HCI Guidelines to Reduce Cognitive Distraction Among Drivers

by

V MANOJ KUMAR
202011036

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

July, 2022

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

V Manoj Kumar

Certificate

This is to certify that the thesis work entitled **"HCI Guidelines to reduce Cognitive Distraction among drivers"** has been carried out by **V Manoj Kumar, ID-202011036** for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.

Dr. Saurabh Tiwari
Thesis Supervisor

Acknowledgments

I am extremely grateful to Dr. Saurabh Tiwari for his continued support and guidance. It has been a pleasure to work under him. He motivated me every time I got stuck in the research work. I would not have been able to finish my research work if not for his inputs.

I would like to specially mention Laxmi Soujanya, Guru Charan Teja and Nithin Teja Reddy for helping me find the volunteers for the experiment. Most of the volunteers for the experiment have been brought on board by them. I would also like to thank all the volunteers for taking time to perform the experiment. They all made it easier to conduct the experiment during peak COVID wave.

Last but not the least, I would like to thank my parents for supporting me throughout my research work. Thank you for giving me a strong moral support.

Contents

Abstract

Driver Distraction is a very common phenomenon observed in automobile drivers. This can be due to several factors ranging from mobile phone usage to talking to co-passengers, etc. In recent years, most cars are being fitted with screens on the dashboards. These dashboard screens serve various purposes. They are used for marketing purposes, infotainment access, car information, etc. Regardless of the purpose, these screens are becoming increasingly prominent in cars. Even though these screens offer functionalities, the fact is that they cause distraction among drivers. This is extremely fatal.

In this thesis, we focused on identifying guidelines for designing the interfaces in automobiles that will reduce cognitive distraction. We first conducted a literature review to find out the reasons for cognitive distraction. Cognitive Distraction is caused because of overload in cognitive abilities that distracts a person from performing a task. Cognitive Load is typically classified into three categories: 1) Intrinsic Cognitive Load 2) Extraneous Cognitive Load and 3) Germane Cognitive Load. Extraneous Cognitive Load is caused due to how information is presented to users. If the design of the interfaces is poor, then it causes Extraneous Cognitive Load to increase which in turn causes distraction. Intrinsic Cognitive Load and Germane Cognitive Load cannot be controlled. But efforts can be made to reduce the Extraneous Cognitive Load since it is in the hands of the designer how they want to present the information. Subsequently, a literature review is conducted in Human-Computer Interaction. HCI is the study of designing, implementing and evaluating interfaces that humans use to communicate with computers and vice versa. HCI guidelines help in designing an effective and user-friendly interface. But there are hardly any guidelines that apply to distracted driving scenarios. We went through different researches and identified a set of principles that apply to distracted driving scenarios. Finally, an experiment was conducted to check the effectiveness of the set of principles identified. 63 participants were surveyed in this experiment. We found that the principles we have identified would help designers in creating effective interfaces for drivers in automobiles.

List of Tables

List of Figures

CHAPTER 1
Introduction

1.1 Thesis Objectives and Problem Statement

The way people use interfaces has evolved throughout time. Experts were the only ones who utilised computers in the past. It's because operating these computers came with a learning curve. Computers have gradually become more accessible to the common public. New obstacles arose as a result of this. The demand for interfaces to make computer use easier arose. Efforts have been made over the years to ensure that the interfaces necessary to access computers become fewer and less complex. Keyboards were used on the first computers that were made available to the general public. The interface still required the users to learn and remember commands. The mouse was then developed. The interfaces' usability was considerably improved as a result of this.

Interfaces have evolved significantly throughout the years. The reason for this improvement is to make sure that the interfaces are simple to use and that learning and remembering commands is minimal, reducing the cognitive load on the users. The number of memory resources utilised for a given task is referred to as cognitive load. Only once the working memory has processed the information will it be stored in permanent memory. It's similar to a computer's RAM.

The number of instances in which interfaces are utilised grows as technology develops. Interfaces were created with the intention of being used in idle situations when the user's attention is focused on the interface. However, in recent years, interfaces have become practically ubiquitous. Automobile industry is one significant industry where interfaces are becoming increasingly popular. Automobiles have been equipped with interfaces for accessing car features, infotainment, and other tasks in recent years. They are being installed on the dashboard so that the driver will have easy access to them. The driver must concentrate on the road while driving. If a driver experiences cognitive overload while driving, it could be fatal. The driver must divert their focus to use the interface while driving. It is a source of distraction and can be quite harmful.

It is critical that the interfaces found in automobiles do not cause cognitive distraction. As a result, our research into HCI in distracting environments gains importance. Existing HCI principles no longer accommodate distracted situations, necessitating the development of new HCI concepts.

1.2 Thesis Contributions

Thesis contributions are as follows:

- **Literature review on Cognitive Science.** Research about cognitive load, Measurement of Cognitive load is done. This is important due to the fact that distraction is caused due to cognitive overload. Understanding of cognitive load will help in not only understanding the source of cognitive distraction but also helps in deriving a solution for the same.
- **Literature review on HCI Guidelines.** Research about HCI guidelines, evaluation techniques, etc is done. HCI guidelines help in improving the experience of user. This is done to help with the research problem which is to identify HCI guidelines to reduce cognitive distraction.
- **User's perspective.** We conducted an experiment to check the effectiveness of the guidelines identified. A total of 63 subjects have participated in this experiment.

1.3 Organisation of the Thesis

The thesis report is divided into the following chapters.

- Chapter 2 presents the Literature review of Cognitive Science and Cognitive Distraction.
- Chapter 3 presents the Literature review of Human Computer Interaction.
- Chapter 4 presents the identified interface guidelines, experimental setup and the survey questions.

- Chapter 5 presents the results of the survey and analysis of the results.
- Chapter 6 documents the conclusions and future research scope.

CHAPTER 2

Literature Review of Cognitive Science

In 1955, Harvard University's George A Miller published a study[13] claiming that the human brain has a finite amount of working memory. He said that young adults can process information up to 7 items in chunks. Later, it was revealed that the number of items depends on the type of chunk used. If this memory is fully utilised, new tasks will cause cognitive overload. When cognitive overload occurs, it is impossible to multitask. Cognitive Load refers to the amount of working memory resources used by the brain to do a particular task.

2.1 Cognitive Load Theory

In 1988, John Sweller developed Cognitive Load Theory (CLT)[24]. Over the next two decades, Sweller has published numerous research studies in CLT. Cognitive Load Theory, which is widely utilised for evaluating learning settings and interpreting empirical results, has been one of the most powerful and contentious frameworks in educational research during the last few decades. Any information should be processed by the working memory and then it is stored in long-term memory. One of the key assumptions of CLT is that working memory is restricted both in terms of capacity and time when it comes to retaining information. CLT's second premise is that long-term memory is basically limitless. The value of CLT is that it may prescribe instructional design based on these specific features of human cognitive architecture. Successful instruction should allow learners to manage working memory load and so learn well if they follow these prescriptions.

According to CLT, Cognitive Load can be divided into 3 categories [25].

- Intrinsic Cognitive Load
- Extraneous Cognitive Load
- Germane Cognitive Load

Intrinsic Cognitive Load:

The inherent complexity or difficulty of some tasks or materials is referred to as intrinsic cognitive load. Some tasks are more difficult to master than others. They can produce an internal cognitive strain if they are more demanding. Solving a complex calculus question, for example, is far more difficult than a simple mathematical equation like 4 + 4.

The nature and subject matter of the activity or problem that the learner finds difficult and hard is characterised as intrinsic cognitive load. Intrinsic load is determined by the quantity of items that interact with one another and are all processed at the same time, making the operation more difficult. When a person must attend to a large number of elements simultaneously, the intrinsic load will be larger than when there is low element interactivity. This may be seen in the example of graph interpretation, where a basic bar graph is easier to comprehend than a complicated histogram with far too many parameters.

The content's fundamental difficulty cannot be modified, but it can be regulated based on the learner's past understanding of the material, aptitude, and learning capacity. The inherent cognitive load may "only be altered by changing the nature of what is acquired or by the process of learning itself," according to John Sweller. For example, instead of a cursive, scribbly typeface, a more basic and legible one can be used to lighten the reader's load. The amount of working memory load imposed is determined by the number of elements that must be processed in working memory at the same time, and the number of elements that must be processed at the same time is determined by the degree of element interactivity. Anything that has been or needs to be learnt is referred to as an element. When noninteracting pieces can be learnt independently, intrinsic cognitive load is minimal because the task's intrinsic nature reduces working memory demand.

Low-element interactivity tasks allow for independent learning rather than simultaneous learning of elements. Without holding more than a few pieces in working memory at a time, the activities can be fully grasped and learned. Interactivity tasks, with a lot of elements, require the working memory to hold these elements in the memory at the same time. Several elements should be processed in working memory at the same time to understand and learn these tasks. For example, vocabulary or a language is easy to learn since learning every word is an independent task. But to frame a grammatically correct sentence, one needs to know where to place each word. So, different elements are processed simultaneously to get a tangible result. This is one of the reasons why we often see new language speakers say all the words in a sentence but in the wrong order.

Extraneous Cognitive Load:

The type of load caused by the way information is shared to a person is known as extraneous cognitive load. The cognitive load is caused by the person being exposed to unneeded or useless content. For example, let's consider that a person wants to explain what a square is to a student. The instructor can choose to list out all the properties of a square to the student like it is a parallelogram, all the sides are equal, all the angles are equal and 90°, etc. This is useless information that can be easily avoided. Instead one can simply draw the diagram of a square and show it to the learner. This will make it easy to understand and generate less load.

