MASTER'S THESIS 2023

Introducing Voice Control in a Graphical User Interface Using a Keyword-Based Approach

Matilda Jansson, Ellinor Nelderup

DEPARTMENT OF DESIGN SCIENCES
LTH | LUND UNIVERSITY

EXAMENSARBETE
Interaktionsdesign

Introducing Voice Control in a Graphical User Interface Using a Keyword-Based Approach

Introduktion av röststyrning i ett grafiskt användargränssnitt med en metod baserad på nyckelord

Matilda Jansson, Ellinor Nelderup

Introducing Voice Control in a Graphical User Interface Using a Keyword-Based Approach

Matilda Jansson
`ma1423ja-s@student.lu.se`

Ellinor Nelderup
`el7858ne-s@student.lu.se`

May 26, 2023

Master's thesis work carried out at
the Department of Design Sciences, Lund University.

Supervisors: Kirsten Rassmus-Gröhn, `kirsten.rassmus-grohn@certec.lth.com`
Felix Kaaman, `Felix.Kaaman@axis.com`

Examiner: Johanna Persson, `johanna.persson@design.lth.se`

Abstract

Speech interaction is a growing field within human-computer interaction. The introduction of several voice-user interfaces on the market has raised the question of whether voice control could be a suitable feature for a video management system. This thesis investigated how voice control could be designed and integrated with a graphical interface in order to add value to the user experience. User research showed that the most common workflows included navigating the system as well as reviewing and exporting footage. Hence, these workflows were prioritized. The design of the voice control solution was developed with an agile approach, where implementation was alternated with testing and evaluation. The final solution was keyword-based, and allowed for a seamless integration between manual interaction and voice commands. This was complemented with graphical support to meet the challenges regarding the invisible nature of voice interaction. As a result, the user was able to perform the most common workflows by voice independently.

Keywords: Voice control, VUI, surveillance, integration, keywords

Sammanfattning

Röstinteraktion är ett växande område inom människa-datorinteraktion. Flera röstgränssnitt har redan introducerats på marknaden, och detta har väckt frågan om huruvida röststyrning hade kunnat vara ett lämpligt komplement till en befintlig programvara för videohantering. Detta projekt undersökte hur röststyrning skulle kunna designas och integreras med ett sådant grafiskt användargränssnitt för att bidra till användarupplevelsen. Genom användarstudier identifierades de vanligaste arbetsflödena, vilka innefattade såväl navigering i systemet som granskning och export av inspelat material. Dessa arbetsflöden prioriterades i utvecklingen av röststyrningsfunktionen. Projektet utgick ifrån en agil metodik, där implementering och tester genomfördes iterativt. Den slutgiltiga designen av röststyrningsfunktionen baserades på nyckelord och möjliggjorde en sömlös integration mellan manuell interaktion och röstkommandon. För att adressera utmaningarna kring det faktum att röst är en osynlig interaktionsform inkluderades även visuellt stöd i lösningen. Resultatet gjorde det möjligt för användaren att självständigt utföra de vanligaste arbetsuppgifterna med röststyrning.

Nyckelord: Röststyrning, röstgränssnitt, övervakning, integration, nyckelord

Acknowledgements

We would like to thank Kirsten Rassmus-Gröhn for her support and feedback throughout the project.

We would also like to thank Felix Kaaman, who has helped us find efficient solutions to a variety of issues. His role as contact person has also been much appreciated.

Thanks also to Axis Communications for giving us the opportunity to carry out this Master's Thesis work.

Finally, we want to thank everyone that helped drive this project forward by participating in tests or observations, providing us with valuable insights, or simply encouraged us along the way.

Contents

Chapter 1

Introduction

1.1 Background

The interest for speech interaction is growing within the field of human-computer interaction. Multiple speech interfaces have already been released to the market such as Google Assistant, Amazon Alexa, and Siri [17]. Several studies have shown that speech interactions via Voice User Interfaces (VUI:s) have a lot of potential, but also come with many challenges [21] [17] [9]. One of these challenges is the unavailability of established design guidelines [17] which are used to ensure the usability of the interface. However, a well designed VUI has the potential to increase the usefulness of a system in several different ways, such as reducing the cognitive load, enabling hands-free usage, and increasing accessibility, efficiency, and mobility [21] [17] [9]. These positive effects have sparked an interest for voice interactions among many different companies, one of those being Axis Communications.

Axis Communications is a company that develops surveillance systems. This includes network cameras, network audio, access control, wearables etc [7]. In addition, they provide video management softwares which are designed to match Axis' other products. One of these softwares is AXIS Camera Station (ACS) [6] which is a tool used to manage the surveillance system. For example, ACS can be used to review and export recorded footage, or to configure cameras. It should fit a wide range of installations, which means that the software could be used to manage more or less complex surveillance systems. ACS is constantly being maintained and further developed with new features. The introduction of several VUI:s on the market and their associated potential has raised the question whether voice control could be a suitable feature for ACS.

1.2 Aim & Method

How could voice control be designed and integrated with the Swedish version of ACS in order to be a valuable contribution to the user experience in the current interface? This thesis aims to answer this question by three main activities: User research, implementation and evaluation. Designing a good voice control feature that contributes to a positive user experience requires knowledge about the users. This knowledge will lay the foundation for a design of the feature which will then be implemented, partially or completely, depending on technical limitations. The result will be evaluated and the process of designing and implementing will be repeated to achieve the most satisfactory result possible.

1.3 Disposition & Delimitations

This thesis will begin with describing relevant literature, followed by an account of the methodology used. Further, it will explain the process flow, consisting of user research, conceptual design, and the iterative work with implementation and evaluation. Finally, the resulting product will be presented and discussed.

The project will focus on the process of designing a voice control feature with already available tools for speech recognition. It will in other words not cover the machine learning or processing which is required for the speech recognition to work. Furthermore, this thesis will focus on managing existing functions in ACS by voice. Hence, new advanced functions that are not currently available will not be implemented.

It is also important to recognize that the focus in this project will be to develop a voice control feature, an add-on, to the current graphical interface of ACS, and not a completely new interface driven by voice.

1.4 Work distribution

The project was mainly carried out through collaborative work, and both project members actively participated in designing and implementing the solution. Exceptions to this were the initial search for relevant literature and the creation of the concept videos, where the work was divided equally. The report work was also initially divided, but the final processing of the report was done collectively.

Chapter 2

Literature & System description

2.1 Speech recognition and natural language processing

Speech recognition is a field within computational linguistics that deals with the ability of a computer to identify spoken words. More specifically, a speech recognition system converts audio input into text. Such a system needs to be trained, which is done by feeding it with vocabulary and speech patterns of the target language [24]. Machine learning methods such as deep neural networks have long been considered a natural tool to perform this type of training. In fact, speech was one of the first applications of deep learning [18]. Today, the cognition of speech recognition systems is comparable to that of humans in quiet environments. However, in noisy environments this level of cognition has not yet been achieved [24].

To enable the computer to draw conclusions from text produced by the speech recognition system, the concept of natural language processing [24] comes into play. Natural language processing is the analysis and interpretation of a text that is written in a human language. It involves dividing the text into structures (paragraphs, sentences and words), analyzing the structures in relation to each other and finding the dictionary meaning of them [24]. The goal is often to extract keywords or key phrases [10] to get an overall understanding of the text. Things that need to be considered by a natural language processing algorithm include the dependence consecutive sentences have on each other, and context, which requires knowledge of the real world. The greater the amount of text that needs to be processed, the more complex the analysis becomes [24].

2.2 Voice control & VUI:s

Speech recognition technology and natural language processing makes it possible to create voice control systems [24], where actions are performed based on spoken instructions from the user. Today, there are many examples of systems that use voice interaction, sometimes as the primary interaction type and other times as a complement to other interaction types. Intelligent home assistants are examples of primarily voice-based interfaces [21], and are often referred to as VUI:s or conversational agents. These typically lack a visual interface and respond to the user through synthesized speech. On the other hand, mobile devices mainly use voice interaction as a supplement to touch [9]. In this context, voice control is rather used as a way of enhancing or facilitating the user experience, as it enables hands-free interaction.

2.3 Challenges and opportunities

VUI:s comes with many opportunities, but also with a number of challenges to consider. This section will cover some of the main challenges that often are discussed in the context of VUI:s.

2.3.1 Design guidelines

A general problem when it comes to VUI:s is the lack of established design principles. On the other hand, there are many established guidelines for graphical interfaces, where Norman [20], Nielsen [19] and Shneiderman [26] are commonly mentioned. There have however been summaries of the challenges with VUI:s in relation to the existing guidelines for graphical interfaces [17] and how these can be complemented to make them more suitable for voice interfaces. Some of the insights are presented below:

- *Visibility/Transparency/Feedback.* Often, the user does not know when it is their time to speak. There is a lack of visibility of when and how the user should respond to the interface [17]. It is also hard to know what the interface is capable of since it often lacks transparency and feedback.

- *User control/Freedom.* A big problem is the lack of control users often feel regarding VUI:s. This leads to frustration and fear that they might miss important parts of the interaction [17]. Providing users with a sense of control might therefore improve user satisfaction. Allowing the user to take control over the interaction by interrupting a current dialogue could provide such a sense of control.

- *Cognitive load/Recognition rather than recall.* Audio output is presented serially, and the user thereby might need to remember long pieces of information. Hence, if audio is the only output modality it could increase the cognitive load [17]. The users might also struggle if there are too many options to remember. More than 5 options can be considered a break point [17]. Furthermore, it is often not intuitive how the user should structure their speech to a VUI, which makes it hard to remember how to give the interface instructions [17].

- *Efficiency.* VUI:s could increase efficiency since they allow the user to say an instruction instead of searching through a graphical user interface for a certain task [17].
- *Error recovering.* It is generally hard to recover from speech recognition errors [17] without creating new ones. It leads to frustration not being able to edit misunderstood queries and it is hard to undo an action and go back to default settings or the home menu.
- *Guidance/Documentation.* An interactive tutorial often increases the user's performance with the VUI. The help can be provided progressively and preferably contextually within the interaction [17].
- *Privacy.* Using VUI:s introduces a privacy problem since the shared information can be heard by everyone within earshot. It is also unclear what information is being collected since the system often lacks transparency [17].

Other important design principles are *learnability* and *discoverability*. While learnability is a measure of the user's ability to learn how to use a system, discoverability refers to how easy it is to discover features within it [12]. The invisible nature of VUI:s makes it difficult for a user to discover how they work. In this way, the level of discoverability can impact the level of learnability [13].

2.3.2 Accessibility

An area where voice control has big potential is accessibility. VUI:s have a great opportunity to make different types of devices accessible for more kinds of users. For those who have limited hand dexterity or other motor impairments in hands and arms, VUI:s could be a solution for operating devices that otherwise would be out of range. This would benefit these groups with increased independence and freedom. However, voice control is today often offered as an alternate control modality during situational impairments [9] such as limited vision, and limited hand availability while driving. Solely voice as interaction modality is not recommended [27] since physical interactions are considered more efficient. This will however decrease the accessibility for certain user groups such as those with motor impairments.

2.3.3 Context & Environment

The interaction with a VUI is not considered a natural interaction by users, which can cause the user to feel uncomfortable to interact with the system vocally in front of others [17]. This is related to "accountability", which in this context means that the user feels a need to explain herself to others before using the system. I.e. a request to the VUI is preceded by an announcement of the user's intention, to make the interaction feel less awkward. This is referred to as "accounting work" [21] and could make the interaction with the system less effective because of the extra time spent explaining the intention to others. It is also worth mentioning that the performance of speech recognition varies depending on the environment. It works better in environments that are calm and silent, while it performs worse in loud, busy environments [24]. Hence, the risk of a failed interaction with a VUI is higher in noisy environments, which could further reinforce the feeling that accounting work is needed.

2.4 AXIS Camera Station

AXIS Camera Station [5] is a video management software used for video surveillance and access control. It is used for live monitoring, managing the surveillance cameras, reviewing recordings and exporting video. An overview of the user interface can be seen in figure 2.1. The basic functionality revolves around the display of video material, which can be accessed by choosing specific cameras or views. A view includes one or several cameras and is created and customized by the user. Below the video display there is a timeline that shows the date and time of the current video content. The time marker can be dragged along the timeline to fast forward or rewind, also referred to as "scrubbing". For larger time jumps, there is a calendar that enables the user to set both date and time. To export video, the user sets export markers that define the start time and stop time for the desired video sequence, and clicks the export button.

ACS also includes more advanced features that can be used when looking for specific details in the recordings. For example, it is possible to display events of different kinds on the timeline. An event refers to something that has been detected by the camera, like motion. There is a filter function that can be used to highlight tags such as motion events and bookmarks. ACS also offers an intelligent search function, *Smart search*, where the operator can search for specific clothing colors, vehicles, etc. More extensive systems may incorporate access control, and for this application there is a feature called *Secure entry*. This makes it possible to identify people before letting them into a building.

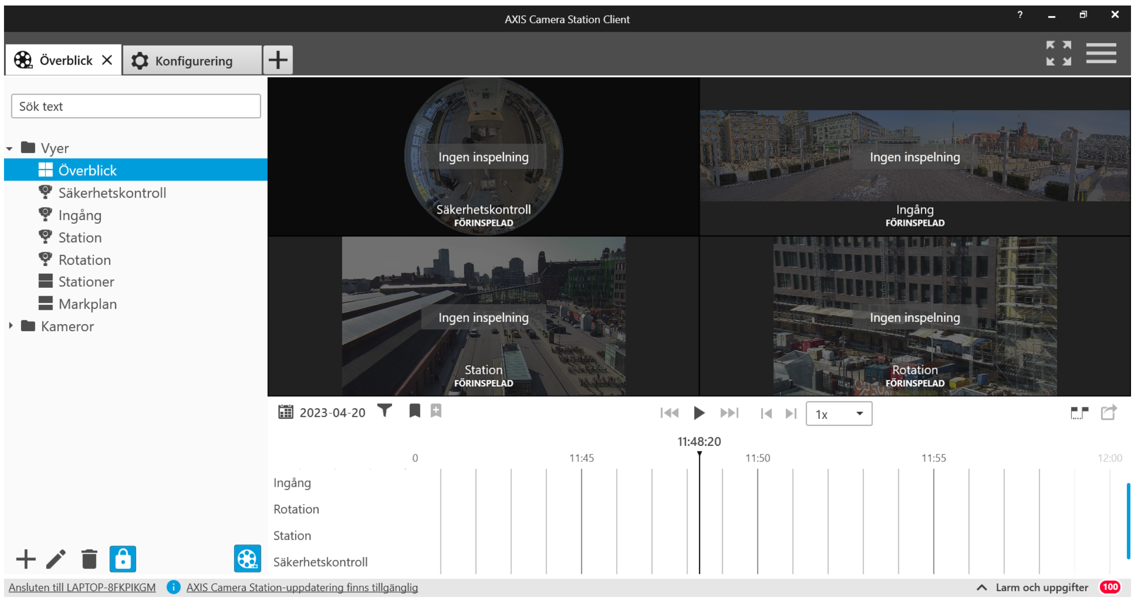


Figure 2.1: AXIS Camera Station Client

Chapter 3
Methodology

3.1 Agile working method

The aim of Agile software development [2] is to deliver a finished, working software solution, where sufficient time has been allocated to ensure that the code is properly written and reliable. This is achieved by dividing the project into a series of smaller projects, each of which leads to a new version of the product [3]. By employing frequent iterations, the solution can be tested and feedback be received within short time intervals [3], leading to a product that continuously evolves. This approach allows for flexibility of the requirements during development, which facilitates in finding a solution that satisfies the users' needs and in turn, generates value for the user. Agile methods can be summarized by the four values defined in The Agile Manifesto [2]:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

3.2 User-Centered design

The international standard ISO 9241-210 describes the development of interactive systems with a Human-Centered or User-Centered approach. The goal with a User-Centered focus is to develop useful and usable systems. The standard provides a number of principles which should be followed during the design process in order to obtain such a system [11]:

- The design is based upon an explicit understanding of users, tasks and environments

- Users are involved throughout the design and the development
- The design is driven and refined by user-centered evaluation
- The process is iterative
- The design addresses the whole user experience
- The design team includes multidisciplinary skills and perspectives

The User-Centered Design process consists of four activities [11] as portrayed in figure 3.1. The first step is to understand and specify the context of use, the second to specify the user requirements, the third to produce design solutions to meet the specified requirements, and lastly to evaluate the design against the requirements. These four activities are carried out with iterations when it is appropriate [25].

