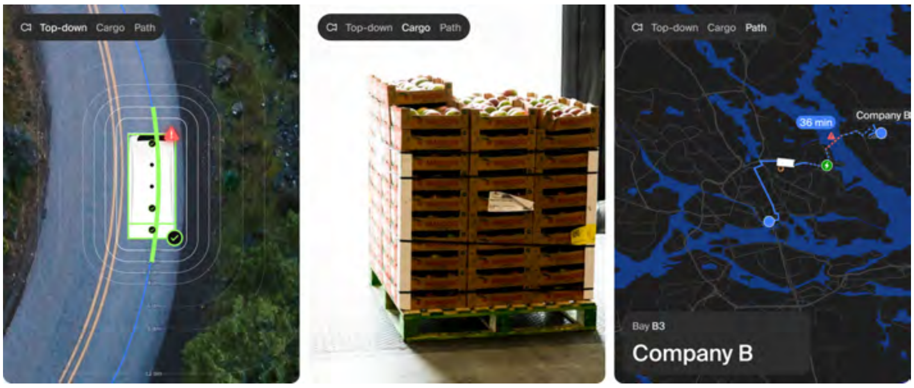


(a) Minimized overview and event log components with positive state.

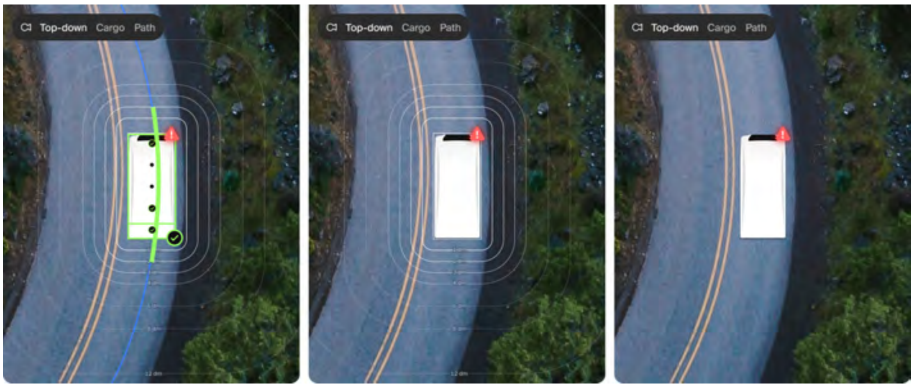
(b) Maximized overview and event log components as well as some errors and warnings.

Figure 7.4: Top left area of the screen displaying technical information about vehicle state, the overview, and event log components.

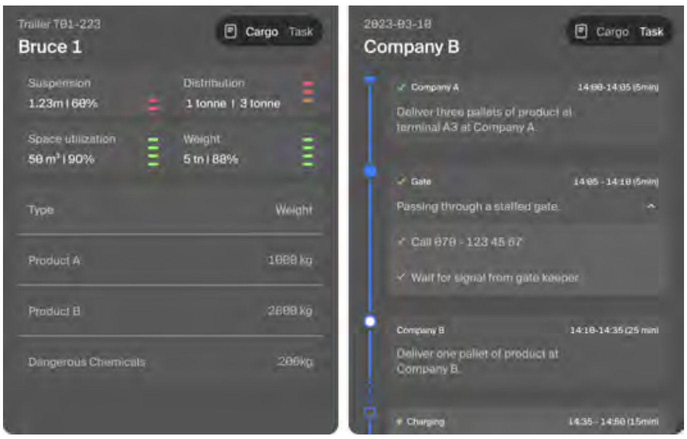
The overview and details components, as can be seen in multiple variations in figure 7.5, were intended to provide the operator with SA at different levels. The overview component was intended to provide, as is implied, an overview and was therefore based on primarily visual information in the form of video, as well as satellite images. Moreover, the top-down view provides a number of different overlays, providing some flexibility in terms of how much information should be shown. The details component on the other hand was a text-based component displaying more detailed information, partly related to what is shown in the overview component. This component was furthermore intended for the operator to use when looking to gain a deeper comprehension of certain aspects of the situation. This can for instance be seen in the task view, which was intended to complement both the top-down view and the path view through similarly coded progression in combination with additional information.



(a) The views of the overview component. From left to right: Top-down, Cargo, Path.



(b) The top-down view with different overlays applied.



(c) The views of the details component. To the left: Cargo, to the right: Task.

Figure 7.5: The views of the overview component, the top-down view with different overlays enabled as well as the views of the details component.

The dashboard component, as seen in figure 7.6a gathers all relevant information for driving, among other things showing current speed, brake/throttle gauges, gear, and autonomous state. It also displays current system degradation if present, and provides both a maximized and minimized form. The dashboard banner, as can be seen in figure 7.6b, also has multiple uses, functioning both to display the current autonomous state, as well as providing the operator with instructions if needed, such as for performing checklists.

(a) To the left the dashboard is shown in its maximized and minimized state. To the right, the dashboard in its maximized and minimized state shows degradation in terms of steering angle and speed.

(b) The dashboard status banner in different states, both providing information about the autonomous state and acting as a checklist.

Figure 7.6: The dashboard component in its final iteration accompanied by the dashboard banner in multiple states.

7.2 Guidelines

It was realized that although the guidelines covered many important topics, they were both too detailed and extensive to be helpful. Consequently, during the last iteration, the guidelines were thoroughly reworked to present the final guidelines in a more cohesive, precise, and concise form. Due to the complex nature of the problem, guidelines can be presented in endless ways as there are multiple possible perspectives for each concept, which also entails that some level of repetition is inevitable. One notable change made from the previous iteration is the removal of the first guideline which was combined into other guidelines for clarity and conciseness. Additionally, guidelines were added as a result of usability testing, including advice to help operators gauge distances and visualization of system degradation. Minor changes also include improved language, reduced repetition, removal of considerations too unspecific or irrelevant to RO GUIs, removal of non-constructive advice, and other general restructuring.

These guidelines aim to provide helpful advice to tackle challenges that have specific importance to GUI design for RO of AVs. For general design considerations, the readers are instead referred to established literature and best practices such as those provided in this report by for example Norman and Draper [53], Ware [50], Endsley and Jones [15], or Cooper [47]. The guidelines presented in this thesis introduce the concepts of **Prioritization** (G1), **Layout** (G2), and **Coding** (G3), as the primary aspects of presenting information to the user to achieve high SA in RO GUIs. Additionally, the guidelines cover specific considerations for **Feedback** (G4), **Interaction** (G5), and **Trust** (G6) as these areas greatly affect the usability of RO GUIs. Furthermore, each major guideline is complemented with sub-guidelines in an attempt to provide more concrete advice.

7.2.1 Guideline 1 - Prioritization

*Utilize **Prioritization** to provide information and interactions based on contribution to telepresence, frequency of use, urgency, and importance to the user's goals.*

a. *Aim for a stable and moderate **Cognitive Load** to achieve appropriate **Vigilance** and high **Situation Awareness (SA)***

The first guideline focuses on prioritization of what information and interaction to provide and when. First and foremost, it urges the designer to prioritize information that is relevant for increased telepresence, such as video. Moreover, what information is displayed should be prioritized based on frequency and urgency, which can be accommodated through applying Shneiderman's mantra [52] in combination with epistemic actions. Additionally, this guideline also implies that what information should be prioritized and shown can change depending on the user's goal and context. Furthermore, this guideline is particularly critical due to this type of GUI being applied in high-stakes and high-cognitive effort situations.

An example of how to apply the first part of the guideline, i.e., prioritize what information to display can be seen in figure 7.1. In this case, only a few elements are persistently displayed. These are the top bar, which is utilized for displaying important system information while minimally covering video information, and the dashboard which is showing only highly relevant driving information while covering video information of lower importance.

G1a encourages the designer to keep the user's SA high by managing the perceived workload of the user by in turn managing the cognitive load of the user. A low workload means that the user is less engaged in their tasks, and this can lead to low vigilance, affecting SA negatively [15]. A high workload on the other hand increases vigilance, where too much vigilance can lead to increased cognitive load which in turn affects SA negatively as well. Moreover, low SA could lead to a non-realistic driving feeling, reducing the user's telepresence, and making drivers perform differently compared to actually sitting in the vehicle [6]. Low telepresence might for instance result in operators having a hard time empathizing with road users which could have disastrous effects. The key is therefore to balance cognitive load to achieve maximum telepresence and SA.