This load may be the consequence of ineffective teaching practises that confuse and complicate learning needlessly. If a graph and its related text are not connected effectively, for example, the individual will have to exert additional cognitive work to switch back and forth between the graph and the text, which will impede learning.

Background noise has the effect of causing extraneous burden. Consider how distracting it is to concentrate when you hear dogs barking, automobiles honking, loud music, or people conversing. These diverse noises are considered superfluous load since they obstruct the accomplishment of the work. To aid learning, it is not wholly necessary to eliminate superfluous cognitive burden. Moderate extraneous load has been shown to aid learners in several circumstances. Noise and concentrated concentration on learning interact in a complex way that has both positives and cons.

Because working memory may be significantly exceeded, a combination of high intrinsic and high external cognitive load may be harmful to learning. Because intrinsic cognitive load cannot be changed, instructional design that decreases extrinsic cognitive burden may be necessary.

Germane Cognitive Load:

Not all cognitive load is harmful. Germane cognitive load is the consequence of a constructive approach to information processing that promotes learning. As a result, germane load describes the effort that goes into creating a long-term knowledge or schema repository. This drastically speeds up the learning process. Creating flowcharts in presentations to clarify complex subjects is a good example. It is simpler to learn and recall knowledge when it is organised in a systematic manner.

The development of thoughts or behaviour patterns to organise categories of information is known as germane load. The more you practise using these behavioural schemas, the more natural the behaviour will become. This is demonstrated while imagining various circumstances, such as earthquake or fire drills in schools. In such instances, people must consider a variety of options before constructing an efficient solution and carrying it out in a systematic manner. The persistence of such training helps people learn how to respond and behave in real-life situations. This kind of cognitive load is encouraged.

Germane cognitive load could be increased by using mnemonics such as acrostics, rhyme schemes, and other techniques that make learning simpler. For example, in mathematics, the acronym BODMAS is used to remember the order of operations, which is as follows: brackets, orders (powers, square, and cube roots), division, multiplication, addition, and subtraction.

So, while driving, performing a task using an interface is a secondary task. The primary task would be to drive the automobile. The interface could be used for tasks ranging from changing the temperatures of the car seats to music features, etc. These tasks have a certain level of difficulty associated with them. So these tasks would cause Intrinsic Cognitive Load. Content will be displayed on the screen in the dashboard in the automobiles' interfaces. This content will cause Extraneous Cognitive Load. Now, the person will have to learn how to use the interface properly. They will have to go through a learning curve. This will cause the Germane Cognitive Load. After certain time, because of repetition, the interaction process with the interface will be stored in the permanent memory of the driver and will become easier to use the interface.

Therefore, the interfaces in the automobiles should be designed in such a way that the Extraneous Cognitive Load is reduced. We cannot do anything about the intrinsic and germane cognitive load in the driving situation. It depends on the task and the person's ability to learn respectively. But the way in which information is presented to the drivers lies in the hands of the developers. If the design of the interface is good, then Extraneous Cognitive Load can be reduced and in turn distraction will be reduced.

2.2 Cognitive Distraction during driving

Cognitive Distraction is a situation where a secondary task causes cognitive overload and causes distraction from the primary task. In a driving scenario, it could be anything ranging from talking to a person, using a mobile phone, etc. Driving a car is one of the most dangerous actions that most people do on a regular basis. Driver distraction has been around for a long time. Eating, talking to passengers, looking after children, smoking, etc. But with the advent of technology, a whole new set of distractions got added to the list.

Mobile phones, wireless devices etc are being used extensively by drivers in recent years. It is also possible that these new distractions are much more dangerous because these new tasks are more cognitively intensive and are performed for significantly bigger duration of time. Strayer et al., 2011 [23] published the most popular research regarding cognitive distraction during driving. It gives an insight into how dangerous cognitive distraction is and what are the causes of this distraction.

There are tasks that place none to little demand on cognitive resources. For example, a pre-programmed radio station playing through the car infotainment system usually puts little load on cognitive load since it is just listening to the radio station and not performing any task simultaneously. There are tasks that place medium load on cognitive resources. For example, talking to a passenger in the car. As it is talking, it requires processing the information the passenger is saying and then forming a response in the head and then speaking the sentences. Although the tasks are intensive, talking is something people normally do and due to the familiarity with the task, the cognitive resources used will not be strained extensively. According to the study, there are tasks that put a significant strain on cognitive resources. The authors opined that using screens while driving puts a bigger strain on cognitive resources. They gave an account of a case where a driver had been killed because he was trying to update information on Facebook using a mobile phone.

The authors gave an account as to the factors affecting driver's attention while driving. They say that there are two factors that significantly negatively affect the attention span of drivers. The first factor would be the duration of activity. The capacity of a driver to effectively estimate driving demands declines as the length of engagement with a device grows. Changing the radio station, for example, places demands on visual and manual resources, although the impairment is only temporary. A cell-phone discussion, on the other hand, can last many minutes, and the conditions that existed at the start of the call may alter significantly during that time. Dual-tasking activities that use resources for longer periods of time will, on average, cause more cumulative impairments than shorter-duration activities. The second factor is the exposure rate of a task. When the frequency of a task increases it increases the risk to public safety. For example, consider a task that requires the driver to do multiple smaller tasks to accomplish the goal. Then to achieve a goal, the driver has to frequently perform interaction which increases the risk.

The authors tried to find out the reasons for the increase in distraction while using mobile phones. The authors have stated that hand held devices and hands-free devices cause the same level of distraction. In some states in the USA, a ban was imposed on using these devices in 2009. But this ban did not increase public safety as compared to the states that did not have any kind of restrictions on handheld devices. Since, hands-free devices cause the same level of distraction, the authors stated that the source of distraction is cognitive in nature. The authors have suggested that using an interface causes inattention blindness where using the mobile phone diverts the attention of drivers from processing the information and restricts them from properly driving the automobile.

The authors also suggested that different people have different levels of multitasking. They suggested that there are supertaskers that can easily multitask without any distraction while driving. But during their experiments they found that this subset of people is very low. They also suggested that there are people who consider themselves as supertaskers. These kinds of people say that they don't feel distracted even if they do get distracted. The authors suggest that supertaskers are a rarity and the odds of being supertaskers is significantly low. The authors conclude that cognitive distraction is a phenomena observed in most people while only few percentage of people find multitasking while driving to be less distracting.

2.3 Cognitive Load Measurement

Measuring cognitive load garners importance because it helps researchers in estimating if cognitive overload occurs or not. Numerous methods have been proposed to measure cognitive load. But the effectiveness of these methods are subjected to a lot of debate among researchers. Some of the methods to measure cognitive load are as follows.

Self Reporting Measures:

In self reporting measure [17][3], the subject self reports how much mental effort they have put in to do the task. The most widely used tool for assessing cognitive load is Paas' rating scale [18]. Paas' assessment has only one item, and subjects are asked to respond using a 9-point Likert scale ranging from very minimal mental effort (1) to very high mental effort (9). Self reporting measures are very economical since you don't need any additional equipment. The self reporting measuring scale can be changed according to the needs of the experiment. One of the drawbacks of the self reporting measure is that it is subjective in nature. The way in which a question is framed affects the perceived meaning of the question and different subjects may answer differently. Another drawback is that, if the measurement is done retrospectively, then there is a chance that the subjects might have forgotten how much effort they put in for the task.

Psychological Measures:

As indications of cognitive load, a variety of physiological measures have been proposed. Heart rate [19], pupil dilation [26], hormone measurements [7], and other physiological indicators are often studied. For heart rate, it is observed that the heart beats much faster during cognitively demanding tasks. One can measure the heart beat during an activity and compare the results to much less demanding tasks. Coming to pupil dilation, researchers have found that the pupil dilation is higher during the cognitively intense tasks. In hormone measurement, it is observed that there is an occurrence of hormonal imbalance during intense tasks. It is difficult to determine what prompted physiological processes and, as a result, to interpret data. Furthermore, the measures are very much intrusive, inefficient, costly and the issue is that it is difficult to determine whether ICL, ECL, or GCL is being assessed.

Dual Task Measures:

The dual-task paradigm requires a learner to do two activities at the same time to assess cognitive load. When the primary job, i.e., the learning task, gets more demanding, it is believed that performance on the second task will deteriorate. There are two methods for conducting dual-task assessments. On the one hand, accuracy and response times can be measured in an observation task that must be completed while the learning task is being performed [4]. (2) On the other hand, a second work must be completed concurrently with learning [20]. This could be anything as simple as tapping your feet in a repeated pattern. Impairments in the secondary tasks could be used to measure increasing demand in the first task.

Dual-task load assessments have the benefit of being objective and mirroring the entire learning process, allowing you to collect extensive data. The most obvious downside is that such methods are intrusive; they disrupt the process of learning and impose load on their own. It's also a question of resources: A secondary task may not be as taxing on learners with strong working memory capacity as it is on learners with poor working memory capacity. This would constantly necessitate the regulation of learners' working memory requirements. Another problem is that the kind of load that is being measured cannot be identified.

