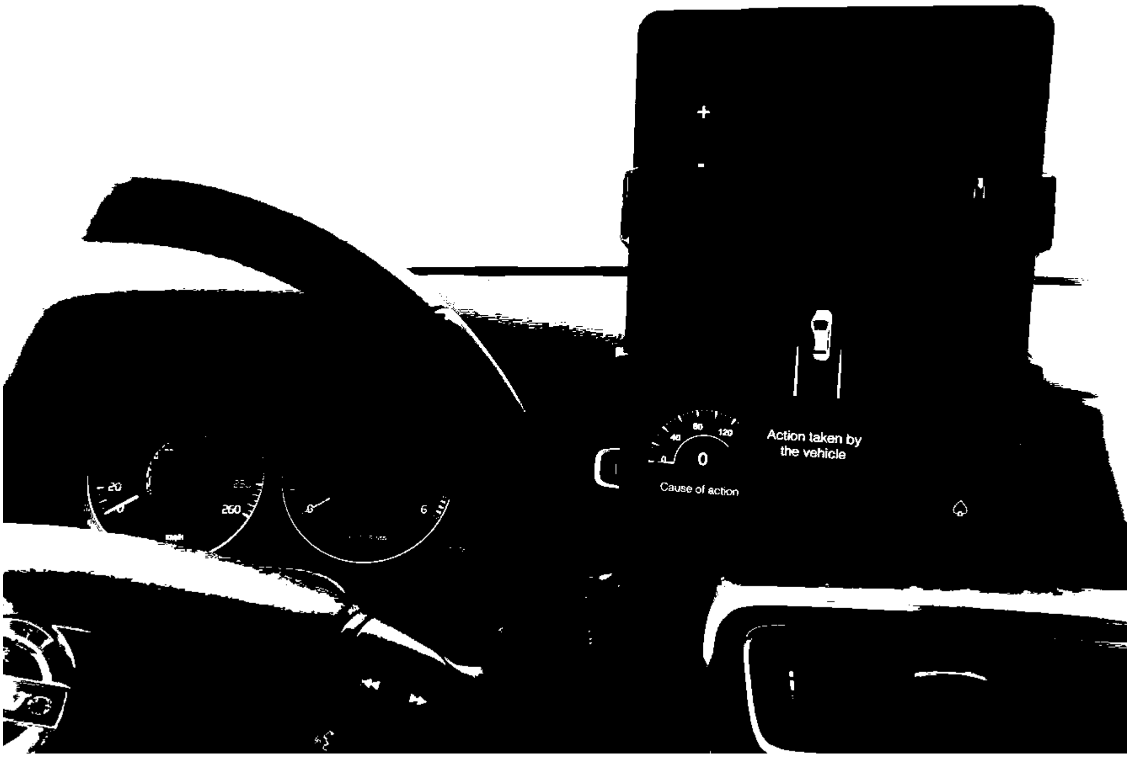
CHALMERS
UNIVERSITY OF TECHNOLOGY



HMI Design for an Automated Vehicle

Constructing an HMI for the Grand Cooperative Driving Challenge

DAVID BERGSTRÖM
AXEL NIKLASSON
LINNEA RUDENIUS

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Bachelor's Thesis

HMI for an Automated Vehicle

Constructing an HMI for the Grand Cooperative Driving Challenge

DAVID BERGSTRÖM
AXEL NIKLASSON
LINNEA RUDENIUS

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals and Systems
Chalmers University of Technology
Gothenburg, Sweden 2016

HMI Design for an Automated Vehicle

DAVID BERGSTRÖM
AXEL NIKLASSON
LINNEA RUDENIUS

Supervisor: Paolo Falcone, Department of Signals and Systems
Examiner: Jonas Fredriksson, Department of Signals and Systems

Bachelor's Thesis
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Gothenburg

Cover: Volvo S60 interior with the HMI mounted at the centre console.

Typeset in LaTeX
Gothenburg, Sweden 2016

HMI for an Automated Vehicle

DAVID BERGSTRÖM
AXEL NIKLASSON
LINNEA RUDENIUS
Department of Signals and Systems
Chalmers University of Technology

Abstract

This thesis describes the construction of a human machine interface (HMI), located at the centre console of an automated vehicle. The interface is to be part of a Chalmers team vehicle in a competition called the *Grand Cooperative Driving Challenge*. This challenge consists of cooperative, autonomous vehicles performing three pre-defined scenarios whilst being judged by a jury on site. The scenarios focus on realistic traffic situations, making them relevant to the development of autonomous connected vehicles. The purpose of the construction was to create an HMI that meets the demands of the challenge, as the interface determines 20% of the final score. The aim for the HMI was to present the most important information to the driver, and not to implement an infotainment system or other functionality you might find in a manually driven vehicle today. HMI is an important factor in the autonomous vehicle, as it can provide a feeling of safety and trust if designed in a proper way.

The HMI was implemented as a web application. A Raspberry Pi 3 was used to serve the application, while an iPad Air 2 mounted to the centre console was used to present the HMI to the driver. The iPad was connected to the server using a Wi-Fi network set up in the vehicle. To create a design for the HMI, four activities were used; establishing requirements, designing alternatives, prototypes and evaluation. The main purpose of the design was to establish a trust between the driver and the vehicle. The final design did, however, heavily rely on the available signals from the vehicle as well as chosen approach for implementation.

The constructed HMI supports fast real-time updates in test environment and presents relevant information on the screen. While the trust between vehicle and driver was the most important factor in the design process, evaluations of these feelings are difficult to conduct. The final design is to be seen as a prototype for general information presentation, and is to be further evaluated.

Keywords: HMI, human machine interface, GCDC, Grand Cooperative Driving Challenge, automated, autonomous, vehicle, web, application.

Glossary

API Application Programming Interface. Defined set of methods used for building web applications.

Create JS JavaScript library used to draw graphics in web applications.

DOM Document Object Model, a tree-like representation of a web page

GCDC Grand Cooperative Driving Challenge

HMI Human Machine Interface. In this thesis, referring to a screen located at the centre console of a vehicle.

JavaScript Programming language for web applications.

Node.js Server implementation in JavaScript.

React JavaScript library for creating user interfaces.

Socket.io Web socket implementation.

UDP Transport layer protocol for sending data packets.

Web socket Protocol enabling a communication channel over a TCP connection.

Contents

List of Figures

List of Tables

1
Introduction

Autonomous vehicles are designed to drive without the help of a human operator. These vehicles have a lot of advantages, such as a reduced number of car accidents as well as positive effects on energy consumption and pollution, and are currently being tested by a number of companies like Google, Tesla and Audi (Keating 2015). There are two types of self-driving vehicles today; the first and most well-known is the autonomous vehicle which uses on-board sensors to find its way. The second type is the connected (cooperative) vehicle which, in addition to on-board sensors, relies on the communication with other vehicles as well as the road infrastructure. Because of its connectivity, cooperative driving has positive effects on traffic flow, road safety and emissions (Rathenau Instituut 2015).

There are different perceptions of the autonomous vehicle; some predict the autonomous vehicles will be on the roads at the end of 2016 and others predict it will not happen in our lifetime (Keating 2015). In order for the autonomous vehicle to be on our roads, the technology has to be fully developed. Another important aspect is whether humans will trust vehicles enough to let go of control. One way to begin establishing a trust could be by letting the driver interact with the vehicle, and vice versa.

A Human-Machine Interface (HMI) is a component that allows human-machine interactions. Usually, user inputs are translated into signals for the machine and the machine presents the result to the user. A vehicle HMI consists of many elements such as a steering wheel, a gearshift lever or a screen located at the centre console, and can provide both information and entertainment for the driver. In an autonomous vehicle, the HMI becomes an important part of establishing trust between the driver and the system (Jong Kyu Choi and Yong Gu Ji 2015). As this trust is essential for future progress, the subject of HMI is highly relevant in the development of autonomous vehicles.

A vehicle HMI includes everything that makes it possible for the driver and system to interact with one another. However, in this project, the only part of the vehicle HMI considered is a screen located at the centre console of the vehicle. This is also what is being referred to when HMI is mentioned in this thesis.

1.1 The i-GAME Project

The i-GAME project is an initiative that aims to speed up the development of cooperative vehicles during a three year project scope (iGAME 2016). I-GAME uses a two way approach to design and construct automated driving systems. The first part is the construction of a functional architecture which is later used in the second part of the project; a multi-vendor driving challenge called Grand Cooperative Driving Challenge, referred to as GCDC. This challenge consists of three scenarios taking place on a highway where two of these will be graded by a jury on site. The scenarios focus on realistic traffic situations, making them relevant to the development of autonomous connected vehicles. The scenarios both promote a realistic introduction to automated vehicles in real traffic situations and serve an academic purpose. By studying the implementation of autonomous cooperative vehicles in this environment, hopefully an understanding of how these techniques could be introduced in real life will be gained.

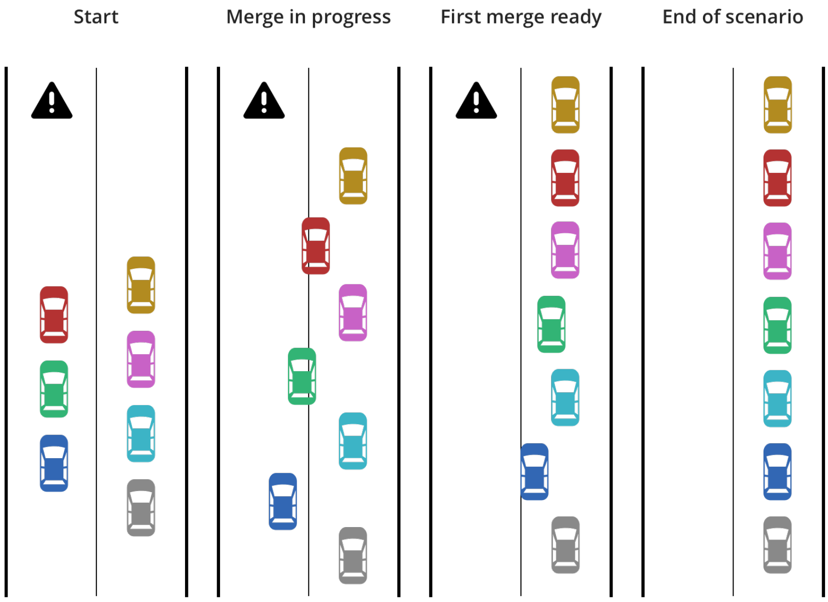


Figure 1.1: Illustration of Platooning Scenario

The first of the GCDC scenarios involves two platoons merging into one platoon on a highway. The two platoons are approaching a construction site and have to merge into one lane in order to be able to pass the site. The merge should take place before the construction site and in a specific area. The scenario is illustrated in figure 1.1. The second scenario consists of a T-intersection and three vehicles trying to pass it. All vehicles are approaching the intersection from different directions, and the goal is for the vehicle not on the main road to enter it. In order to do this, the two vehicles on the main road have to adjust their speed and let the incoming vehicle pass. This scenario is illustrated in figure 1.2. The third and final GCDC scenario involves, again, two platoons on a highway. This time, an emergency vehicle is approaching the platoons from behind. The vehicles in the platoons have to make room for the emergency vehicle to pass. This scenario is illustrated in figure 1.3.

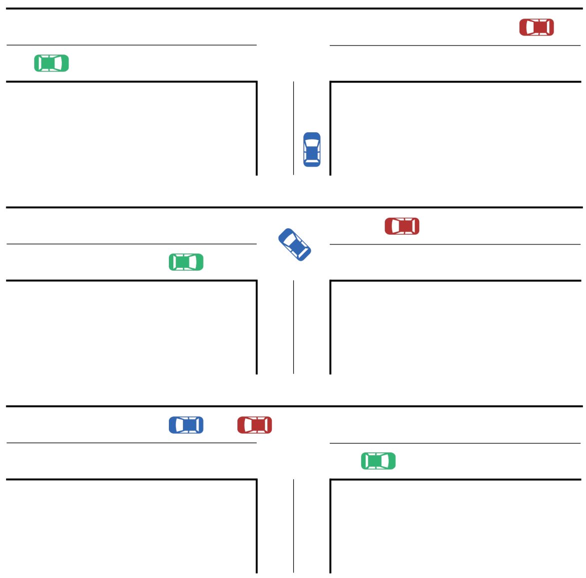


Figure 1.2: Illustration of Intersection Scenario

Chalmers will participate in GCDC 2016 with two teams this year. This is the first time that an in-vehicle HMI is one of the requirements and a part of the judging criteria. It will determine 20% of the final score, and it will promote things such as intuition and proper handling of safety information.