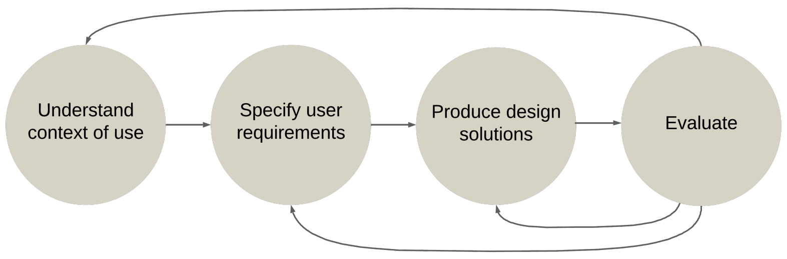


Figure 3.1: User-Centered Design Process

3.3 Agile-UX with a sequential approach

Working with an Agile approach limits the time spent on producing requirements that may become irrelevant at a later stage, and instead uses that time to produce working software. This reduces the risk of the initial analysis becoming too comprehensive [25]. However, stand-alone Agile methods are software-focused and do not provide information on how user requirements should be satisfied [25]. To ensure this is considered, it can be beneficial to integrate User-Centered Design methods with Agile methods [8]. This integration is called Agile-User Experience Design or Agile-UX [25]. Since this project deals with both design and implementation of a software solution within a relatively strict time frame, adopting an Agile-UX methodology is concluded to be beneficial. Achieving a working solution at an earlier stage becomes an important part in ensuring the voice control solution can be demonstrated and evaluated in a realistic way.

An Agile-UX approach should explain how the work is to be organized. There are several proposals of process designs defining how the development- and design tasks can be divided [25], but commonly the tasks are parallelized. However, this assumes the existence of two teams working with each part, which is not applicable for this project. Consequently, we will instead adopt a sequential approach as described by Deuff et al [25]. This process consists of recurrent, iterative design- and development phases, and is concluded by a final test. In addition to the final test, user tests are conducted in each iteration. With this approach, the project can move forward in both the design- and development phases concurrently, while still working sequentially.

Chapter 4
Process flow

The development of the voice control solution needs to consider both how the human-computer interaction can be optimized, and what the technical limitations are. An optimal solution should meet as many of the user's needs as possible, while making sure that the technical implementation is well-functioning and robust. In the early stages of the project, we conducted an analysis that resulted in insights about who Axis' users are, what they want to achieve and how their needs can be satisfied. This analysis followed the structure of a design process consisting of four main activities: user research, conceptual design, implementation and evaluation [22]. The user research- and conceptual design steps were first conducted more extensively to produce an initial design from a set of user requirements. We also selected an appropriate voice control library to work with to implement the initial design. This design was then evaluated and updated iteratively according to the Agile-UX approach. Through the implementation of this solution, there was also a general exploration of what was technically possible and reasonable in relation to the quality of the library, our own ability and the time frame for the project.

4.1 User research

The purpose of the user research was to get an introduction to the different user groups connected with AXIS Camera Station and to identify their needs. The research also aimed to provide information about what unites the user groups and what separates them from each other.

To find the starting position for the user research, it was first concluded that ACS already is a product on the market and thus the user groups were already well defined. The challenge would instead be to investigate how these previously established users could benefit from a voice control feature. Nevertheless, it was still deemed important for the project that we perform our own user research tasks so that we as designers could attain a good level of understanding of the users.

Interviews were frequently used as a data gathering method. This was motivated by the situation of having a lot of knowledgeable people like UX designers and system experts in our immediate working environment. Additionally, an observation with a real user was conducted. The combined insights from these project activities are summarized in section 4.1.3.

4.1.1 Personas

Theory

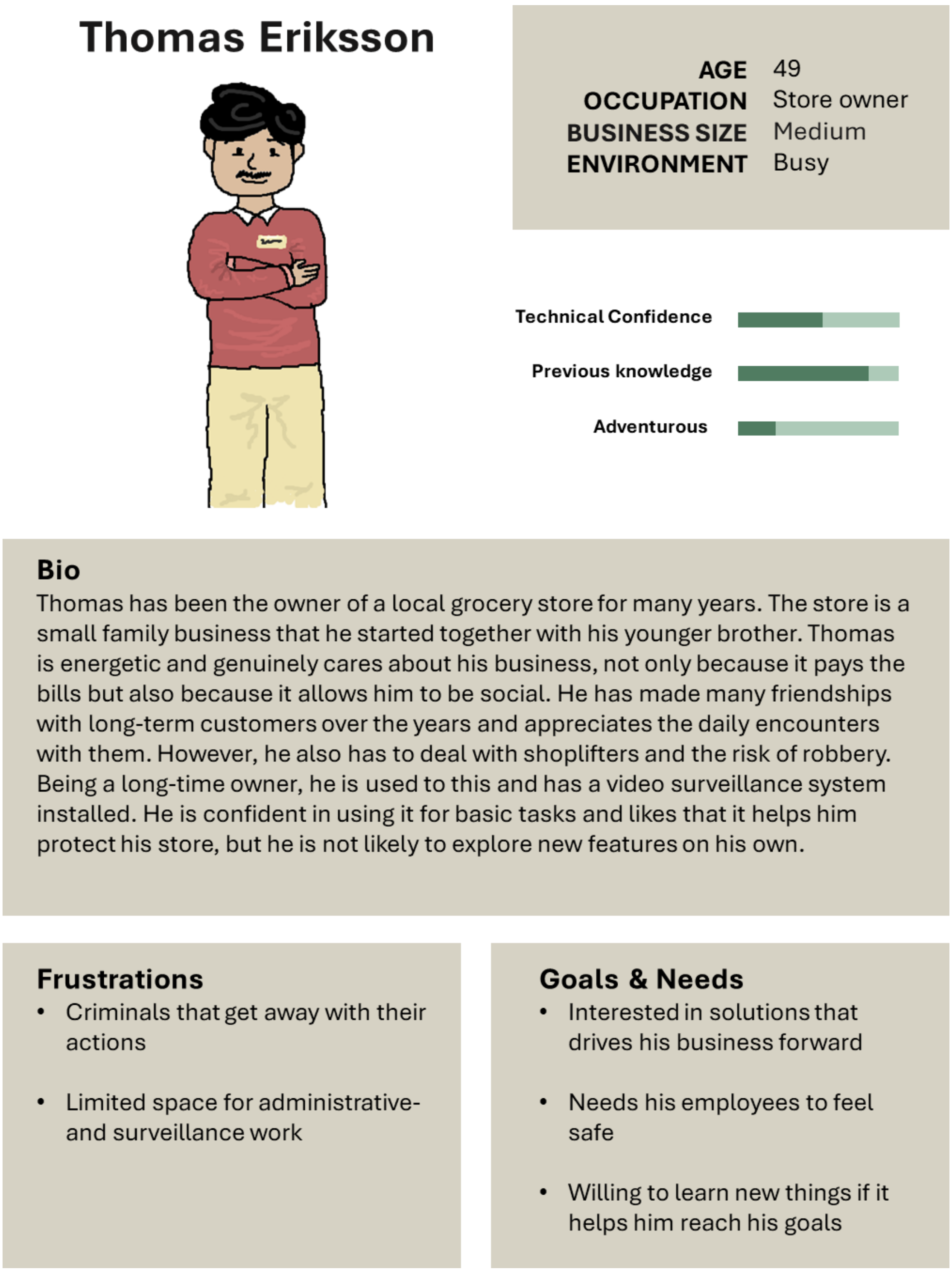
The characteristics of different user groups can be summarized into a number of user profiles, which are commonly brought to life by the transformation into personas. Personas are relatable characters accompanied by detailed descriptions [22] which designers can use throughout the design process to keep in mind who they are designing for. One persona typically represents multiple users which are united by the same goals. Therefore, it is often beneficial to create more than one persona to better capture the variability of users.

Workflow

To gain a first insight into the different user groups that are important to Axis, and specifically to ACS, an unstructured, exploratory interview with a Usability Engineer at Axis was conducted. Axis has developed their own personas so that it is clear to employees what they are working towards, and these were frequently used during our meeting to communicate the insights that they have learned so far.

The introduction to Axis' users and their personas laid the groundwork for the creation of our own personas. In order to contribute with something new, these were specifically developed with the voice control aspect in mind. The personas were created as different types of operators, i.e. the end users of the video surveillance system. The operators are typically owners of local, small- or medium-sized businesses. They are characterized by their entrepreneurial personality and their interest to keep their business safe. Generally, they are not particularly confident or interested in technology, but can appreciate it for the value it adds to their business. In addition to the operators, there are system integrators. They install surveillance systems and support the end users.

Below is a representation of the three personas created. Figure 4.1 presents Thomas Eriksson, the grocery store owner who cares a lot about his business. Figure 4.2 presents Hanna O'Sullivan, a professional investigator who enthusiastically learns about new technologies and ways to become more efficient at work. Lastly, figure 4.3 presents the board member Berit Lundqvist, who struggles to deal with a newly installed surveillance system at her local activity club.



Thomas Eriksson

AGE	49
OCCUPATION	Store owner
BUSINESS SIZE	Medium
ENVIRONMENT	Busy

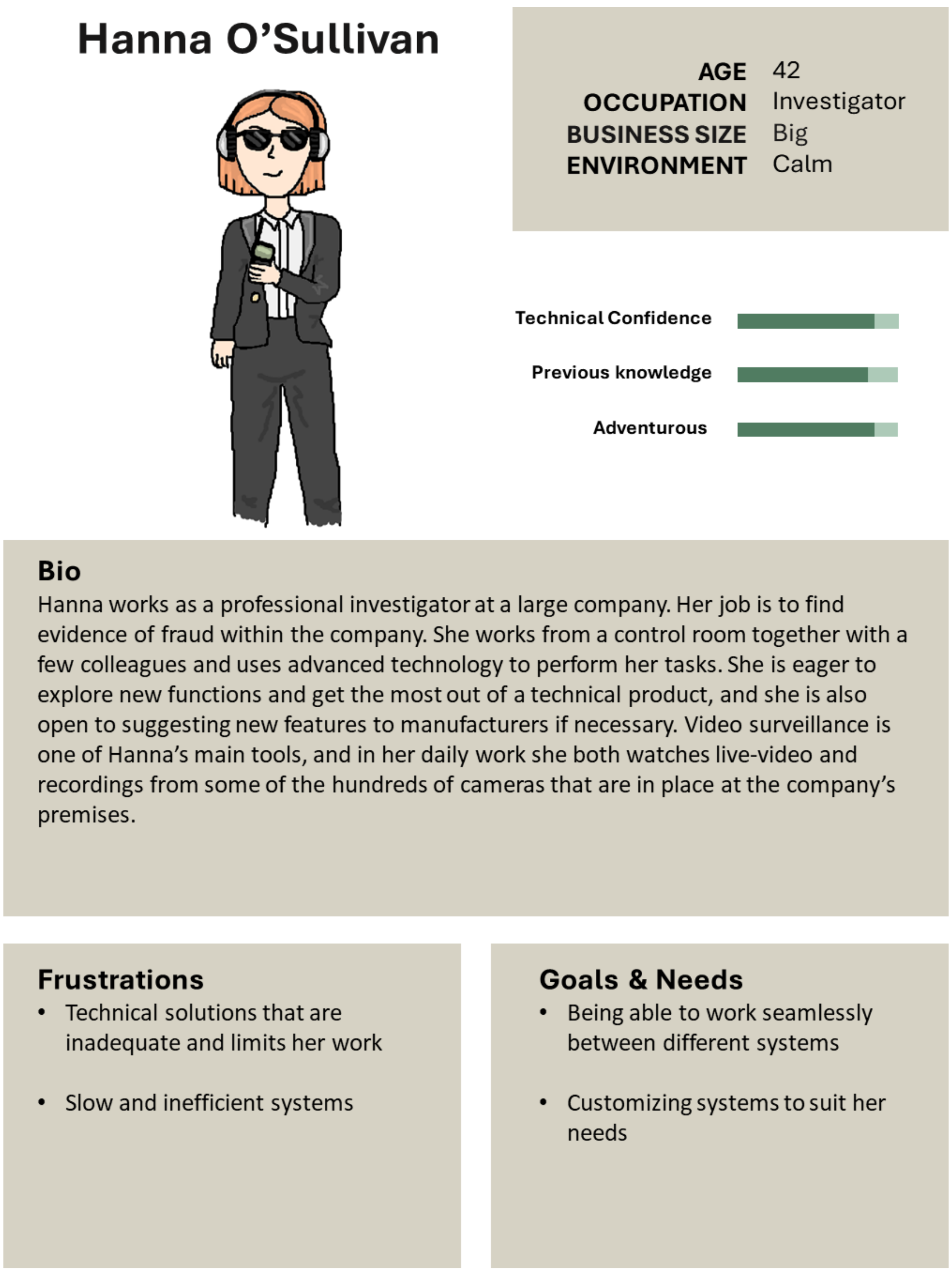
Technical Confidence
Previous knowledge
Adventurous

Bio
Thomas has been the owner of a local grocery store for many years. The store is a small family business that he started together with his younger brother. Thomas is energetic and genuinely cares about his business, not only because it pays the bills but also because it allows him to be social. He has made many friendships with long-term customers over the years and appreciates the daily encounters with them. However, he also has to deal with shoplifters and the risk of robbery. Being a long-time owner, he is used to this and has a video surveillance system installed. He is confident in using it for basic tasks and likes that it helps him protect his store, but he is not likely to explore new features on his own.