Measuring Cognitvie Load in a differentiated way:

This measuring method [11] can be useful to differentiate between the different types of loads namely, intrinsic, extrinsic and germane cognitive load. The authors devised and evaluated two distinct methodologies for measuring cognitive load: (1) Informed rating: The authors taught subjects how to distinguish between different types of cognitive load so that they could score them correctly. (2) Naive rating: The authors created a questionnaire with two to three items for each category of cognitive load for this sort of rating. The identical learning circumstances have to be scored with this questionnaire. Although informed rating appears to be a potential technique for assessing many components of cognitive load, it does not appear to be cost-effective or feasible for bigger research, and standardised training would be required. The authors improved the naive rating and the enhanced version of the nave rating proved to be a valuable, practicable, and trustworthy tool.

Measuring cognitive load in driving domain:

Over the years, many methods have been proposed to evaluate cognitive load during driving. Self reporting and Detection Response Task are most commonly used in the driving domain. While self reporting can be termed as qualitative results, DRT could be classified as quantitative analysis. Self reporting is the most cost effective method. Since, it does not require any additional equipment and subjects can easily fill the questionnaire. Self reporting as mentioned previously, should not be applied blindly and should be modified accordingly. Efforts must be made to make sure the questions are not ambiguous. The wording of the questions should be clear and precise.

Detection Response Task (DRT):

The Detection-Response Task [9] is a way for determining how cognitive load affects attention in a driving situation. Every 3–5 seconds, drivers are given sensory stimulation and prompted to reply by pressing a button on their finger. Reaction times and success rates are used as indications of the cognitive load's attentional effect. The stimuli might be visual, tactile, or auditory, and they are selected based on the sort of in-vehicle system or equipment being tested.

Reaction times and success rates are used as metrics of the cognitive load's attentional effect. Response times are determined as the duration of time between when the stimuli is presented to the subject and the response from the subject is recorded and hit rates are computed as the ratio of correctly answered stimuli out of all presented stimuli. As a result, it's critical to pick the proper stimulus modality and placement so that they're always detected and either hidden by the environment or overlooked owing to driving-related duties. A DRT measurement is done separately and simultaneously with the task in order to determine the imposed load of a given secondary task. Higher cognitive distraction is indicated by larger variations in response times between the two situations and lower hit rate ratios. There are different ways to perform this method. For example, while driving, the drivers could be subjected to visual stimuli in the peripheral vision. A blinking light could be used periodically. The driver then presses a button on his finger in response to the visual stimulus. If the driver is experiencing higher cognitive load, then the response time will be higher and also the hit rate will be lower.

The method's major drawback is that it has an impact on the driver's performance and secondary task completion times. The response task itself has a certain level of cognitive load attached to it. Compared to self reporting, this method is costly because of the detection sensors and stimuli equipment. Nonetheless, this is a simple to use and deploy strategy that provides for relevant in-vehicle system assessment and evaluation. Researchers can obtain valid and meaningful results on the attentional effects of cognitive load on drivers by following the recommendations and taking into consideration the constraints.

Oculomotor metrics:

A study [21] has found a correlation between eye movements and cognitive load of a driver. Because secondary visual and cognitive task demands have been demonstrated to influence driving performance in different ways, it's critical to isolate each to research their effects on driving performance. The allocation of attention is inextricably tied to eye movements. As a result, if high cognitive load disrupts attention networks, we can predict alterations in eye movement behaviour. Increases in blink frequencies, higher saccade peak velocities, and a considerable reduction in fixation spread along the horizontal axis are all indicators of increased cognitive task stress. This research is a first step in using eye movement patterns as a testing tool for detecting cognitive distraction in drivers.

CHAPTER 3

Literature Review of Human Computer Interaction

The study of developing, implementing, and evaluating interactive interfaces used by humans and computers to interact with each other to achieve a job is known as HCI (Human-Computer Interaction). HCI is made up of three elements: a human, a computer, and their interaction. An interface is a way for two unconnected entities to communicate.

A Human is a single person or a group of users who collaborate to complete a task.
Any computing equipment or technology that performs a task is referred to as a computer. It receives information through the interface, processes it, and then outputs.
When humans communicate with a computer through an interface, this is referred to as an interaction.An interface is a way for two independent things to communicate. A computer interface is a hardware device or system that serves as a communication channel between a computer and another entity. A mobile phone screen, for example, is a computer interface between a human and the phone.

Significance of HCI

The most widespread misperception about HCI is that it exists solely to improve the aesthetics of an interface. This is not correct. HCI is concerned with creating interfaces that:
Boost your output
Improving the user experience
Reduce the chance of safety-critical systems failing.

Productivity: The amount of work accomplished in a given time period is referred to as productivity. A nice interface can enhance productivity in the same way that a pleasant working environment can. For example, consider a proper calculator A and a calculator B with the buttons arranged randomly. One can finish a task with calculator A faster because the buttons are in orderly fashion. Due to bad design in calculator B, the amount of time taken to finish a task increases. Hence, productivity decreases.

Improving the User Experience: The user experience refers to how a person feels while using a computer. Good design contributes to a positive user experience, and vice versa. Consider what would happen if someone mistakenly erased a document they had been working on instead of saving it. It would be incredibly aggravating. It may have happened because the two tabs were close to each other or because they didn't know what each one did. A terrible design might be difficult to work with for a variety of reasons.

Reduce the chance of safety-critical systems failing: Bad design can be more than annoyance and frustration. It can also be dangerous and even fatal at times. The dangers connected with safety-critical systems are just too great, and the repercussions of poor interfaces can be disastrous. Here are some instances of systems in which the interface is important:

- Aviation control systems
- Industrial control systems
- Automated medical equipment

For example, on December 20, 1995, a normal aircraft from Miami, Florida, to Cali, Colombia, crashed into a mountain in Buga, Colombia. The catastrophe claimed the lives of 159 of the 163 individuals onboard, including all passengers and crew members. The pilot was found to have pressed the incorrect navigational fix, according to the probe. The R button on the pilot's control panel was coded to make a left turn, but the R button on the computer management system database was designed to go straight. The plane was meant to fly straight when the pilot pressed R, but instead it took a left turn and crashed into the mountains.

3.1 Modes of Interaction

3.1.1 Norman's Mode of Interaction

The most influential model is Norman's interaction model [16], also known as the execution-evaluation cycle. According to this approach, the user sets the goal first, then uses the system interface to carry out actions to achieve it. The system displays the output on the interface after the actions have been completed. The user examines the interface and determines whether the result matches the aim. The cycle ends when the goal is met. Otherwise, a new objective is set, and the cycle begins again.

Norman's model has seven steps The above-mentioned interaction process can be broken down into seven stages.

- Determine your objective.
- Create your intention.
- Determine the action sequence.
- Perform the action.
- Perceive the current state of the system.
- Interpret the current state of the system.
- Evaluate the current state of the system.

For example, consider the goal is to write a paragraph with font size 14. Let's imagine the paragraph has been typed, but the user now wishes to enlarge a portion of it. Let's split this task down into Norman's seven stages.

Determine your objective: Text font size should be increased.
Create your intention: From the toolbar's font choice, choose a font size that is a few sizes larger than the existing font size.
Determine the action sequence: To enlarge the text, select it. In the font choice on the toolbar, select the current font size. From the dropdown menu, choose a font size.
Perform the action: Complete all of the activities listed in the previous step.
Perceive the current state of the system: Perceive (or, in this example, examine)

the chosen text.
Interpret the current state of the system: Examine the content and check for any noticeable differences in font size from the prior edition.
Evaluate the current state of the system: Compare the current text size to the desired text size. The cycle ends when the current size meets the user goal. If the font is significantly larger than required, a new target is set to reduce the font size. If the font size is still insufficient, a new target to increase font size is set.

Norman utilised this model to demonstrate why some user interfaces can be problematic. He summed up the reasons as follows::

Gulf of Execution: The discrepancy between the activities stated by the user and those permitted by the system is known as the Gulf of Execution. A more effective interface has a narrower gulf of execution. An effective interface, more particularly, is one that allows a user to complete an operation without being limited by the system's limits.
Gulf of Evaluation: The discrepancy between the display of system output and user expectations is known as the Gulf of Evaluation. A reduced gulf of evaluation refers to an effective interface that allows the user to quickly assess the product depending on their objectives.

Norman described how problems might originate on the system's side of an interaction, but errors can also occur on the user's side. Human error can be divided into two categories: slip and mistake.
Slip: When a user understands the system and aim, and formulates appropriate actions, but fails to carry them out properly. This can include things like typos and hitting the wrong button. This can be avoided by enhancing the user interface.
Mistake: When a user does not comprehend the system, interface, or aim, they make a mistake. This can include an incorrect interpretation of icons. A greater understanding of the system can help avoid this.

Limitations of Norman's model:

Norman's approach is simple and intuitive, allowing us to grasp the concept of interaction. It does, however, have some drawbacks, such as:

- It just sees the system as a user interface. Interaction problems can also emerge on the system's side. As a result, understanding how the system works is crucial.

- focuses solely on the user's perspective of interaction, ignoring how the system interprets, processes, and translates the user's actions, as well as how the system updates its state.
- makes no mention of how the system utilises the interface to display its current status as a response to the user's action.