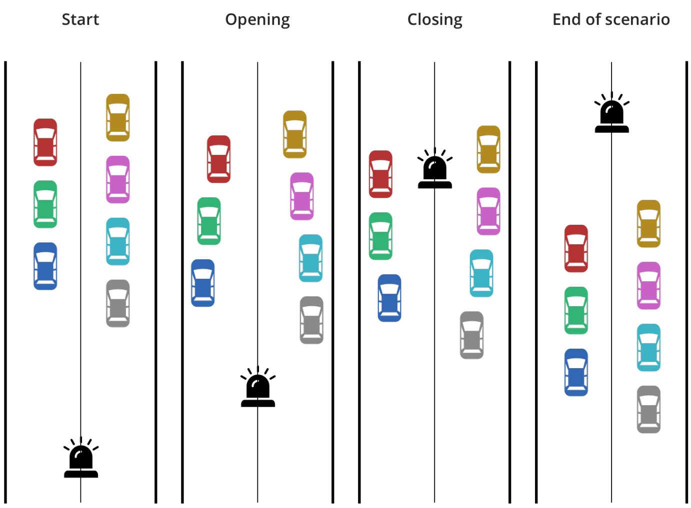


Figure 1.3: Illustration of Emergency Scenario

1.2 Purpose

The purpose of this project is to create an HMI that meets the demands of the Grand Cooperative Driving Challenge, described in section 2.2.1.1. The HMI will be developed for one of the Chalmers teams participating in GCDC 2016, and be part of the vehicle that this team will enter the competition with. The HMI should give the driver an opportunity to get the idea of the state of the system, as well as providing perceptions and planned behaviour of the vehicle. The HMI will be a part of the final judging in the GCDC, and the aim of this project is to get the highest score possible.

1.3 Scope

This project is focused on creating an HMI that covers the three different GCDC scenarios, presenting the most important information to the driver. The project will neither aim to develop an infotainment system, nor any other extra features normally found in a vehicle HMI. The HMI will mainly support a one way communication (vehicle to driver), and if there is time left, two way communication will be implemented.

Although implementing the HMI is required for GCDC, the area of implementing an HMI in an automated vehicle is not elaborated on in this thesis. The primary study focuses on designing an HMI for an automated vehicle.

1.4 Thesis Outline

Chapter 2 describes the design process of the HMI for GCDC, using an acknowledged interaction design process as a guideline. In Chapter 3, the implementation of the HMI design is described together with the installation in the vehicle. Chapter 4 contains a discussion regarding the project as a whole, followed by a conclusion in Chapter 5.

2
Designing an HMI for an Automated Vehicle

HMI is an important part of autonomous cooperative driving, and requires taking areas into consideration that you may not have to when designing for a non-autonomous vehicle. Besides regular functions, such as GPS or infotainment system, the autonomous vehicle's HMI also needs to include all information that the driver normally gets from manual maneuvering. Due to this, the process of designing an HMI for an autonomous vehicle includes analysing the driver's needs and requirements. In a manually driven vehicle, the driver's main tasks are perceiving, deciding and acting. With these tasks delegated to the system, the duty of the HMI is to present the outcome of the system's perception of surroundings, decision making and decided execution. The awareness of the driver is different depending on who the driver is, which means you need to consider a wide range of personas in order to choose the right information for an HMI that should satisfy all possible users. Another problem regarding the design of the HMI is determining how the information should be displayed. The design should be simple enough for any driver to understand, but still show a number of complex operations. The driver should also be able to easily interact with the HMI, just like in a a non-autonomous vehicle.

2.1 The Process of Interaction Design

The subject of HMI in autonomous vehicles is quite new, which makes it hard to know exactly what the HMI should contain. A lot of the conducted research in the area is not publicly available due to confidentiality, making it even harder to find relevant information. In order to choose a proper design for the HMI, the process of interaction design from Preece, Rogers, and Sharp (2012) was used. It consists of four basic activities that can be carried out iteratively; Establishing Requirements, Designing Alternatives, Prototyping and Evaluation.

2.1.1 Establishing Requirements

The activity of establishing requirements has two goals. The first one is understanding as much possible about the users and their goals so the system can help achieving them. The second is to form a good basis for the design process. The requirements need to be as specific and clear as possible in order to make the verification of fulfillment easier (Preece, Rogers, and Sharp 2012). For this project, three different categories of requirements were used:

- **Information Displayed**
  The different types of information that should be presented by the HMI.
- **Functionality**
  Functionality that the HMI should contain.
- **User Experience**
  Requirements concerning the user experience. What the aim is for the user to feel or think when using the HMI.

2.1.2 Designing Alternatives

The activity of designing alternatives aims to suggesting ideas for meeting the requirements. It is usually divided into two sub-activities: conceptual design and concrete design. For the conceptual design, a conceptual model is created. A conceptual model describes an abstraction outlining what people can do with a product, and what concepts are needed to understand how to interact with it. Concrete design is about generating alternatives for concrete elements in the design such as colours, icons or font (ibid.).

2.1.3 Prototyping

A prototype is a manifestation of a design that allows for interaction and exploration of suitability. They are an effective way to explore design ideas, and support designers in choosing alternatives (ibid.). For this project, two types of prototypes were created; prototypes by sketches and prototypes by implementation.

Prototypes by sketches: A low-fidelity prototype is a prototype that does not need to look like or provide the same functionality as the final product. They are usually simple, cheap and easily modified to support exploration of design ideas, and are not meant to be integrated in the final product (ibid.). The prototypes by sketches in this project can be seen as low-fidelity prototypes. They are created for the exploration of design ideas and provide no animations or functionality.

Prototypes by implementation: A high-fidelity prototype looks like the final product and provides more functionality than a low-fidelity prototype. It is useful for selling ideas and for testing technical issues (ibid.). The prototypes by implementation in this project can be seen as high-fidelity prototypes that are created from the low-fidelity ones, and looks similar to the final product. They are created to test technical limitations and provide a simulation for evaluation.

2.1.4 Evaluation

Evaluation focuses on both the usability of a product, and on the user experience when interacting with it. The activity of evaluation can be carried out in various settings and with shifting focus (ibid.). This project will focus on using semi-structured interviews in a controlled setting involving users, which is a setting where the evaluator control what the user does. The benefits of this type of settings is that they are good at revealing usability problems, while the disadvantage is poor capturing of context of use. The semi-structured interview uses both closed and open questions. A basic script for guidance is used, starting with preplanned questions and then encourages the participant to say more (ibid.).

2.2 GCDC HMI Design Process

Using the process of interaction design described in section 2.1, the GCDC HMI design process was carried out iteratively and with the user in focus. All activities were adjusted to be relevant for the design of an HMI, and conducted with the purpose of meeting the GCDC demands.

2.2.1 Establishing Requirements

The activity of establishing requirements was approached in three different ways; *GCDC Judging Criteria Analysis*, *Getting to Know the User*, and *Examining Scenarios*. The three approaches resulted in three lists of requirements, which were summarised into one final list of requirements.

2.2.1.1 GCDC Judging Criteria Analysis

The GCDC provided a set of judging criteria containing guidelines for designing and developing a good HMI. The guidelines encouraged a multi-directional communication process, meaning that the HMI should not only allow for the vehicle to communicate with the driver, but the driver should also be able to communicate with the system. The following list contains issues the judging criteria suggested the teams should consider in the development of the HMI, as well as evaluation criteria.

Vehicle to Driver Communication

- Vehicle Status Communication: Speed, Acceleration, Driving Mode.
- Goal Communication: Planned Route, Planned Behaviour, Actions to be taken during planned behaviour.
- Verification of Communication: Make sure that the information is well received by the driver.

Driver to Vehicle Communication

- Influencing the vehicle behaviour: Enable drivers to influence planned behaviour based on personal preferences like speed, fuel efficiency or quantity of overtakings.
- Present parameters that driver can alter, such as how actions will be decided by the cooperative system.
- A convenient way of transferring control between system and driver.
- The driver should be able to trust vehicle decisions.

Evaluation Criteria

- Transparency: It should be clear what the information means.
- Innovation: The interface should contain novel interaction methods.
- Aesthetics and minimalism: The interface should provide a good user experience and avoid clutter.
- Coherence/Consistency: The interface should give a well-integrated impression.
- Vision: The interface should reflect a clear vision on what to achieve and how.
- Innovation: The interface should provide an innovative way to interact with the driver.

The GCDC Judging Criteria Analysis resulted in the requirements listed in table 2.1.

Table 2.1: HMI Requirements gathered from GCDC Judging Criteria

InformationDisplayed	*Functionality*	*UserExperience*
• Speed of Vehicle • Acceleration/ Deceleration of Vehicle • Driving Mode of Vehicle • Planned Route • Planned Behaviour • Actions to Be Taken	• Multi-Directional Communication • Changing Personal Preferences • Alter Vehicle Parameters • Smoothly Transfer Control • Innovative Interaction Techniques	• Communication Well Received • Trust in Vehicle Decisions • Easy to Understand at Quick Glance • Easy to Understand Functionality • Visually Attractive • Well-Integrated Impression • Clear what Vehicle Wants to Achieve and How.

2.2.1.2 Getting to Know the User

In order to understand the user and the user's expectations and needs, the following questions were answered.

Who is going to use the HMI?
As for today, the possible user of the HMI would be anyone with a drivers licence. In the future, however, one might think there would not be a need for a drivers licence when riding an autonomous vehicle. Because the aim of the project is to succeed in the GCDC, the main user of the HMI to consider in this project will be the judge of the challenge. It is unknown who this person is, but the driver will probably have more knowledge of autonomous vehicles than most other people.

How is the HMI going to be used?
The HMI consists of a touch screen located at the centre console of the vehicle. The HMI is mainly supposed to present information to the driver, but hopefully it will also allow for the driver to interact with the system. The driver of the autonomous vehicle should be able to look at the HMI and gain an understanding of what the system is thinking or planning to do next.

What in the design might help people?
The HMI is in the vehicle to inform the driver of what is going on. By doing this in a proper way, it will allow for the driver to feel safe and confident in the system. When the driver is confident enough, he will be able to relax and focus on other things while riding the car.

Table 2.2: HMI Requirements Gathered from User Analysis

InformationDisplayed	*Functionality*	*UserExperience*
• Planned Behaviour • Current Behaviour	• Driver Able to Interact With System • Fast Updates • Safe to Use	• Easy to Understand • Driver Recognized the HMI as a Vehicle HMI. • Feelings: Safe, Confident, Relaxed, Aware • Driver Able to Focus on Other Things • Visually Attractive • Easy to Learn • Easy to Remember

What are the user experience goals?
The HMI should be visually attractive and make the user feel safe, confident, relaxed, and aware. The user should not feel the need to look out the window to see if the system is correct.

What are the usability goals?
The HMI should be easy to learn and understand, but also easy to remember. The updates of the HMI should be fast and it should also be safe to use, that is not allow for the user to make dangerous operations.

The answers to these questions resulted in the requirements listed in table 2.2.

2.2.1.3 Examining Scenarios

The GCDC consists of three already defined scenarios, and it is important for the HMI to be able to handle these. However, the HMI is also supposed to be able to handle things not included in the three scenarios. During the challenge, the HMI will be presented to the jury, showing parts of it not covered by the challenge scenarios. In order to be able to show that the HMI was constructed with more scenarios in mind, three additional scenarios were developed.

The three additional scenarios were developed with two main goals. The first one being that all three scenarios should be different, both from one another and from the challenge scenarios. They should all show some unique functionality. The second goal was that all three scenarios had to consist of actions that were possible for the available vehicle to perform without any time consuming implementation.

The actions included in the the six scenarios, planned behaviour and some supplementary information were translated into requirements of the first category; information displayed. The actions also generated some additional requirements of the other categories.

GCDC Scenarios
Scenario 1: Cooperation on Highway
Two platoons are approaching a construction site on a highway. A road site unit sends a message to oncoming traffic that they are approaching a construction site along with information about position and speed limit. The vehicles in the two platoons have to merge into the available lane for passing the site.

Scenario 2: T-intersection
Three vehicles are approaching a T-intersection from different directions. The goal is for the two vehicles on the main road to let the third vehicle enter the main road. They are to do this by adjusting their speed, desirably not coming to a full stop.

Scenario 3: Emergency Vehicle
An emergency vehicle approaches two platoons on a highway. When the vehicles in the platoons receive a message with information about this, they are to move so that

the emergency vehicle can pass. Depending on what lane the vehicle approaches in, and the width of the road, the platoons make space in the middle, to the left or to the right.

Custom Scenarios

Scenario 1: Transition from Manual Driving to Autonomous Driving

The aim of this scenario is to show that the HMI considers the aspect of the transition from manual to autonomous driving. The driver controls the vehicle manually and the HMI lets the driver know that the system is ready for autonomous driving. The driver would have to initiate the transition and confirm it before it would actually happen. Looking at the HMI, it should be clear to the driver what he has to do and when the transition is complete.

Table 2.3: HMI Requirements Gathered from Scenario Development

$InformationDisplayed$	$Functionality$	$UserExperience$
• Planned Behaviour: Lane Change, Letting Vehicle in, Go to Side of Road, Go Back to Road, Let other Vehicle Turn Left at T-intersection, Safe Stop, Overtaking. • Surrounding Vehicles • Construction Site • Emergency Vehicle • Autonomous Driving Possible • System Malfunction / Autonomous Driving Not Possible • Necessary Transition to Manual Driving	• Driver Initiates and Confirms Transition • Alert if Driver Does Not Make Manual Transition if Necessary. • Abort Overtaking	• Driver is Calm and Relaxed • Driver Knows What is Going to Happen • Driver Knows Difference Between Own Vehicle, Other Vehicles and Emergency Vehicles. • Trust in Vehicle Decisions • Driver Knows What Mode He is In. • Clear Transitions.