Cognitive load can be increased by occupying the operator with preparation and meta-tasks and should not be increased through excise. Through such preparation and meta-tasks, the operator could work proactively to gain control over future situations for a better user experience. One way this is done in the designed software prototype is through the task view in the top right corner, where the operator can gain access to upcoming tasks for this particular vehicle as can be seen in figure 7.2. Moreover, this image also shows the closed event log displaying contextually relevant messages even if the event log is closed. The closed event log thus acts as a filter, only displaying information that is urgent at this moment while also relevant to the user's current goals. This furthermore combats the "hours of boredom punctuated by moments of terror"-phenomenon described by Sheridan and Parasuraman [54], providing better conditions for flow while reducing anxiety, fatigue, and data overload. Something that is especially important due to the sovereign posture of these types of applications, indicating long periods of use.

7.2.2 Guideline 2 - Layout

*Design for a **Layout** that minimizes distance between frequently used and urgent information.*

a. *Consider the trade-off between covering critical video information and reducing the distance between elements.*

b. *Consider **Attentional Tunneling** and decreased **Useful Field of View (UFOV)**.*

While **G1** considers if information should be presented, **G2** instead relates to where the information should be presented, i.e., the layout of the information chosen to display. This guideline asks that the designer considers the distance between elements that are frequently used in combination. This should be minimized to the greatest extent possible in order to reduce excise and cognitive load while also considering what information might be lost or occluded.

There are some additional aspects to consider for the design of a layout, particularly attentional tunneling which is a phenomenon that appears when users "lock in on certain aspects or features of the environment they are trying to process" [15, p.32]. When doing this, they stop performing otherwise regular systematic scans of their surroundings, and their UFOV is reduced. The designer must predict what information might be prone to this phenomenon, and design the layout such that even if an area is subject to attentional tunneling, its distance to more critical information is minimized. For instance, one possible way of breaking attentional tunneling is through interrupts of suitable strength in order to force the user to restore SA by scanning their surroundings. A layout would account for decreased UFOV by making sure elements that risk attentional tunneling are placed close to where such interrupts might occur.

An example of how to work with the general layout in the context of highly prioritized and maximized usage of video information can be seen in figure 6.7. The choice to place the dashboard in the yellow area was also made in order to combat attentional tunneling. This way, if the dashboard becomes the focus of attentional tunneling, it is still in close proximity to the most important video information, and therefore more easily interrupted by any changes in the video. How to position specific elements based on relevancy can be seen in multiple components, such as the dashboard gathering all information related to driving in one place, and most of the technical information being placed in the top left corner, both of which can be seen in figure 7.1. An additional example of this, and how it can change during a design process is the changed positioning of side view cameras as can be seen in figure 7.3.

7.2.3 Guideline 3 - Coding

*Utilize information **Coding** to make frequently used, urgent, and critical information more salient.*

a. *Use semantic, redundant, and consistent coding while balancing conventional and new idioms.*

b. *Utilize color, motion, orientation, and size to allow for the use of **Preattentive Processing**.*

c. *Color*

   i. *Use colors that portray seriousness.*

   ii. *Use color sparsely and deliberately.*

d. *Code system messages according to response/message type and severity for an appropriate degree of **Perceived Urgency**.*

While **G2** relates to where information should be presented, **G3** instead relates to how information should be presented, i.e., the **coding** of the information chosen to display. This is in order to direct the user towards the most important information and to group related elements. Poor coding risks misplaced salience making visual search difficult which can be highly detrimental to SA. **G3a** encourages the use of semantic, redundant, and consistent coding, i.e., the designer should consider any cultural or cross-cultural associations or conventions associated with for example colors, such as described in section 3.2.4, shape, icons, phrasing, layout, and general design. This is in order to increase understandability such that warnings are perceived as warnings, and buttons communicate the correct level of affordance such that they portray that they are interactive. Failure to consider semantics and consistency of coding risks producing errant mental models where the operator has the wrong understanding of information. Moreover, this guideline also encourages the use of redundant coding, such as using both icons and color for identification in order to achieve basic accessibility. Furthermore, this should be presented through the use of both old and new idioms in order to achieve a certain degree of familiarity, while also breaking free from any physical limitations traditionally put on designing interfaces for vehicles.

G3b focuses on utilizing different attributes, such as color, motion, orientation, and size in order to aid visual search by enabling preattentive processing [50], and thus increased SA and reduced excise. **G3c** further emphasizes the important role of color. This is done by imploring the designer to be sparse in their color usage, partly due to sovereign applications often being used for longer periods of time, thus demanding modest color usage to reduce cognitive load. Especially when used in combination with video which is often already rich in color. This is also done in order to imply some level of gravity for the actions performed within the application.

G3d urges the designer to code system messages in an appropriate manner such that they are correctly grouped, communicate effectively, and have an appropriate degree of salience, and thereby also an appropriate degree of perceived urgency. This is particularly important given the context of RO where the operator is unable to physically check for any potential issues and is completely reliant on system feedback.

One of the most evident examples of coding used in the design presented in this thesis is the use of color, where the established convention of traffic lights has been applied. This was used for coding system messages in terms of them being positive, warnings, or errors. Moreover, this color was also applied to other elements in order to provide a consistent use of color, such as the statuses of different systems. Color was also used in combination with symbols portraying the same meaning, providing redundant coding which further aids search and supports accessibility for color blindness, as can be seen in figure 7.4a. Preattentive processing is somewhat enabled through the coding used in the event log, especially in its open form if searching for a particular message. Moreover, error coding in the form of symbols and colors primes the user for the preattentive processing of such events, allowing them to more easily detect new errors.

7.2.4 Guideline 4 - Feedback

Prefer Rich Modeless ***Feedback****, display potential consequences of interactions as well as useful interruptions when necessary.*

a. *Utilize motion to grab attention, especially in the periphery.*

   i. *Adapt magnitude of motion according to the desired level of annoyance and amount of other movements present.*

b. *Design to mitigate* ***Change Blindness*** *by grouping related elements and components while limiting the number of simultaneous changes, as well as indicating change for as long as relevant.*

   i. *Introduce elements before their use in critical situations.*

c. *Compensate for physical detachment from the vehicle by*

   i. *helping the operator gauge real-world distances.*

   ii. *presenting the location of cause (physical or otherwise).*

   iii. *visualizing system degradation in place.*

The fourth guideline is focused on keeping the user informed of the system state through rich modeless feedback and grabbing attention to urgent system state through interruptions. **G4a** encourages the use of motion for interruptions due to peripheral human vision being color blind [50], while also asking that such interrupts are not too aggressive, which would make the application unpleasant to use.

Change blindness can be disastrous to SA, where a person does not notice big changes due to saccadic suppression, blinking, or not specifically paying attention to the area concerned [50]. **G4b**, therefore, presents a number of ways to mitigate this phenomenon, heavily inspired by the techniques presented by Budiu [87] and Ware's [50] four basic visual requirements for interrupts. Moreover, it also encourages that elements are introduced before they are required in critical situations due to humans' poor ability to perceive any change that occurs in unexpected locations or using unexpected presentation [50]. However, this must as always be balanced in terms of cognitive load and the amount of information being displayed simultaneously.

G4c asks designers to help the user to accomplish their tasks despite being physically detached from the vehicle and therefore reliant only on sensor data. This means that the operator is unable to for example exit the vehicle in order to check things such as distances, potential system degradation, or cause of errors or interruptions. This guideline, therefore, focuses on providing assistance in these particular aspects.

Potential consequences of actions are somewhat difficult to display for actions dealing with physical changes to the vehicle or its surroundings. One could however argue that this is provided in the design through the use of arrow indicators for the direction of the vehicle, see figure 7.6, as well as the vehicle's intended path shown as a blue line in the top-down view, see figure 7.4b. Multiple applications of motion are used in the design presented, such as the top-bar status indicators shaking when in an error state, see figure 7.4b, and the event log showing new messages despite being closed 7.4a. The amount of motion applied in these cases has been balanced in accordance with their urgency but still requires more testing. Moreover in order to mitigate change blindness, these are displayed for as long as they are relevant, as in the case of the ASB in figure 7.4b, which utilizes motion to indicate that it is damaged until fixed.

Most elements are displayed before they are required in critical situations as they are always displayed either in their minimized or maximized form. There are however some exceptions to this, notably the temporary status indicators displayed on top of the dashboard as can be seen in figure 6.15. It was decided that these statuses occur infrequently and always portray some semi-urgent divergence from the normal operation of the vehicle. Moreover, these statuses all indicate that a system is either turned “on” or “off”, and therefore only needs to be introduced once for a new user to learn while also being strongly tied to the conventional dashboard. Something that can easily be done in an onboarding process. The gain from this is a significantly reduced cognitive load, as well as a more modular GUI. Finally, examples of how to display system degradation can be seen in the dashboard, see figure 7.6a.