3.1.2 Abowd and Baele's Interaction Framework

The interaction framework [8][1] places a greater emphasis on the system, resulting in a more realistic description of the interaction. The drawbacks of Norman's approach are addressed in this model. The interaction is divided into four parts in this model:
User
System
Input
Output

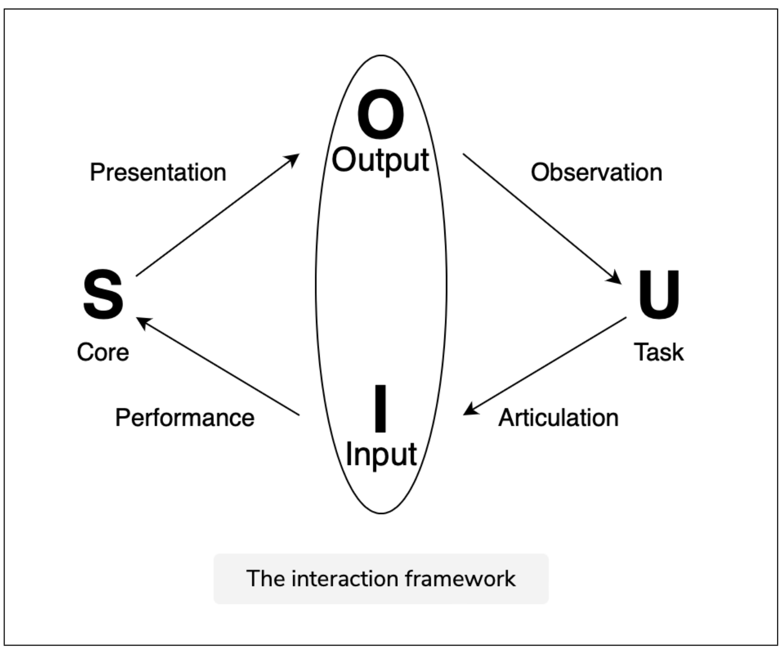

Figure 3.1: Abowd and Baele's Interaction Framework

- Input and output components have their own language in addition to the user's task language and the system's core language. The input component's language refers to the methods and activities that the system allows for data input via the system interface. The output component's language describes how the system outputs and shows information on the system interface.

- An interface is formed by both input and output.
- Language translation between components allows components to interact.

Because an interaction has four components, it requires four translations, one from each element to the next, to complete one cycle. Articulation, presentation, performance and observation.

Articulation Translation: The translation of the user's task language into the interface's input language is known as articulation translation. The user must first define the goal and the procedures required for achieving it. The user then performs the activities and converts them into input component stimuli. When a user formulates or performs incorrect actions, this is known as an articulation error.

For example, if a user copies a text using shortcuts, good articulation would be that the user correctly uses the shortcuts and bad articulation would be that the user incorrectly uses the shortcuts.

Performance Translation: The translation of the input language into the system's core language is known as performance translation. The input reaction is translated into stimuli for system execution. The system detects actions, executes them, and updates the state of the system. When the system either does not allow the user to accomplish the planned activity or fails to understand the intended actions, a performance error occurs.

For example, if a user types a name into the word processor, then The system interpreting that it's a name and does not show a spelling error is a good performance translation. If the system shows error, then it is poor performance translation.

Presentation Translation: The translation of the system's core language to the output language is known as presentation translation. The state of the system is updated after the actions have been completed. This new state is translated into stimuli for the interface to represent the output component. The system's interface displays the current state. When the system fails to update the interface in accordance with its current state, or when the revised interface delivers insufficient or unclear information, a presentation error occurs.

For example, A user chooses a mobile application to install and begins the process. After a certain amount of time, an error notice displays and the installation is terminated. Good presentation would be “Unable to install the application. Phone memory is full. Please empty some space and try again later”. Bad presentation would be “Could not install the application”.

Observation Translation: The translation of the output language into the user’s task language is known as observation translation. The user translates this reaction into stimuli for personal understanding to assess the result after the output component has presented the system’s updated state.

For example, when the user closes the text editor without storing their work, the system prompts them for confirmation in the form of a dialogue box with two options: "save and close" and "close." Good observation would be the user choosing “save and close” option and bad observation would be the user choosing “close” option.

3.2 HCI Guidelines

3.2.1 Norman’s seven principles

Based on the interaction principles, Don Norman identified 7 fundamental principles of design [16]. They are

Visibility: Reduce the gulfs of execution and evaluation by making things visible to the user. The more visible an object or option is, the easier it is for a user to learn about it.

Feedback: The purpose of giving feedback is to let the user know what actions they’ve taken and what actions they’ve accomplished. Any form of perception, such as visual, aural, or tactile, can be used to provide feedback. Auditory icons and earcons, for example, make a sound to signal that an action has been completed successfully.

Constraints: The activities that can be performed on the present system state are limited by constraints. They constrain the user to just doing the right things in the right order. A user interface with an infinite number of available actions can be confusing. As a result, they are more likely to make mistakes. For example, you cannot specify a departure/arrival date prior to the current date on Emirates' online booking system.

Mapping: The mapping of a control's layout to the device to be managed is called mapping. Mapping should appear as natural as possible, with activities that represent our daily duties. A slider feature in an interface, for example, has strong mapping. Moving it to the left lowers the value, while moving it to the right raises it.

Signifiers: The usage of signifiers effectively promotes discoverability as well as clear and understandable feedback.

Affordance: The design of items in an interface should reflect their fundamental features, making it easy for the user to recognise and understand how to utilise them. This Apple logo, for example, does not have a strong affordance. A beginner user will not be able to tell that this is a menu icon that can be clicked.

Conceptual model: The design projects all of the information required to construct a good conceptual model of the system, resulting in comprehension and a sense of control. The conceptual model improves both discoverability and results evaluation.

3.2.2 Schneiderman's Eight Golden Rules

According to Ben Schneiderman [22], there are eight "golden guidelines" that can be applied to most interactive systems. These principles, developed over two decades of experience and refinement, require validation and adjustment for specific design domains. Although no list can be complete, students and designers have found it to be a useful resource.

Strive for consistency: This principle is the most often broken, although adhering to it can be difficult due to the various types of consistency. In similar scenarios, comparable action sequences should be required; identical wording should be used in alerts, menus, and help screens; and similar colour, structure, fonts, and other design elements should be used throughout.

Ensure universal accessibility: Recognise the demands of many users and develop for flexibility, allowing for content transformation. The spectrum of design requirements is broadened by novice-expert disparities, age ranges, impairments, and technological diversity. Adding features for amateurs, such as descriptions, and features for experts, like shortcut keys and faster pacing, can improve perceived system quality by enriching the interface design.

Provide useful feedback: There should be system feedback for every user activity. The response can be small for regular and minor acts, but it should be more significant for uncommon and major ones.

Create dialogues that result in closure: Action sequences should be divided into groups with a start, middle, and end. Useful feedback at the end of a set of actions gives operators a sense of accomplishment, relief, the signal to forget about contingency plans, and the sign to get ready for the next set of actions. E-commerce websites, for example, guide customers through product selection to checkout, concluding with a clear acknowledgement page that completes the purchase.

Prevent errors: Design the system as much as possible so that users cannot make severe mistakes. If a user makes a mistake, the interface should identify it and provide straightforward, constructive, and precise recovery instructions. Erroneous activities should either leave the system state unaltered or provide guidance on how to restore it.

Easy reversal of action: Actions ought to be reversible as much as feasible. This feature reduces anxiety because the user knows that mistakes may be reversed, encouraging them to try new things.

Encourage a strong internal locus of control: Experienced operators crave the feeling that they are in control of the interface and that it responds to their actions. Unexpected interface actions, laborious data entry sequences, inability to access or difficulty in collecting relevant information, and inability to perform the intended action all contribute to anxiety and discontent.

Reduce short-term memory load: Because of the limitations of human short-term memory, displays must be kept simple, multiple-page displays must be consolidated, window-motion frequency must be minimised, and appropriate training time must be allowed for codes, mnemonics, and action sequences.

3.2.3 Principles to support Usability

In 1998, Dix et al., [6], presented 14 principles which were broadly classified into 3 categories. They are

- Learnability
- Flexibility
- Robustness

Learnability

The convenience with which beginners can start interacting effectively and achieve peak performance. Principles affecting Learnability are:

Predictability: Support for determining the impact of future actions based on previous interactions.
Synthesizability: Support for assessing the impact of previous operations on the present state.
Familiarity: When dealing with a new system, the extent to which a user's knowledge and expertise from other real-world or computer-based domains can be used.
Generalizability: Support for the user to apply their understanding of a certain interaction within and across programmes to other scenarios that are comparable.
Consistency: Similarity in input–output behaviour as a result of similar contexts or task objectives.

Flexibility

The several methods in which the user and the system interact with each other. Principles affecting Flexibility are:

Dialog initiative: Allowing the user to be free of the system's artificial limits on the input dialogue.
Multi-threading: The system's ability to allow simultaneous user involvement with several tasks.
Task migratability: The ability to delegate authority over the execution of a task so that it can be internalised by the user or shared between them.
Substitutivity: Allowing equivalent input and output values to be substituted for each other at will.
Customizability: The user's or the system's capacity to modify the user interface.

Robustness

the level of assistance provided to the user that support in goal achievement and evaluation. Principles affecting robustness are:

Observability: The user's ability to assess the system's internal state based on its perceivable representation.
Recoverability: Ability of the user to remedy an error once it has been identified.
Responsiveness: The user's perception of the system's communication rate.
Task conformance: The extent to which the system services enable all of the tasks that the user wishes to complete and in the manner that the user understands.

3.3 HCI evaluation Techniques

The goals of HCI Evaluation are to-

- Determine the user's experience of the system interaction.
- Determine potential errors or problems in the system.

There are two main types of evaluation based on the participants/users involved in the evaluation. The first type is one that involves experts as participants. In the other type, participants from the actual user groups are selected.