Scenario 2: Making a safe stop

If something were to happen so that autonomous driving is no longer possible, such as a system malfunction or end of road, the HMI should let the driver know that he has to make a transition to manual driving. If the driver has not taken over control of the vehicle in a given time frame, the vehicle should make a safe stop. The safe stop should be conducted by turning on the hazard lights and the vehicle should pull over.

Scenario 3: Overtaking

The vehicle is in autonomous driving mode and approaches another vehicle that it for some reason needs to pass. The HMI should notify the driver of the upcoming overtaking and during the entire process in order to ensure that the driver feels safe.

The scenarios resulted in the list of requirements presented in table 2.3.

2.2.1.4 Requirements Results

Three lists of requirements were established with different aspects in mind, and the final task was to summarise these lists. While some requirements were unique, some were also repeated in the different lists. Because there were many requirements and not much time, the requirements were divided into different priorities. The repeating requirements were determined to be high priority, while requirements such as the ones included in the three additionally created scenarios were determined to be low priority. The rest were determined to be desired requirements, which resulted in the final list containing three different types of priorities; required, desired, and potential requirements. These are presented in table 2.4, 2.5 and 2.6.

Table 2.4: Final HMI Requirements: Required Attributes

$InformationDisplayed$	$Functionality$	$UserExperience$
• Speed of Vehicle • Acceleration/ Deceleration of Vehicle • Surrounding Vehicles • Coming Up: Construction Site, Emergency Vehicle • Actions To Be Taken • Current Behaviour • Planned Behaviour: Lane Change, Letting Vehicle In, Go to Side of Road, Go Back, Let Other Vehicle Turn Left At T-intersection.	• Fast Updates • Safe To Use	• Communication Well Received • Trust in Vehicle Decisions • Easy to Understand: Easy to Understand at a Quick Glance, Easy to Understand Functionality, Clear What Vehicle Wants to Achieve, Easy to Understand Differences Between Vehicles etc. • Easy to Learn • Easy to Remember • Visually Attractive • Well-Integrated Impression • Driver Recognizes the HMI as a Vehicle HMI. • Feelings: Safe, Confident, Relaxed, Aware, Calm.

Table 2.5: Final HMI Requirements: Desirable Attributes

InformationDisplayed	*Functionality*	*UserExperience*
• Planned Route • Current Driving Mode • Autonomous Driving Possible • System Malfunction/Autonomous Driving Not Possible • Necessary Transition to Manual Driving • Planned Behaviour: Safe Stop, Overtaking	• Driver Initiates and Confirms Transition • Alert if Driver Does Not Make Manual Transition when Necessary	• Clear Transitions

Table 2.6: Final HMI Requirements: Potential Attributes

InformationDisplayed	*Functionality*	*UserExperience*
• Preferences • Parameters • Abort Overtaking	• Multi-Directional Communication • Changing Personal Preferences • Alter Vehicle Parameters • Abort Overtaking	

2.2.2 Designing Alternatives

The activity of designing alternatives was conducted in this project with the aim of getting an overall picture of how the HMI should look and work. The activity was carried out in three steps; suggestions for meeting requirements, conceptual design and concrete design.

2.2.2.1 Suggestions for Meeting Requirements

It was decided from the beginning that the HMI should show a bird's-eye view of the vehicle and the vehicles surrounding it. With this in mind, ideas were discussed for each one of the requirements. For the requirements found in the User Experience category, you can find different solutions depending on how you approach the problem, making it difficult to come up with concrete examples of how to satisfy them. What makes one person feel safe and comfortable, may make another person feel completely opposite. It is important to understand that you can not design a user experience, only design *for* a user experience (Preece, Rogers, and Sharp 2012).

Because of this, ideas were not discussed for each User Experience requirement, but rather the user experience as a whole. The established requirements were, however, taken into consideration during the discussions of the other category requirements as well.

Speed of vehicle
The speed of the vehicle could be presented as a number, or it could be presented by a speedometer. The speedometer alone or in combination with the speed displayed with numbers might be what people are used to seeing in a vehicle. In the HMI of this project, the speedometer might make it easier for the driver to be aware of the speed even when focusing on some other part of the screen, while the speed presented in numbers would give a more precise indication.

Information Displayed: Acceleration/Deceleration
For the acceleration and deceleration, it is no use presenting the information in numbers. As acceleration is measured in meter per second squared, the value is very hard to relate to and the number would not give the driver any sense of how much the vehicle was accelerating. However, there are multiple other ways of displaying the acceleration. One could use some sort of a bar that fills up as the acceleration increases, or a circle showing both amplitude and direction of the acceleration/deceleration.

Information Displayed: Current Driving Mode
There are two driving modes to be considered; manual mode and autonomous mode. It is important that it is clear to the driver what mode is currently activated. While in autonomous mode you need to show a lot of information about what the vehicle is doing, you do not have to show as much in manual mode. The GCDC scenarios only covers autonomous driving, and since the main purpose of the project is to design an HMI for these scenarios, there will not be much focus on the manual driving mode. It is, however, important that there is a significant difference between the driving modes. An example would be having the HMI show a map with vehicle position in manual mode, and then zoom in to show the vehicle and surrounding vehicles in autonomous mode. There should also be some indicator on the screen saying what mode the vehicle is in.

Information Displayed: Surrounding vehicles
In order for the driver to feel safe, the driver has to learn to trust the vehicle. Two alternatives were considered in order to make sure the driver could feel safe and relaxed.

The first alternative was to show everything surrounding the vehicle as accurate as possible. The driver would be able to see exactly what the vehicle was seeing, which could provide a feeling of trust. If the driver were to compare what he saw with what was presented by the HMI, he would probably see that the vehicle was very aware of it's surroundings, which might provide a feeling of safety. However, there was a couple of problems with this approach. First of all, looking at a picture of what is surrounding the vehicle is not the same thing as looking out the window. If the driver were to see something and the HMI presented it in a way he did not agree with, the driver would probably not feel very safe anymore. Also, there are already a lot of information that needs to be presented on the HMI. If it also was supposed to give an accurate picture of the surroundings, it would be hard keeping a minimalistic design.

The second alternative was to not show any information of the surroundings of the vehicle. the HMI would only present to the driver that a certain sensor had spotted something. The driver would not know what the sensor had spotted, only that something was there. Doing his, the driver would not be able to compare what the HMI was presenting and what he saw when looking out the window. This could make the driver feel safer, but not knowing what the vehicle actually sees could also have the opposite effect.

Information Displayed: Coming up: Construction site, Emergency Vehicle
A construction site upfront could be shown by the road work sign, which you often see while driving. Most people probably know what this sign means, and only a quick glance would be needed to understand what the HMI was trying to say. Some sort of progress bar could also be used, showing distance to the road work. An emergency vehicle could be shown in a similar way, with an icon that is easy to understand along with a progress bar showing the vehicle approaching.

Information Displayed: Actions to be taken
Somewhere on the screen it needs to say what actions the vehicle is about to take. This information needs to be short and easy to understand at a quick glance. To accomplish this, the actions could be presented by an icon along with a short description.

Information Displayed: Current behaviour
The current behaviour could be shown just as the upcoming actions, with an icon along with a short description. If the current behaviour is just going straight forward, the HMI could show something like the heading of the vehicle instead.

Information Displayed: Planned behaviour: Lane change
To show that the vehicle is planning a lane change, the HMI could show an arrow in front of the picture of the vehicle on the HMI, showing where the vehicle is going. During the lane change, the picture of the vehicle in the HMI could follow this arrow. The problem with this could be that at the time the arrow is drawn, there might be another vehicle at the position our vehicle is going to enter. Although the vehicle is aware of that, and waits for a gap before making the lane change, it might make the driver feel unsafe seeing that there already is a vehicle in the desired position. Another problem with this approach is that the picture of the vehicle in the HMI has to move. It is desirable that the own vehicle is always shown on the same coordinates of the screen, in order to avoid confusion.

Another approach to this lane change is to not show any arrows, but relying on that the information from actions to be taken and current behaviour is sufficient enough to make the driver understand what the vehicle is about to do. In this case, the picture of the vehicle could start blinking when the lane change is about to happen and instead of moving the picture of the vehicle when the lane change is happening, everything else on the screen could be moved.

Information Displayed: Planned behaviour: Letting vehicle in
In order to show the driver that the vehicle knows that another vehicle in going to enter in front of the own one, the HMI could show a transparent picture of the entering vehicle at the position it is going to claim. However, this approach gives the same problems as mentioned in the lane change scenario. What if, at the time the transparent picture is drawn, there is another vehicle in the desired position? Would this make the driver feel insecure? It might be enough just showing in current behaviour that another vehicle is going to merge in front of the own one.

Information Displayed: Planned behaviour: Go to side of road, go back
As mentioned before, it might be a good idea to have the picture of the own vehicle at the same coordinates of the screen at all times. In this case, all the surroundings will be moved at the screen when the vehicle goes to the side of the road and back.

Information Displayed: Planned behaviour: Let vehicle turn left at T-intersection
The T-intersection is a bit trickier than letting another vehicle change the lane. In the other behaviours, you only need to show vehicles that are driving next to one another, and not a vehicle coming from a different direction. In order to show the T-intersection on the small screen, it might be a good idea to zoom out when approaching it. The exact road may not have to be printed out, but it is important for the driver to understand that the vehicle is aware of the intersection. The vehicle entering the road has a safety distance that the other vehicles are not supposed to enter. This safety distance could be drawn as a circle around the incoming vehicle.

Information Displayed: Planned behaviour: Turn left at T-intersection
If the own vehicle is planning to make a left turn at a T-intersection, the HMI needs to make sure that the driver knows that the vehicle is aware of all the surrounding vehicles. Again, it would probably be a good idea to zoom out and let the driver see a larger part of the road. Also, the intentions of the vehicle have to be clear, and the driver has to understand that the vehicle is about to take a left turn.

Information Displayed: Planned Route
The planned route could be shown in a part of the screen not covered by the birds eye view, however, it might not be room for that. The planned route should be shown on a map, and the map have to be big enough for the driver to be able to interpret the contents. To solve this, the planned route could be part of information that is not always visible, but shows when pressing a button on the screen. For the manual mode, however, a map could always be visible as there would be no

presentation of the surroundings of the vehicle.

Information Displayed: Autonomous driving possible
This information should be displayed when the vehicle is ready to enter autonomous mode. The HMI could present the information in text and with an icon that the driver presses to confirm that he wants to start the transition.

Information Displayed: System Malfunction/Autonomous driving not possible/Necessary transition to manual driving
For any type of system malfunction, the driver should be notified immediately. This notification has to be very clear and make the driver react immediately. For this, it could be good using an alarming sound, and change the colors of the HMI. No functionality will be implemented where the driver uses the HMI to take back control of the vehicle, this will be done by taking over the steering wheel or gas/break pedals. Because of this, the HMI could present a picture and a text of how the driver takes over control.

Information Displayed: Planned Behaviour: Safe stop
The vehicle should make a safe stop if the driver does not take over control when necessary. The HMI should because of this continue to alert the driver, as well as show the safe stop maneuver. If the picture of the vehicle on the screen is always at the same coordinates, this safe stop could be shown by moving the picture of the vehicle from its set coordinates and making it red. The red color would show that something is wrong, and moving the picture would mean that the vehicle is no longer on the road.

Information Displayed: Planned Behaviour: Overtaking
This maneuver might make the driver feel very unsafe if not enough information is displayed in the HMI. Just as with the T-intersection, it would probably be a good idea to zoom out and present a bigger part of the road to the driver. The driver needs to see that the vehicle is planning a safe overtaking, and that there are no other vehicles that might be in the way.

Information Displayed: Abort Overtaking
If the driver feels insecure, he should be able to abort the planned overtaking. The driver should be given this option as soon as possible, and it should be presented in a way that makes it clear to the driver that he has the choice to abort. An alternative to this approach is to always let the driver confirm before conducting an overtaking.

Information Displayed: Preferences/Parameters
Changing the parameters or preferences is a good feature, but may not be as important as showing surrounding vehicles or planned behaviour. Because of this, these options could be hidden inside a menu. The driver should no how to get there without any problems, but the options would not be visible at all times. In this way, the design could still be simple and minimalistic yet provide the desired functionality.

Functionality: Fast Updates
This requirement is attended to in Chapter 3.

Functionality: Safe to Use
In order for the system to be safe to use, and not allow for any dangerous driver operations, the system will be quite closed. Functionality will only be implemented if it is certain that this functionality will not in any way compromise the safety of the system. This requirement is also attended to in Chapter 3.

Figure 2.1: Volvo S60 Centre Console

Functionality: Driver initiates and confirms transition
To make sure that the driver is aware of the transition, a proper way of handling this transition has to be implemented in the HMI. First of all, the HMI may present the alternative to enter autonomous driving, but the driver has to be the one initiating the transition. Due to technical restrictions, the transition will happen when the driver presses a physical button not located on the screen. However, it would be possible for the driver to initiate the transition on the HMI screen, and then confirm by pressing the physical button. The initiation could be done by pressing a button

on the screen and holding for a couple of seconds, in order to prevent driver pressing the button by mistake.