7.2.5 Guideline 5 - Interaction

*Provide operator agency through **interaction** that works towards the user’s needs and goals.*

a. *Utilize the strengths of automation and human characteristics respectively depending on the task.*

b. *Considering the specific capabilities of the user by for example providing multiple command modalities.*

c. *Provide error handling when the automated system fails.*

d. *Accommodate seamless state transition between autonomous and remote operation.*

e. *Design for high **Learnability** by providing memorization vectors.*

f. *Support the operator’s meta tasks.*

g. *Allow for dynamic camera behavior and indicate physical locations of camera views.*

The fifth guideline focuses on the interaction with the system afforded to the human operator while emphasizing user-centered design. Designers are specifically encouraged to utilize the operator for tasks they are good at while relying on automation where automation excels with the operator monitoring the automation. One way to consider the strengths of the operator is to consider the specific capabilities of the current user for instance by providing multiple command modalities to ensure that the user can interact with the application according to their ability and skill. Further considering the interplay of humans and automation, Endsley and Jones argue that a human will always be required to “plug the holes” of automation, and therefore needs to be able to intervene [15]. Behymer and Flach add to this by claiming that "all autonomous systems will eventually face problems their designers did not anticipate" [92, p. 105].

In the event of automation facing issues it cannot handle, there are two important factors to consider which are that the user has access to the correct tools as well as that they are immediately able to intervene. **G5c** focuses on the former, and in addition to traditional error handling relevant to sovereign applications, this type of software is also prone to errors regarding the physical vehicle. This in turn could make the operator dependent on external parties for correcting some issues. Error handling for such problems could for example involve providing means of communication with the immediate environment of the vehicle or direct access to external partners. **G5d** handles the latter and one critical aspect regarding state transitions is that there is a trade-off to be made regarding the level of involvement of operators, where high involvement leads to lower unpredictability but also a higher cognitive load. Conversely, less involvement and more unpredictable automation transitions can severely damage SA [54]. The key is to put the contextually relevant needs and goals in focus when providing interaction to avoid complexity creep which could overwhelm the operator, leading to reduced SA. **G5e** argues for memorization vectors described by Cooper *et al.* [47]. This is in order to as quickly as possible move new users from being novices to becoming intermediate users, and as such improve their general performance and SA.

Whereas both the main task of the vehicle and operator is likely to safely maneuver from A to B, the operator has many additional tasks in order to do this, i.e., the "meta-tasks" that **G5f** focuses on. These can range from things such as planning, ensuring that the vehicle is conducting its task in an effective manner, or for example handling communication with third parties such as loading bay or gate personnel. Access to such meta-tasks could be provided through for example epistemic actions. Additionally, worth noting is that operators' goals change dynamically depending on the situation, and interactions should therefore be provided to support this [15].

G5g describes the need for dynamic camera behavior such that camera usage is better suited for the operator's current task. This could for example be done by dynamically changing sizes or display of certain cameras, or preferably by allowing the operator to control the camera behavior. Moreover visualizing the physical locations of cameras can be useful for reducing the risk of confusion regarding vehicle positioning [15].

In terms of utilizing human and automation strengths respectively depending on the context, this software prototype shows the automated state in the default, green, and therefore positively coded mode, as can be seen in figure 7.4a. This means that as long as everything is green, the system deems that the automation can solve the issues at hand, better than the operator. Once this changes to orange, or red, as seen in figure 7.4b, the system is no longer confident in the automation and therefore requests assistance from an operator. As such, the operator is available to "plug the holes" of automation. Moreover, state transition is handled through indicators in the top-left, as seen in figure 7.4b and also displayed in the dashboard, as seen in figure 7.6 in order to ensure that the operator is made aware of any changes to the state. Additionally, while the design was intended to be used both for observing and operating the vehicle, there was no difference in the GUI between the two states except for the autonomous state indicator being changed. This is a conscious trade-off as it provides a smooth transition between states while only relying on the autonomous state indicator as a means of communicating the current state.

Meta-tasks are mainly supported through multiple views, such as the event log, map, and task view, providing information about what is going to happen, and what has happened. This allows the user to keep track of what other tasks are available in order to ensure that the primary task of both vehicle and operator is successful as can be seen in figure 7.2.

Due to prioritization regarding value at this stage of prototyping, no memorization vectors nor multiple command modalities were provided. These are however very important in a sovereign application and should thus be included in a launch-ready application in order to accommodate accessibility issues and improved usability. The same can be said for providing error handling, which was also not explored. This is partly due to the authors not having access to any specifications of how such errors should be handled according to the company guidelines. Moreover, dynamic camera behavior is only partly provided in terms of changing the sizes of views depending on the context, such as increasing the size of the rear-view when changing gear to reverse, as can be seen in the differing size between figure 7.1 and 7.2. Wanted behavior in this regard would be to provide more direct control of the camera angles and positioning by the operator, something that was not possible due to technical limitations. The physical location of cameras was also not provided in this iteration of the design. Instead, a small label indicating which part of the screen contained video from which camera allowed operators who were familiar with the technical setup to easily remind themselves of this. This compromise yielded good results with very little effort.

7.2.6 Guideline 6 - Trust

Design for suitable levels of ***trust*** *in the automated system to achieve appropriate vigilance.*

a. *Increase trust through the use of good computer etiquette by:*

   i. *avoiding unnecessary interruptions.*

   ii. *allowing for safe exploration.*

   iii. *adapting to the users' goals and strengths.*

   iv. *being patient.*

b. *Avoid over-trust by designing an automated system that is transparent*

   i. *that explains what decisions have been made and why.*

   ii. *by presenting users with the capabilities and intentions of the autonomous system.*

   iii. *regarding its level of confidence in decisions made.*

The final guideline focuses on designing an automated system that achieves appropriate vigilance through balancing levels of trust. This guideline provides multiple ways of increasing trust through **G6a** using aspects of computer etiquette while avoiding inappropriately high levels of trust through increased system transparency. A balance is required since over-trust in a system leads to lower SA and more accidents [54], while under-trust in a system can lead to too high vigilance, and high cognitive load, acting as a stressor that with time leads to lowered SA.

In short, some important aspects of achieving good computer etiquette include trying not to break flow through unnecessary and rude interrupts [56]. Interrupts have the potential to contribute to operator anxiety and stress through an increased workload, and thus contribute to lower SA. Furthermore, the sporadic nature of work in teleoperation in combination with operators often not being in direct control is likely to reduce operator performance, similar to Endsley and Jones's example of being a passenger in a vehicle [15]. The user should therefore be kept in control of the situation and decision-making, while to the greatest extent possible allowing safe exploration. Additionally, the system should adapt to the capabilities and goals of the operator to the greatest extent possible [57], to achieve two-way communication between man and machine, thus improving accessibility and collaboration. Since human operators in general are slower than computers, patience is one such highly important adaption. Conclusively, given that good computer etiquette can compensate for low system reliability, increasing performance regardless, is an incredibly important tool for increasing trust [54]. That said, the best performance is achieved through both good computer etiquette and high system reliance.

In general, operators choose to rely on their own abilities versus automation depending on their own abilities relative to automation reliability [56]. Therefore, the goal is to achieve a healthy collaboration between the operator and the automation through a moderate workload. Additionally, transparent communication between the operator and system should be provided so that the operator may take control in a controlled manner in situations where automation fails. Some important things for automation to communicate to the operator in a given situation include what is going on, what the system can do, what the system is going to do, and why. Additionally, it should convey its level of confidence in the decisions it has made and will make. Displaying confidence levels needs to be very clear since data displayed on screens has an inherent sense of truth to them [50]. Through transparent communication the automation becomes less unpredictable and therefore reduces the risk of out-of-the-loop syndrome, thus increasing SA.

The design created in this research works to increase trust by only interrupting the operator when strictly necessary and works with rich modeless feedback where possible. The user's attention is somewhat shifted to different elements of the GUI using various degrees of interruptive behavior through movement or changes in color relatively frequently. While this is used to indicate for instance autonomous state, additions to the event log, their level of interruption is depending on urgency. The only interrupts which are truly disruptive are the strongest and most urgent interrupts indicating immediate danger, such as situations bringing the vehicle to an immediate stop while the corresponding system indicators shift in color, turning red, while also using motion to draw attention, thus indicating why the stop has happened.

Providing redo/undo operations for this type of interface is difficult due to many of its actions affecting the physical world. Moreover, while this software prototype did not provide any interactions with real consequences in terms of vehicle state, all forms of exploration and interactions could be deemed "safe". Safe exploration is therefore not accommodated in any specific way in this design. This should however be highly prioritized in order to build trust between operators and automation while providing some level of transparency. Especially in a situation where interactions actually yield real consequences.