3.3.1 Evaluation Based on Expert Analysis

The earlier mistakes are identified, the less expensive they are to correct during the design process. However, user testing can be costly. Getting the design evaluated by professionals or system designers early in the process is a preferable method. The widely used evaluation techniques used are Heuristic-based evaluations, Model-based evaluations.

Heuristic-based evaluation: Heuristic-based evaluations are one type of evaluation performed by experts. They use rules of thumb or heuristic guidelines to evaluate a design.

Nielsen's ten heuristics [15]
Visible system state
Mapping between system actions and the real world
Offer user control and freedom
Consistency and standards
Prevent errors
Recognition rather than recall
Flexible and efficient to use
Minimal design
Provide help to recognize and recover from an error
Provide help through documentation

Define heuristics, depending on system specifications, Nielsen's heuristics could be used directly, could be modified, or a completely different set of heuristics could be developed.

Model-Based Evaluation: This approach uses cognitive and design models. These models help to identify project specifications along with design evaluation.

The GOMS Model [10]
Goals: This describes user goals, i.e. what the user wants to achieve.
Operations: The set of basic actions that users can perform to interact with the system and achieve a goal.
Methods: These are the different ways of achieving the same goal. Method X will have a different set of subgoals from method Y. Both methods can be used to achieve the goal.
Selection: These are the conditions and situations that predict which method will be used.

The Keystroke-level model [5]
The Keystroke-level model, abbreviated as KLM, is a model that predicts user performance on the basis of a deep understanding of the human motor system. It focuses on the time taken by the user to perform basic unit tasks of simple functions. For example, it can be used to evaluate individual unit tasks performed for functions like changing font size, searching and replacing. Complex functions such as making a graph or a diagram will be divided into subtasks with subgoals just like in the GOMS model and then these subtasks can be evaluated using KLM. KLM assumes that the user has decided and established how to perform a task in the mind before initiating actions.

3.3.2 Evaluation Based on User Participation

Experimental Evaluation
The interface is put to the test in an experiment. End users are among the participants. In a controlled experimental context, they are allowed to engage with the interface. This evaluation is novice since it is conducted in an experimental setting that would be considerably different in the real world.

Observational Evaluation
Observational techniques are evaluations based on user analysis. The user is obliged to either accomplish a planned set of behaviours or simply engage with the system in a casual manner, exploring whatever they like. The evaluator watches and records the behaviours of the users.

Think-aloud evaluation [12] - This is an observational technique in which the user is requested to think aloud, that is, to talk about their activities as they occur. "Now I'm going to click the add to cart button," or "I'm returning to the previous page," for example. The evaluator observes and records the user's behaviour and actions. The most significant benefit of this method is its simplicity. Also, whether it's a paper-based prototype or a fully functional application, this evaluation may be done at any stage of the design process. The review provides insight into how users will really use the technology.

Post-task walkthroughs [14] - The information gathered from observing the user may not be sufficient for interpretation. Additionally, this strategy does not allow for the collection of information such as why a given method was chosen over all other choices. "I'm going to undo this activity," for example, but the person does not explain why they are doing so. In post-task walkthroughs, the recorded data is presented back to the user and the evaluator discusses what persuaded the user's actions and their reasons.

CHAPTER 4
Identified Guidelines and Experimental Setup

As we have discussed in chapter 2, there is nothing we can do about intrinsic and germane cognitive load. It depends on different persons. But extraneous cognitive load is caused by the way in which information is presented. An effective interface causes less extraneous cognitive load on the user.

An organisation has to make a decision whether they require an application interface that will be used in automobiles. Application interfaces in automobiles are different from idle conditions. There are different ways to use the interface. First would be to reach an agreement directly with the automobile company. One can install the application interface directly onto the dashboard screen. Other would be to make the software compatible with android auto. You have to connect your mobile phone to the interface. The application will then need to switch to a special mode which gets displayed on the screen. There are some manufactures that allow for docking a mobile smartphone instead of a screen in the dashboard. In this case special mode should be made available in the smartphone application itself. Whatever the method you use, the interface should be different from when used under idle conditions.

4.1 Guidelines

We have discussed in chapter 2 that the two factors that affect driver distraction are duration of activity and rate of activity. Based on these principles and HCI guidelines, we have identified the following principles that are applicable to distracted driving conditions.

- Consistency
- Lesser Duration of Activity

- Lesser Frequency of Interaction
- Robustness

Consistency

The interface should be consistent with the normal interface. Efforts should be made to make sure that the colour palette, icons, etc should be consistent with the normal interface. This helps in making sure that there is less learning to be done and the users can depend upon their memory to easily figure out what different elements of the interface are supposed to be.

Lesser Duration of Activity

It is reported that if a task's duration is higher, then the cognitive load increases[4]. So, the interface should be designed in such a way that the duration of the activity is as low as possible. For example, in music apps, changing the volume requires the user to slide a button to increase or decrease the volume. This means that the user has to keep interacting with the interface for a significant time. So, the duration of the activity should be made as low as possible.

Lesser Frequency of Interaction

Decrease in the frequency of interaction also decreases the distraction levels of a driver. It is quite possible to achieve this by making sure that all the elements on the interface provide an endpoint towards a goal. If the user clicks on an element and the interface goes to another state to complete the task, the user will find it more distracting. But if the elements on the interface offer an endpoint to an activity, then the frequency of interaction with the interface decreases.

Robustness

The interface should be made to easily switch between the two modes. Developers can make it automatic but the ability to switch to another mode should be made easily accessible.

These principles provide a starting point for designers to develop interfaces in distracting environments. According to necessary requirements, these principles should be modified or extended. One of the most important points for the developers to remember is that these interfaces should not undergo drastic changes over time. The reason is because new interfaces take time to get used to and learning to use them again would increase the cognitive load. But in distracting environments, efforts should be made to remove this kind of learning curve.

How to evaluate these interfaces? Normal interfaces are usually evaluated by experts using a number of techniques mentioned before. These special mode interfaces require a system to get user feedback for the first few iterations. The user feedback is very costly and time consuming. That is why it is not widely used. But these interfaces are not to be changed drastically over the years. Hence, user feedback can be used for the first few iterations and will bring down the cost. An alternative is to use observational techniques which will keep the cost of evaluations down.

4.2 Experimental Setup and Process

The goal of this experiment is to check if the principles are effective or not. We assess the effectiveness of the principles by recording the responses of the users in similar environments. By effectiveness, we mean that we are checking if the cognitive load is lowering or not.

We asked a group of 63 people in the age group 18-33 years to participate in the experiment. According to studies [2], this is the age group that uses screens often during driving and this is also the age group that has a higher percentage of distracted driving related accidents. They were asked to fill a questionnaire that was designed to check the effectiveness of the principles. The questionnaire can be found in table 4.1.

Table 4.1: Questionnaire of the experiment

Sr. No.	Questions	Data Collection Format
Q1	In normal mode, while trying to play a song, did you make mistakes while driving? For example, moving away from the line, Not being able to make a turn, etc.	Yes/No
Q2	In normal mode, is it difficult/distracting to play the next song using the interface?	Yes/No
Q3	In normal mode, is it difficult/distracting to read the song name using the interface?	Yes/No
Q4	In normal mode, is it difficult/distracting to pause the song using the interface?	Yes/No
Q5	In normal mode, is it difficult/distracting to resume the song using the interface?	Yes/No
Q6	Since the play button is bigger, were you able to easily spot the play button without feeling distracted?	Yes/No
Q7	Were you able to use the interface to play the next song in car mode without feeling distracted?	Yes/No
Q8	Were you able to read the song name from the interface in car mode without feeling distracted?	Yes/No
Q9	Were you able to pause the song in car mode without feeling distracted?	Yes/No
Q10	Were you able to resume the song in car mode without feeling distracted?	Yes/No
Q11	In which mode, did you make more mistakes while driving?	Normal Mode/Car Mode/Didn't feel any difference
Q12	In which mode, did you feel comfortable using the interface while driving?	Car Mode/ Normal Mode/ No preference
Q13	In car mode, some of the features of the normal mode are sacrificed like equaliser, ability to separate playlists, etc. Were you ok with that sacrifice in order for improved experience?	Yes/No

Table 4.1: Questionnaire of the experiment

Sr. No.	Questions	Data Collection Format
Q14	In car mode, were you immediately able to figure out what each element of the player does? (Like the forward button, pause button, etc)	Yes/No
Q15	In normal mode, did it take more time to accomplish a task than in car mode?	Yes/No
Q16	Will you be ok if software developers sacrifice some additional features while preserving the core functionality of the app for better usability in driving?	Yes/No/Doesn't Matter
Q17	Do you want the interaction elements on the screen to be endpoints/actions ? (Do you want every element on the screen to perform an action or not. It is possible that an element might take you to another interaction interface. So you will have to do more tasks to get to an endpoint.)	Yes/No/Doesn't Matter
Q18	If the length of interaction with the interface is less, will it be a good experience? (This implies that you do minimal work to get a task done but you will not have access to additional features.)	Yes/No/Doesn't Matter
Q19	Does the experience become better if the non-interactive elements like colour scheme, logos etc are similar to that of the normal interface?	Yes/No/Doesn't Matter
Q20	Is it better if the system automatically changes to special mode than you manually turning it on?	Yes/No/Doesn't Matter
Q21	Will you use the interface in the real world while driving? Is there anything you would want the developers to add?	Subjective

We asked the subjects to get familiar with an online driving simulator[1]. We have used Amazon's Prime Music App as they have a special mode called 'Car mode' which is aimed to ease the use of the app during driving. We asked the subjects to perform certain tasks in both 'Car mode' and the normal mode while

[1]The online simulator can be found here.

driving in the simulator. The tasks include playing the song, reading the name of the song from the player, pause the song, play the next song, and resume the song. We then asked the participants to self report their mental efforts via the questionnaire to assess their difficulties in the two modes. We made efforts so that the questionnaire is easy to read and the questions can be answered in a simple yes no format. It is done to remove ambiguity and decrease the difficulty level of the questions for everyone to understand.