Functionality: Alert if driver does not make manual transition when necessary
The alert should be presented both visually and with sound.

Functionality: Multi-Directional Communication, Changing Personal preferences, Alter vehicle parameters, Abort overtaking
It is desirable that the user can interact with the system in such ways as altering parameters or setting preferences. However, in order to implement this functionality, rigorous testing would be needed as well as help from the other team members. Because of this, there will probably not be time for the implementation of the functionality prior to the competition.

Figure 2.2: Volvo S60 Dashboard

User Experience
In order to keep the design simple, the HMI could present only the most important information. The less important information, that does not have to be presented to the driver at all times, could be hidden inside a menu. Information should, when possible, be presented with icons or pictures instead of text.

A vehicle HMI often has a very minimalistic design, using dark colors and metallic textures that make the HMI look well-integrated in the vehicle. The same style could be used when designing the HMI, in order to make the user recognize it as an in-vehicle HMI. Tim Smith, design specialist, states that the problem with current internally designed in-car HMIs is that they either look hard to use or they are hard to use (Smith et al. 2014). The HMI in this project will not have nearly as many functions as a regular in-vehicle HMI, but it is still important to keep it simple and intuitive.

The dashboard and centre console of the vehicle that the HMI will be implemented in can be seen in figure 2.1 and 2.2. The knowledge of how the interior of the Volvo should be of use when designing the HMI in order to achieve a well-integrated impression.

2.2.2.2 Conceptual Model

During the discussions of suggestions for meeting the requirements, there were a lot of thoughts about how the HMI should look in terms of basic layout. It was decided that a conceptual model should be created, where the basics of the HMI should be defined. The first model was created for an iPad in landscape mode. The reason for choosing landscape mode was that it would give more space for showing information. However, when using this mode, the space in front of the own vehicle became very small. Later, when drawing multiple vehicles in the surroundings, it turned out more space would be needed. The conceptual model was re-evaluated and a new one, where the iPad was placed in portrait mode, was created.
The first conceptual model, created for an iPad placed in landscape mode can be seen in 2.3. The latter, created for an iPad placed in portrait mode can be seen in 2.4.

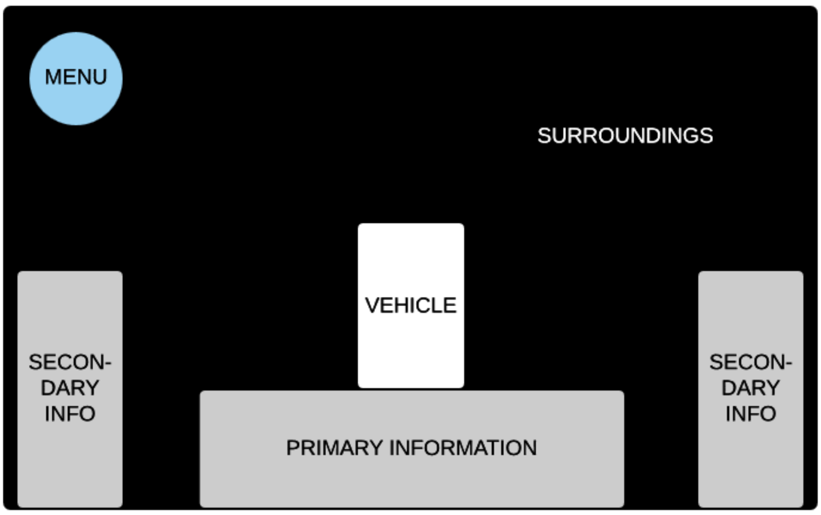


Figure 2.3: Conceptual Model for the HMI using an iPad placed in landscape mode.

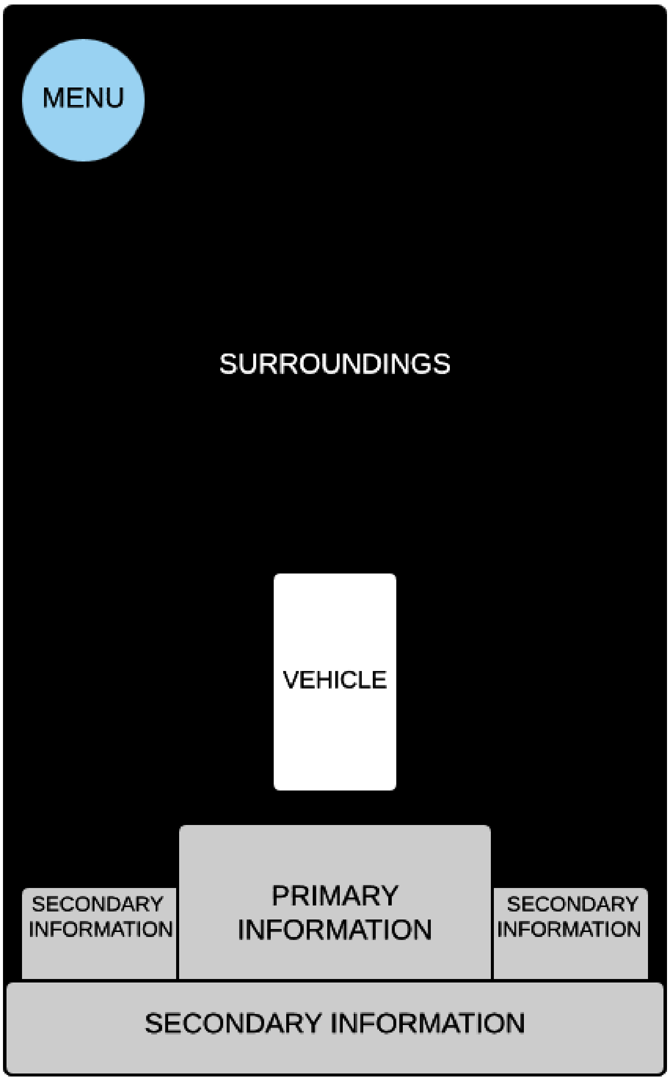


Figure 2.4: Conceptual Model for the HMI using an iPad placed in portrait mode.

2.2.2.3 Concrete Design

During this step of the designing alternatives, decisions about the more concrete design were made with the help of a report on design guidelines for safety of in-vehicle information systems (Stevens et al. 2002). The report was found on the Transport Research Innovation Portal, a portal for information on all transport research and innovation conducted at European and national levels.

Location: The iPad should be placed at the centre console of the vehicle. It should be easy to reach, but not encroach upon or interfere with normal leg, hand or arm movements (ibid.).

Menu: The menu should be used for when the driver wants to alter some parameter or view more detailed information about the system. It should be easy to navigate and simple to return to start. The number of choices in the menu should be limited to three or four (Stevens et al. 2002).

Colours: The choice of colours for the HMI was based on what people are used to seeing in a vehicle HMI, as well as the interior of the Volvo S60. It was decided that the background of the screen should be black, just as the centre console is, so that the iPad would look well-integrated. The colour of the picture of the own vehicle should be similar to the colour of the Volvo, and the colour of other vehicles should be easy to distinguish from the own vehicle. Use of red colour should only be used when alerting. A list of the colours that used in the final design of the HMI is presented in table 2.7.

Table 2.7: Table of Colors Used in the HMI

HMIElement	*Color*	*ColorCode*
Background		RGB(0.0.0)
Text Color		RGB(255.255.255)
Own Vehicle		RGB(219.202.184)
Information of Surroundings		RGB(0,128,181)

Font: Stevens et al. (ibid.) implies that sans serif fonts are recommended, and that no more than two different fonts or two types of emphasis should be used. Because of this, it was decided that the HMI should use Helvetica, plain and bold.

Icons: The icons used should be appropriate to their function and conform to stereotypical norms. When unfamiliar symbols are used, a text label of limited text should accompany them, but should otherwise be avoided (ibid.). The icon that will be used for the own vehicle is designed by Freepik from www.flaticon.com.

Alerting: The HMI should alert by both an alerting tone as well as a visual blinking image that attracts attention. For blinking, a frequency of 2-3Hz would be used. (ibid.).

Information Presentation: The information displayed in the areas of secondary and primary information should be short and accompanied by an icon or visualization making it easier for the user to understand the information by a quick glance. The information regarding planned behaviour should be presented for as long time as possible. The vehicle does not make decisions about the far future, so this might be for a shorter time than what might be sufficient for the driver to be able to observe and understand the information. However, it could be possible for the HMI to store the previously conducted action and have this available for the driver to read if he did not have time to understand it. The quantity of information presented to the driver should not be excessive (ibid.).

2.2.3 Prototyping

The activity of prototyping was carried out with the conceptual design and the concrete design alternatives in mind. The first prototypes were created as sketches, and the later prototypes were created for the implemented system.

2.2.3.1 Prototyping by Sketches

The first prototype was created as sketches, displaying several views that were to be expected in the actual implementation. It was designed to handle basic events in all three GCDC scenarios in order to get an overview of the complete interaction flow.

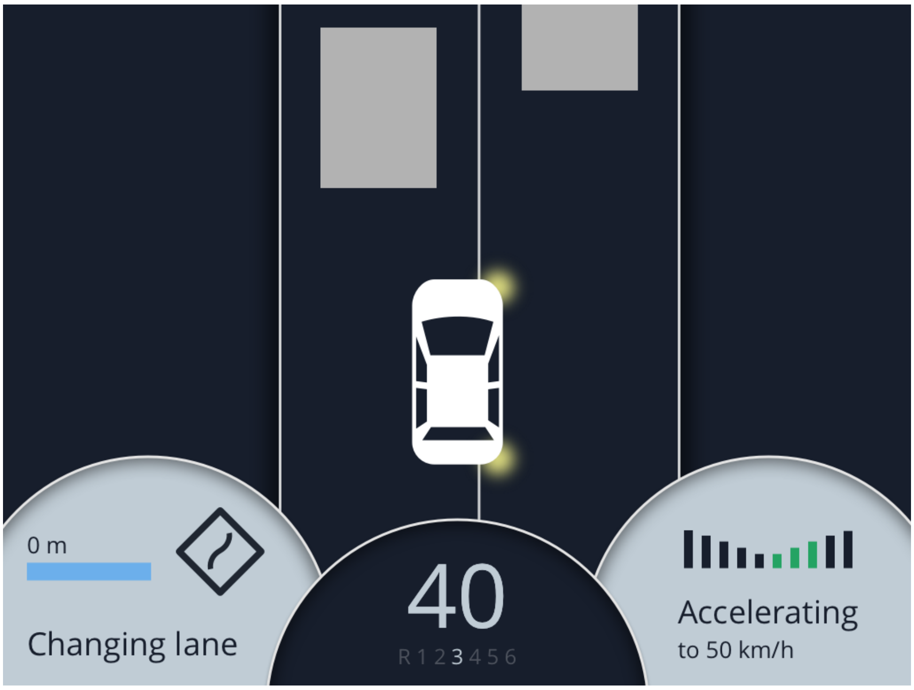


Figure 2.5: First Prototype of Lane Change Scenario

The first sketch of the lane change scenario is presented in figure 2.5. In this sketch, the speed was decided to be the primary information, whereas acceleration and information about upcoming actions were considered to be secondary information. The speed of the vehicle was presented as a number since a speedometer already exists in the vehicle. The acceleration was presented visually, but clarified with a text informing the driver about the desired speed. The information of the lane change is presented with an icon and text, along with a progress bar indicating the remaining time until the action occurs.

For showing the surrounding vehicles, a combination of the two design alternatives presented in 2.2.2.1 were used. The HMI does display the surrounding vehicles, but not any information about what kind of vehicle it is. In a poll from 2014, 70% participants answered yes to the question "Would you like to be able to see that the autonomous car can see?", with the majority stating that the feature would increase their sense of control and reassurance (Basani 2014). This contributed to the decision of actually showing the surrounding vehicles and their position, but by not showing too much information, the driver will not have the same possibility to compare what he sees out the window with what is presented in the HMI.

The lane change is presented as suggested in section 2.2.2.1 With the own vehicle staying at the same position, with the road moving instead. Also, the turn indicators of the own vehicle is presented in the HMI, to show that the vehicle is making a maneuver to the right (or left).