Adaption to the users' goals and strengths considered in the design by allowing the user to take the time needed to achieve satisfactory SA. This includes for instance keeping a log of transpired events such that the user, when they have time can catch up to past events. Moreover, this is also part of the system being patient, and not forcing the user to respond to system messages unless strictly necessary. This is also done through extensive use of modeless feedback, and using precise and plain language.

Transparency in the design is achieved through a few components such as the event log which allows the operator to gain an understanding of what has happened and why, as can be seen in figure 7.4a. The vehicle's exact capabilities and intentions are, however, not clearly defined or shown. Its intended path, or trajectory is however displayed in the top-down view through the blue line shown in figure 7.4b. In some ways, the task view also provides some insight into the intentions of the autonomous system, but only on a high level, as can be seen in figure 7.5, and 7.2.

8
Discussion

This chapter discusses the thesis project in retrospect, and as such discusses things such as if the chosen methods were useful, the result, as well as potential suggestions for future research both in terms of how to conduct it and what to focus on. Furthermore, while this report has focused on presenting the process as it was conducted, this chapter also covers some of the changes made to the process and insights gained as a result of overcoming unexpected challenges.

8.1 Method Discussion

One of the earliest choices made for this thesis was to utilize the primary research of Flising & Widerstrand [12] in an attempt to spend more time on the latter stages of design, i.e. prototyping. This approach was successful as the thesis mainly followed the plan. However, some concerns regarding this were brought up relatively early in the first iteration where the authors realized that a lot of time was going to pass without any direct feedback from users. However, an opportunity to perform an expert review soon presented itself, where both users and experts were involved which worked to validate the early design, ensuring that the work was progressing in the right direction. This was greatly appreciated both by the authors and the participants, as it meant that important feedback could be taken into consideration relatively early in the design process and that stakeholders felt involved early in the process. Had this not been done, users would not have been involved until the final stage of the prototyping of this thesis. As such the authors recommend that future designers keep this in mind, and at the very least plan for some sort of involvement of actual users early on, even if the project is an extension of previously conducted user research.

Performing an extensive literature review proved useful for achieving the end result. Furthermore, the fact that the literature was based on contextually relevant insights from Flising & Widerstrand [12], successfully allowed this project to leverage their findings. Taking a broader perspective during the literature study was however key in improving upon their results. Furthermore, the field trip to AstaZero proved useful, providing more contextually specific insights such as the difficulty in measuring distances when teleoperating. The recommendation is then that while a heavy focus on second-hand information can be useful if parts are highly contextually relevant, it should be complemented with at least some first-hand empathizing early on.

During the define phase of the first iteration, the plan was to solely rely on affinity diagramming, however, the creation of a system diagram was also found necessary to complement some of its weaknesses. Although not initially planned, this combination of affinity diagramming and system diagramming proved effective. Through affinity diagramming some general clusters, and affinities could be uncovered, and through the system diagram, the complex relations between these could be modeled. This both made the dynamics of the problem much more clear while providing an overview of what concepts were important. The authors, therefore, recommend this combination of methods for distilling large amounts of latent information into a more digestible form for improved understanding. However, one potential issue with this method is that it further processes already-processed second-hand information. This down-sampling risks losing important insights and skewing results, similar to how information is skewed in the telephone game. One way to mitigate this could be to include first-hand information, such as insights from a field trip, which could help keep information grounded in truth.

Flising & Widerstrand mention in their thesis, that had they redone their design process, they would have involved users already in the ideation phase. This would have served as a form of empathizing, producing insights into what the users want, while also generating an abundance of ideas from the perspective of someone with a lot of domain knowledge. However, it is worth mentioning that there is a risk of designing too much for what the users believe they want and need as they often cannot formulate what they want due to a lack of design knowledge. Furthermore, basing the design on a few target users risks overfitting the design to these particular individuals. This is especially relevant considering that the target audience might change in the future and that the legal and technical environment may rapidly change. The method instead applied in this thesis where theory was the primary basis for ideation and where users were involved in confirming or denying findings from theory, might therefore have been suitable in the end. If however involving users in ideation as proposed by Flising & Widerstrand, it is likely important to filter the results through some kind of theoretical foundation.

Many of the initial designs of the prototype did not prioritize urgency as intended due to its color usage. As a consequence, relatively much time was spent ensuring a suitable degree of color usage, while also following established color conventions. This process mostly consisted of removing color from areas that were deemed to contain too much color, therefore affecting the perceived urgency and visual search negatively. As such it might be useful to in the future start out with a GUI that is designed in grayscale, slowly and deliberately introducing color. While this was done to some degree by designing the layout using placeholder elements, this approach could have been followed more strictly.

During the part of prototyping that included a software development project, an initial attempt was made at agile work with Scrum, utilizing a Scrum board and creating user stories and tasks. However, it was quickly realized that this workflow would only add administrative work to the project due to the small team size. The authors kept in contact daily and the need for a formal workflow was therefore excessive as long as good communication was kept up.

The software prototype that was developed was limited in that it replayed videos of driving, where it would have been better to develop a GUI that would interface with the systems of an actual vehicle. While this would have allowed more accurate usability tests, this solution comes as close as possible without having access to a real vehicle. Moreover, the development of the software prototype could possibly have gone more smoothly had some more research gone into the exact data available, as the current setup required quite a bit of manual work to realistically represent the vehicle state.

A commonly raised point by participants, and an observation made by the authors, was that it was somewhat difficult at times to evaluate the feedback of the GUI due to not being in control of vehicle maneuvering. As such, it would have been preferable to compensate for this by including more interactive tasks in the usability testing as this would at least have increased the level of engagement. In addition to this, given more time, it would have been interesting to include the RO station's physical modes of interaction in the testing, i.e., including the stream deck, dial, and lever and allowing the user to interact with the GUI using those. This would however have required some re-structuring of the tests and therefore also the prototype, and would have required more focus to be put on this particular aspect. It would also have made the GUI less generalizable, as this would have imposed even more limitations on the design. However, it is possible that some method akin to Wizard of Oz [64] could have been explored and would thus have allowed for more involvement by the users.

It is worth noting that the specific background of the participants might somewhat have affected the test results, where many were familiar with the test track used in the scenarios. Another point to be made is the fact that although many had experience with the current GUI, they were effectively beginner users during testing as this was their first time in contact with it. This too put a high strain on the design to be easy to learn and could have contributed to somewhat unrealistic results. One way to combat this issue could have been to explain the GUI before performing the tests, however, this would likely still not solve the issue entirely and could bias the results in the opposite direction.

8.2 Result Discussion

This section presents a discussion regarding the results in its two forms, the final guidelines and the software prototype in addition to a discussion regarding their generalizability and validity.

8.2.1 Software Prototype Discussion

Features were implemented in the software prototype partly based on the ability to test said features and the time required to implement them during the project. As such there are some potentially valuable features that were conceptualized and planned but that were never implemented. One such feature was providing an escape hatch, which would close all open views, and was described in section 6.2.6.

This is a highly relevant part of designing for safe exploration and while not implemented in the design it was still included in **G6**. Furthermore, it is difficult to provide safe exploration through undo and redo functionality since a physical object is being manipulated through the interface. However, the system can be designed to compensate for obviously poor commands from the operator. Aside from that, user customization or configuration has been mentioned multiple times throughout this thesis, both by the authors as well as by the participants during usability testing. This is however also not present much in the design more than allowing users to display the information they want. Focus was instead put on creating a general base design that could be further expanded, and could as such provide customization in future iterations. One possible way of implementing this would have been to allow users to construct presets regarding the positioning and size of elements, or by for example providing multiple versions of different components, such as throttle/brake gauges. Allowing for this type of customization must however be implemented with great care. Especially considering the high-stakes environment, and should therefore only allow limited customization to avoid counter-productive customization.

In addition to acting as a design artifact to exemplify and aid in the development of the guidelines, a secondary informal objective was initially that the prototype software could serve as a tool for rapid prototyping. The tool would have provided a testing environment where the designed GUI, implemented in a modern front-end framework, would run over any combination of data/video logs. This would have allowed developers to rapidly develop the GUI and continuously and safely test it in any scenario. The potential for the prototype to act as a flexible tool was however somewhat negated due to the amount of manual interference required to set up any new scenario. Moreover, more things that would have contributed to a better prototype include having access to more data regarding the decision-making of the autonomous system, and any internal error messages which could have provided more familiar content for the participants.

It was found during testing that the layout assumptions made early in the project seem to have been quite sound. The response was positive as participants expressed that they could find information where it was expected. Another aspect of the layout that seemed to make sense was the top bar portraying overarching and general information. Participants would usually focus on the top bar right away, starting at the top left, in accordance with the western top-to-bottom, left-to-right scanning pattern. Furthermore, a point can also be made that the overview information is usually more urgent than the detailed information to the right, this was reflected during testing as participants would often keep the left components open and the right closed. This means that the right side of the interface will be clear much of the time, and allows operators to notice street signs and traffic lights which appear on the right side of roads within the geographical context. As such, localization of the layout by considering these aspects might be worth investigating.