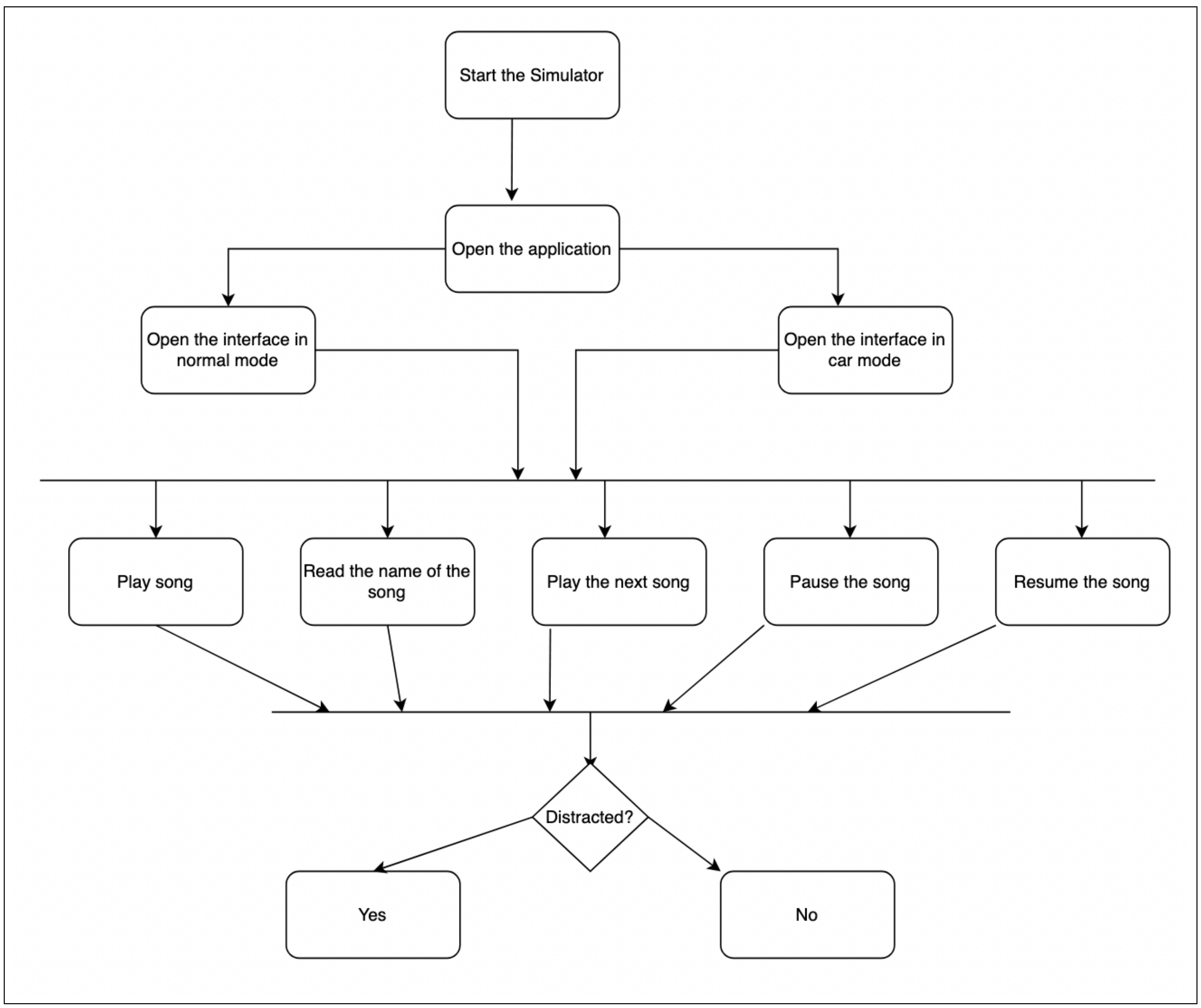

Figure 4.1: Flow of Experiment

First we asked them to perform the tasks in normal mode while driving the simulator. We didn't specify any specific driving route as the randomness would more likely represent a real world scenario. Then we asked the participants to perform tasks using a special mode called 'car mode' which is available in the application itself. It can be toggled on in settings. We asked them to perform the same tasks above in this new interface. We asked them to use the phone the same way they use in driving scenarios. The flow of the entire experiment can be found in figure 1. The sensitivity of the simulator is very less on lower speeds. So, we asked participants to drive the simulator in the 85-100 speed range.

The responses in normal mode are used as a base line and are compared to the responses of the 'car mode'. The questionnaire is prepared to find out how the participants feel if the above mentioned principles are implemented in the interface.

CHAPTER 5
Results and Analysis

Figure 5.1 shows how the interface looks in normal mode and in car mode. One can notice that player buttons are much larger in car mode and it doesn't have a lot of features like share button, chromecast button, etc.

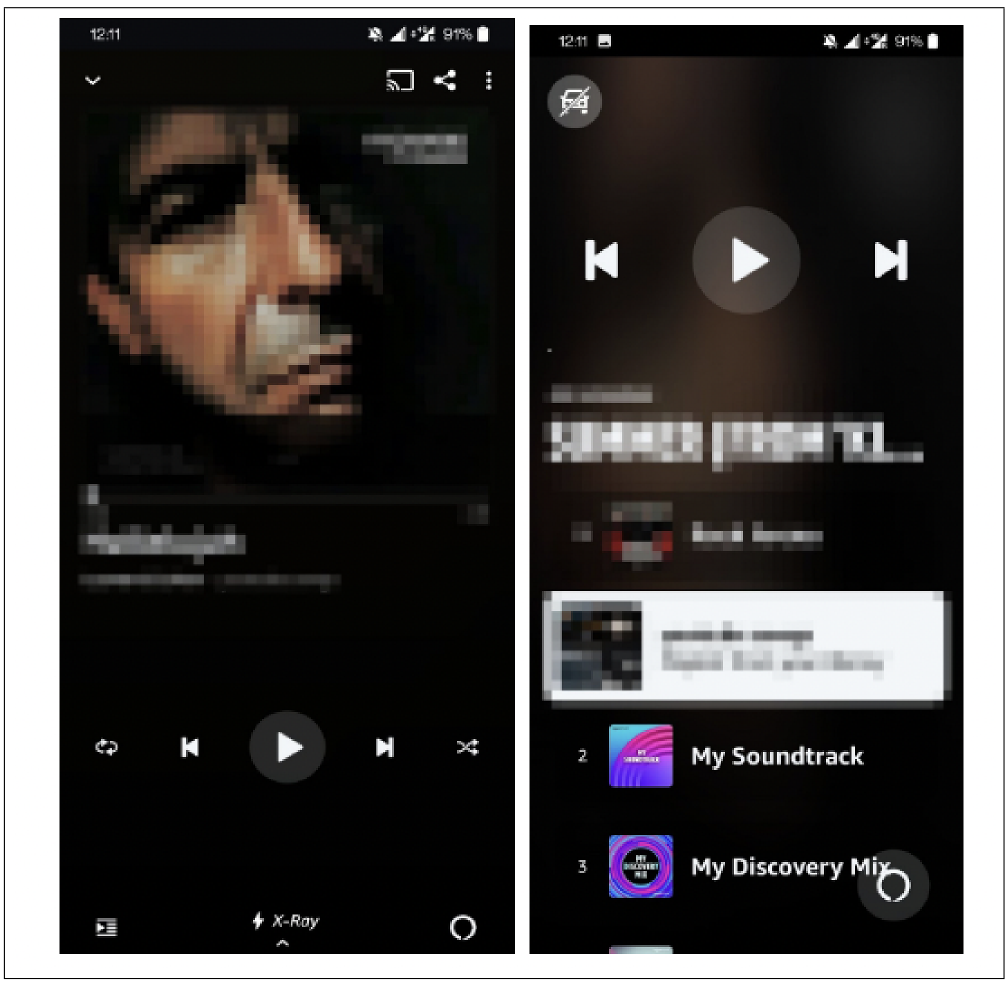

Figure 5.1: Normal mode (left) and Car mode (right) in the amazon prime music app

The questionnaire was designed in such a way that Questions 1-5 recorded how the subjects felt using the normal mode. Questions 6-10 recorded how the subjects felt with car mode. Questions 11-15 recorded the comparison between the two modes.

5.0.1 Results

From table 5.1, we can observe that more than 65% of the participants felt distracted performing all the 5 tasks. We can see that interacting with the interface in normal mode to pause, play, resume, read the name of the song, play the next song are distracting to most of the people. We can conclude that the interface is distracting to use to most users in normal mode. The interface requires some changes to make it less distracting. In this app, a car mode is implemented which aimed at reducing the distraction of the drivers.