Frustrations
- Criminals that get away with their actions
- Limited space for administrative- and surveillance work

Goals & Needs
- Interested in solutions that drives his business forward
- Needs his employees to feel safe
- Willing to learn new things if it helps him reach his goals

Figure 4.1: The first persona: Thomas



Hanna O'Sullivan

AGE	42
OCCUPATION	Investigator
BUSINESS SIZE	Big
ENVIRONMENT	Calm

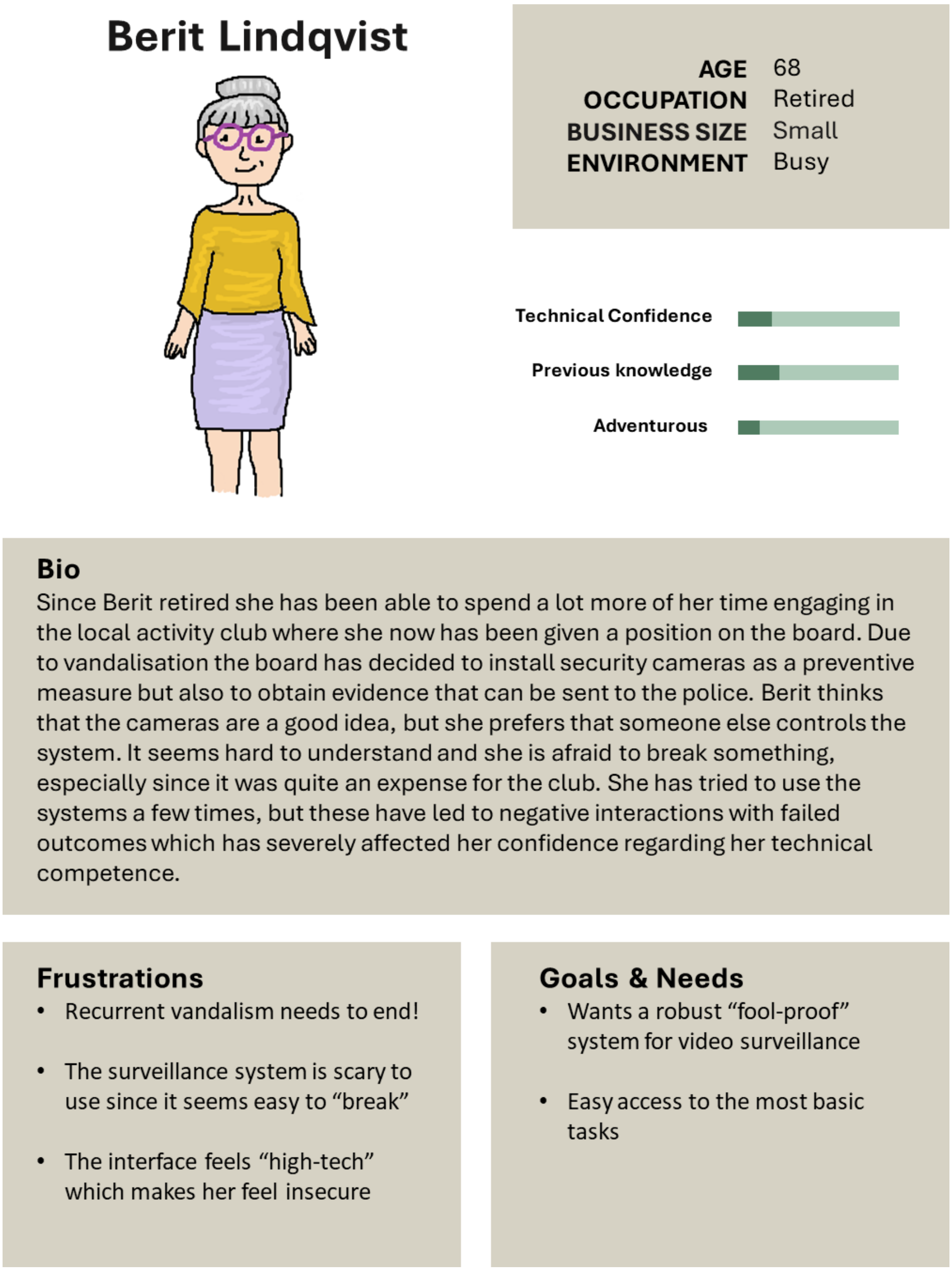
Technical Confidence
Previous knowledge
Adventurous

Bio
Hanna works as a professional investigator at a large company. Her job is to find evidence of fraud within the company. She works from a control room together with a few colleagues and uses advanced technology to perform her tasks. She is eager to explore new functions and get the most out of a technical product, and she is also open to suggesting new features to manufacturers if necessary. Video surveillance is one of Hanna's main tools, and in her daily work she both watches live-video and recordings from some of the hundreds of cameras that are in place at the company's premises.

Frustrations
- Technical solutions that are inadequate and limits her work
- Slow and inefficient systems

Goals & Needs
- Being able to work seamlessly between different systems
- Customizing systems to suit her needs

Figure 4.2: The second persona: Hanna



Berit Lindqvist

AGE	68
OCCUPATION	Retired
BUSINESS SIZE	Small
ENVIRONMENT	Busy

Technical Confidence
Previous knowledge
Adventurous

Bio

Since Berit retired she has been able to spend a lot more of her time engaging in the local activity club where she now has been given a position on the board. Due to vandalisation the board has decided to install security cameras as a preventive measure but also to obtain evidence that can be sent to the police. Berit thinks that the cameras are a good idea, but she prefers that someone else controls the system. It seems hard to understand and she is afraid to break something, especially since it was quite an expense for the club. She has tried to use the systems a few times, but these have led to negative interactions with failed outcomes which has severely affected her confidence regarding her technical competence.

Frustrations

- Recurrent vandalism needs to end!
- The surveillance system is scary to use since it seems easy to "break"
- The interface feels "high-tech" which makes her feel insecure

Goals & Needs

- Wants a robust "fool-proof" system for video surveillance
- Easy access to the most basic tasks

Figure 4.3: The third persona: Berit

4.1.2 Observation & Interview

Theory

Interviews are a common technique used to gain knowledge about the users. However, they might be insufficient on their own. In interviews the users talk about how they do things, but in observations they can show how they actually do it [4]. Sometimes there can be quite a difference between the two. Therefore observations are a good complement to interviews in the pursuit of truly understanding the user. There are several different observation techniques available, one of them being the think-aloud-technique [4]. In this case the user will complete a task while they explain outloud what they are doing. It is important to capture where the problems and friction arise during the interaction with the system. Things to look for are the order in which things are done, required cognitive activities, how information is presented, if there are any workarounds, etc.

Workflow

We had the opportunity of meeting a real user, and decided to arrange an observation followed by a semi-structured interview. This operator could be categorized as a "Thomas", who works at a grocery store and uses the surveillance system regularly to control incidents in the store. Since this operator is used to the system we suspected that he could work in the system quite efficiently. Therefore, we chose think-aloud as the observation technique in order to capture the workflow.

The aim of the observation was to gain further insight into how Axis' customers work and interact with the system. Therefore, we prepared a task that was supposed to symbolize a common activity, i.e. a fictive scenario, in the real working environment. This task was to find a certain incident in the store's recorded footage, with the help from a tip. The tip was that around a certain time, a girl in beige jacket would enter the store and pick up some tea. By the tea shelf, she would put on a hat and then take it off again. The intention was that this movement would mimic the movement of a shoplifter. The task for "Thomas" was to find this incident with the help of ACS, while explaining his work by thinking aloud. After the observation, a semi-structured interview was held for further analysis. The insights gained from the observation and the interview were summarized and divided into different categories deemed fit, in order to improve the overview of the information. These categories are presented below:

Representation. The task provided was said to be a good representation of how ACS is normally used. However, the information regarding an incident might be more or less specific according to the interviewee. This time the area of the incident was known, hence the starting point of the search was known as well. However, this is not always the case. A person acting suspicious needs to be followed by the surveillance system for the duration of their visit in order to determine whether they stole something or not.

Number of cameras. To be used as evidence, the video material must cover the entire time period from the moment the thief picks something up, until they leave the store. This requires a lot of cameras. In our case, where the "thief" walked straight from the tea-shelf to the checkout, numerous cameras were required in order to capture the entire lapse. This involves a lot of flickering between different views, and a lot of processing in order to export the sequence.

Value. To gather and export evidence is time consuming, so it was explained that it needs to be worth the time. If the information regarding an incident is too unspecific it will be too inefficient to pursue the matter further.

The interviewee also highlighted a more extensive problem related to the value of a surveillance system at a community level. Even if there is video evidence of the entire lapse of a theft, and the thief is caught in the act, it does not always lead to conviction due to limited police resources. This makes the time consuming work of gathering and exporting the evidence, writing the report to the police, as well as the surveillance system in itself, rather meaningless. This is of course a huge problem which needs to be solved at a community level, but it is nonetheless an important issue to consider.

Secure entry. The system is not only used for video surveillance. It also includes the feature *Secure entry*, where ports and doors can be opened remotely with a phone. However, this process was perceived as quite inefficient by the interviewee, since it involves many steps and clicks. This feature could benefit from an automated workflow that could be activated using voice control.

Attitude. ACS is not the main focus for the end user. This might seem obvious, but as a developer it is important to remember that ACS is merely a tool for the end user. Therefore it is important that the system is effective and performs well. New features, such as voice control, will not be used if they do not perform well enough or add some kind of value to the user experience. However, it was explained during the interview that voice control as concept was not perceived as something deterrent. The attitude was rather neutral. The main focus was whether it would be useful or not.

Environment. Another factor to consider is the environment where ACS is used. In this case it was an office where four other persons were working, which made it a rather cramped and noisy space to work in.

Concentration and memory load. Exporting a certain video sequence requires some effort from the user. In the task provided, the interviewee had to search for a thief in the recordings based on a certain event. Thereafter he had to follow this person through the store until they left. This task required high focus, and he had to fixate his eyes while looking through the recordings. When he later wanted to export the recordings, he needed to remember the time stamps from when the thief was leaving camera 1, which would be the time they entered the view of camera 2. Hence, even central tasks in ACS can be demanding in regards to concentration and memory.

4.1.3 Conclusion of user insights

The insights learned through the user research process were summarized into a mind map as seen in figure 4.4. This format enabled us to get a better overview and clarity of what should be considered when moving into the next phase of the design process. The insights were divided into the categories *users, tasks, goals, attitudes* and *conditions*. In *users*, the two main types of users (system integrators and operators) are found, as well as the personas Thomas, Hanna and Berit. Out of these three, the main focus going forward will be on Thomas. He represents a large user group and has an intermediate knowledge of ACS, meaning he confidently uses the basic functions of the system. The core functions like live viewing, looking at recordings and exporting are important for all users, so by helping Thomas, we hope to help all user groups. The *tasks* category includes the main tasks that users perform and the most commonly used functions in ACS. It can be seen that replaying footage and exporting it, as well as watching live video and controlling doors are common tasks. Further, adding bookmarks with comments, and scrubbing exemplifies the usage of individual functions to complete the overall tasks. The *goals* category focuses on what users want to achieve with the system. For most users, technology is simply a tool that can be used to achieve a particular goal, like security. They invest in technology with the expectation that it will deliver value to their business or organization. An investment in surveillance technology is both expensive and time consuming, and this needs to be weighed against the positive outcomes. Further, *attitudes* can vary greatly between users. It should be kept in mind that not all users are voluntary users, and that systems like ACS can appear daunting to some. Lastly, the *conditions* category is a collection of insights about the context in which ACS is used and what the prerequisites of the users are. Many of these insights were learned through a semi-structured interview with a system expert at Axis. A video surveillance system includes a lot of cameras, which results in many different views to switch between. This combined with the fact that the goal often is to look for details in the footage means it is visually intensive to work with ACS. In addition, the spaces housing the system are often small offices where other tasks are performed as well. Finally, most users are reactive, meaning that they use ACS when something has happened. It needs to be considered that these can be stressful scenarios, especially if the usage is otherwise infrequent.

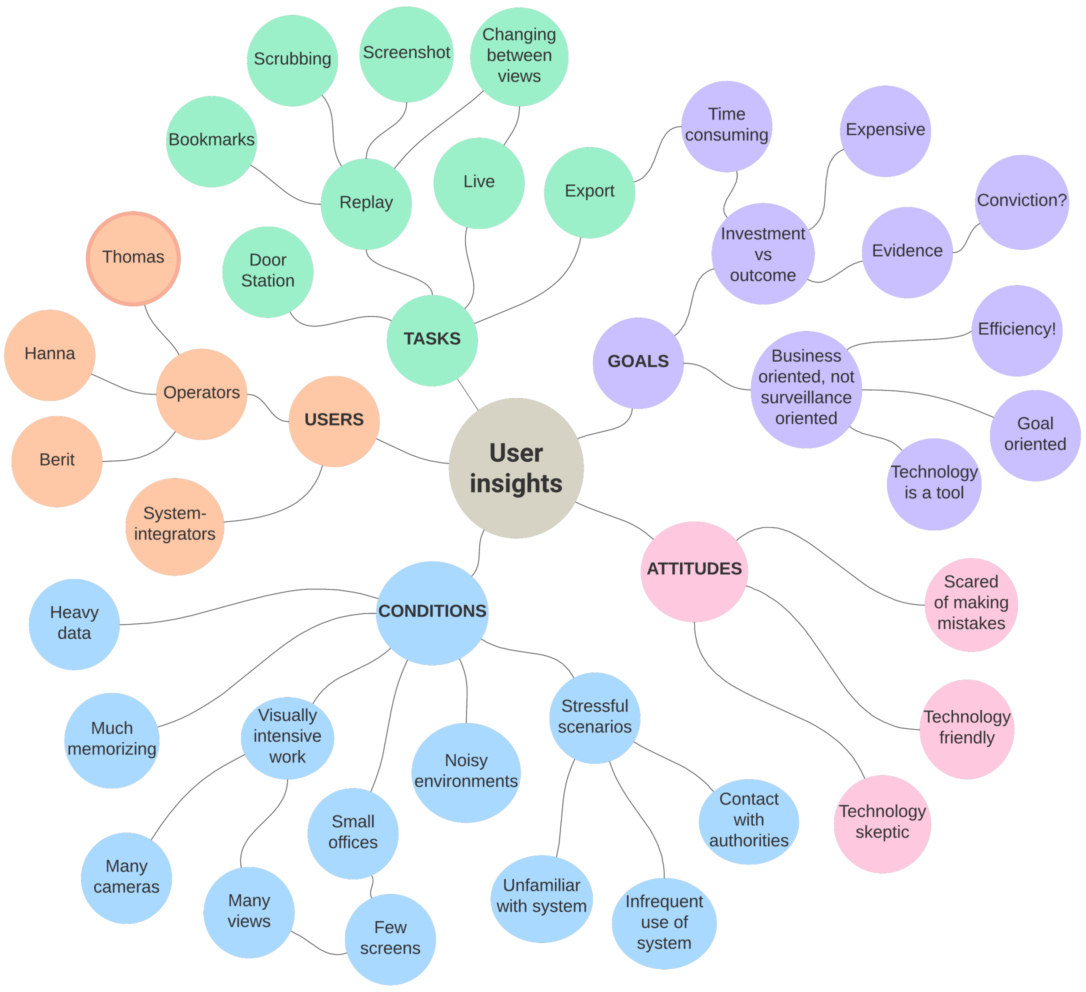


Figure 4.4: Mind map of user insights

4.2 Conceptual design

The conceptual design process was initiated by a brainstorming session focusing on listing all functions that could be controlled by voice. This was followed by the creation of concept videos, which were used to communicate to each other how we imagined the integration of voice control in ACS. The discussions that followed resulted in the establishment of a few general design requirements.