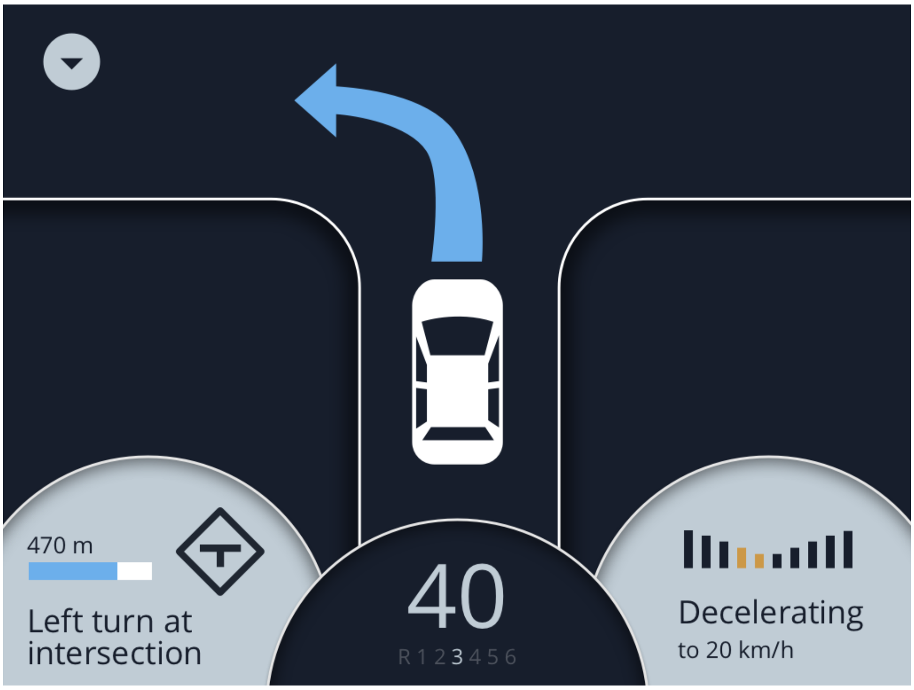


Figure 2.6: First Prototype of Intersection Scenario

The first prototype of the intersection scenario is presented in figure 2.6. The same principles for speed, acceleration, and information of upcoming action is the same as mentioned for the lane change scenario. In this prototype sketch, a button is added in the top left corner. This button, when pressed, is thought to show a menu. To show that the vehicle is going to turn left, an arrow is drawn to show where the vehicle is going.

For the second prototype, the iPad was put in portrait mode to show more of the road ahead. Also, the design was changed to look more integrated in the Volvo.

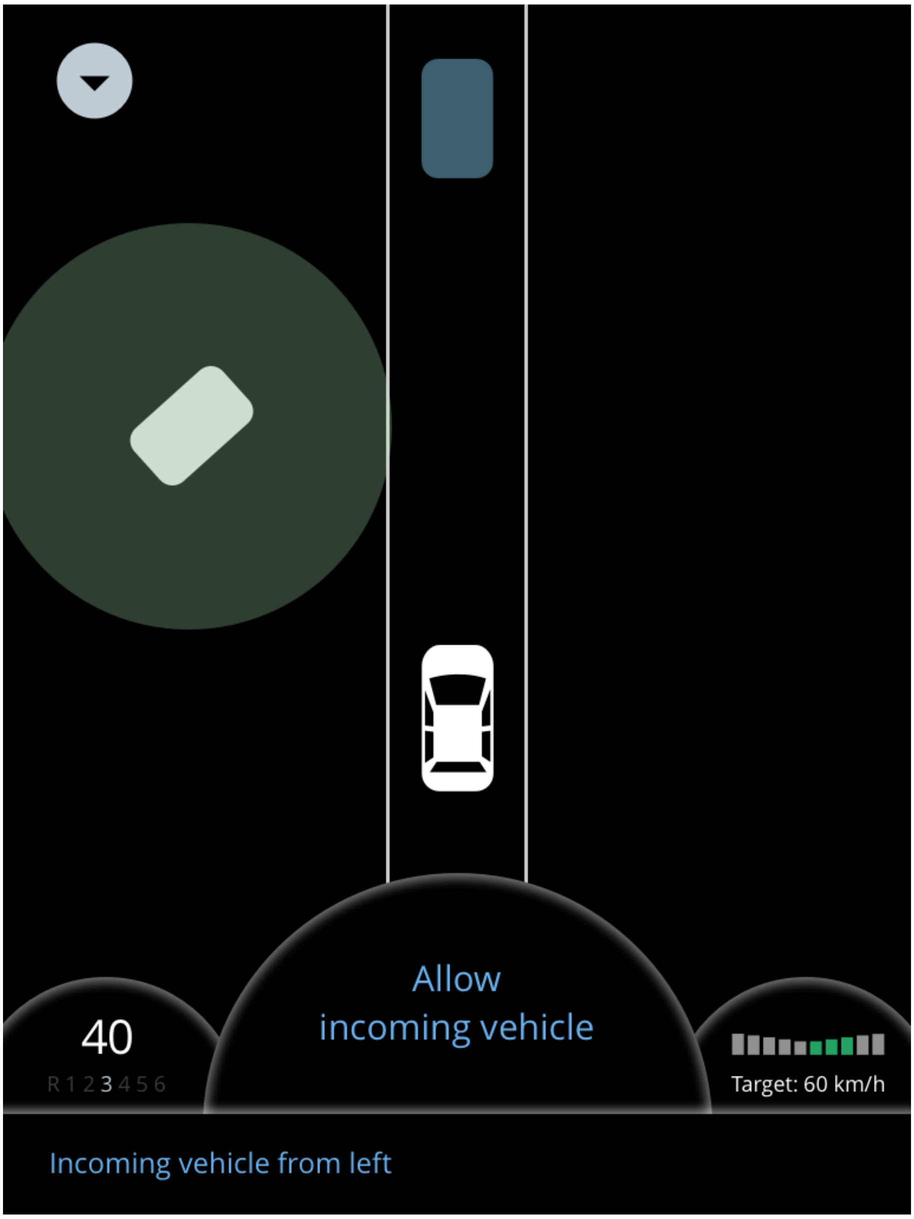


Figure 2.7: Second Prototype of Intersection Scenario

The second prototype for the intersection scenario is presented in figure 2.7. The information displayed was reevaluated and it was decided that the current action should be primary information, while speed and acceleration along with cause of vehicle behaviour should be secondary information. At this point, it turned out there would not be information about the upcoming action until just seconds before it was going to occur. Because of this, the progress bar showing how far until the action occurred was removed.

The intersection scenario in this mode shows a more zoomed out view of the road, as suggested in section 2.2.2.1. Also, the intersection is not displayed in the HMI, only the road of the own vehicle. This was due to the fact that it perhaps would not be possible to get information about the upcoming road to the HMI. The circle around the incoming vehicle is a safety distance, which the own vehicle should not enter.

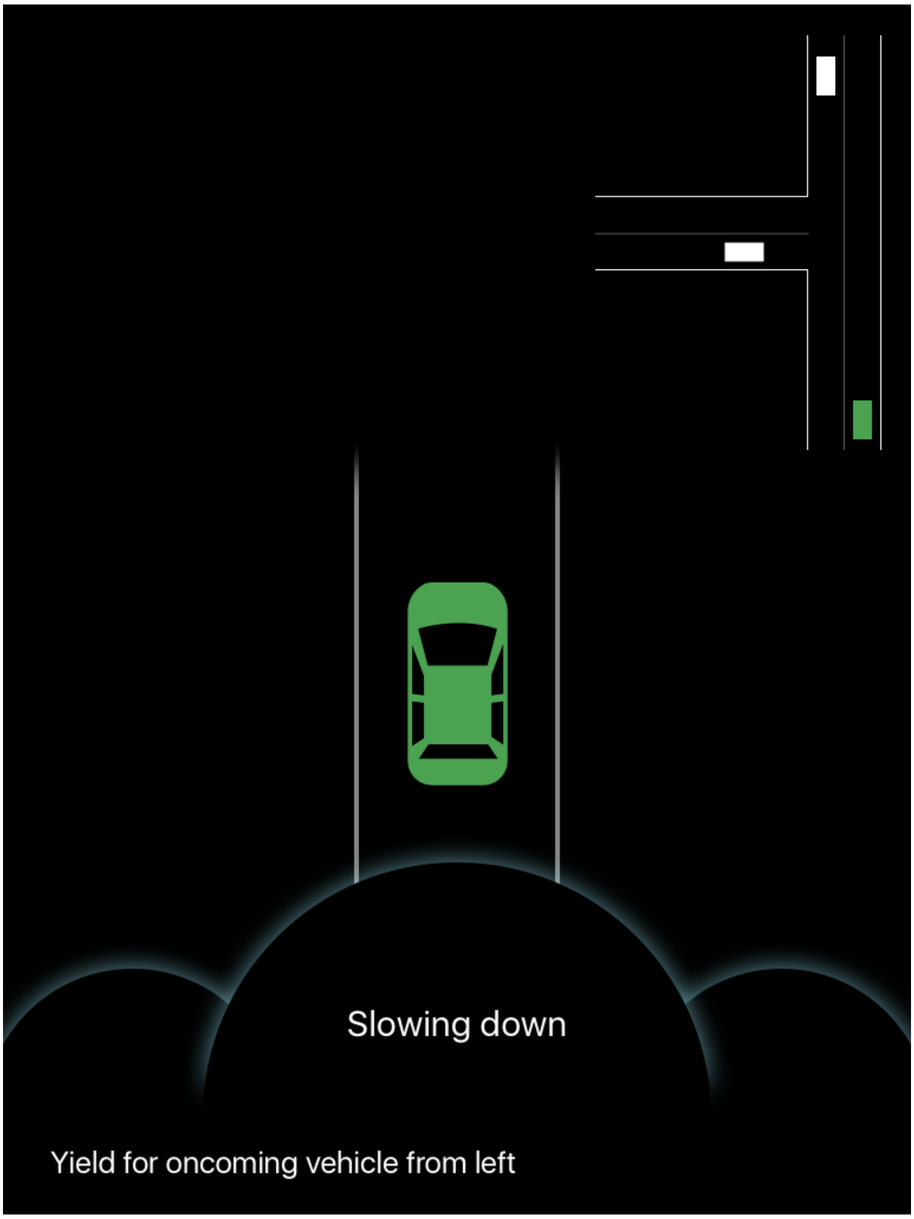


Figure 2.8: Third Prototype of Intersection Scenario

A third prototype of the intersection scenario is presented in figure 2.8. This prototype considers the problem of showing the upcoming intersection in a new way. Instead of zooming out to show the road ahead, which might be a problem when it comes to the implementation, it shows a minimap at the top right corner of the HMI. The minimap displays a more zoomed out view of the upcoming intersection, while the regular view stays the same. This might give the user a better overview of what the vehicle is planning to do. This prototype does not present speed or acceleration, but is mainly an idea for how to present information.

2.2.3.2 Prototyping by Implementation

In order to confirm that the elements of the proposed prototype could be implemented, the second stage in the prototyping process was to create an implemented prototype.

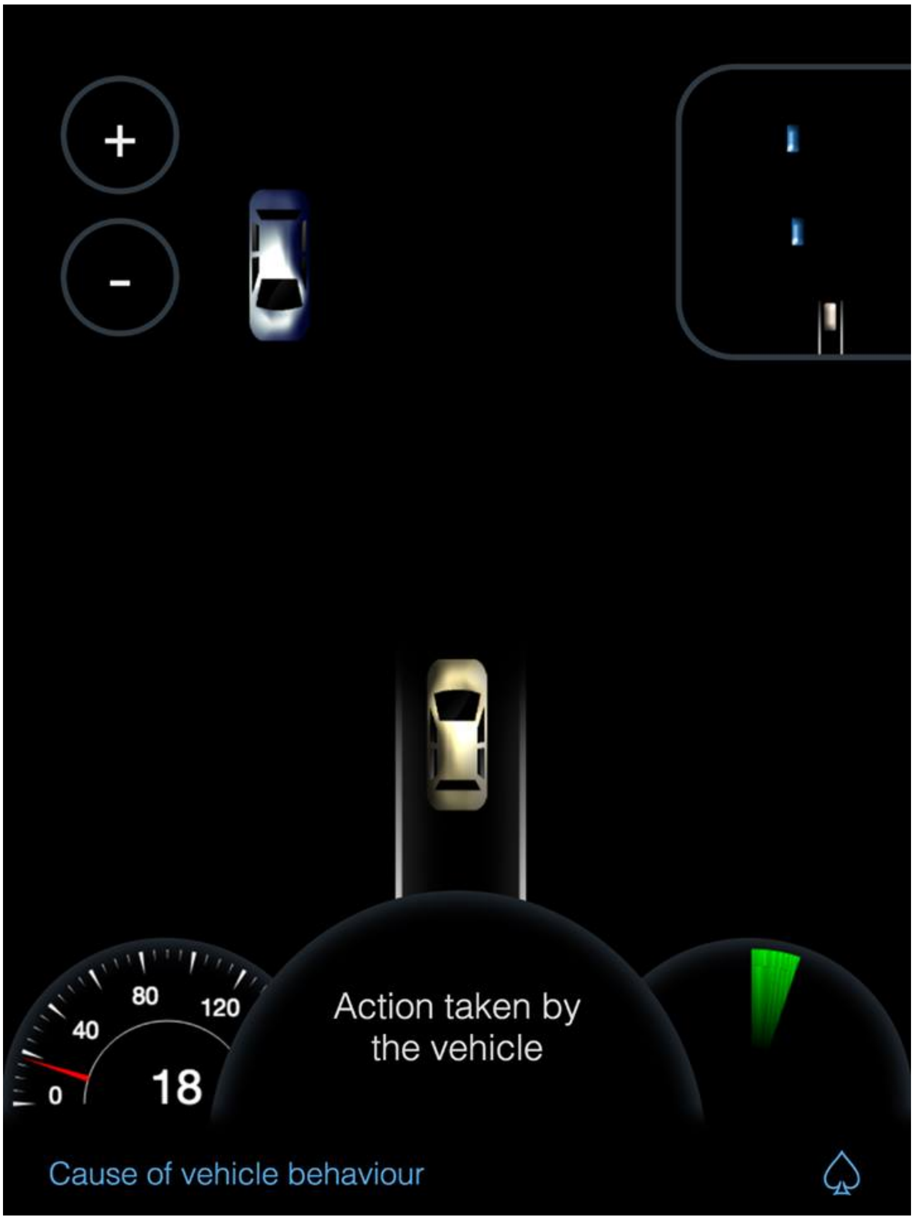


Figure 2.9: Prototype in implemented system

The prototype created for the implemented system is presented in figure 2.9. The prototype uses the same layout as in the earlier prototypes, but with a slightly different look. Shading and highlighting were added to the vehicle icons, to make them less flat and to get more depth in the picture. A speedometer was added in complement to the numeric representation, as suggested in section 2.2.2.1 The speedometer was design to look similar to the existing speedometer in the Volvo S60. For showing the acceleration of the vehicle, the previous bars were replaced with a circle showing how the vehicle is accelerating or decelerating. The circle fills up with green to the right for acceleration and as the acceleration increases, the angle grows larger. For deceleration, the circle fills up with red to the left.