Table 5.1: Responses of experience in Normal Mode

	Question	Yes	No
1	Q1	79.4%	20.6%
2	Q2	81%	19%
3	Q3	65.1%	34.9%
4	Q4	69.8%	30.2%
5	Q5	68.3%	31.7%

From table 5.2, we can observe that in car mode, all the tasks are easier to perform without distraction. Reading the name of the song was found difficult by 30.2% of participants. The buttons are much bigger in this mode. One can see that playing, pausing and resuming a song is not at all distracting to more than 93% of the participants. This is due to the bigger pause, resume button available on the interface. Playing the next song was found distracting by 19% of the people. From figure 5.1, we can see that interaction buttons are bigger and the text is much more readable and bigger. Due to these changes, the users felt the tasks are much less distracting in car mode.

Figure 5.2 shows that when asked about which mode was more distracting, 79.4% of the participants said it was Normal mode. While the remaining 20.6% said they did not feel any difference. It could be either because they have a higher threshold for cognitive load or they have used the interface and have gone

Table 5.2: Responses of experience in Car Mode.

S.no	Question	Yes	No
1	Q6	93.7%	6.3%
2	Q7	81%	19%
3	Q8	69.8%	30.2%
4	Q9	98.4%	1.6%
5	Q10	98.4%	1.6%

through the learning curve to use the interface. Even if the users are used to the normal mode, we can see that none of them said it was the car mode. We can conclude that for some people the experience may not be better but it does not get worse. The experience remains same.

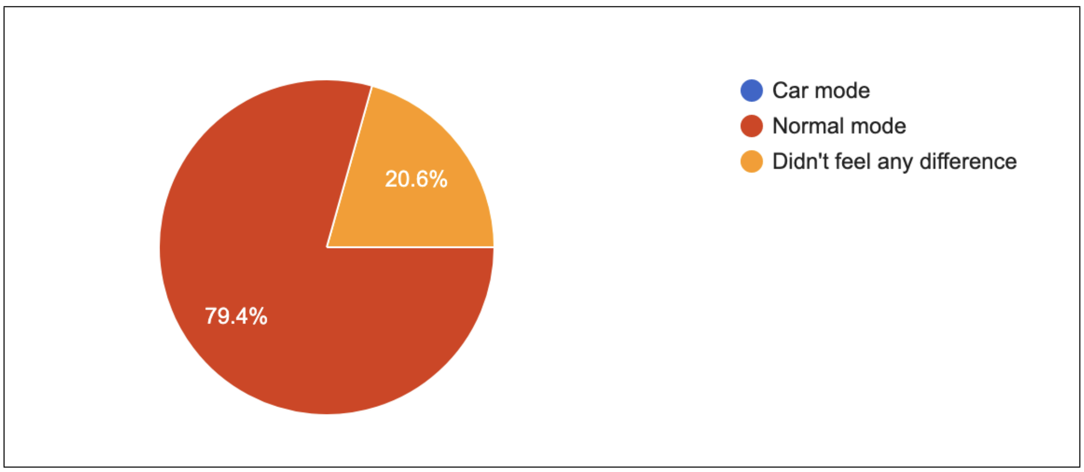

Figure 5.2: Responses to which mode was more distracting.

Figure 5.3. Shows that 76.2% of the participants felt more comfortable using the car mode. While the remaining said they don't have any preference and felt comfortable with both modes. These are the people who said that they have been using screens for a long time while driving and they trained their brain to use the screens.

Coming to the special mode, from figure 5.1, we can see that the colour palette, logos, symbols, etc of the car mode is same as that of the normal mode. So, consistency is maintained. In the car mode, one need not interact with interface more than once to get a task done. So, lesser frequency of interaction and lesser duration of interaction are maintained. The special mode is easily accessible from settings button and when connected to a car, this mode is automatically switched on. Hence, robustness is achieved. Table 5.3 shows the participants' response to

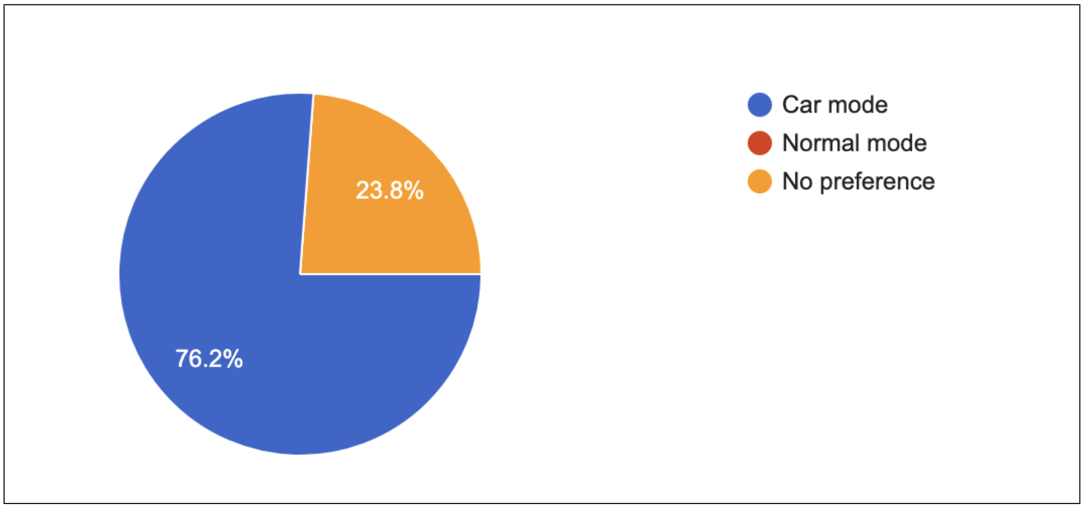

Figure 5.3: Responses to which mode is comfortable to use.

the questions which are designed to check the effectiveness of these principles. 77.8% say that if the frequency of interaction is less, it would enhance experience. 77.8% say that lesser duration of interaction enhances experience while only 1.6% said that it does not improve the experience and the remaining 20.6% of the participants said that it does not matter to them. 65.1% of people say that they would like the same colour palette, logos, etc which indicates consistency. Finally, 60.3% say that they would like the system to automatically switch while 11.1% said they would like manual control to switch the modes. The remaining 28.6% of the participants said they don't have any preference.

Table 5.3: Responses to check the effectiveness of principles.

S.no	Question	Yes	No	Doesn't Matter
1	Q16	79.4%	4.8	15.9%
2	Q17	77.8%	-	22.2%
3	Q18	77.8%	1.6	20.6%
4	Q19	65.1%	-	34.9%
5	Q20	60.3%	11.1	28.6%

While designing the questionnaire we checked if the participants would be fine with only preserving the core functionality of the application and removing extra features. For example, in the music app, in car mode, many extra features like chromecast, live lyrics, equaliser, etc are removed. Since these features are useless in driving scenarios, 79.4% of the participants felt that removing them did not affect the experience of the UI. Only 4.8% of the users said they want these features to be present in the car mode while the remaining 15.9% said they don't mind if these features are present or not. So, we can also add Reduction to Core Functionality to the list of guidelines.

5.0.2 Analysis and Takeaways

We have already seen in previous chapters that there have been a number of researchers who have proposed various HCI guidelines. Some of the guidelines overlap while some of them are new guidelines. These guidelines have been subjected to refinement and modifications for decades. While researchers proposed different principles, all of them agree that these principles are not irrefutable. They will have to be modified according to the situation.

One of the major motivation for this research has been the lack of HCI guidelines for driving environments. 98.8% cars released in 2020 in USA have screens fitted into the dashboards. These screens offer different functionality ranging from accessing car features to display information, etc. According to American Automobile Association's director, these screens are a safety hazard and the safety hazard is not due to the screens but due to the design of these screens. Guidelines for designing these interfaces are non-existent. Hence, a set of guidelines are to be identified for better usability in distracting environments.

In the previous chapter we have seen Norman's principles, Shcneiderman's Eight Golden rules, Alan Dix's principles. One principle that is common in all the three set of guidelines is Consistency. Internal consistency is important in driving environments. When users use the application in idle conditions, they get familiar with the application's colour palette, logos, interaction elements, etc. In distracting environments, users will not have time to familiarise themselves with new color scheme, interaction elements. Hence, it is important that the interface is consistent with the interface that is used in idle conditions. In the experiment there two questions to check the effectiveness of consistency. *In car mode, were you immediately able to figure out what each element of the player does? (Like the forward button, pause button, etc)* and *Does the experience become better if the non - interactive elements like colour scheme, logos etc are similar to that of the normal interface?*. 65.1% of the participants have said that the experience would be better if the special mode maintained consistency with normal mode.

In the subjective question, when asked if they have any suggestion 8 of the participants have said that they felt something is off while playing a playlist. They said that they expected the interface to go into the playlist so they can choose which song to play. But in car mode, the the playlist automatically plays and does not allow one to pick the song they want to play from the playlist. Thus, we can conclude that consistency is a rather important principle. Even when not explicitly mentioned, people expect interfaces to be consistent.

There are different ways to access the applications in interfaces in dashboard. If the interface is equipped with Android Auto, then the users need to connect the phone with the interface via USB. Then the users can directly access the application from the screen itself. There are cars which use screens just to display information. One cannot use applications with these screens. Cars have the option of docking their mobile screens on to the dashboard. In this option, the mobile screen itself is an interface that the users use while driving. There are different ways to interact with the screens while driving. Therefore the systems should be robust enough so that the users can use their preferred mode. It is also important that these support systems are either easily accessed or automatically turned on. In the experiment, we asked the users, if they would like the system to automatically change to special mode on connecting with interface in car. 60.3% said they want the system to automatically turn on the special mode while remaining said it wouldn't matter and they can turn it manually. Hence, the system should maintain robustness to switch to different modes of interfaces easily.