4.2.1 Brainstorming

Theory

Brainstorming is a method that can be used to generate many ideas in a short amount of time [4]. An important principle during brainstorming is that no idea should be discarded or criticized. Crazy ideas should instead be encouraged since the goal is to generate many ideas. The evaluation should take place afterwards.

Workflow

We chose to use brainstorming as a method to generate ideas of how voice control could be used in ACS. The brainstorming was centered around the question "which actions/functions could be controlled by voice?". We gave ourselves about 10 minutes to write down and come up with ideas. Thereafter we discussed our ideas and compiled them in an affinity diagram which can be seen in figure 4.5. The diagram served as foundation and inspiration to succeeding activities in this process step.

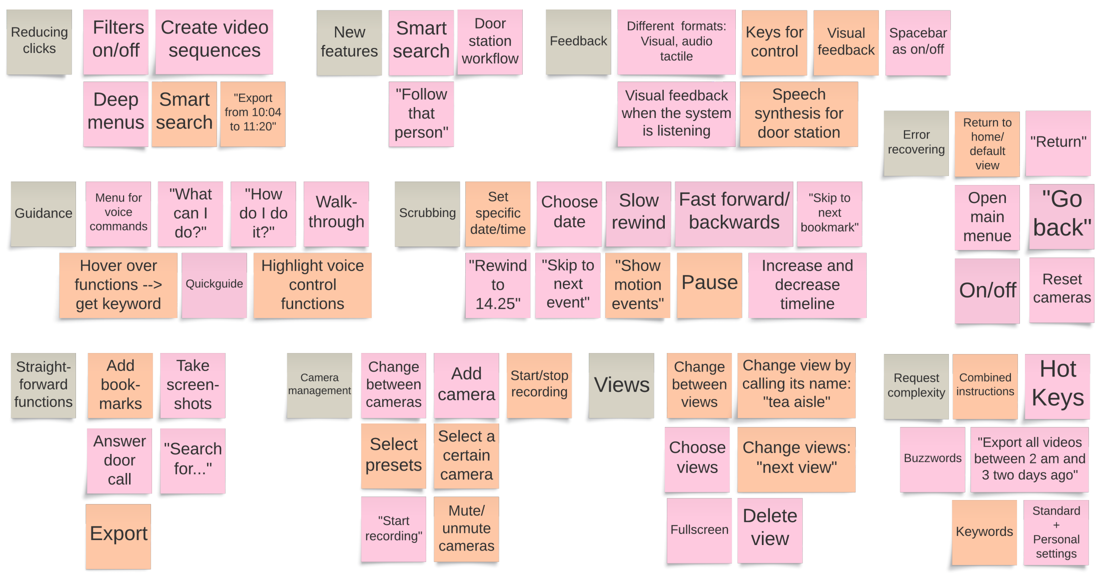


Figure 4.5: Affinity diagram from brainstorming

4.2.2 Concept videos

Theory

A prerequisite for developing a conceptual model of the design is that the project members have the same perception of it. In other words, the mental models of the members within the project group should be aligned. A mental model is the internal perception of how an object works [22], and this affects the way one makes predictions about how to interact with it.

Workflow

To communicate our mental models to each other, each project member was given the task of creating a concept video representing their perception of how the voice control solution should function. The essence of these videos were captured in an image series format, see figure 4.6 and figure 4.7. The first video focused on demonstrating how the integration between manual commands and voice commands could work. The idea was that voice commands could function as a complement to the manual maneuvering, especially when doing visibly intensive work like scrubbing to find specific details in the video. In addition, it showed how the system could help the user discover functions supported by voice control. The focus area for the second video was to demonstrate how voice commands could be given and how the user could get feedback on how they expressed their instructions. The idea was that voice commands should be built upon hot keys, i.e. keywords that the user could define. The feedback should be displayed visually in the user interface and provide continuous information of the state of the video management system. The user should know when the system is listening and when it is processing a command, as well as how a spoken command was interpreted by the system.

While some details of our demonstrations slightly varied, the overall concept was concluded to be very similar. Most differences could simply be explained by the fact that we focused on different aspects of the voice control solution in our videos. This turned out to be advantageous going forward, since most aspects had already been addressed to some extent.

4.2.3 General design requirements

As a final step, we summarized our concept videos and insights from our user research into some general design requirements. These served as guidelines for what we wanted to achieve with the solution. First of all, it was decided that the voice control solution should be implemented in Swedish. Overall we wanted the system to be managed with short voice commands that could be used as an alternative to manual commands. We wanted to achieve a smooth integration between the voice commands and the manual commands in order to obtain a seamless experience for the user. Hence, the resulting product should have two different interaction types, manipulating and instructing [23]. It should be easy to switch between and work with both alternatives. An example of this would be giving voice commands of a certain date when looking at recordings (instructing), while fine tuning the rewinding by dragging the time marker by hand (manipulating). In addition, we aspired to build the voice control solution as an add-on that can be turned on or off depending on the user's needs. We aimed to support the most common workflows in ACS.

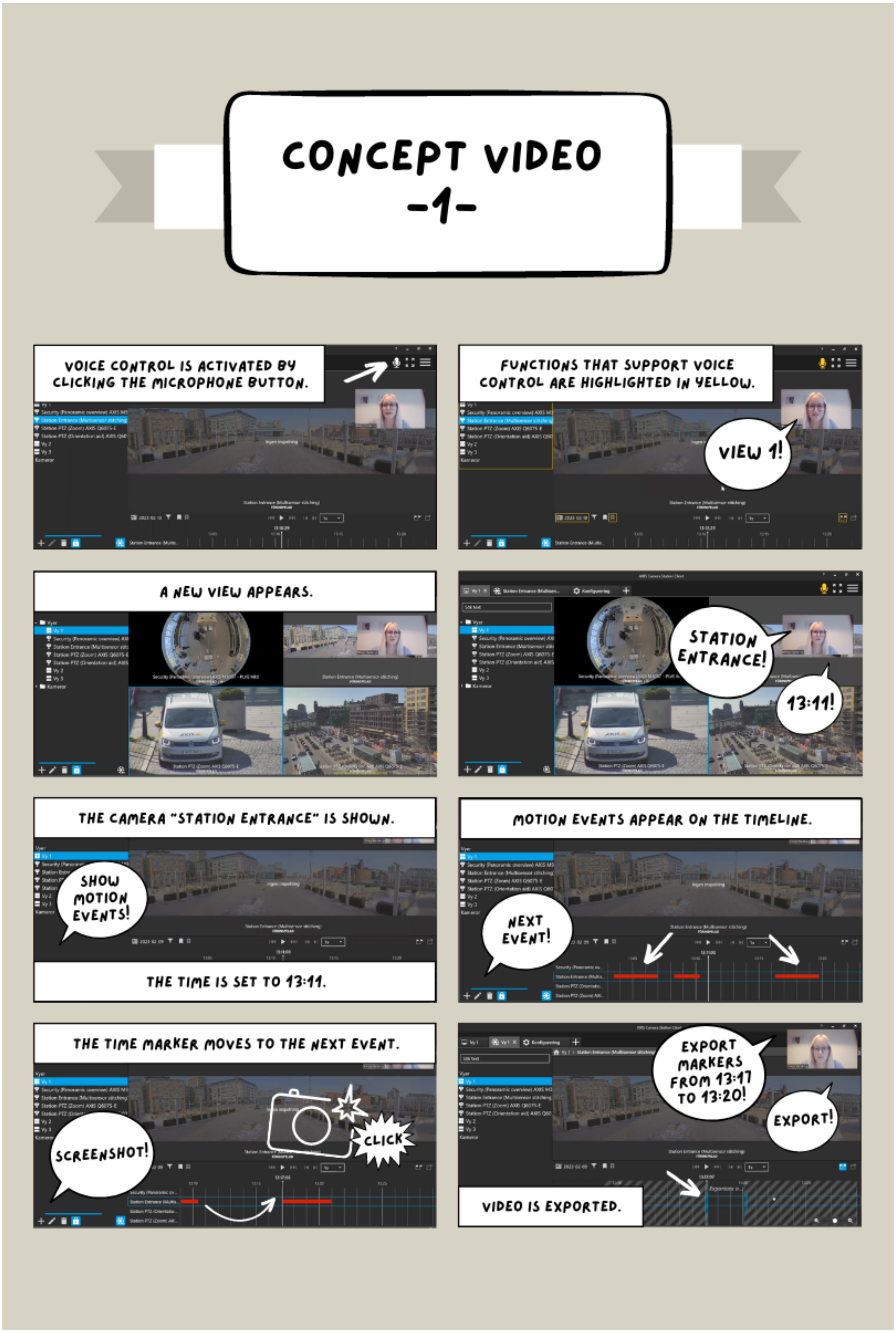

CONCEPT VIDEO -1-
VOICE CONTROL IS ACTIVATED BY CLICKING THE MICROPHONE BUTTON.
FUNCTIONS THAT SUPPORT VOICE CONTROL ARE HIGHLIGHTED IN YELLOW.
VIEW 1!
A NEW VIEW APPEARS.
STATION ENTRANCE!
13:11!
THE CAMERA "STATION ENTRANCE" IS SHOWN.
SHOW MOTION EVENTS!
THE TIME IS SET TO 13:11.
MOTION EVENTS APPEAR ON THE TIMELINE.
NEXT EVENT!
THE TIME MARKER MOVES TO THE NEXT EVENT.
SCREENSHOT!
CLICK
EXPORT MARKERS FROM 13:17 TO 13:20!
EXPORT!
VIDEO IS EXPORTED.

Figure 4.6: Concept video by Ellinor

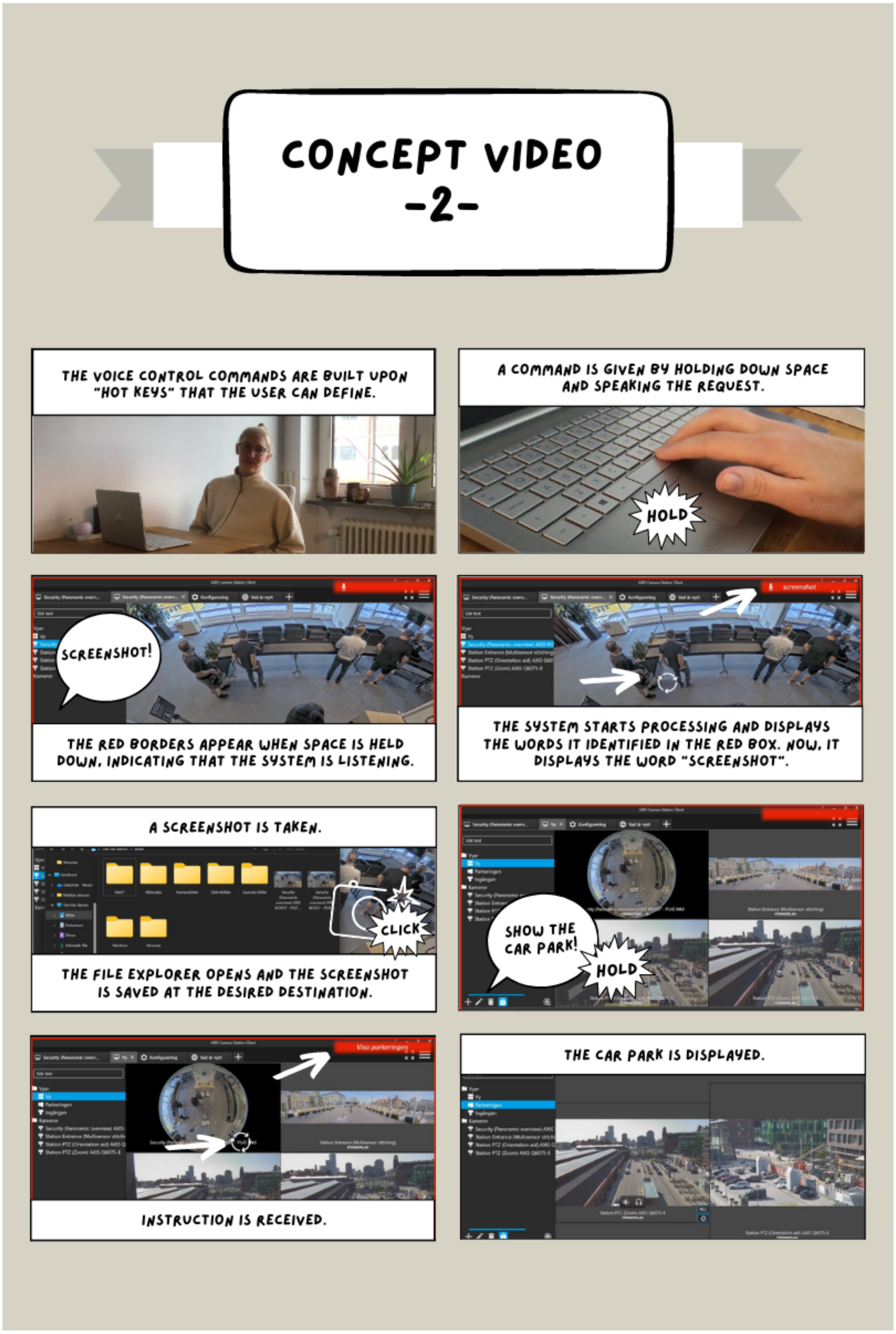

CONCEPT VIDEO
-2-
THE VOICE CONTROL COMMANDS ARE BUILT UPON "HOT KEYS" THAT THE USER CAN DEFINE.
A COMMAND IS GIVEN BY HOLDING DOWN SPACE AND SPEAKING THE REQUEST.
HOLD
SCREENSHOT!
THE RED BORDERS APPEAR WHEN SPACE IS HELD DOWN, INDICATING THAT THE SYSTEM IS LISTENING.
screenshot
THE SYSTEM STARTS PROCESSING AND DISPLAYS THE WORDS IT IDENTIFIED IN THE RED BOX. NOW, IT DISPLAYS THE WORD "SCREENSHOT".
A SCREENSHOT IS TAKEN.
CLICK
THE FILE EXPLORER OPENS AND THE SCREENSHOT IS SAVED AT THE DESIRED DESTINATION.
SHOW THE CAR PARK!
HOLD
Visa parkeringen
INSTRUCTION IS RECEIVED.
THE CAR PARK IS DISPLAYED.