Something that was noticed during usability testing in regards to **G1**, was that participants seemed to keep a quite stable and moderate cognitive load during long boring stretches of road. A couple of aspects of the design aided in this such as the event log which at relatively continuous intervals would inform the participants of system behavior and thus keep them moderately occupied. The operator was kept in the loop by investigating what was happening, while the chronological order of logs furthermore aided in comprehending the why of what was happening. The participants could also look at the task view when the workload was low to plan for the future. Furthermore, having interactive components to toggle and gain optional information from seemed to in itself provide some manner of baseline optional cognitive load to mitigate boredom, low vigilance, and poor SA.

8.2.2 Guideline Discussion

The guidelines were created using a bottom-up approach where the first iteration of guidelines were extensive and they were subsequently trimmed down across iterations. This meant that whereas the first iteration included most insights gained through researching and defining the problem, the final guidelines only provide the most useful and specifically relevant insights. Through this trimming, exhaustiveness was traded for clarity and conciseness. The resulting guidelines are based on second-hand peer-reviewed research and other trusted second-hand sources in combination with insights gained during this project and could thus be considered somewhat tested. With that said, it is not claimed that the guidelines are perfect in any way and could likely be further improved through further design iterations and evaluation.

While the guidelines were developed in parallel to the design of the GUI, as well as being based on the same research and insights, they were always intended to be more general than the designed GUI. This is because the design is limited to the context of a specific technical setup and resources. Moreover, guidelines should never be followed blindly, but rather provide a base to adapt depending on context. Therefore, the GUI design can be seen as an attempt to follow the guidelines even under restricted forms, and as such provides an example of the potential validity and usefulness of the guidelines as a whole. With that said, there are aspects of the guidelines that were not represented and tested in the GUI design but were still deemed important enough to be included. Some examples of this can be seen in section 7.2.

8.2.3 Generalization & Validity

The resulting design and software implementation is heavily dependent on the context of this thesis and as such are adapted to any technical specifications of both the autonomous system and truck as well as the modes of interaction and desk setup. As such, the design as a whole and software prototype is not particularly generalizable. However, the guidelines produced were designed to be as general as possible and should be useful even outside this specific context. Furthermore, while the guidelines at times refer to video information, they should be adaptable to any other type of visual information such as 3d-visualizations of lidar data or similar. As such, the aim is that they should be useful for anyone designing a GUI for RO of AVs.

While the guidelines are based on insights from research that should generalize quite well to applications in monitoring and supervision, the prototype was designed with direct control as the primary use case and monitoring as the secondary. Furthermore, the prototype was intently designed with the future prospect of 1:N monitoring in mind. One example of how this was considered is through the use of the top-bar, displaying the name of the vehicle and location. This in turn could be used in a minimized form to display a larger view used in the 1:N operation displaying the status of vehicles and if they are in need of assistance, and once selected expand into the 1:1 view presented in this thesis.

The focus on 1:1 operation means that the guidelines have only been deliberately validated for their applicability within 1:1 direct control and monitoring, and it is clear that they require to be further scrutinized and tested by other designers than the authors in order to validate their usefulness, as mentioned in section 8.2.2. The only way to truly test the generalization and validity of the guidelines is to repeatedly design interfaces using them as support and for heuristic evaluation while documenting the process. Additionally, even if an attempt was made to generalize and clarify them during the project, they are still relatively verbose, and could therefore likely be improved on this front as well.

8.3 Ethics

By contributing to improving this technology, and thus its business case, one is promoting and increasing the probability of worldwide adoption. This means that risks and flaws of the technology are propagated and exposed, regarding for instance general road safety, data privacy, and ethically questionable use cases.

One central aspect of a successfully designed RO GUI is in terms of its ability to facilitate safety for everyone involved. The aim was to create an interface that would be as safe as possible which was important for both ethical reasons and for the general success of the product. In general, SA could be considered the primary aspect aimed for during the design and is strongly related to the operator's performance in terms of safely and efficiently bringing the vehicle from A to B. Moreover, telepresence was a central aspect considered for achieving high SA and it has furthermore a direct relation to perceived urgency and effect on empathy for road users. Although the achieved level of telepresence was not confirmed, due to the formative nature of this project, much thought went into raising it. This mainly took the form of following **G1** prioritizing what information to display and when, while keeping a stable and moderate cognitive load, as well as **G2** through careful layout to avoid occluding video information that was deemed critical for telepresence.

In addition to actually increasing safety, for general adoption, it is important that the general public trusts the systems. While the reliability of the autonomous system is important, much falls on the GUI to purvey a sense of trust. One thing to consider is to which degree the public trusts the system, relative to the actual safety of the system. This dynamic would have to be explored through safety tests where vehicles are maneuvered using the new GUI. The reliability could be cross-checked with the perceived sense of trust of the GUI, whereupon trust in the system could be managed using **G6**. Furthermore, in the unfortunate event of an accident, it falls on legislators and policy-makers to decide on accountability. One idea could be to introduce a standardized metric for RO GUIs, where they could be graded based on criterion that look similar to the guidelines presented in this thesis. This would however be difficult as it would strongly depend on the context and operator but might anyhow provide a guide for where accountability is to be placed.

Another important discussion to have is around the potential application of the results of this thesis in the arms industry, which brings up some ethical issues. Teleoperated vehicles have an incredibly potent and dangerous application as weapons and have been researched for decades [11]. In recent times, such weapons have been broadly deployed, surfacing the ethical discussion around detached violence at a distance. While it might serve to decrease the deaths of allied troops, it also risks killing innocents through lost empathy and SA. With that said, any improvement to the technology is likely to end up being applied to such weapons. One aspect of the result of this thesis is that it might produce more effective weapons. However, as previously mentioned, telepresence and safety have been central aspects of the design process which is reflected in both the final guidelines and design artifact. Through for instance increased empathy for people behind the screen due to increased telepresence and minimized malpractice due to increased SA, deaths, and especially accidental deaths, might be prevented.

For the successful widespread application of the resulting guidelines, accessibility is important to consider since every operator will have a different context. Although accessibility was a guideline of its own in the earlier iterations, its intention was instead combined into other guidelines for conciseness and to maintain specificity to the context of RO. This means that by following the guidelines, designs produced should indirectly be taking accessibility into account without explicitly mentioning it. For instance, guidelines such as **G6a** detail how the system should consider the strengths of the user and be patient, this implies that the system should consider the specific context of the user and adapt depending on capabilities. Furthermore **G5a**, describes how the system should consider the specific capabilities of the user when providing interaction. And **G1a** which considers how cognitive load should be managed, which differs from person to person. During the research phase, multiple command modalities as well as redundant coding were found to be important practical measures of achieving baseline accessibility. Redundant coding was therefore generally applied throughout the resulting design artifact. However, multiple command modalities were not explored due to time restrictions.

During usability testing it was important to make it clear to the participants well beforehand what the test was about, to make sure they knew what to expect, and to inform them that the test data would be anonymized. This was done through a form that was sent before the test, the submission of which could be seen to mark their acknowledgment and consent of the context of the test. After data had been gathered it was anonymized by processing individual comments and observations into greater insights.

8.4 Future Work

There is some substantial future work to be done within this field. The natural next step of this thesis would be to continue the design process in a more realistic setting, by conducting tests using real trucks and workstations allowing for real input. This includes, as the iterative process continues, ensuring that all aspects of the guidelines are applied, including some of which were less focused on in this iteration, such as those covering learnability, accessibility, error handling through third parties, and confidence levels of the automation. Another potential future test would be to evaluate the GUI using eye tracking to quantitatively evaluate the assumptions regarding eye gaze, video importance, salience, and layout. Additionally, while not exactly a continuation of this thesis, conducting more research into the end-users, who they are, and what qualities they should have would have the potential for major positive implications.

In addition to continuing the design process with more iterations, it would be interesting to combine the design of GUI and guidelines intended for 1:N operations with the results from this thesis to see how these would integrate. This would be highly relevant since the most likely application of this product is within a scenario where the operator jumps between a 1:N to a 1:1 GUI based on need. Additionally, conducting this type of research and design in parallel with actually constructing the systems underlying the GUI would provide the ability to display system transparency more easily. This would allow more flexibility in design as it could be based on any wanted data rather than being somewhat confined within what data is available. Specifically, the design could then further explore confidence levels of the autonomous system, uncertainty, false messages, and more.