In chapter 2 we have discussed the factors affecting the cognitive distraction in drivers. First one is rate of interaction. From figure 5.4 we can see that 87.3% of the participants have said that they took more time to accomplish the task in normal mode. It is question number 15 in the questionnaire.

Why did it take more time to finish a task in car mode than in normal mode for majority of the participants? This is because in car mode, the frequency of interaction is much less. All the interacting elements on the interface are endpoints. In question no. 17, we asked users what they felt about rate of interaction. *Do you want the interaction elements on the screen to be endpoints/actions ? (Do you want every element on the screen to perform an action or not. It is possible that an element might take you to another interaction interface. So you will have to do more tasks to get to an endpoint).* 77.8% of the users have said that they would like the rate of interaction to be as less as possible. As we have seen, the users reported that they have made

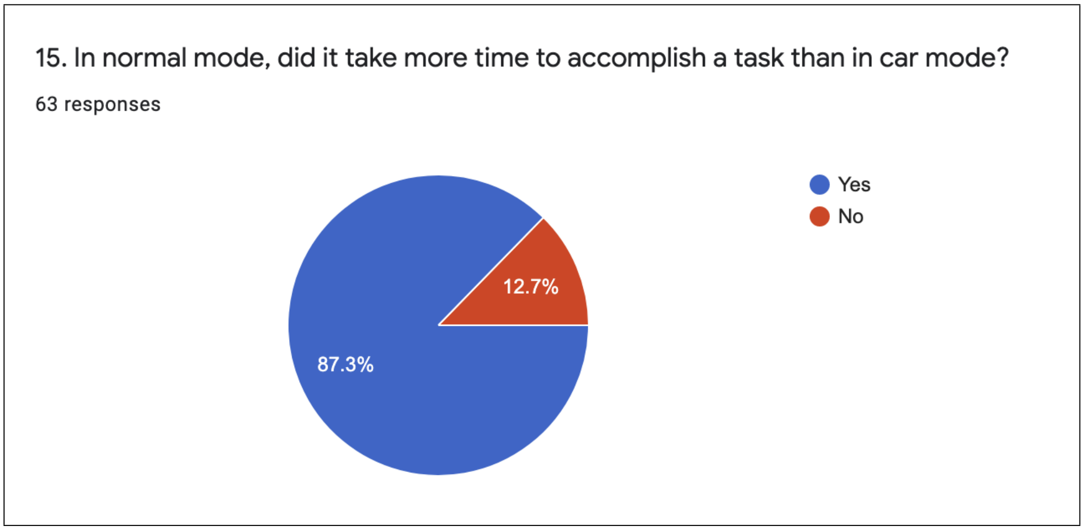

Figure 5.4: Responses to which mode took more time to accomplish a task.

less mistakes while using the car mode, it garners importance that lesser rate of interaction indeed does help improve user experience.

The other factor that affects distraction is the length of interaction. The interface in car mode has all the interaction elements available on the homepage itself. One doesn't need to interact with the interface for longer periods of time. Furthermore, all the buttons, text, etc are much bigger in size and there is less scrolling to access other elements. In the questionnaire, question no.18 was asked to get the responses on length of interaction. *If the length of interaction with the interface is less, will it be a good experience? (This implies that you do minimal work to get a task done but you will not have access to additional features)*. 77.8% of the participants have responded that the experience would be better if they have to do minimal work to get things done. With doing minimal work to get things done, users' focus will not shift much during driving.

From figure 5.1, one can see that in normal mode, the text is smaller, buttons are smaller, etc. One can see that there are additional features like sharing button, chrome-cast button, etc on the interface. In car mode, all these additional features are removed. In the questionnaire, question no 13 and 16 were asked regarding how the users felt when these features are removed. *In car mode, some of the features of the normal mode are sacrificed like equaliser, ability to separate playlists, etc. Were you OK with that sacrifice in order for improved experience?* and *Will you be OK if software developers sacrifice some additional features while preserving the core functionality of the app for better usability in driving?* We can see that in car mode, the app is reduced to it's core functionality. All the interface does in the car mode is to play music, pause, resume, etc which are all basic interactions with any music player. By removing the extra features and keeping it consistent with the normal mode of the music player, 79.4% of the participants felt that the experience is better. So, it doesn't matter how many extra features one has in their interface, while driving, the interface should be reduced to it's basic functionality. It significantly improves the experience.

So, what are the takeaways from this experiment? The guidelines which are being proposed will be very useful in creating user interfaces in driving scenarios. Now, another question arises. Are these guidelines enough? The simple answer is no. This is a fairly new problem that arose because of the presence of screens in the cars. As technology advances, we may find new ways of interaction and new guidelines will be required in the future. The guidelines in this section are proved to be effective. They provide a strong base for the developers to rely on. There should be feedback channels to get user feedback.

CHAPTER 6
Conclusions and Future Work

From the results we have also found that participants don't mind having extra features like equaliser, live lyrics, etc removed. In distracting environments, users don't have time to use these features. Reduction to Core Functionality should also be added to the list of principles. From the experiments we can conclude that the principles in this paper could provide a starting point for designing interfaces for distracting environments. The guidelines are:

- Consistency
- Lesser frequency of interaction
- Lesser duration of interaction
- Reduction to Core Functionality
- Robustness

Just like every HCI guidelines, these principles will have to be refined over time through further research studies and user feedback.

In our research, the sample size of the experiment is 19 because these are the first steps taken to propose guidelines in this domain. A much bigger sample size can be used to ratify the guidelines proposed in this thesis. When developers use these principles to create interfaces, they should keep feedback channels open to get responses from users in real life. It would help to evaluate which principles are working in real life.

The experiment is conducted using online simulator. For further confirmation, a real car can be used to check the effectiveness of the guidelines.

References

[1] G. D. Abowd and R. Beale. Users, systems and interfaces: A unifying framework for interaction. In *HCI*, volume 91, pages 73–87, 1991.

[2] N. H. T. S. Administration et al. Young drivers report the highest level of phone involvement in crash or near-crash incidences. *Traffic safety facts: Research note (Report No. DOT HS 811 611). Washington, DC: National Highway Traffic Safety Administration. Retrieved from http://www. nhtsa. gov/staticfiles/nti/pdf/811611. pdf*, 2012.

[3] O. Bratfisch et al. Perceived item-difficulty in three tests of intellectual performance capacity. 1972.

[4] R. Brünken, S. Steinbacher, J. L. Plass, and D. Leutner. Assessment of cognitive load in multimedia learning using dual-task methodology. *Experimental psychology*, 49(2):109, 2002.

[5] S. K. Card, T. P. Moran, and A. Newell. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410, 1980.

[6] A. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human-computer interaction*. Pearson Education, 2004.

[7] F. T. Eggemeier, G. F. Wilson, A. F. Kramer, and D. L. Damos. Workload assessment in multi-task environments. *Multiple-task performance*, pages 207–216, 1991.

[8] J. Finlay and R. Beale. Neural networks and pattern recognition in human-computer interaction. *ACM SIGCHI Bulletin*, 25(2):25–35, 1993.

[9] I. O. for Standardization. Road vehicles–transport information and control systems–detection response task (drt) for assessing attentional effects of cognitive load in driving, 2016.

[10] R. Gong and J. Elkerton. Designing minimal documentation using a goms model: A usability evaluation of an engineering approach. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 99–107, 1990.

[11] M. Klepsch, F. Schmitz, and T. Seufert. Development and validation of two instruments measuring intrinsic, extraneous, and germane cognitive load. *Frontiers in psychology*, page 1997, 2017.

[12] C. Lewis and J. Rieman. Task-centered user interface design. *A practical introduction*, 1993.

[13] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[14] J. Nielsen. Usability inspection methods. In *Conference companion on Human factors in computing systems*, pages 413–414, 1994.

[15] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, 1990.

[16] D. Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[17] F. G. Paas. Training strategies for attaining transfer of problem-solving skill in statistics: a cognitive-load approach. *Journal of educational psychology*, 84(4):429, 1992.

[18] F. G. Paas. Training strategies for attaining transfer of problem-solving skill in statistics: a cognitive-load approach. *Journal of educational psychology*, 84(4):429, 1992.

[19] F. G. Paas and J. J. Van Merriënboer. Variability of worked examples and transfer of geometrical problem-solving skills: A cognitive-load approach. *Journal of educational psychology*, 86(1):122, 1994.

[20] B. Park and R. Brünken. The rhythm method: A new method for measuring cognitive load—an experimental dual-task study. *Applied Cognitive Psychology*, 29(2):232–243, 2015.

[21] S. Savage, B. Tatler, and D. Potter. Oculomotor signatures of cognitive distraction. 08 2013.

[22] B. Shneiderman. Designing the user interface: strategies for effective human-computer. *Interaction*, 3, 1998.

[23] D. L. Strayer, J. M. Watson, and F. A. Drews. Cognitive distraction while multitasking in the automobile. In *Psychology of learning and motivation*, volume 54, pages 29–58. Elsevier, 2011.

[24] J. Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2):257–285, 1988.

[25] J. Sweller. Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction*, 4(4):295–312, 1994.

[26] P. W. Van Gerven, F. Paas, J. J. Van Merriënboer, and H. G. Schmidt. Memory load and the cognitive pupillary response in aging. *Psychophysiology*, 41(2):167–174, 2004.