In the prototype created for the implemented system, a different approach was made than for the sketched prototypes. Because of a limited simulation data as well as a limited number of signals from the vehicles, this prototype was created generically and meant to work for every scenario. With no information about road curvature, it was decided that the road in front of the vehicle should not be printed to the HMI. To make sure that the driver does understand where the surrounding vehicles are located and feels comfortable with the vehicle decisions, a minimap was added to the top right corner of the screen. The minimap shows a more zoomed out view of the surroundings, making it easier for the driver to get a quick overview of what the vehicle is seeing. In addition to this, zoom buttons were added to the top left of the screen, which the driver could use to zoom out and get more information about surroundings or zoom in to get a better look at what is going on close to the vehicle.

With this approach, messages would show telling the vehicle what action the vehicle is currently taking, and the cause of the action. However, the look of the HMI would not be different for the different scenarios. This makes the layout easy to learn and easy to remember, and with the messages it would hopefully still be clear to the driver what the vehicle is doing and why.

2.2.4 Evaluation

For evaluating the HMI, a few people were selected for a semi-structured interview. Because the intended user for the HMI mainly is the judges of GCDC, the people selected had all had their drivers licence for more than five years, and were driving on a daily basis. They were also people that were considered to have an above average knowledge of modern vehicles. The prototype used in this evaluation is the one presented in figure **??**

2.2.4.1 Evaluation Procedure

The interview was created with the help of a checklist for assessment of in-Vehicle information systems (Stevens and Cynk 2011). The checklist considered manually driven vehicles, and focused on a lot of things not applicable on this particular HMI, so only a small selection of relevant questions were used. Stated in the document was that the checklist may be used to assess individual functions, and may be applied to bench "mock-ups" or simulations. Because of this, the checklist was decided to be relevant in spite of only using a small selection.

Stevens and Cynk (ibid.) recommended that assessors familiarise themselves with both the user instructions and operation of the in-vehicle information system prior to undertaking an assessment. Implementing this recommendation in the interview procedure, the HMI was first presented to the participant and briefly explained. The reason for this was that the GCDC would allow for a card with HMI instructions, so it made sense that the participants would also get information about the different parts in the HMI. After examining the first view of the HMI, the participant got to see a simulation of the intersection scenario for as many times as they would like to.

The selected questions from the checklist had three possible responses; No concern, minor concerns or serious concerns. For making the decision easier, each question was followed by a short list of statements, were the answer could be true or false to each statement. An extra question in the same format as the questions from the checklist was added that focused on the surrounding vehicles, which is something that may not be presented in an HMI for a manual vehicle. In addition to these questions, a couple of open questions were created so that the participant could state reasons for concerns, an overall assessment, and also recommendations. Recommendations could be answered both by participant and interviewer. This was added to make the interview more qualitative, and give the participant an opportunity to give all their opinions on the HMI. A list of the questions used in the interview can be found in appendix A.1.

2.2.4.2 Evaluation Results Summary

The main concern most participants of the interview had with the HMI turned out to be the minimap at the top right corner. It was stated that it does not present any new information, as the user can access the exact same information by using the zoom buttons. For resolving this issue, one alternative is to completely remove the minimap, or zoom out even more. If the minimap were to be removed, an alternative to show that the own vehicle sees another vehicle that does not fit on the screen could be to use some kind of colour indication at the screen edge. As the vehicle got closer, colour could grow brighter, and eventually the other vehicle would appear on the screen.

Another comment received from the evaluation was that the HMI should show reasons as to why the vehicle is breaking. This was information that was not currently provided by the vehicle, and would therefore be hard to implement. Breaking could, however, be presented more clearly. For instance, breaking lights could be shown on the own vehicle in the HMI, and additional text could be presented. One participant wondered whether the HMI looked the same when the vehicle was reversing, which was something not yet considered. In reverse, the driver might want to see what is behind the vehicle, instead of what is in front of it.

Even though the evaluation focused more on finding issues with the HMI, some positive feedback were also provided. The interface was considered to be clean and with a nice overall style. One participant who had been driving Volvo for a long time stated that the interface design looked similar to the ones used in her vehicles, which is positive in terms of fulfilling the requirement of the interface looking well-integrated in the Volvo S60.

2.3 Final Design

The final design of the HMI presents a generic interface that looks the same for each scenario. Due to lack of time and simulations, focus has been set on fulfilling the most important requirements. The final design is presented in figure 2.10.

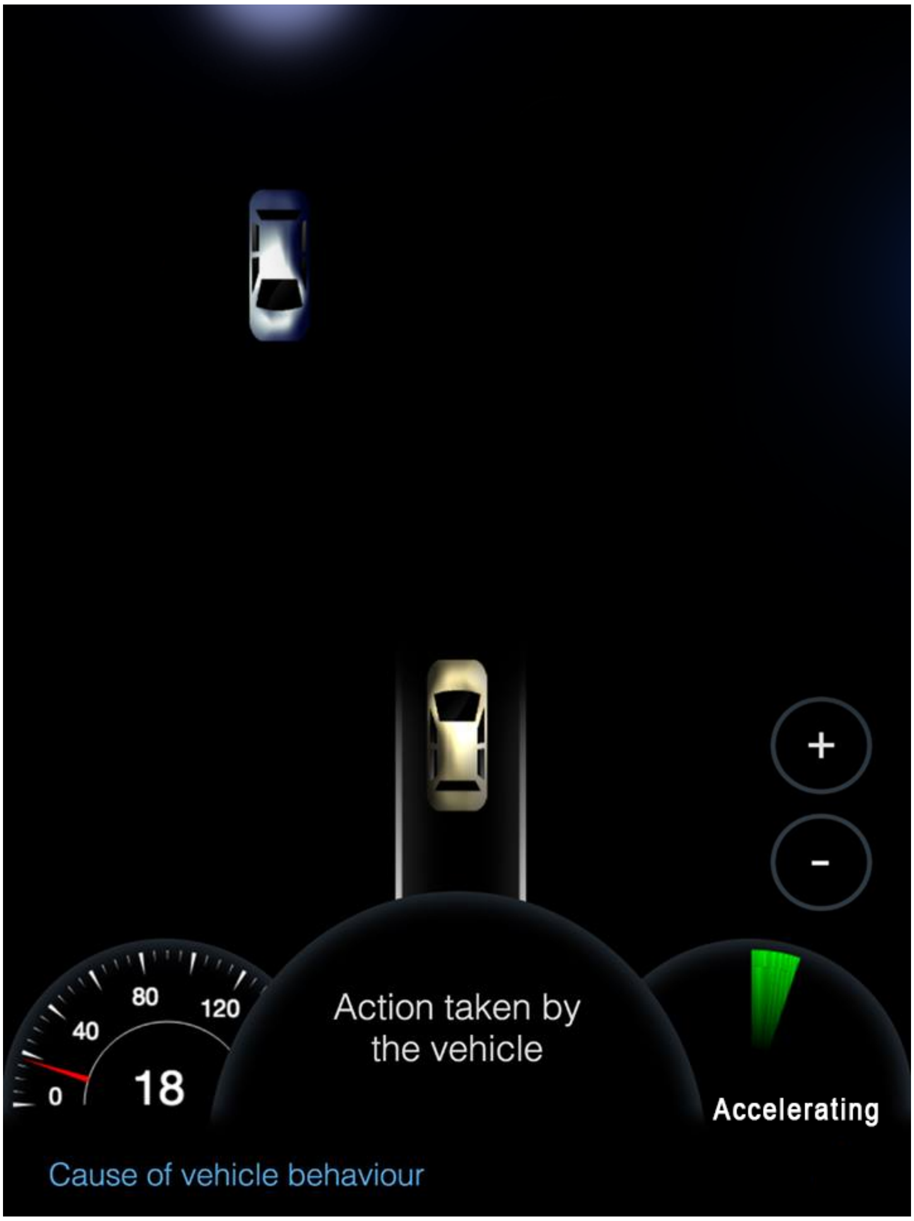


Figure 2.10: Final HMI Design

2.3.1 Speedometer

The speedometer is located in the left bottom corner, and resembles the one found in a Volvo S60. It has an analogue indicator in order to display fluctuating values, and to provide the driver with a familiar interface for the speed. To get a more precise reading, the current speed is also displayed as a digital number.

Requirements Fulfilled: Information Displayed - Speed of vehicle, User Experience - Easy to understand, well-integrated impression, driver recognizes HMI as vehicle HMI, visually attractive.

2.3.2 Accelerometer

An accelerometer is placed in the right bottom corner. If the vehicle is accelerating the accelerometer will display a green gradient to the right of the center. If the vehicle is slowing down there will be a red gradient to the left of the center. Accompanying the accelerometer is a text, telling the driver if the vehicle is accelerating, decelerating or actually breaking.

Requirements Fulfilled: Information Displayed - Acceleration/Deceleration of Vehicle, User Experience - Easy to learn, easy to remember, visually attractive.

2.3.3 Information: Cause

At the bottom, there is an information text field. This is where every cause behind the vehicle's actions are displayed. The field mainly consists of a text message, but if relevant, a symbol will be displayed along with the text message.

Requirements Fulfilled: Information Displayed - Current Behaviour, Actions to Be Taken, Planned Behaviour, User Experience - Communication well received, Trust in vehicle decisions, Easy to understand what vehicle wants to achieve, easy to learn.

2.3.4 Information: Action

In the bottom center, between the speedometer and the accelerometer, information about the current action of the vehicle is displayed. This will always tell the driver what the vehicle is doing at the moment. In combination with displaying the cause of this action at the bottom, the driver will be fully aware of what is going on. If relevant, a symbol is displayed above the text message.

Requirements Fulfilled: Information Displayed - Actions to Be Taken, Current Behaviour, Planned Behaviour, User Experience - Communication well received, Trust in vehicle decisions, easy to understand, clear what vehicle wants to achieve, easy to learn.

2.3.5 Own Vehicle

In the center of the HMI, the own vehicle is located. It has the same color as the actual vehicle to make it easily recognizable. The position of the vehicle is fixed in the center of the screen. All of the surroundings, both other vehicles and other lanes, move in relation to the own vehicle.

Requirements Fulfilled: User Experience - Communication well received, easy to understand, Easy to Remember, Easy to Learn, Visually Attractive

2.3.6 Zoom Buttons

At the bottom right of the screen, two zoom buttons are placed. An important part of the HMI is for the driver to be able to get a good overview and a larger perspective on what is happening. When pressing either button, the scale of the whole screen is changed. There is a limit to how much it can be altered in order for the HMI to always display relevant information. The zoom buttons are placed so the driver can still see the screen when pushing them.

Requirements Fulfilled: User Experience - Easy to understand functionality, Easy to learn, Easy to Remember.

2.3.7 Surrounding Vehicles

The surrounding vehicles are presented with the same icon as the own vehicle, but with a blue colour to make them easy distinguishable from the own. As mentioned before, the the surrounding vehicles move in relation to the own vehicle. When a vehicle is detected that does not fit on the screen, blue colour is painted on the screen in the direction of that vehicle. As the vehicle gets closer, the colour gets brighter, until the vehicle appears on the screen.

Requirements Fulfilled: Information Displayed - Surrounding Vehicles, User Experience - Communication Well Received, Trust in vehicle Decisions, Easy to understand at a quick glance, Easy to understand differences between vehicles.

2.3.8 Requirements conclusion

While some requirements are easy to see as fulfilled or not, others are harder to measure. The User Experience goals that are marked as fulfilled are based on the evaluation of the prototype. The User Experience goals in required attributes that are not yet mentioned are the experiences that aims for a certain feeling. These experiences will always be different depending on who the user is, and will because of this not be marked as fulfilled. Other requirements not yet mentioned are those that concern functionality. These requirements will be attended to in the implementation of the HMI.