Figure 4.7: Concept video by Matilda

The solution should also consider feedback, guidance and error prevention. It must be clear whether the system correctly interpreted the voice command, if the specified command exists and if the system heard the right thing. It must also be clear how the system responds to a command so the user knows if it has been executed or not. We also wanted to achieve explicit feedback regarding the status of the system. It should be clear whether the system is listening, processing, etc. Here, visual feedback was of great interest.

4.3 Implementation: Proof of Concept

In the previous process step we defined some goals and guidelines of what we wish to achieve with the system and how it should work conceptually after some iterations. The goal for the initial implementation was to implement a few important core-functions with focus on building a good structure. This was done with the intention of facilitating the work for further expansions, both in number of commands and in complexity.

4.3.1 Technical approach

The first step was to find a library for speech recognition. After some research we found a library from Microsoft called Speech Service [15], which offers methods to translate speech into text. We chose to start with this library since it seemed quite robust and relatively accurate. It also supports several languages, which is useful if we want to use a mixed set of Swedish and English keywords as commands. We downloaded the library into Microsoft Visual Studio, which was the environment we worked in. After a first setup we were able to translate spoken words into a string which could be further processed to make the system perform a certain task.

ACS is a large system with complex code, and a long learning period is required before new developers are able to understand and work with it. Since this thesis was limited to 20 weeks, we did not have the time to make a fully integrated solution to the current system. Instead we chose an approach were we built a stand-alone solution which could work more as an add-on to ACS. Our solution would communicate with ACS through *deep links* in order to control and manage the client. A deep link is a link that directs the user to some content inside an application or website [1]. Hence, it was possible to perform some tasks through deep links such as change between cameras, change between views, go to a chosen timestamp in recorded material, etc. The advantage of this approach was that we could build a system with core functions quite quickly since some deep links had already been created by Axis employees. The disadvantage was that we were limited to a one-way communication with ACS. Our solution could send information to ACS, but the information that ACS could send back was limited. However, since it left us with enough flexibility to implement the core functions, we decided to continue with this approach. This meant that we now had a tool to receive speech as input, and a tool to manage and give instructions to ACS. The next step was to implement the connection between the two.

4.3.2 From speech to action

We decided to manage ACS through a number of keywords, see table 4.1. In this way a certain keyword was connected to a certain task. A key-value dictionary was created where the keywords and their corresponding deep links were listed. Next, the speech input was scanned and if it contained any of the listed keywords, a process would start where ACS navigated to the related deep link. This was straightforward in some cases where we could use already complete deep links such as navigating to the configuration page. However, other cases required some further processing. For example, in order to change between different cameras we had to supply additional information in the deep link such as identification number of the chosen camera. This information could be collected through an API request that returned a list of available options. This enabled us to build complete deep links that could be connected to a keyword i.e. the name of each camera. This made it possible to change between different cameras simply by saying their name. We continued to implement a few more core-functions such as rewinding, shifting between live and recorded material, changing between views, etc. This was overall done in the same way as the function for shifting cameras, by collecting the needed information and building a deep link that would make ACS navigate to the desired page. The main focus was to build a good initial structure and to cover core functions that are essential in ACS. A schema of the structure is presented in figure 4.8. We also started to consider how feedback could be implemented to facilitate the usage. The aim was to produce an effective system where short commands can be given rapidly. Instead of having continuous input, we implemented a key control to gain control of when the system should and should not be listening. When a key is pressed on the keyboard, the system starts receiving input. This was considered more favourable than using a wake word since it would allow for quicker input. Repeatedly pressing a key feels more natural than repeatedly using a wake word during rapidly given commands, and it is also more efficient. As a temporary solution we also implemented some visual feedback in the console window from where we could manage the voice control solution. The text "Press key and speak into your microphone" will be shown when the solution is ready to listen to new speech input, and the text "...Listening..." will be shown when the system actively registers the input. It will then display the recognized input.

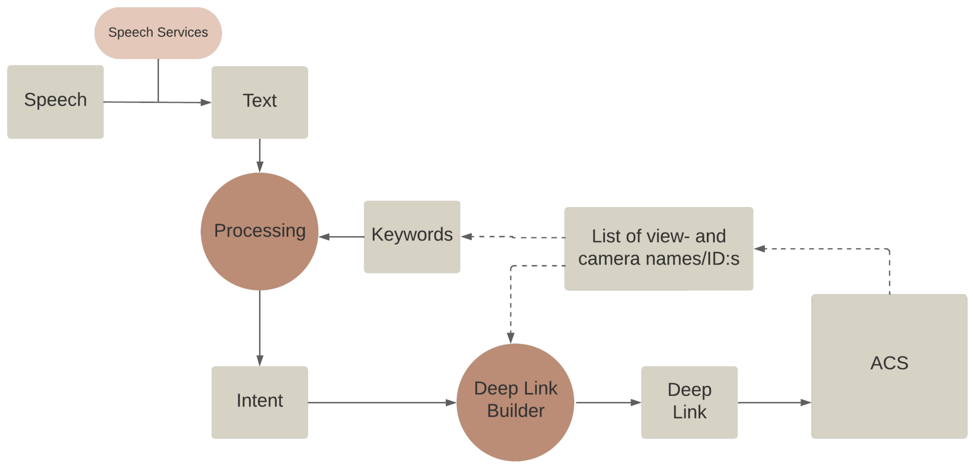


Figure 4.8: Schema of structure

Table 4.1: Overview of the solution

Function	Description	Keyword
Key control	Press key to give commands	
Configuration	Change to configuration page	Configuration
Cameras	Change between cameras	Name of camera
Views	Change between views	Name of view
Live	Change to live view	Live
Recordings	Change to recordings	Recording
Scrubbing	Select timestamp for video	Rewind
Date	Select date for video	Date
Feedback	**Description**	
Ready for speech input	Text in console window	
Listening	Text in console window	
Recognized words	Text in console window	

4.3.3 Evaluation

Reflections

After the first round of implementation work, we took some time to reflect in order to evaluate the work so far. These reflections were our own thoughts and experiences related to different areas of the solution.

Flexibility. We had already achieved some flexibility in the system since we chose to build a structure that scanned the input for keywords. This meant that as long as the recognized input contained a keyword, the command would work despite what other content the input contained. This made the system less fragile in noisy environments. However, it also made it gullible, since it did not interpret the meaning of the input. The structure enabled some additional flexibility since we could add several synonyms as keywords for the same function. However, this had to be done manually, and the solution required that the exact word existed in the list of keywords. This meant that different inflections of a command would not work if they are not listed as keywords. A limitation in flexibility was the way that instructions for different timestamps and dates had to be given. Currently, a date had to be given as the number of the month, followed by the date. For example, the twelfth of February had to be given as "02 12". A certain timestamp had to be given sequentially, starting with the hour followed by the minutes and seconds. This might not feel natural to all users and may result in a longer learning curve.

Feedback. A simpler form of feedback was given through the console window. As previously presented, this was text-based feedback that indicated when the system was ready to receive a new command, when it was listening, and when it was done listening. It also reported what words it actually recognized. This type of feedback served as an important tool during the development process to facilitate debugging, etc. However, we wanted to keep the concept for the end user as well since it facilitates the learning of how the user needs to formulate their commands, and how clearly they need to articulate their speech for the speech recognition to work.

Performance. At this point, the system worked rather well. The time it took to process the speech and send the instructions to ACS was short enough to feel effective, less than one second. However, it might differ in speed on different computers.

Limitations. There was, as previously mentioned, some limitations in the chosen approach. This was a stand-alone solution which made it quick and easy to implement, but also limited in some areas. For example, we could not make changes to the current interface of ACS in order to make it more customized to voice control. We were also limited to the, at the time, existing deep links, since it would take time and extensive comprehension of ACS to develop new ones. This limited which functions we could implement.

Since this approach was more or less limited to a one-way communication, we had some problems regarding missing information. For instance, our solution would only have information that had passed through the speech recognition (apart from the camera- and view id:s). This meant that if you changed camera in the recordings by hand and then decided to give a voice command to change the timestamp, our solution would not have the information that the camera had been changed and would hence show the requested timestamp, but for the camera that was last mentioned by voice. As a consequence, it was difficult to achieve the seamless experience between the manual and the voice interface with this approach. It is also worth mentioning that ACS would open a new tab for each invoked deep link. This meant that a new tab would open every time a new camera, timestamp or date was requested, resulting in a great number of open tabs. This was not a limitation for conceptually showing how voice control could be used in ACS and was subsequently left in its current state, but it was worth declaring for future reference.

Functions. We had chosen to implement a few core-functions that could be considered central to the system. As long as the user was working with the most common flows the voice control could be considered relatively comprehensive, but outside of this scope it was not particularly integrated. Some continued exploring of which functions that were suitable to control by voice was hence needed. The key-control was currently implemented to give the user control over the system. It might be worth to examine how it effects the cognitive load since it forces the user to switch between mouse and keyboard.

General comments. Our own experience of the current solution was that it was relatively easy and comfortable to use. The solution felt more natural and safe to use with increased experience, especially regarding choice of dates and timestamps. The user did not have to search in the graphical interface in order to navigate which reduced the cognitive load, as long as they have good knowledge of their views and cameras. However, we might had some knowledge bias since we were the ones developing the system.

Demo

The initial solution was evaluated through our participation in a demo session at Axis. The goal of our demo was to investigate how our knowledge bias affected the perceived ease of use of the system, and to gain insight into how the voice control solution could be made stable enough for further testing. The agenda for the demo was as follows:

Brief introduction. The audience, consisting of around ten ACS colleagues, was given an overview of the voice control project. The majority of them had little or no previous knowledge of what we were working on.

Demonstration. A demonstration of the working voice control solution was performed. The audience got to see a workflow that included shifting between views and cameras, changing the date and time, and navigating between live video and recordings. This was done without any further explanations of how the system worked or what kind of commands it accepted.

Participant testing. One person from the audience was asked to try to finish a pre-determined task in ACS, using only voice control. The instructions for the task specified the state (recording), camera ("Station"), date (2023-02-20) and time (10:23), which this test participant was meant to navigate to.

Discussion & feedback. The audience was asked to come up with synonyms to the keywords used, as well as other functions that could benefit from voice control. There was also time for a more unstructured discussion and feedback.

So far, the solution accepted a very limited amount of keywords, and it was not flexible enough to accept variations like grammatical tense. We believed that this limitation would make the system quite unintuitive, and that it would lead to a lot of errors when tested by someone who had only gotten a short demonstration. This was considered to be something positive, since it would give an insight into what the error handling could look like. For example, it could provide us with more synonyms to keywords. However, the test participant carried out the task perfectly. The feedback from the audience also suggested that the learning curve was quite short, and the keywords were perceived to be intuitive. The only context in which more variations were clearly requested was for choosing the time and date. They wanted to use formats like "15th of february", "yesterday", "thursday" and "ten o'clock".

4.4 First Iteration: Expansion and Stability

The insights from the initial implementation and its' evaluation step, laid the foundation for the first iteration. The main focus for this process step was to improve and expand the solution. The improvement revolved around a number of bugs that were found during testing. For example, the solution mistakenly changed the camera-view when the user shifted between live and recordings for a certain camera. A few other bugs similar to this were resolved in order to obtain a more stable solution with a reliable performance, so that future observations could be focused on the user experience rather than the performance of the system.

The expansion of the system revolved around a core-function that could not be implemented in the first round of implementation due to lack of time, namely the export. This was implemented in a way that aimed to provide quite a flexible usage. For example, the user could choose to complete the entire workflow by voice, by first setting the start- and stop time for the video sequence and then exporting it. Another option would be to give the export command first and then manually drag the export markers to define the start- and stop time. The second option is an example of how the solution makes it possible for the user to combine different interaction types to suit their preferences.

In addition to the export function, a few other functions were introduced, and some were improved. Table 4.2 presents an overview of the current solution. As a response to some of the continuous feedback that we got from colleagues, it was made possible to go back to the previous command. This was implemented to work like an undo function. Another addition to the solution was the reset command, which was supposed to help with error handling. After zooming in on or moving a camera, there is no efficient way of manually going back to the default view. The reset command solves this by providing a direct way of instructing the system to go back to default mode.

Table 4.2: Overview of the solution

Function	**Description**	**Keyword**
Key control	Hold key to give commands	
Configuration	Change to configuration page	Configuration
Cameras	Change between cameras	Name of camera
Views	Change between views	Name of view
Live	Change to live view	Live
Recordings	Change to recordings	Recording/Recordings
Scrubbing	Select timestamp in video	Rewind/Time/Go to
Now	Select present time in video	Now
Date	Select date in video	Date
Today	Select today's date in video	Today
Export	Export video sequences	Export
Export marker, start	Set the start marker for export	Start/Start time
Export marker, stop	Set the stop marker for export	Stop/Stop time
Return	Undo most recent command	Return
Reset	Go back to default camera mode	Reset
Feedback	**Description**	
Ready for speech input	Text in user interface	
Listening	Text in user interface	
Recognized words	Text in user interface	
Error message	Text in user interface	

There were some improvements made relating to keyword flexibility and user control. With the added flexibility, the user could define dates in several new formats. For example, the date 03-15 could now be defined as 15th of March or 15/3. In addition, the keywords "today" and "now" for date and time were added, and the supported time formats were expanded so that the solution could understand times given as an hour, hour-minutes or hour-minutes-seconds. Regarding user control, the main goal was to change the way the speech recognizer was controlled practically. Previously, the speech recognizer was turned on by a simple key press and turned off by silence. The new idea was to let the user hold the key down for the duration of the spoken command, meaning that the speech recognizer would be turned off by the release of the key. The reason for implementing two versions of the control was to interest to test them against each other at a later stage. Furthermore, we needed to address an issue where the console window stopped detecting key presses when it became unfocused. This happened when the user clicked outside of the console window, for example somewhere in ACS. By resolving the issue, it became possible to demonstrate a seamless integration between manual interaction and voice control. However, the implementation of both these improvements turned out to be technically challenging. To get access to the functionality needed, we had to go from a console application to an application with a user interface (UI) [16]. While this was time-consuming, it did provide us with a good foundation for developing visual feedback via the UI. Visual feedback was originally planned to be the theme of the next iteration, but since the UI was already an integrated part of the new application, we had to transform the already existing console window feedback into feedback presented in the UI. However, in this iteration the feedback and design were kept simple and minimal.

4.4.1 Evaluation

The purpose of this evaluation was to examine how well the improved voice control solution functioned when accompanied by simple feedback. We expected insights about how a user would explore the voice control feature, and what the challenges are for a person learning how to use it. By observing explorations, assigning the participants specific tasks, and having a discussion about how different keywords can be interpreted, we also expected to collect more information about desired functions and keyword synonyms. These insights were supposed to contribute to the knowledge of how to make the solution more flexible and intuitive.

Test structure

This evaluation step consisted of one large test divided into three smaller parts. The test was conducted in Swedish with two participants, both colleagues, one at a time. They both had previous knowledge of ACS. The following is a description of the different parts of the test:

Free exploration. The participant was asked to freely explore the voice control solution by managing ACS solely by voice. The goal was to find as many functions as possible that were supported by voice control. The participant was encouraged to keep trying for at least three minutes, but was allowed to go on for a longer time.