Considering that this thesis was set within the specific Einride context and their current technical setup, it would be interesting to explore other types of hardware and see how they would affect the design and the use of the proposed guidelines. Exploring potential other technical setups, such as additional screens, was also mentioned by multiple participants during usability testing. Using an additional, fourth screen, featuring touch controls could be useful for displaying detailed information and supporting a multitude of interactions. This way the design would not have to make any trade-offs in terms of covering important video information, while still being able to display equal amounts or possibly even more information. Although, while not covering video information this would require operators to move their eyes for greater distances, a trade-off that would be interesting to explore. On the same topic, this thesis has mainly focused on the visual senses, and other types of sensory data such as sound and haptic feedback would be interesting to add. This would most likely result in a more accurate sense of urgency and higher telepresence of the operators.

9

Conclusion

The purpose of this thesis has been to contribute to best practices within the field of remote operation of autonomous vehicles based on existing theory through an iterative design process exploring the design of a GUI for RO. Through this process, the aim has been to answer the following research question:

What should be considered when designing graphical user interfaces for the remote operation of autonomous vehicles?

This wicked problem was tackled through RtD and an iterative design process, producing a set of guidelines that together constitute an answer to this question. The process, consisted of a four-week-long pre-study, followed by two design iterations with an explicit focus on the final prototyping phase where a software prototype was implemented in code. The goal was to answer the research question through the simultaneous design of a high-fidelity software prototype (see section 7.1) and the development of a set of guidelines. The prototype served as a design artifact and an example of how to apply the design guidelines in a particular context, while also functioning as an evaluation tool for the guidelines. The process resulted in a design that was well received by the target audience, who expressed a desire that it would be deployed into the production environment promptly. It also yielded the following set of guidelines as an answer to the presented research question:

G1. *Utilize **Prioritization** to provide information and interactions based on contribution to telepresence, frequency of use, urgency, and importance to the user's goals.*

   a. *Aim for a stable and moderate **Cognitive Load** to achieve appropriate **Vigilance** and high **Situation Awareness (SA)***

G2. *Design for a **Layout** that minimizes distance between frequently used and urgent information.*

   a. *Consider the trade-off between covering critical video information and reducing the distance between elements.*

   b. *Consider **Attentional Tunneling** and decreased **Useful Field of View (UFOV)**.*

G3. *Utilize information **Coding** to make frequently used, urgent, and critical information more salient.*

a. *Use semantic, redundant, and consistent coding while balancing conventional and new idioms.*

b. *Utilize color, motion, orientation, and size to allow for the use of **Preattentive Processing**.*

c. *Color*

   i. *Use colors that portray seriousness.*

   ii. *Use color sparsely and deliberately.*

d. *Code system messages according to response/message type and severity for an appropriate degree of **Perceived Urgency**.*

G4. *Prefer Rich Modeless **Feedback**, display potential consequences of interactions as well as useful interruptions when necessary.*

a. *Utilize motion to grab attention, especially in the periphery.*

   i. *Adapt magnitude of motion according to the desired level of annoyance and amount of other movements present.*

b. *Design to mitigate **Change Blindness** by grouping related elements and components while limiting the number of simultaneous changes, as well as indicating change for as long as relevant.*

   i. *Introduce elements before their use in critical situations.*

c. *Compensate for physical detachment from the vehicle by*

   i. *helping the operator gauge real-world distances.*

   ii. *presenting the location of cause (physical or otherwise).*

   iii. *visualizing system degradation in place.*

G5. *Provide operator agency through **interaction** that works towards the user's needs and goals.*

a. *Utilize the strengths of automation and human characteristics respectively depending on the task.*

b. *Considering the specific capabilities of the user by for example providing multiple command modalities.*

c. *Provide error handling when the automated system fails.*

d. *Accommodate seamless state transition between autonomous and remote operation.*

e. *Design for high **Learnability** by providing memorization vectors.*

f. *Support the operator's meta tasks.*

g. *Allow for dynamic camera behavior and indicate physical locations of camera views.*

G6. *Design for suitable levels of **trust** in the automated system to achieve appropriate vigilance.*

   a. *Increase trust through the use of good computer etiquette by:*

      i. *avoiding unnecessary interruptions.*

      ii. *allowing for safe exploration.*

      iii. *adapting to the users' goals and strengths.*

      iv. *being patient.*

   b. *Avoid over-trust by designing an automated system that is transparent*

      i. *that explains what decisions have been made and why.*

      ii. *by presenting users with the capabilities and intentions of the autonomous system.*

      iii. *regarding its level of confidence in decisions made.*

There are a number of interesting fields for potential upcoming iterations of this design process. This includes, but is not limited to applying the design guidelines to other contexts and technical setups, conducting more usability tests in even more realistic settings where the user is afforded control over the vehicle, as well as evaluation of the current design using eye-tracking.

In conclusion, this thesis has answered the research question by providing a set of guidelines that designers can utilize for future design, while also providing an example of how to apply these guidelines given a particular context.

Bibliography

[1] *Press Releases*, [Online; accessed 16. May 2023], May 2023. [Online]. Available: https://www.einride.tech/press.

[2] L. Mead, [Online; accessed 10. Jan. 2023], May 2021. [Online]. Available: https://www.iisd.org/system/files/2021-05/still-one-earth-sustainable-transport.pdf.

[3] World Bank, *Over 1.25 Million People are Killed on the Road Each Year*, Jan. 2023. [Online]. Available: https://blogs.worldbank.org/opendata/over-125-million-people-are-killed-road-each-year.

[4] United Nations, *Mobilizing sustainable transport for development*, Oct. 2016. [Online]. Available: https://sustainabledevelopment.un.org/content/documents/2375Mobilizing%20Sustainable%20Transport.pdf.

[5] S. Djavadian, R. Tu, B. Farooq, and M. Hatzopoulou, "Multi-Objective Eco-Routing for Dynamic Control of Connected & Automated Vehicles," *arXiv*, May 2020. DOI: 10.1016/j.trc.2020.01.002. eprint: 2005.00815.

[6] S. Gnatzig, F. Chucholowski, T. Tang, and M. Lienkamp, "A system design for teleoperated road vehicles," vol. 2, Jul. 2013. [Online]. Available: https://www.researchgate.net/publication/256946185_A_System_Design_for_Teleoperated_Road_Vehicles.

[7] Einride, *Einride intelligent technologies for movement*, Einride. [Online]. Available: https://www.einride.tech/.

[8] *Einride Autonomous vehicles Gen 2*, Jan. 2023. [Online]. Available: https://www.einride.tech/autonomous/vehicles.

[9] *About Einride*, [Online; accessed 10. Jan. 2023], Jan. 2023. [Online]. Available: https://www.einride.tech/about.

[10] S. Posada, G. Magno, and S. Ridella, *Preliminary evaluation - autopilot & first responder scenes*, Jun. 2022. [Online]. Available: https://static.nhtsa.gov/odi/inv/2022/INOA-EA22002-3184.PDF (visited on 12/05/2022).

[11] T. Fong and C. Thorpe, "Vehicle Teleoperation Interfaces," *Autonomous Robots*, 2001. [Online]. Available: https://www.semanticscholar.org/paper/Vehicle-Teleoperation-Interfaces-Fong-Thorpe/da1f8c956a28425090238695cf54e57fbdc52f68.

[12] J. Flising and E. Widerstrand, *Visualisation and interaction for remote driving*, 2022. [Online]. Available: https://odr.chalmers.se/items/acd6a7f2-e74b-4d6d-9d10-8b7a5ca8e72b.

[13] SAE, *J3016c: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles - sae international*, www.sae.org, Apr.

2021. [Online]. Available: https://www.sae.org/standards/content/j3016_202104/.

[14] *SAE International Releases Updated Visual Chart for Its Levels of Driving Automation Standard for Self-Driving Vehicles*, [Online; accessed 17. Jan. 2023], Jan. 2023. [Online]. Available: https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-%E2%80%9Clevels-of-driving-automation%E2%80%9D-standard-for-self-driving-vehicles.

[15] M. R. Endsley and D. G. Jones, *Designing for Situation Awareness: an Approach to User-Centered Design, Second Edition*. Crc Press, 2016.

[16] *United Nations Treaty Collection*, [Online; accessed 25. Jan. 2023], Jan. 2023. [Online]. Available: https://treaties.un.org/pages/ViewDetailsV.aspx?src=TREATY&mtdsg_no=XI-B-1&chapter=11&Temp=mtdsg5&clang=_en.

[17] *United Nations Treaty Collection*, [Online; accessed 25. Jan. 2023], Jan. 2023. [Online]. Available: https://treaties.un.org/pages/ViewDetailsIII.aspx?src=TREATY&mtdsg_no=XI-B-19&chapter=11.