3 Implementing the HMI

In the GCDC challenge, the HMI is ultimately to be thought of as an integrated component in the vehicle dashboard or centre console. Although this is the purpose of a fully functional HMI, a prototype might be constructed as an external unit. This creates the opportunity to choose a platform for the HMI that does not need to be limited to the current vehicle system.

In the case of creating an HMI for the Chalmers team vehicle, the main requirement is that the HMI should be able to receive and handle signals from the vehicle. There are several signals that are broadcasted from the vehicle, and these signals determine the information that can be presented by the HMI. The platform needs to be able to read and interpret signals from the vehicle and present these visually in real time. This requires an approach that is stable, fast, and able to communicate with minimum delay.

The HMI also needs to be visually attractive, and it is desired to show different images and symbols to enhance the user experience. All images must be drawn in real time, which requires them to be rendered often and quickly in order to avoid delay and misguided positions. Therefore, the HMI needs to be constructed on a platform that both supports good visual design and fast real time updates.

3.1 Web Application Development

One approach that meets the requirements listed above is building the HMI as a web application. A web application can run on any device that has a browser. Currently, this means almost all desktops, laptops, and mobile devices with displays.

3.1.1 JavaScript in Web Development

JavaScript is the main programming language of the web. It is built into all the major web browsers, and runs in the browser of whoever is visiting the web page (Klimushyn 2015). It is mainly a programming language for the client side of web development, but has come to be very popular on the server side as well (Rowinski 2015). JavaScript uses a browser API called the DOM to create interactions with the web application. The DOM is a tree-like representation of the entire web page including all elements and their attributes.

3.1.2 Server-Client Model

Web applications consist of two main components – the server and the client.

A server is a program or device that serves data to other programs or devices, called clients (Nations 2014). In practice, this could be a program on a computer that presents information in a browser on another computer, the browser being the client. In addition to providing clients with data, a server also executes calculations and is responsible for the communication with other systems.

There are a lot of software designed to implement web servers. A popular server software today is called Node.js. Node is designed to build applications that easily scale and also provides an environment to produce complex applications rapidly (Foundation 2016). With Node, the web server is written in JavaScript. With JavaScript being the main programming language for web applications (Rowinski 2015), Node is very popular among web developers.

In web development, a client is most of the time considered to be a browser, such as Google Chrome, Safari or Mozilla Firefox. The client only stores the state of the user's interactions, which is reset when the page reloads. Any data that is modified needs to be sent to the server in order to be stored if the page reloads.

3.1.3 Server-Client Communication

A common way to implement a web application is having the server and the client located at different geographical locations. They communicate through a connection established over the Internet, and use different transport layer protocols to transmit data.

3.1.3.1 Sending Packets over UDP

UDP, or User Datagram Protocol, is a transport layer protocol defined to provide a "best-effort" datagram service to an end system (Fairhurst 2008). The most well-known characteristic of UDP is that it does not establish a connection between the two ends, it only focuses on sending packets as fast as possible to the set destination. The lack of connection establishment before each packet is sent makes it possible for UDP to provide a very efficient way of sending large amount of data in minimal time. The main drawback when using UDP is that there is no guarantee that the data is actually received at the other end. However, during transfers where the number of packets are high enough, the transfer speed of UDP is more important than establishing connections. Therefore, some packet loss is acceptable.

3.1.3.2 Streaming Data Through a Web Socket

To transfer data from a server to a client in real time, a web socket is typically used. Web sockets establish a two-way connection and were originally introduced in web browsers, and are today widely used (Banks 2016). One of the most well-performing and well-used web socket frameworks is called Socket.io, which provides functionality for real-time data streaming (Socket.io 2016). Socket.io lets the server emit events along with data and the client can in turn subscribe to these events. The client is notified when new data is pushed from the server and receives the updated data in real time.

3.1.4 Presenting the Application

Presenting the web application to an end user is the responsibility of the client. There are several frameworks and libraries that can be used when implementing the client. One JavaScript library that lately has grown in popularity is React (StackOverflow 2016).

3.1.4.1 Real Time Rendering with React JS

React is a JavaScript library for client development (Facebook 2016). It uses a component based approach and divides the application into small and maintainable components. The components are structured like a tree, where larger components hold several smaller components.

One of the things that React does best is fast re-rendering of the web application. Normally, when data changes, the whole DOM-tree needs to be rebuilt. With React, only components that contain changed data are rebuilt. This is especially favorable for web applications dealing with real time data updates.

3.1.4.2 Drawing Graphics with Create JS

In HTML5, the canvas element is a tool to draw graphics in web applications. The canvas element is a container in which graphics can be drawn. The graphics, themselves, are drawn by using a scripting language, such as JavaScript. To simplify the process of drawing graphics using JavaScript, several JavaScript libraries have been created. One of these is called Create JS. Create JS supplies the client with an API that is intuitive and easy to use.

3.2 Implementing the HMI for GCDC

The design of the HMI for GCDC is based on the established requirements. For the implementation, the set requirements were that the HMI should be safe to use and have support for real-time updates. In order to meet these requirements, a web based approach was chosen.

The resulting implementation is presented in two stages. First, the hardware installation is presented, for both the server and the client, along with a list of signals given by the vehicle system. Secondly, the web application construction and data flow is described.

3.2.1 Server implementation

To serve the web application, the server software Node.js was used. Node is a server with many useful plugins and other additional software. The server contained different modules, optimized for each GCDC scenario. All scenario modules shared a parsing module, used when parsing data sent from the vehicle.

3.2.2 Client Implementation

The client side of the application was built with React and used Create JS to draw graphics in the browser. Choosing React is based on the requirement of having fast real time updates in the client. React also creates a maintainable application structure through its module like approach. Create JS is used because of its easy-to-use API towards the canvas element. It has great documentation as well as an intuitive syntax, and enables good visual design.

3.2.2.1 React Component Architecture

The React application has a simple component tree architecture, as seen in figure 3.1. *App* is the root component that holds the React application. *Top* is the component where the main content is displayed. *MainScreen* is the canvas component in which all vehicle graphics are drawn. The *Bottom* component holds the *Dashboard* and the *InfoBox*, which displays information about the cause of vehicle behavior. The *Dashboard* component contains the *Speedometer*, the *Accelerometer*, and the *InfoActions* component, which displays information about any actions taken by the vehicle.

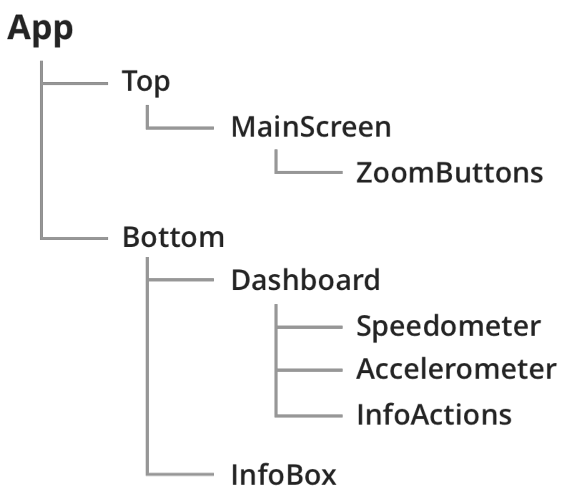


Figure 3.1: React Component Architecture

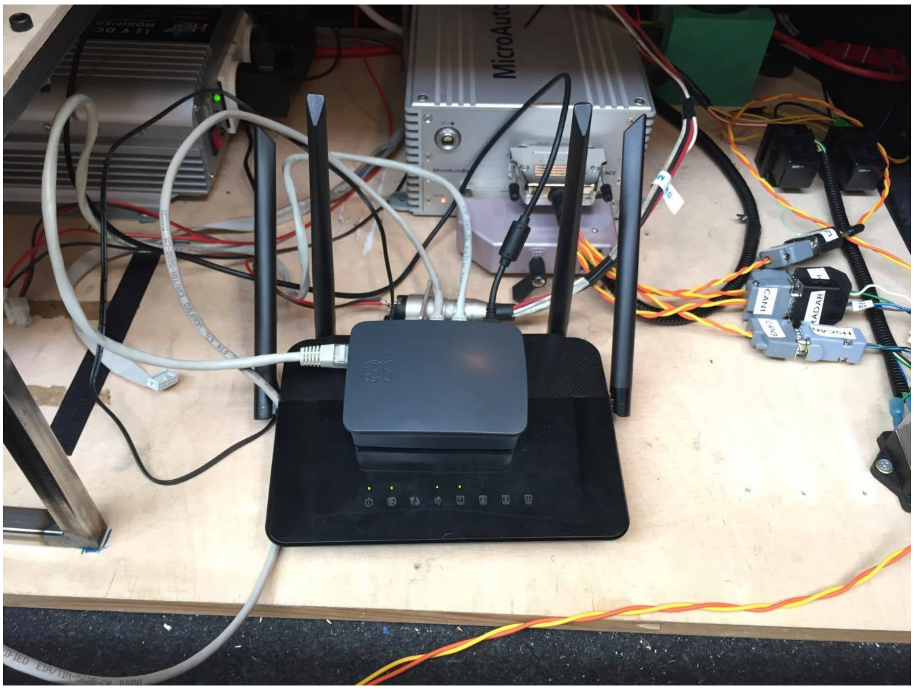


Figure 3.2: Raspberry and router mounted in vehicle

3.2.2.2 Graphics

React components that consist of a single canvas element, such as *Speedometer* and *Accelerometer*, use Create JS to render graphical elements. Canvas elements create a type of coordination system for accurate positioning and placement of vehicles and other graphical components. Within the canvas element, Create JS is used to draw icons, images, and geometrical shapes.

3.2.3 Vehicle Setup

A Raspberry Pi 3 was used to run Node.js and serve the web application. The Raspberry, being a small portable computer, was placed in the trunk together with the other parts of the vehicle system. It was connected to the internal network of the vehicle using a router as seen in figure 3.2. The router provided a Wi-Fi in the vehicle and assigned a static IP address to the Raspberry, enabling client-server connection via Wi-Fi.

An iPad Air 2 was used to present the HMI to the driver. It was mounted next to the steering wheel in a portrait position as seen in figure 3.3. The iPad connected to the server through the Wi-Fi network setup by the router. The web application ran in a browser on the iPad and was displayed in full screen mode.

Table 3.1: Signals available in own vehicle

Signal	Usage
ID	To identify the vehicle
Speed	To display the current speed of the vehicle
Acceleration	To indicate whether the car is accelerating or de-accelerating
Heading	To calculate direction
UTM Easting	To calculate the position of the vehicle
UTM Northing	To calculate the position of the vehicle
Width	To draw correct width of the vehicle
Length	To draw correct length of the vehicle
Distance to lane center	To know the vehicles position between the lanes
Road width	To draw the lanes correct in relation to the vehicle

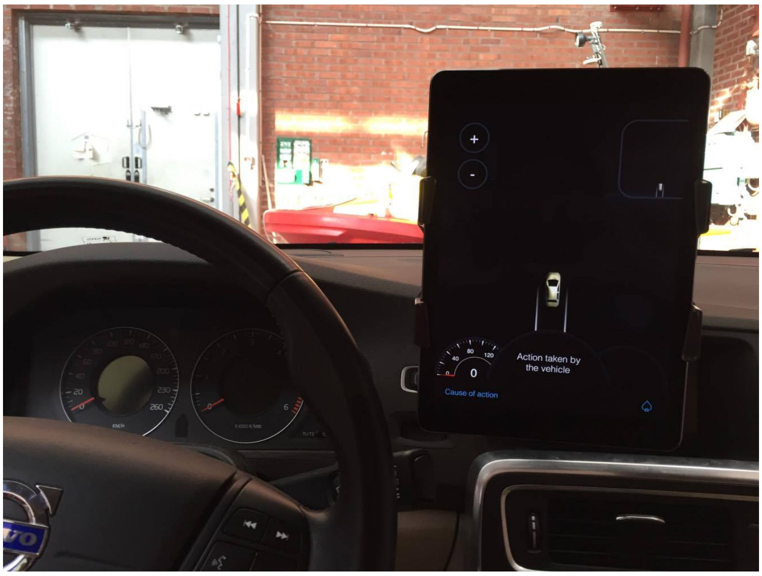


Figure 3.3: iPad setup in the vehicle

3.2.4 Vehicle Signal Composition

The vehicle provided an interface modelled in Simulink, where it was possible to send signals from the vehicle to the server. There were many signals available, but not all were relevant for the HMI. Signals used to present the own vehicle in the HMI are presented in table 3.1

For surrounding vehicles, a subset of these signals are used, these are presented in table 3.2

Table 3.2: Signals available in other vehicles

Signal	Usage
ID	To identify the vehicle
Speed	To display the current speed of the vehicle
Acceleration	To indicate whether the car is accelerating or de-accelerating
Heading	To calculate direction
UTM Easting	To calculate the position of the vehicle
UTM Northing	To calculate the position of the vehicle
Width	To draw correct width of the vehicle
Length	To draw correct length of the vehicle

Apart from the described vehicle signals, were are also a given number of flags available. The flags were used to indicate whether specific events should be displayed in the HMI, such as when the driver was braking, using the turn indicators and so on. They were also used to display appropriate messages when the car made certain decisions, for example when deciding to merge into a platoon.