Context-based test. The participant was given smaller tasks to complete in a way that felt intuitive to them. The first task involved returning to the default view after having zoomed in on a camera. The second task touched upon rewinding and fast forwarding in the video material. The third task was to go back or undo an action in the contexts of navigating cameras and views, selecting dates and exporting video sequences.

Keyword discussion. The participant was asked to give an explanation to what the Swedish equivalent of the following command words meant to them, and what result they would expect from using them: return, reverse, backwards, forwards, reset, undo, export (verb) and export (substantive).

Insights

The tests provided several valuable insights and highlighted some challenges regarding the solution. The first part of the test, the free exploration, showed the difficulty of finding the functions supported by voice control. From observing the participants, we deemed it counter-intuitive that some features were supported but others were not. In combination with the lack of feedback it was hard for the participants to determine whether a function did not exist or if they just gave the command incorrectly. This led to multiple attempts to perform a certain action before realising that the function was not supported. Hence, it was necessary to improve the discoverability and feedback of the solution in order to assist the user when they are learning how to use the system. This part of the test also led to a number of new, suggested functions that could be controlled by voice, such as zoom, play, and pause.

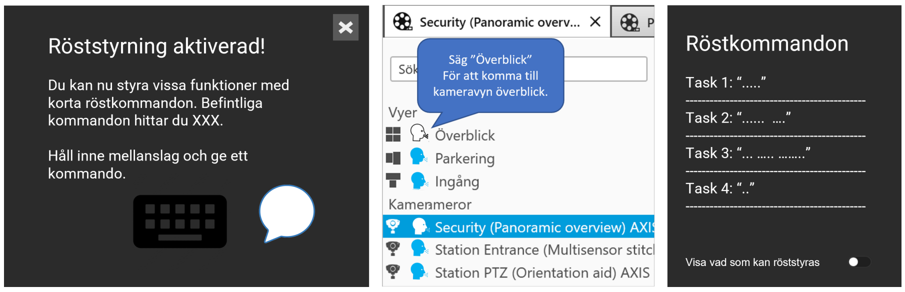
The second and third parts of the test showed the difficulty of choosing appropriate keywords for certain tasks and actions. Some words have a strong contextual meaning, which complicates the work of building an intuitive solution. For example, the word "back" could mean "rewind the video", but also "go back to the previous page". Another example was the complex task of distinguishing between dates and times if the user only provides a number such as "03:25". It is obvious for the user since they know what they are referring to, but it is impossible for the solution to tell if the user wants to change the date or the time. Some words could also have an entirely different meaning for different people depending on their background. For example, one participant could not find the word "reset" in order to go to the default view for a zoomed in camera. For this participant, the word "reset" had a much more drastic meaning, such as restoring the camera's configuration rather than just going back to the normal, zoomed out view. This shows that there was a fear of loosing control of the system, and that the user could be afraid of accidentally performing a task that would mess with the system's configuration. Thus, a solution built upon keywords rather than a complete speech interpretation comes with some challenges. It encourages the user to give short commands instead of normal sentences, which makes it more difficult to correctly interpret the user because of the missing context. However, short and fast commands could feel more effective than longer and slower conversing commands, but it requires the command to be given in a certain way and that the user is aware of this. Consequently, the effectiveness comes with a prize of a higher learning curve.

Another insight from the second part of the test was that none of the participants could finish the task where they were supposed to export a short sequence of footage. They were however able to finish the task with some guidance. This might imply that the export-function could be further refined in order to achieve a more intuitive workflow, but it could also be supported by a more extensive feedback and contextual guidance. The feedback in this version of the solution was limited, since this test focused on the strengths and weaknesses of the voice control feature without comprehensive graphical support. Thereby the feedback was confined to the recognized words, and a few number of error messages. However, the design did not encourage the user to make use of the feedback to improve the quality of their input or correct mistakes. The reason could be a combination of too generic error messages that did not supply any guidance, and a font that was simply too small and difficult to read.

Lastly, the overall performance of the solution was quite fast and stable. The speech recognizer misheard the input only a few times, and the solution worked as it was supposed to, just not as intuitively as intended. There were only a few mishaps that needed to be addressed in the following iteration.

4.5 Second Iteration: Graphical Support

The main objective of the second iteration was to explore and concretize the concept of visual feedback as a support to the voice control solution. This involved improving and expanding the existing visual solution, which was limited to simple feedback messages printed in a text box. The aim was to explore how different alternatives of graphical support could improve the user experience by improving discoverability, visibility, error recovering, feedback and guidance. In order to develop helpful graphical support, we performed a short brainstorming session within the project team centered around two questions: *what* can be controlled by voice, and, *how* should the commands be given? See section 4.2.1 for a more detailed explanation of the brainstorming method. The ideas were discussed and then illustrated with pictures and drawings in order to clarify the different suggestions. A handful of these can be seen figure 4.9, which contains drafts of a pop-up guidance window, a tool for discovering new functions, and a quickguide listing valid voice commands. The benefits and drawbacks of each suggestion were discussed and then further processed. Some ideas were scrapped and others were improved or combined. This resulted in seven distinct graphical support alternatives. When doable, these were implemented into the current solution, and otherwise they were visualized through images or slideshows. It was important to visualize these concepts both for ourselves and for the participants that would later evaluate the different options. The refined versions of the seven suggestions are presented and described below. While these were all made to look like real ACS features, it should be noted that they were actually placed as windows on top of the screen.


Röststyrning aktiverad!

Du kan nu styra vissa funktioner med korta röstkommandon. Befintliga kommandon hittar du XXX.

Håll inne mellanslag och ge ett kommando.

Security (Panoramic overv...

Säg "Överblick" För att komma till kameravyn överblick.

Sök

Vyer
- Överblick
- Parkering
- Ingång

Kamenmeror
- Security (Panoramic overview) AXIS
- Station Entrance (Multisensor stitc
- Station PTZ (Orientation aid) AXIS

Röstkommandon

Task 1: "....."

Task 2: "...... ...."

Task 3: "... ...."

Task 4: ".."

Visa vad som kan röststyras

Figure 4.9: Ideas from the brainstorming session

Voice control button. A button for activating/deactivating the voice control feature was placed in the top-right corner of ACS. The design of the button can be seen in figure 4.10, which also shows how it appears in different states. The first image represents the state of the button when voice control is deactivated. The second image shows how the button lights up when hovering on it, and the third image represents the activated state. The design of the button and its states was created in accordance with the visual design of ACS.

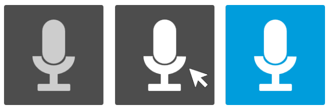


Figure 4.10: The different states of the voice control button

The reason for implementing an on/off button was that the solution required some space in the interface regarding text boxes for feedback etc. When voice control is not being used, it should not take up a lot of space since it might cover important information. It might also provide the user with a sense of control. Unless voice control is activated, they do not risk being recorded.

Text box. In terms of functionality, the original text box was kept. It was still used to indicate the state of the speech recognizer, to display error messages, and to display the recognized speech. How this information was presented in the text box was also further refined, for example by formatting times and dates. Error messages were also improved by making them more context based. For example, if the keyword for changing time was given, but the time format was incorrect, the error message would instruct the user how to give a valid time.

In terms of appearance, two new designs of the text box were created. Both were visually altered to blend in with the design of ACS, and fixated at their intended location. The text box in figure 4.11 was placed in conjunction with the on/off button in order to communicate that they were related. The advantage of this placement was that it did not cover any video content, however, it decreased the available area for open tabs. If the operator is working in a large amount of tabs, all of them would not be visible if the voice control feature is turned on. Thereby an alternative design was made, presented in figure 4.12. This text box was placed below the on/off button. The drawback of this design was that it covered some of the video area. Since this area can be split into several different parts, showing numerous cameras at the same time, the text box is at risk of covering a large part of one camera. This was resolved by increasing the transparency as well as introducing an animation. The text box will appear from the right when a command is given, and will disappear a few seconds after a given command. The intention was to minimize covering of the interface simply by making the text box visible only in concurrence with a voice command.

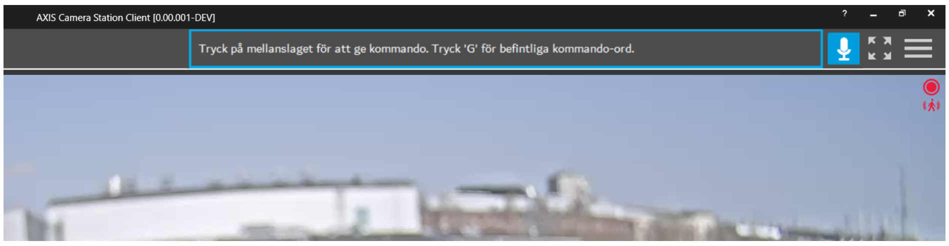

AXIS Camera Station Client [0.00.001-DEV]
Tryck på mellanslaget för att ge kommando. Tryck 'G' för befintliga kommando-ord.

Figure 4.11: Design alternative 1 for text box

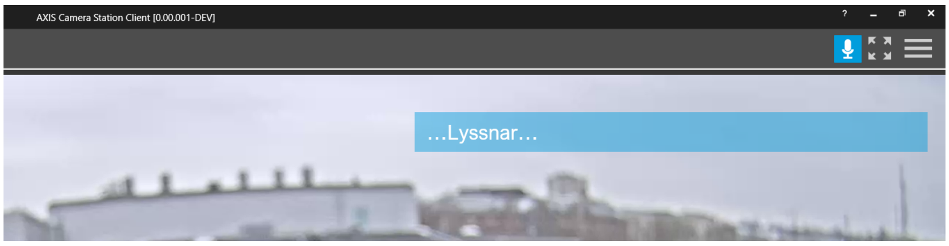


Figure 4.12: Design alternative 2 for text box

Discovery mode. The discovery mode was created with the intention of helping new users discover which functions are supported by the voice control feature. This mode is activated by default, but can be turned on/off. When voice control is activated, this mode highlights all functions in the ACS user interface that can be controlled by voice. When hovering over a supported function, the user can see which voice command is associated with it. This feature was created as a prototype, and figure 4.13 shows an example of what it could look like in dark theme.

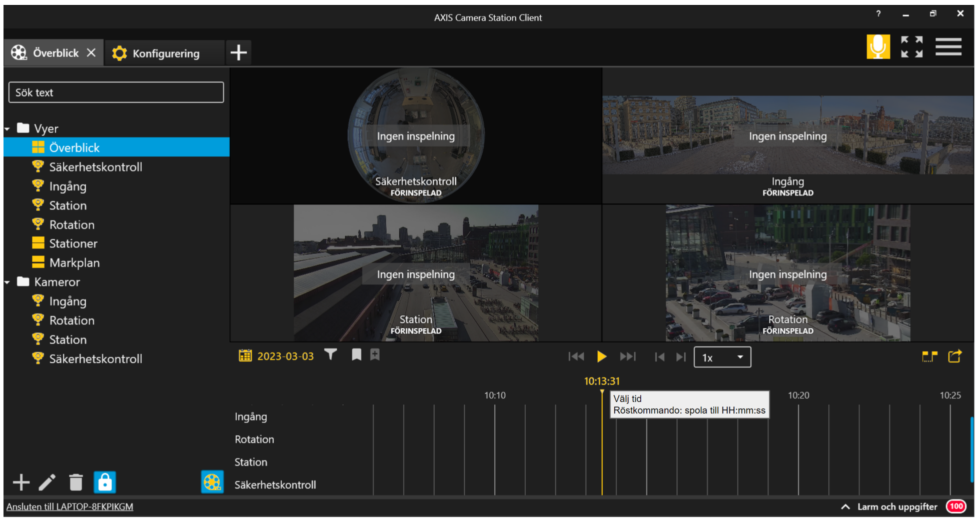
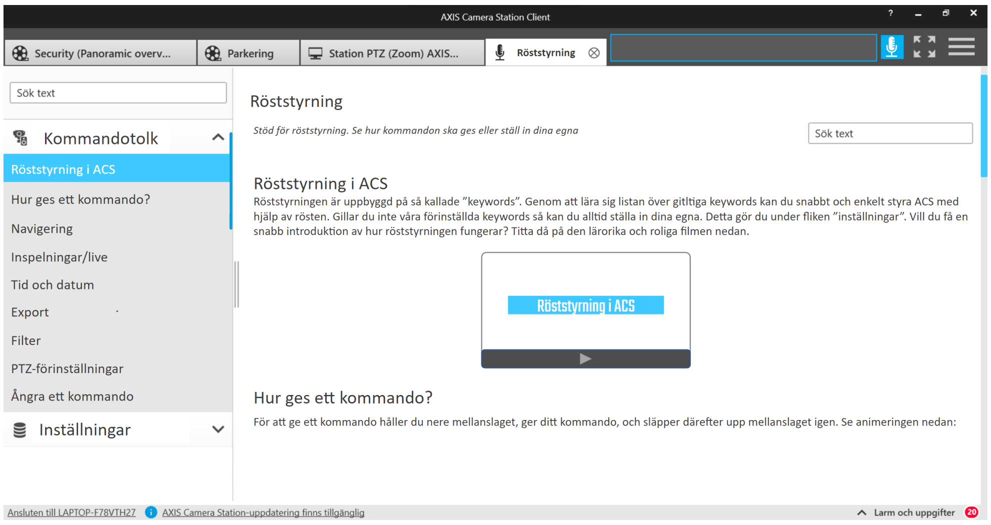


Figure 4.13: Discovery mode

Voice control tab. ACS uses a tab system where the user can navigate between different pages like the regular camera view, configuration page, smart search etc. Accordingly, a tab for voice control was created, see figure 4.14. This was made as a prototype, but conceptually the page should contain the full documentation on the voice control function. It should also offer in-depth guidance on how to use the feature. The tab is intended to be opened, but not in focus, when voice control is activated.



Röststyrning

Stöd för röststyrning. Se hur kommandon ska ges eller ställ in dina egna

Röststyrning i ACS

Röststyrningen är uppbyggd på så kallade ”keywords”. Genom att lära sig listan över gitltiga keywords kan du snabbt och enkelt styra ACS med hjälp av rösten. Gillar du inte våra förinställda keywords så kan du alltid ställa in dina egna. Detta gör du under fliken ”inställningar”. Vill du få en snabb introduktion av hur röststyrningen fungerar? Titta då på den lärorika och roliga filmen nedan.