[18] *EUR-Lex - 32019R2144 - EN - EUR-Lex*, [Online; accessed 25. Jan. 2023], Jan. 2023. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32019R2144.

[19] Ilková, Viktória and Ilka, Adrian, *Legal aspects of autonomous vehicles - an overview*, [Online; accessed 25. Jan. 2023], Nov. 2022. [Online]. Available: https://publications.lib.chalmers.se/records/fulltext/249781/local_249781.pdf.

[20] Transportstyrelsen, *Automated vehicles*, [Online; accessed 25. Jan. 2023], Jan. 2023. [Online]. Available: https://www.transportstyrelsen.se/en/road/Vehicles/self-driving-vehicles.

[21] Transportstyrelsen, *Transportstyrelsens föreskrifter och allmänna råd om tillstånd att bedriva försök med automatiserade fordon*, [Online; accessed 25. Jan. 2023], Jan. 2023. [Online]. Available: https://www.transportstyrelsen.se/TSFS/TSFS%202021_4k.pdf.

[22] *Einride Unveils Disruptive Freight Mobility Solutions at November Release Event*, [Online; accessed 30. Jan. 2023], Nov. 2022. [Online]. Available: https://www.einride.tech/press/einride-november-release.

[23] *Delivering on autonomous transport*, [Online; accessed 17. Jan. 2023], Jan. 2023. [Online]. Available: https://www.sae.org/news/2019/06/einride-delivers-on-autonomous-transport.

[24] Einride, *Einride remote operation: Human-led autonomy*, Einride. [Online]. Available: https://www.einride.tech/autonomous/remote-operation/ (visited on 12/05/2022).

[25] *Chalmers University of Technology | Chalmers*, [Online; accessed 15. Jan. 2023], Jan. 2023. [Online]. Available: https://www.chalmers.se/en/Pages/default.aspx.

[26] S. Jafarnejad, L. Codeca, W. Bronzi, R. Frank, and T. Engel, "A Car Hacking Experiment: When Connectivity Meets Vulnerability," *ResearchGate*, Dec. 2015. DOI: 10.1109/GLOCOMW.2015.7413993.

[27] National Governors Association, *NGA 2017 SUMMER MEETING Introducing the New Chair's Initiative "Ahead of the Curve"*, [Online; accessed 11. Jan. 2023], Jul. 2017. [Online]. Available: https://www.youtube.com/watch?v=2C-A797y8dA&t=3577s.

[28] C. Menon, "Ethical Considerations for Autonomous Vehicles," *ResearchGate*, Aug. 2017. [Online]. Available: https://www.researchgate.net/publication/337421926_Ethical_Considerations_for_Autonomous_Vehicles.

[29] J. Morris, "Why Is Tesla's Full Self-Driving Only Level 2 Autonomous?" *Forbes*, Mar. 2021. [Online]. Available: https://www.forbes.com/sites/jamesmorris/2021/03/13/why-is-teslas-full-self-driving-only-level-2-autonomous/?sh=2097f6c86a32.

[30] B. Duignan, "Trolley problem | Definition, Variations, Arguments, Solutions, & Facts," *Encyclopedia Britannica*, Sep. 2021. [Online]. Available: https://www.britannica.com/topic/trolley-problem.

[31] mars. nasa. gov, *Home | Curiosity NASA's Mars Exploration Program*, [Online; accessed 9. Jan. 2023], Jan. 2023. [Online]. Available: https://mars.nasa.gov/msl/home.

[32] *Autonomous Trucks | Self-Driving Truck | Driverless Vehicles*, [Online; accessed 23. Jan. 2023], Dec. 2022. [Online]. Available: https://www.tusimple.com.

[33] *Frequently Asked Questions | TuSimple*, [Online; accessed 23. Jan. 2023], Nov. 2021. [Online]. Available: https://www.tusimple.com/faq.

[34] *Self-Driving Technology | Automated Transportation Systems*, [Online; accessed 25. Jan. 2023], Oct. 2021. [Online]. Available: https://www.tusimple.com/technology.

[35] *About us Roboauto*, [Online; accessed 10. Jan. 2023], Jan. 2023. [Online]. Available: https://roboauto.tech/about-us.

[36] *Teleoperation Roboauto*, [Online; accessed 10. Jan. 2023], Jan. 2023. [Online]. Available: https://roboauto.tech/solutions/teleoperation.

[37] *Phantom Auto - Remote Operation for Logistics*, [Online; accessed 15. Jan. 2023], Dec. 2022. [Online]. Available: https://phantom.auto.

[38] *Distanced Driver Training | Phantom Auto - Remote Operation for Logistics*, [Online; accessed 15. Jan. 2023], Dec. 2022. [Online]. Available: https://phantom.auto/solutions/distanced-driver-training.

[39] *ITS Conglobal*, [Online; accessed 15. Jan. 2023], Oct. 2020. [Online]. Available: https://itsconglobal.com.

[40] C. Kettwich, A. Schrank, and M. Oehl, "Teleoperation of Highly Automated Vehicles in Public Transport: User-Centered Design of a Human-Machine Interface for Remote-Operation and Its Expert Usability Evaluation," *Multimodal Technol. Interact.*, vol. 5, no. 5, p. 26, May 2021, ISSN: 2414-4088. DOI: 10.3390/mti5050026.

[41] *Welcome to the future of road safety - AstaZero*, [Online; accessed 15. Jan. 2023], Oct. 2022. [Online]. Available: https://www.astazero.com.

[42] *About us - AstaZero*, [Online; accessed 15. Jan. 2023], Jan. 2022. [Online]. Available: https://www.astazero.com/en/about-us.

[43] *Sveriges forskningsinstitut | RISE*, [Online; accessed 15. Jan. 2023], Jan. 2023. [Online]. Available: https://www.ri.se/sv/om-rise.

[44] *Active safety - AstaZero*, [Online; accessed 15. Jan. 2023], Jan. 2022. [Online]. Available: https://www.astazero.com/en/research-innovation/active-safety.

[45] *Test Site - AstaZero*, [Online; accessed 15. Jan. 2023], Feb. 2022. [Online]. Available: https://www.astazero.com/en/tracks-facilities/test-site.

[46] H. Sharp, Y. Rogers, and J. Preece, *Interaction design : beyond human-computer interaction*, 5th ed. Wiley, 2019.

[47] A. Cooper, R. Reimann, and D. Cronin, *About Face: The Essentials of Interaction Design*, 4th ed. John Wiley and Sons, 2014.

[48] W. H. Organization, *Disability and health*, Dec. 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/disability-and-health.

[49] E. Petrick, "Curb cuts and computers: Advocating for design equality in the 1980s.," *Design Issues*, vol. 35, no. 4, pp. 23–32, 2019, ISSN: 07479360. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=bsu&AN=138818914&site=ehost-live&scope=site&authtype=sso&custid=s3911979.

[50] C. Ware, *Information visualization: Perception for design.* Morgan Kaufmann, 2021.

[51] L. Bartram, A. Patra, and M. Stone, "Affective Color in Visualization," in *CHI '17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA: Association for Computing Machinery, May 2017, pp. 1364–1374, ISBN: 978-1-45034655-9. DOI: 10.1145/3025453.3026041.

[52] B. Shneiderman and C. Plaisant, *Designing the user interface : strategies for effective human-computer interaction.* Addison-Wesley, 2010.

[53] D. A. Norman and S. W. Draper, *User centered system design: new perspectives on human-computer interaction.* Lawrence Erlbaum, 1986.

[54] T. Sheridan and R. Parasuraman, "Human-automation interaction," *Reviews of Human Factors and Ergonomics*, vol. 1, pp. 89–129, Jun. 2005. DOI: 10.1518/155723405783703082.

[55] U. N. E. C. for Europe, *Human factors challenges of remote support and control a position paper from hf-irads*, [Online; accessed 6. Mar. 2023], Dec. 2021. [Online]. Available: https://unece.org/fileadmin/DAM/trans/doc/2020/wp29grva/GRVA-07-65e.pdf.

[56] R. Parasuraman and C. A. Miller, "Trust and etiquette in high-criticality automated systems," *Commun. ACM*, vol. 47, no. 4, pp. 51–55, Apr. 2004, ISSN: 0001-0782. DOI: 10.1145/975817.975844. [Online]. Available: https://doi.org/10.1145/975817.975844.

[57] D. J. Rea, "Now you're teleoperating with power: learning from video games to improve teleoperation interfaces," Ph.D. dissertation, University of Manitoba, Feb. 2020. [Online]. Available: https://mspace.lib.umanitoba.ca/xmlui/handle/1993/34559.

[58] J. Forlizzi, J. Zimmerman, and S. Evenson, [Online; accessed 19. Jan. 2023], Aug. 2016. [Online]. Available: https://arl.human.cornell.edu/879Readings/Research%20through%20Design.pdf.