3.2.5 Data Flow

The entire data flow is described in figure 3.4, where the main components in the system are modelled as boxes with arrows indicating the data flow.

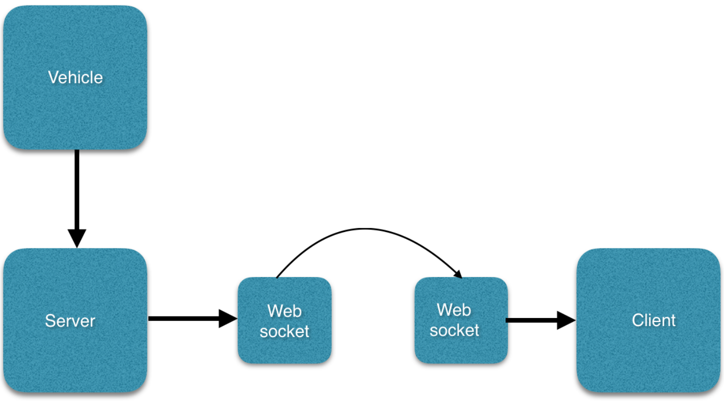


Figure 3.4: Data flow

Signals from the vehicle system are transmitted to the server over UDP at a set rate of 10Hz. All data is unpacked and parsed using constructed algorithms. When the data is ready to be sent to the client, the server emits the data over the web socket, making it available for the client.

The web socket connection is established on the client in the *App* component. Data is received from the server and passed down the component tree. Each child component receives selected data, relevant for its purpose, and either passes it on to other children, or uses it to render graphical elements. Each time new data is received, the graphical content in each affected component is re-rendered.

4
Discussion

Conducting this project has presented a couple of difficulties and challenges. Some problems have presented themselves and created minor or major decision points, while others have been factors to consider during the entire project process. In this section, we will try to address the problems and fundamental conditions that have affected the final results of this project the most.

4.1 The Grand Cooperative Driving Challenge

The aim of this project was to create an HMI for GCDC, and the competition has been in focus during the entire process. This has presented a couple of challenges, as the directives from the competition coordinators have been somewhat unclear. In the beginning of the project, we interpreted the task as creating a fully functional HMI for the vehicle. Because of this, a lot of time was spent setting up the server and getting simulations to work. When implementing the HMI, we were limited in the way that we could only present the information provided by the vehicle on the screen. This meant we were not able to address some important questions, such as how to present a transition between manual driving and autonomous driving. Later, it turned out that a functional HMI is only preferrable, and that the design concept might be more important, which has made us question the chosen approach.

Looking back, we can definitely say that we would have approached the project differently if we had had the knowledge of the competition that we have today. If we had known that the HMI concept was perhaps more important than the actual implementation in the vehicle, we would have focused on this to begin with. With this approach, we would not have had to be as dependant on what information was provided from the vehicle.

When setting the requirements for the HMI, we had three categories; required, desired and potential attributes. If we had focused more on only creating a design concept, we might had been able to address all of these attributes. However, with the chosen approach, there was no time for this.

4.2 Procedure

Following a set procedure in this project has been difficult. A large number of people have been part of the GCDC project, and we have at many times had to rely on these people for help. This has, along with the uncertainty of what the result was supposed to be, made the process of creating the HMI difficult to plan.

4.2.1 Implementation techniques

The project was started with the focus on the implementation of the HMI. Early in the process we decided on creating a web application. One alternative to creating a web application was to create a native application. Native is considered to have better performance than web which is relevant for an HMI that requires accurate real time updates. The main reason that web was chosen as the platform was our previous experience working with web. We had an idea of how the implementation would be structured, and with a limited time frame, this was enough to support our decision.
The criteria of a fast real time application set our eyes on React as framework for the client because of its rendering technology. A lot of time was spent on researching React and how the library could be applied to our application. Learning more about this made us discover the need for other frameworks and technologies as well, which extended the period of research extensively.
Naturally, React is not the only option when building user interfaces for web applications. There are many others, and given that some would not require as much research, the outcome of the implementation might have changed whether we had chosen a different framework. However, since there was limited time, it was more suitable for us to select one, and we decided on React given its popularity.

4.2.2 Designing the HMI

The procedure of the HMI design was started quite early in the project, before we knew exactly how the vehicle was "thinking". We had a lot of ideas early on about how to present surrounding vehicles and upcoming actions, which we based our first prototypes on. Later on, it turned out the vehicle was not "thinking" the way we thought it was, and some ideas we could not use at all. With the approach of creating a fully functional HMI, it would probably have been better if we had waited with the HMI design until it was clear what information we would be able to get from the vehicle.

4.3 Product

As mentioned in section 2.3.8, some requirements set for the HMI were harder to see as fulfilled or not than others. Also, mentioned in section 2.2.2.1, you can not design a user experience, only design *for* a user experience (Preece, Rogers, and Sharp 2012). Although we might think that the final HMI design provides the user experience aimed for, another person might feel differently. From the beginning, we had our mind set on presenting the vehicles from a bird's eye point of view, which might have narrowed our view when generating design alternatives. This makes it hard to tell whether our approach is better than any other approach, since we never evaluated anything else.

The problems with seeing if the user experience requirements are fulfilled are not presented themselves during the evaluation of the product. To conduct a proper evaluation of user experience, we would have needed a lot of people with different attributes and driving experiences. However, an evaluation with a lot of participants could also have made it harder to decide what design alternative to go with, or if a certain requirement had been fulfilled, as people tend to have very different opinions.

From the beginning, the HMI design was intended to be evaluated quite extensively. This did, however, present a few difficulties. We did not have the opportunity to evaluate the HMI in the actual vehicle. This meant that the evaluation we could perform would be having someone look at the HMI and pretend to be in the vehicle, which might lower the quality of the evaluation result. Since none of the participants of the evaluation had been in an autonomous vehicle before, it could be hard to imagine what it would feel like.

5
Conclusion

The purpose of this project was to create an HMI that met the demands of GCDC, and would be used in the competition. As the challenge has not yet taken place, it is not possible to tell whether this purpose has been fulfilled or not. There are, however, other conclusions to be drawn regarding the resulting product.

The resulting product is an HMI presented on an iPad, that is intended to be located at the centre console of a vehicle. The HMI provides information of perceptions and behaviour of the vehicle, and covers the three scenarios of GCDC. The HMI design is limited to the GCDC, meaning that only information relevant to the competition is presented on the screen. The design does, however, present an idea of a general HMI for autonomous vehicles. The HMI does not allow driver interaction with the system, as this could not be implemented without proper testing.

As it is not possible to tell whether the HMI will meet the demands of GCDC or not, conclusions regarding the final design are drawn by looking at the established requirements for the HMI. All measurable requirements, such as information displayed, are considered fulfilled. The requirements that are not measurable, such as user experience, can not be regarded as neither fulfilled nor not fulfilled. Because of this, the final design can be thought of as a prototype that needs further evaluation in order to assure all requirements are fulfilled.

A working implementation of the HMI, compatible with the vehicle, has been produced. The implementation supports fast real-time updates in test environment. With the goal of this project being a high score in the Grand Cooperative Driving Challenge, the development of the HMI will continue until the day of the competition. The design of the HMI will continue to be evaluated and improved. The final tests regarding real time and implementation of the HMI will be conducted at the site of the competition.

Bibliography

Banks, Paul (2016). *The State of Real-Time Web in 2016*. https://banksco.de/p/state-of-realtime-web-2016.html. [Online; accessed 11-January-2016].

Basani, Camilla (2014). *Results: How would you like to interface with your car?* http://www.openroboethics.org/results-how-would-you-like-to-interface-with-your-car/. [Online; accessed 6-March-2015].

Facebook (2016). *React – a JavaScript Library for Building User Interfaces*. https://facebook.github.io/react/. [Online; accessed 6-March-2016].

Fairhurst, Gorry (2008). *The User Datagram Protocol (UDP)*. http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/udp.html. [Online; accessed 4-March-2016].

Foundation, Node.js (2016). *About | Node.js*. https://nodejs.org/en/about/. [Online; accessed 16-April-2016].

iGAME (2016). *The i-GAME Project*. http://www.gcdc.net/en/i-game. [Online; accessed 2-February-2016].

Jong Kyu Choi and Yong Gu Ji (2015). *Investigating the Importance of Trust on Adopting an Autonomous Vehicle*. 31:10,692-702. International Journal of Human-Computer Interaction.

Keating, Lauren (2015). *The Driverless Car Debate: How Safe Are Autonomous Vehicles?* http://www.techtimes.com/articles/67253/20150728/driverless-cars-safe.htm. [Online; accessed 28-February-2016].

Klimushyn, Mel (2015). *Web Application Architecture from 10,000 Feet, Part 3 – Frontend Development*. https://spin.atomicobject.com/2015/04/20/front-end-dev-frameworks-libraries/. [Online; accessed 5-March-2016].

Nations, Daniel (2014). *Web Applications*. http://webtrends.about.com/od/webapplications/a/web_application.htm. [Online; accessed 5-March-2016].

Preece, J., Y. Rogers, and H. Sharp (2012). *Interaction Design, beyond human-computer interaction*. 7th. Wiley.

Rathenau Instituut (2015). *Car of the future: Autonomous, or cooperative*. http://www.sciencedaily.com/releases/2015/06/150622124832.htm. [Online; accessed 2-February-2016].

Rowinski, Dan (2015). *JavaScript Is Eating The World*. http://arc.applause.com/2015/11/06/javascript-is-eating-the-world/. [Online; accessed 4-March-2016].

Smith, T. et al. (2014). *Are We There Yet? Thoughts on In-Car HMI*.

Socket.io (2016). *Socket.io information*. http://socket.io/. [Online; accessed 5-March-2016].

StackOverflow (2016). *Developer Survey Results 2016*. `http://stackoverflow.com/research/developer-survey-2016`. [Online; accessed 17-May-2016].

Stevens, A. and S. Cynk (2011). *A safety checklist for the assessment of in-vehicle information sysems*. Crowthorne, UK: Transport Research Laboratory.

Stevens, A. et al. (2002). *Design Guidelines for Safety of In-Vehicle Information Systems*. Project Report PA3721/01. Crowthorne, UK: Transport Research Laboratory.

A
Appendix 1

A.1 Evaluation questions

Following are the list of questions used when evaluating the HMI.

GCDC HMI Design Evaluation

None - "No concerns about the design"
Minor - "Minor concers about the design"
Serious - "Serious concerns about the design"

*Obligatorisk

1. **Gender** *

   Markera endast en oval.

   - Male
   - Female

2. **Drivers licence** *

   Markera endast en oval.

   - < 1 year
   - 1-2 years
   - 3-5 years
   - > 5 years

3. **Backround**

   ....

   ....

   ....

   ....

   ....

4. **Is surrounding vehicles presented appropriately?**

   Markera alla som gäller.

   - [ ] It is clear what surrouding vehicles has been detected
   - [ ] It is clear where surrounding vehicles are poisitioned
   - [ ] Surrounding vehicles are distinguished from the ego vehicle
   - [ ] Information of surrounding vehicles are sufficient
   - [ ] The current zoom levels present a good overview of surrounding vehicles

5. **Concerns: Is surrounding vehicles presented appropriately?** *

   Markera endast en oval.

   - None
   - Minor
   - Serious

6. **Is colour used effectively to aid coding and layout of controls?**
Markera alla som gäller.

- [ ] Red/Green combinations are avoided
- [ ] Blue/Yellow combinantions are avoided
- [ ] The meaning of colour coding is clear

7. **Concerns: Is colour used effectively to aid coding and layout of controls?** *
Markera endast en oval.

- None
- Minor
- Serious

8. **Are colours used effectively in the design and presentation of visual images?**
Markera alla som gäller.

- [ ] Colours are limited to clearly differentiated sets
- [ ] Red/green and blue/yellow colour combinantions are avoided

9. **Concerns: Are colours used effectively in the design and presentation of visual images?** *
Markera endast en oval.

- None
- Minor
- Serious

10. **Is the layout of graphics/representational features appropriate?**
Markera alla som gäller.

- [ ] The choice of graphics/representational features is suitable for what they represent
- [ ] The design of graphics/representational features is not too detailed or complex
- [ ] Functionally related graphics/representational features have a consistent style
- [ ] The apparent size of the display images is appropriate to their function
- [ ] Graphics/representational features are not cluttered

11. **Concerns: Is the layout of graphics/representational features appropriate?** *
Markera endast en oval.

- None
- Minor
- Serious

12. **Is numerical data presented appropriately?**

Markera alla som gäller.

- [ ] An analogue format is used for fluctuating values
- [ ] An appropriate number of decimal places are used

13. **Concerns: Is numerical data presented appropriately?** *

Markera endast en oval.

- ( ) None
- ( ) Minor
- ( ) Serious

14. **Serious Concerns and Reasons**

....

....

....

....

....

15. **Minor Concerns and Reasons**

....

....

....

....

....

16. **Overall Assessment**

....

....

....

....

....

17. **Additional Comments**

....

....

....

....

....

18. **Recommendations**