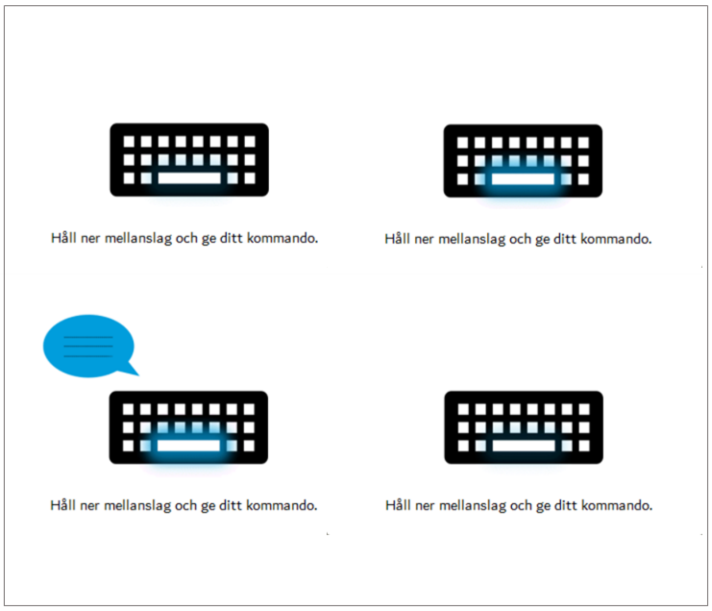
Hur ges ett kommando?

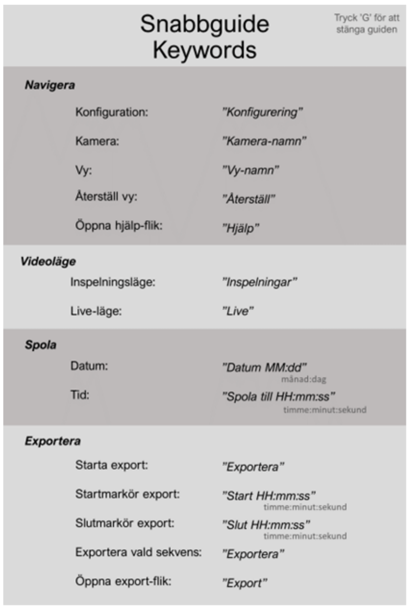
För att ge ett kommando håller du nere mellanslaget, ger ditt kommando, och släpper därefter upp mellanslaget igen. Se animeringen nedan:

Figure 4.14: Voice control tab

Pop-up window. A pop-up window containing an animated instruction on how to give a command was created, see figure 4.15. This appeared each time the user activated the voice control feature. This was made with the intention of giving the user initial guidance of how to control the voice interface. With this knowledge the user is able to give commands, regardless if they are correct or not. This was considered crucial knowledge and was thereby made as a pop-up window which makes it hard to miss. After this stage in the learning curve, more extensive documentation can be supplied which guides the user how to formulate their commands.

Quickguide. A quickguide containing the most common actions and their corresponding keywords was created, see figure 4.15. When voice control is activated, the text box initially displays the sentence "Press space to give a command. Press G for valid commands.". Pressing 'G' on the keyboard opens/closes the quickguide, which appears as a temporary window on the right side of the screen. The intention with this quickguide and how it is accessed was to supply the user with fast instructions of common commands. The user is also referred to this guide in some error messages in order to quickly teach the user how to formulate correct commands.


Håll ner mellanslag och ge ditt kommando.
Håll ner mellanslag och ge ditt kommando.
Håll ner mellanslag och ge ditt kommando.
Håll ner mellanslag och ge ditt kommando.


Snabbguide
Keywords

Tryck 'G' för att stänga guiden

Navigera	
Konfiguration:	*"Konfigurering"*
Kamera:	*"Kamera-namn"*
Vy:	*"Vy-namn"*
Återställ vy:	*"Återställ"*
Öppna hjälp-flik:	*"Hjälp"*
Videoläge	
Inspelningsläge:	*"Inspelningar"*
Live-läge:	*"Live"*
Spola	
Datum:	*"Datum MM:dd"* månad:dag
Tid:	*"Spola till HH:mm:ss"* timme:minut:sekund
Exportera	
Starta export:	*"Exportera"*
Startmarkör export:	*"Start HH:mm:ss"* timme:minut:sekund
Slutmarkör export:	*"Slut HH:mm:ss"* timme:minut:sekund
Exportera vald sekvens:	*"Exportera"*
Öppna export-flik:	*"Export"*

Figure 4.15: Animation sequence (left), Quickguide (right)

In addition to the expansion of the visual feedback, a few other existing functions were improved during this iteration. For example, "home" was added as a keyword for shifting to live view, since this could be considered a default view. In response to the previous evaluation step, the word "undo" was added as a keyword to the return action, and "zoom out" was added to the reset action. Additionally, the keyword "go to" was removed. This was previously connected with scrubbing, but was overused in testing. As intended, the participants used it to go to specific times in the video material, but they also used it for several other actions. It was found that "go to" was a widely used precursor to any command, and was therefore deemed unfit as a keyword. The full description of the current solution is presented in table 4.3.

Table 4.3: Overview of the solution

Function	**Description**	**Keyword**
Button	Press to activate voice control	
Key control	Hold key to give commands	
Configuration	Change to configuration page	Configuration
Cameras	Change between cameras	Name of camera
Views	Change between views	Name of view
Live	Change to live view	Live/home
Recordings	Change to recordings	Recording/Recordings
Scrubbing	Select timestamp in video	Rewind/Time
Now	Select present time in video	Now
Date	Select date in video	Date
Today	Select today's date in video	Today
Export	Export video sequences	Export
Export marker, start	Set the start marker for export	Start/Start time
Export marker, stop	Set the stop marker for export	Stop/Stop time
Return	Undo most recent command	Return/Undo
Reset	Go back to default camera mode	Reset/Zoom out
Feedback	**Description**	
Ready for speech input	Text in text box	
Listening	Text in text box	
Recognized words	Text in text box	
Error message	Text and error symbol in text box	
Guidance	**Description**	
Pop-up window	Animation on how a command is given	
Quickguide	List of actions and corresponding keywords	
Discovery mode	Filter highlighting functions supported by voice control	
Tab	Information page	

4.5.1 Evaluation

All the visual features presented above were evaluated in relation to their role as support tools for voice control. The question we aimed to answer was which of the tools the participants found helpful in their discovery and usage of the voice control feature.

Test structure

The test was divided into four parts and covered specific design preferences, evaluation of visual concepts, and an observation. Testing was done individually. The participants were two colleagues with previous knowledge of ACS, but no previous experience using our voice control solution. Since the test was quite extensive, it was preceded by a pre-test that determined the final structure of it. The details of the finalized test are described below:

Evaluation of discovery mode. The participant was presented with the discovery mode prototype, and was asked which functions they thought were supported by voice control. They were also asked to find the right command for a certain action.

Giving commands. The two different ways of activating the speech recognizer (pressing or holding space) were evaluated against each other in a test inspired by A/B testing [14]. The participant was asked to give two specific commands using each version and then state their preference.

Observation. The participant was given the task of finding a specific video event using voice control. The date and time of the event was provided, as well as the camera on which it appeared.

Text box. The two different versions of the text box were evaluated against each other in a test inspired by A/B testing. The participant tried both versions and was then asked to state their preference.

Insights

From pre-testing, it was concluded that the original observation task was too extensive and added unnecessary amounts of time to the test overall. The participants should not only find, but also export, a specific event. We found that the voice control workflow for the export was not deemed intuitive enough in itself, regardless of the added visual support. Consequently, it was decided to remove this part from the observation, and use the next iteration to rebuild the export function. In addition, a presentation of the voice control tab was removed from the test. Evaluating the tab in its current state was not considered to be valuable. It would have needed further development in order to benefit from testing.

The first visual support feature tested by the participants was the discovery mode, which was generally appreciated. The color shifting was well-noticed, and it was considered an effective way of signaling what can be controlled by voice. However, there were differences in how apparent this was to the two participants. While both noticed the color shifting, it took some time for one of them to understand what it symbolized. There was also a request to better highlight the functions that were marked in blue, like the *recordings* function, see figure 4.13. Regarding the hovering, both had issues discovering this tool. Despite managing to hover on multiple highlighted functions, they did not read the text that appeared until they were given hints. However, the function was appreciated when spotted. It was also quickly adopted, as they tried to use the function in the following test parts when looking for guidance.

The second part of the test introduced the participants to giving voice commands in two different ways. For each version, they got to follow the instructions given in the pop-up window. This worked very well - the combination of animation and text was concluded to be powerful in conveying the message. However, the participants found it annoying that it appeared each time the voice control feature was activated. Regarding the two versions, it was evident that the participants preferred to hold space for the duration of their spoken command. This was motivated as being more intuitive and providing an increased feeling of control. The technique could be seen as a metaphor for a walkie-talkie, making it feel familiar.

The observation went rather well in both test cases. However, the participants were more likely to test their way forward than to utilize the guidance provided in the text box, such as introductory text and error messages. Nevertheless, they did notice the text printed in the text box, because they both read what the speech recognizer had heard and made corrections accordingly. This shows that the user is selective, and might not want to use help just because it is given. One consequence of this was that none of them found the quickguide, which was referenced to in the messages. Once they were specifically asked to read the error messages, opening and using the quickguide was done with ease.

The last part of the test presented an alternative style for the text box. While the participants could appreciate that the alternative text box was distinct and did not permanently cover a part of the user interface, the design and animations were questioned. The design was not thought to be in line with the rest of ACS, and the animations were deemed flashy and annoying. A request was to rather let the error messsages be more highlighted. In favor of the first text box, it was also said that for tools that are frequently used, there is a motivation for giving them designated space. This also applies to voice control.

In general, the graphical support significantly improved the learning curve for first-time users. The participants immediately understood how to give commands without external guidance, and the discovery mode lead to an initial comprehension of what the voice control solution covered. The button used for activating voice control was easily found, and made the participants aware of the adjacent text box. Although they preferred testing over reading error messages, they did use the speech recognizer output to their advantage. This was an important improvement from the last iteration, where they were disregarded. In conclusion, all visual tools tested proved to add something to the learning. Hence, they were kept as part of the solution.

4.6 Third Iteration: Final Adjustments

The last iteration was spent making some final improvements to existing functions. From testing, we concluded that the participants disregarded the error messages they received. Thus, there was an interest to highlight these, while avoiding making them frustrating. With these aspects in mind, we added a smooth, blinking effect surrounding the error text, see figure 4.16. When the user gives a command that can not be apprehended, the error symbol and error message appears. The area surrounding the error text blinks by fading to blue three times, and then disappears. The blue color was once again chosen in accordance with the design of ACS.

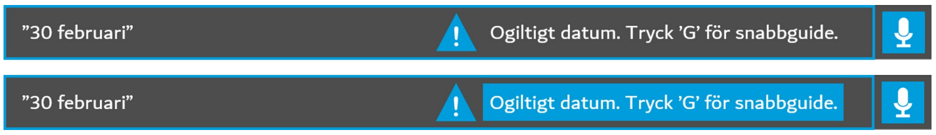

”30 februari”
Ogiltigt datum. Tryck 'G' för snabbguide.
”30 februari”
Ogiltigt datum. Tryck 'G' för snabbguide.

Figure 4.16: Error message animation

Another issue found through testing concerned how information was presented. Firstly, the text in the quickguide was considered too small to be read comfortably, so the guide was magnified in its entirety. Secondly, the participants did not understand what "MM-dd" meant in a date context. It was more intuitive for participants to use the format "12th of April" than "04-12". Since both formats were already implemented, we decided to simply change the format suggested in the quickguide and error messages to "day-month". It was also accompanied by an example, "1 april", in the quickguide. Further, the word "timestamp" was used by one of the participants when scrubbing in the video material. This would be a good contribution to the existing keywords for scrubbing, and was thereby added.

Finally, the workflow for the export was rebuilt. A major issue was the ambiguous usage of the keyword "export", used for both starting the export and later for confirming the chosen time sequence. This combined with the difficulty of finding the keywords for defining the video sequence, "start" and "stop", made it hard to complete the export. The new version of the export targeted these flaws by removing the ambiguity of the keyword "export", and providing contextual guidance for defining the video sequence. As a result, the keyword "export" exclusively referred to starting the export, whereas the keyword "confirm" referred to confirming the export. In between these commands, the user received visual guidance for setting out the time markers. This appeared as a gray window above the timeline as seen in figure 4.17. When a start- or stop time was set, it appeared in the guidance window and replaced the keyword instruction. For example, "Start HH:mm:ss" was replaced by "Start: 15:24:30". This window also informed the user of the keyword "confirm". With this solution, the only command the user had to give before receiving contextual guidance was "export", which had been an obvious keyword among participants.

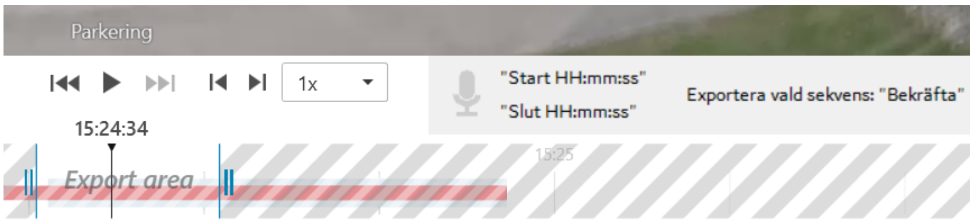


Figure 4.17: Contextual guidance for the export

In addition to the last improvements, this iteration involved creating and conducting a final test, as well as collecting the insights from all tests conducted during the project. This material was then used to suggest a final design for the voice control feature, which is presented in detail in chapter 5.

4.6.1 Evaluation

Pre-test

A pre-test was conducted with one participant to finalize the structure of the test. This participant had a background in design, and had no previous experience with ACS. The planned test structure was concluded to work well, so no changes were made to it. However, we discovered two aspects of the solution that were deemed important to improve. Firstly, the participant made attempts to use commands like "twelve o'clock" when changing the time. Hence, the keyword "o'clock" was added. Secondly, it was found that the error messages still did not grab the user's attention enough to make the information about the quickguide apparent. However, once it was found, it was heavily used and highly valued. In contrast to the error messages, the pop-up window was never disregarded by users. Therefore, the instructions for opening the quickguide were added to the pop-up window, see figure 4.18.

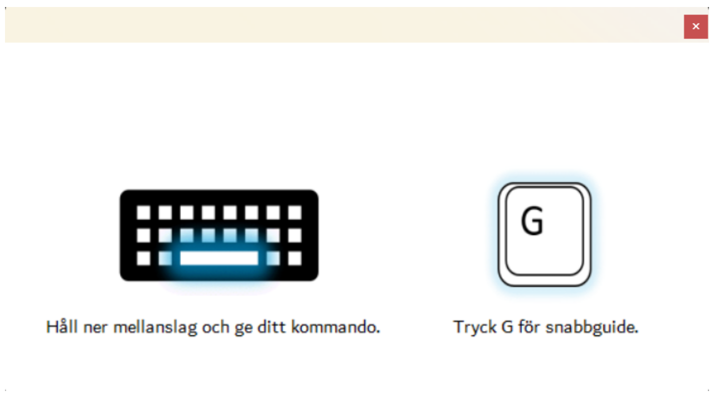


Figure 4.18: New version of the pop-up window

Test structure

The main objective of the final test was to evaluate voice control as interaction form. This was done in relation to the existing manual interaction, so that conclusions could be drawn about their respective strengths and weaknesses. In addition, the test aimed to evaluate improved parts of the solution, including the new export function and the highlighted error messages.