[59] H. W. J. Rittel and M. M. Webber, "Dilemmas in a general theory of planning," *Policy Sciences*, vol. 4, pp. 155–169, Jun. 1973. DOI: 10.1007/bf01405730. [Online]. Available: https://www.sympoetic.net/Managing_Complexity/complexity_files/1973%20Rittel%20and%20Webber%20Wicked%20Problems.pdf (visited on 01/18/2023).

[60] *History*, [Online; accessed 30. Jan. 2023], Jan. 2023. [Online]. Available: https://designthinking.ideo.com/history.

[61] *Framework for Innovation: Design Council's evolved Double Diamond*, [Online; accessed 30. Jan. 2023], Jun. 2022. [Online]. Available: https://www.designcouncil.org.uk/our-work/skills-learning/tools-frameworks/framework-for-innovation-design-councils-evolved-double-diamond.

[62] *Share and engage with the Design Sprint Community*, [Online; accessed 30. Jan. 2023], Jan. 2023. [Online]. Available: https://designsprintkit.withgoogle.com/methodology/overview.

[63] R. F. Dam, "The 5 Stages in the Design Thinking Process," *Interaction Design Foundation*, Jan. 2023. [Online]. Available: https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process.

[64] B. Hanington and B. Martin, *Universal methods of design : 125 ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport Publishers, 2019.

[65] G. Eden, S. Sharma, D. Roy, A. Joshi, J. A. Nocera, and N. Rangaswamy, "Field trip as method: a rapid fieldwork approach," in *IndiaHCI '19: Proceedings of the 10th Indian Conference on Human-Computer Interaction*, New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 1–7, ISBN: 978-1-45037716-4. DOI: 10.1145/3364183.3364188.

[66] *Design Kit*, [Online; accessed 31. Jan. 2023], Jan. 2023. [Online]. Available: https://www.designkit.org/methods/secondary-research.

[67] R. F. Dam and T. Y. Siang, "Stage 2 in the Design Thinking Process: Define the Problem and Interpret the Results," *Interaction Design Foundation*, Jan. 2023. [Online]. Available: https://www.interaction-design.org/literature/article/stage-2-in-the-design-thinking-process-define-the-problem-and-interpret-the-results.

[68] *Design Kit*, [Online; accessed 31. Jan. 2023], Jan. 2023. [Online]. Available: https://www.designkit.org/methods/define-your-audience.

[69] *Design Kit*, [Online; accessed 31. Jan. 2023], Jan. 2023. [Online]. Available: https://www.designkit.org/methods/align-on-your-impact-goals.

[70] *Design Kit*, [Online; accessed 31. Jan. 2023], Jan. 2023. [Online]. Available: https://www.designkit.org/methods/frame-your-design-challenge.

[71] Design Kit, *Brainstorming*, [Online; accessed 30. Jan. 2023], Jan. 2023. [Online]. Available: https://www.designkit.org/methods/28.

[72] Design Kit, *How Might We*, [Online; accessed 30. Jan. 2023], Jan. 2023. [Online]. Available: https://www.designkit.org/methods/how-might-we.

[73] Google, *Crazy 8's*, [Online; accessed 30. Jan. 2023], Jan. 2023. [Online]. Available: https://designsprintkit.withgoogle.com/methodology/phase3-sketch/crazy-8s.

[74] *What is Prototyping?* [Online; accessed 31. Jan. 2023], Jan. 2023. [Online]. Available: https://www.interaction-design.org/literature/topics/prototyping.

[75] *What is Wireframing?* [Online; accessed 7. Feb. 2023], Feb. 2023. [Online]. Available: https://www.interaction-design.org/literature/topics/wireframing.

[76] V. Lenarduzzi and D. Taibi, "Mvp explained: A systematic mapping study on the definitions of minimal viable product," Aug. 2016. DOI: 10.1109/SEAA.2016.56.

[77] *10 Usability Heuristics for User Interface Design*, [Online; accessed 30. Jan. 2023], Dec. 2022. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics.

[78] W. W. Royce, "Managing the development of large software systems: concepts and techniques," in *ICSE '87: Proceedings of the 9th international conference on Software Engineering*, Washington, DC, USA: IEEE Computer Society Press, Mar. 1987, pp. 328–338, ISBN: 978-089791216. DOI: 10.5555/41765.41801.

[79] B. W. Boehm, "A spiral model of software development and enhancement," *Computer*, vol. 21, no. 5, pp. 61–72, 1988. DOI: 10.1109/2.59.

[80] O. Hazzan and Y. Dubinsky, "The agile manifesto," in *Agile Anywhere: Essays on Agile Projects and Beyond*. Cham: Springer International Publishing, 2014, pp. 9–14, ISBN: 978-3-319-10157-6. DOI: 10.1007/978-3-319-10157-6_3. [Online]. Available: https://doi.org/10.1007/978-3-319-10157-6_3.

[81] S. Dhir, D. Kumar, and V. B. Singh, "Success and failure factors that impact on project implementation using agile software development methodology," in *Software Engineering*, M. N. Hoda, N. Chauhan, S. M. K. Quadri, and P. R. Srivastava, Eds., Singapore: Springer Singapore, 2019, pp. 647–654, ISBN: 978-981-10-8848-3.

[82] S. Alsaqqa, S. Sawalha, and H. Abdel-Nabi, "Agile software development: Methodologies and trends," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 11, pp. 246–270, Jul. 2020. DOI: 10.3991/ijim.v14i11.13269. [Online]. Available: https://online-journals.org/index.php/i-jim/article/view/13269.

[83] *Scrum Guide | Scrum Guides*, [Online; accessed 19. Jan. 2023], Dec. 2022. [Online]. Available: https://scrumguides.org/scrum-guide.html.

[84] Figma, *Figma: The collaborative interface design tool*. Figma, 2022. [Online]. Available: https://www.figma.com/.

[85] *React*, [Online; accessed 20. Apr. 2023], Apr. 2023. [Online]. Available: https://react.dev.

[86] S. Amershi, D. Weld, M. Vorvoreanu, *et al.*, "Guidelines for human-ai interaction," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19, Glasgow, Scotland Uk: Association for Comput-

ing Machinery, 2019, pp. 1–13, ISBN: 9781450359702. DOI: 10.1145/3290605.3300233. [Online]. Available: https://doi.org/10.1145/3290605.3300233.

[87] R. Budiu, *Change Blindness in UX: Definition*, [Online; accessed 14. Apr. 2023], Sep. 2019. [Online]. Available: https://www.nngroup.com/articles/change-blindness-definition.

[88] Andersson, Jonas and Rizgary, Daban and Söderman,Mikael and Vännström,Johanna, *Heavy Automated Vehicle Operation Center (HAVOC) - Requirements and HMI design | Vinnova*, [Online; accessed 14. Apr. 2023], Apr. 2023. [Online]. Available: https://www.vinnova.se/en/p/heavy-automated-vehicle-operation-center-havoc---requirements-and-hmi-design.

[89] *What is a context diagram and how do you use it? | MiroBlog*, [Online; accessed 2. Mar. 2023], Dec. 2022. [Online]. Available: https://miro.com/blog/context-diagram.

[90] Microsoft, *TypeScript*, [Online; accessed 27. Apr. 2023], Apr. 2023. [Online]. Available: https://www.typescriptlang.org.

[91] OpenJS Foundation, *Electron*, [Online; accessed 27. Apr. 2023], Apr. 2023. [Online]. Available: https://www.electronjs.org.

[92] K. J. Behymer and J. M. Flach, "From Autonomous Systems to Sociotechnical Systems: Designing Effective Collaborations," *She Ji: The Journal of Design, Economics, and Innovation*, vol. 2, no. 2, pp. 105–114, Jun. 2016, ISSN: 2405-8726. DOI: 10.1016/j.sheji.2016.09.001.

A

Post Usability Testing Questionnaire

At the end of the usability testing, the participants were presented with a questionnaire consisting of eight questions in one way or another related to the guidelines presented in section 6.2.7. These questions were formulated using a seven-level Likert scale, ranging from *Strongly disagree* to *Strongly agree*. The questions used were then used as a basis for a short informal interview, utilizing the responses from the questionnaire to spark discussion, and thus gain a better understanding of the usability of the design.

The questions were formulated as follows:

Q1. There was a suitable level of information being presented.

Q2. I could understand the information presented to me.

Q3. I found the information I was looking for where I expected to find it.

Q4. I could understand what was happening and why.

Q5. I felt as if I was directed to the most important information available.

Q6. I could control the available information to a satisfactory degree.

Q7. The use of color aided my understanding of the situation.

Q8. I was sometimes distracted by elements of the GUI.