The final test was divided into three parts, where each part revolved around exporting a video sequence using a certain type of interaction. The test was conducted individually, with a total of three participants. Two of them had previous knowledge of ACS, while the third participant had just started working with the system. None of them had any previous experience using our voice control feature. The structure of the test is presented below:

Manual interaction. The first task was to export video footage of a red car driving past a parking lot. The participant was asked to complete the task solely by manual interaction, i.e. by clicking in the ACS user interface.

Voice interaction. The second task was to export video footage of a group of people running past a parking lot. The participant was asked to use voice control to the extent it was possible. For example, an exception was the play/pause function, which was not supported by the voice control solution.

Combined interaction. The third task was to export video footage of a white van driving past a parking lot. For each step towards completing the task, the participant was free to choose whether to use manual interaction or voice interaction.

At the end of the test, the participant was asked if they preferred working manually, by voice, or with a combination of the two.

Insights

There was a general consensus among participants about the major advantages of voice interaction compared to manual interaction when working in ACS. When something was difficult to find in the graphical interface, either because it was not visible in a certain context or because there were a large amount of alternatives, participants deemed it more efficient to use a voice command. One example of this was the navigation between cameras and views. In our test setup we only had access to one camera, resulting in a very short list of cameras and views. This made it easy to find the right camera, so most participants chose to simply click on it. However, they stated that in a real user environment, this list would likely be much longer. That would increase the time needed to find the right camera, and in that scenario they preferred to give a voice command. A second example was the selection of times and dates that required repeatedly dragging the timeline or opening the date picker. In these cases, it was preferred to set the date and the approximate time by voice, and then manually make smaller adjustments. Time and date selection in particular was concluded to be the greatest asset of our voice control feature overall, as it was used by every participant when given the choice.

A combined interaction, incorporating both manual interaction and voice commands, was undoubtedly the preferred way of using the system. According to participants, this made it possible to make use of the benefits from both interaction types. From observing the test, we also noticed that the participants seemed confident in shifting between the two. In summary, manual interaction was efficient when a desired action was directly visible to the user. An exception to this was when buttons and other manipulable features were far apart from each other in a workflow, or when they were difficult to handle practically. In these cases, as well as for actions that were less visible, our voice control feature became a valuable contribution to the system.

The new export workflow made interesting progress during testing. The majority of participants did complete it independently, which was a considerable improvement. The good results were likely connected to a combination of more distinct keywords, the contextual guidance window, and the improved visibility of the quickguide. As previously mentioned, instructions on how to open the quickguide were added in the pop-up window after the pretesting. This made the participants aware of the quickguide, which had a significant impact on their ability to independently solve tasks. It was heavily used by all participants, especially during the export, and they appreciated that they could show and hide it with a simple key press. However, there were a few details in the quickguide that confused some participants. The user was prompted to use the names of the cameras and views when navigating between these, and this was communicated as the keyword "camera-name" and "view-name". This lead one participant to believe they had to say "camera" or "view" as keyword and then state the name of the camera/view. Another participant believed that they also had to state the titles in the quickguide, i.e. the context in which the action was connected to, like "navigation". This lead them to say "navigation, camera, parking lot", which became inefficient and prone to error. Consequently, there was potential to make some clarifications. One suggestion we received was to use angle quotation marks around "camera name", implying that contextual information should be inserted.

Another function that was improved during this iteration was the display of error messages. The visibility of these were improved with a blinking effect, which did catch the user's attention to a larger extent. They were read slightly more often than before, but not enough to be relied on. They still remained a complement to other, higher regarded guidance features like the quickguide.

The testing also provided interesting insights regarding the timeline. It was found that the majority of participants viewed the timeline as one long collection of single points in time, and thus, the actions of choosing a date and a time were considered the same. However, our solution separated these processes by using different keywords for date and time. This was confusing to participants, who argued that the keyword "rewind" should be extended to include dates as well. Furthermore, it was concluded that none of the participants used the date format MM-dd. This format had been at risk of being confused with a time in the speech processing, meaning that a keyword was always needed for context. By removing this as a valid date format, it would be possible to use this digit sequence exclusively for times.

Chapter 5
Results

5.1 Resulting proposition

The following section will present a proposition of how voice control could be implemented in ACS. The proposition covers different parts such as functions, graphical supplement, and feedback. The design is based on insights gained during the entire design process, and its advantages and drawbacks will be further discussed in chapter 6.

5.1.1 Basic structure

The resulting voice control solution takes speech as input and translates it to text. This text is processed and scanned for predefined keywords, which are connected to already existing functionality in ACS, for example navigating or scrubbing. An input to the speech recognizer is given by holding a key down, speaking, and then releasing the key. During development and testing this key was specified as the spacebar, however the proposition is that this should be a self-defined key on the keyboard or an external button. The voice control solution as a whole can be turned on or off by a button in the graphical interface of ACS.

5.1.2 Functions

A crucial part of the solution was which functions it should be able to perform. From user research, a number of functions were concluded valuable and were thereby implemented and tested. An overview of these functions and their corresponding keywords is presented in table 5.1. Additionally, several other functions were also deemed applicable but technical limitations prevented implementation, and they could therefore not be tested. These are presented in table 5.2.

Table 5.1: Implemented functions

Function	Description	Keyword
Configuration	Change to configuration page	Configuration
Cameras	Change between cameras	Name of camera
Views	Change between views	Name of view
Live	Change to live view	Live/Home
Recordings	Change to recordings	Recording/Recordings
Scrubbing	Select timestamp in video	Rewind/Time Timestamp/O'clock
Now	Select present time in video	Now
Date	Select date in video	Date
Today	Select today's date in video	Today
Export tab	Open the export tab	Export
Start export	Begin the export process	Export (Swedish verb)
Export marker, start	Set the start marker for export	Start/Start time
Export marker, stop	Set the stop marker for export	Stop/Stop time
Export selected sequence	Confirm and export sequence	Confirm/Done/Finished
Return	Undo most recent command	Return/Undo
Reset	Go to default camera mode	Reset/Zoom out

Table 5.2: Additional recommended functions

Function	Description
Play/pause	Play and pause recorded video.
Close tabs	Close open tabs.
Skip to next event	Fast forward to next recorded event.
Open view/camera folder	Open folder to show available camera/view names.
Screenshot	Take screenshot of the video.
Yesterday	Change date to yesterday's date.
Back x	Rewind x seconds, minutes, or hours in the video.
Playback speed	Change playback speed.

5.1.3 Feedback & Guidance

Testing showed it was necessary to complement the voice control functionality with feedback and guidance in order to make it user friendly. This was done by introducing new graphical elements into ACS. Table 5.3 presents the information that was found to be necessary, and how it is, or could be provided, in the final solution.

Table 5.3: Feedback & Guidance

Necessary information	How the information is provided
System ready for speech input	Text in text box.
System listens	Text "Listening" in text box.
Recognized words	The recognized words in text box.
Occurrence of error	Text, error symbol and blinking light in text box.
How to give commands	Pop-up window with animation.
Formulating commands	Quickguide with a list of actions and corresponding keywords. Text box displaying keywords when hovering over functions.
Existing quickguide	Pop-up window with animation.
Supported functions	Discovery mode with a filter highlighting supported functions.
Collected information about voice control	Information page as a tab in ACS.

An overview of the graphical elements is presented in picture 5.1 below. This is not a proposition of the exact design, but rather a recommendation of which elements that should be included in the solution.

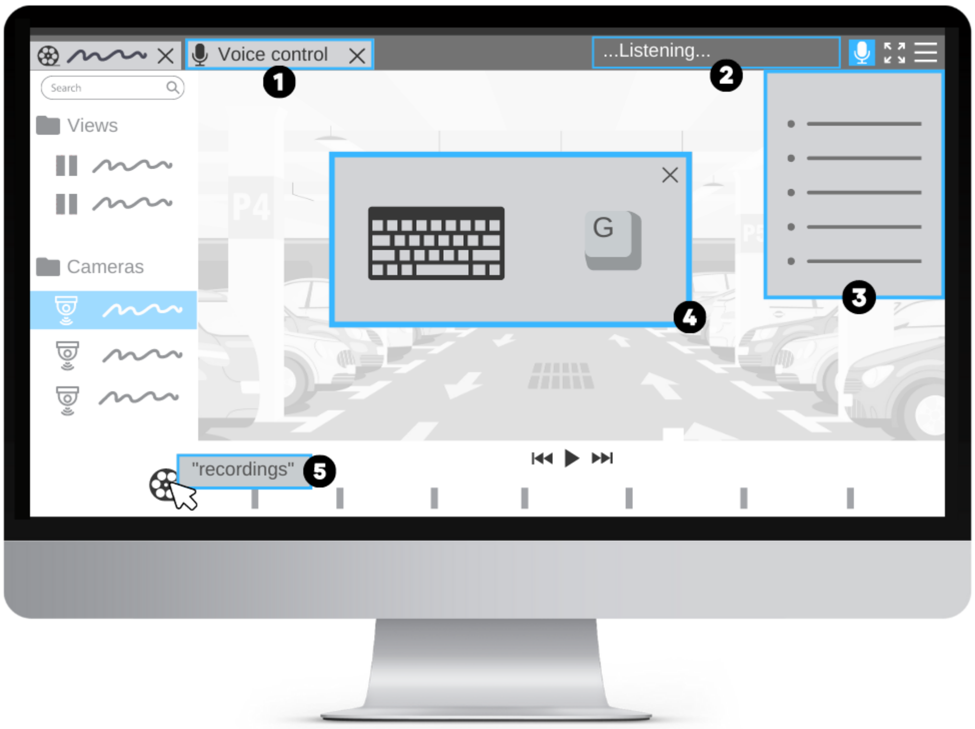


Figure 5.1: (1) Information page, (2) Text box, (3) Quickguide, (4) Pop-up window, (5) Hovering in discovery mode

Chapter 6
Discussion

6.1 Purpose and seamless integration

The purpose of this project was to explore how voice control could be designed in order to be a valuable contribution to the user experience in the current interface of ACS. The premise of the project was the already existing functionality of the system, and our approach was to build voice control as a supplement to its current interface. We strove for small interventions rather than comprehensive redesign. We aimed to make the voice control solution fit ACS, and not the other way around. These incentives were the underlying reason behind many different decisions regarding the basic structure, feedback and guidance. A crucial aspect was the seamless integration between manual commands and voice commands. If voice control is a supplement and contribution to the current interface, it should not impede the manual workflows. Instead it should enhance the workflow, by acting as an alternative to awkward tasks such as changing dates, while smaller adjustments can still be done manually, such as fine tuning the scrubbing. This combination of voice control and manual control has been an appreciated feature in testing.

6.2 System design in relation to personas

Several design decisions have been made during the development process. These decisions were based on a particular persona, "Thomas" see figure 4.1, who has intermediate knowledge of the system and regularly uses the core functions. The reason for focusing on Thomas in this project was that he represents a large user group of ACS. Additionally, efficient workflows for Thomas might benefit other users as well, since the most common workflows are shared between the different user groups. Thereby, our ambition was to make the most common tasks more efficient and intuitive by building voice control as a support to the core functions. If this was succeeded it would benefit users represented by Hanna and Berit as well, see figure

4.2 and 4.3. However, the optimization of the solution as a whole was made in favour of Thomas.

One of the earliest decisions in the design process regarded which type of speech interaction the solution should be built upon. One option was to have a conversation-based interface built on sentences. Another option was to have an interface based on keywords. With our chosen persona in mind, the choice fell on the keyword-based option. This option offered quick commands which could support the workflows and make them more efficient, for example by performing large time jumps by voice, instead of dragging the timeline repeatedly by hand. A conversation-based solution might not exclude short and effective commands, but requires considerably more advanced speech processing. However, it might provide a more flexible solution that could interpret a wider range of commands regarding the same functions. This could benefit users such as Berit, who could explain to the system what she wants to achieve, rather than doing it herself by hand. However, for a user who works in the system regularly, this type of conversational interaction could feel inefficient in comparison to do it by hand. Unlike Berit, Thomas knows where to click and which actions he needs to perform, and thereby it would be faster and easier to do it by hand than figuring out how to formulate a command. A keyword-based solution does not provide the same flexibility, hence forcing the user to learn certain keywords. The advantage of this approach is that the user remembers the commands instead of formulating new ones, which would be slower and more demanding. Accordingly, the keyword-based solution provides the efficiency needed for the solution to become valuable for a user like Thomas. It is also important to consider the environment in which the voice control solution is supposedly used. In Thomas' case, this is often in tiny, crowded and loud offices, which is not the ideal environment for speech recognition, see section 2.3.3. Hence, sentence-long commands in a conversation-based solution has a higher risk of being misheard by the speech recognizer, than shorter inputs such as keywords. In combination with our own technical experience, it was concluded favourable to explore the keyword-based option in this project.

6.3 Activation of speech recognizer

In accordance with the decision that voice control should be implemented as a supplementary tool, it was reasonable to restrict the input fed to the speech recognizer. In turn, this would restrict the possibilities for the voice control feature to perform undesirable actions, and give the control back to the user. The button for activating voice control as a whole could be seen as a first level of restriction. When voice control is turned off, no key press could accidentally activate the speech recognizer. This would prevent unexpected outcomes for users that are not interested in using voice commands at the moment, or at all. The second level of restriction comes into play when voice control is turned on. The user might want to work interchangeably with manual- and voice interaction, possibly in a noisy environment, which requires additional control of the speech recognizer. This control could have been designed in many different ways. The choice fell on activating the speech recognizer with a key, however, it would have also been possible to activate it with a wake word. To make a reasonable comparison between the two, it is relevant to return to the objective of the project and to the design choices that had already been established based on our personas. It had been determined that the voice control feature should enable seamless integration between different interaction types, providing a new, efficient way of working in ACS. This required that the voice commands could be given quickly, which was mainly achieved by the use of a keyword-based approach. If a wake word was used for activating the speech recognizer, it would have had to be uttered each time a voice command is given. This would have increased the length of every voice command, resulting in an inefficient interaction that would likely be annoying to use. On the contrary, a wake word could have been useful if the voice control feature was conversation-based and supported multi-step commands.

Using a key control was generally appreciated by participants in the evaluation steps, and they were involved in determining the final design of it. In the developed solution, the speech recognizer starts listening when the spacebar is held down, and stops listening when it is released. However, as stated in section 5.1.1, the suggestion is to use another key. While the concept of using a well-defined key was appreciated by test participants, many wanted to use the spacebar for other actions such as playing and pausing the video. Consequently, we suggest to use either a key that can be self-defined by the user, or an external button. The advantage of a self-defined key is that no extra equipment is needed. However, some users might want to use voice control primarily for ergonomic reasons. Thomas, who lacks the space needed to be able to work comfortably with ACS, represents a user group that could benefit from an external, mobile button. This would further reduce the keyboard- and mouse interactions needed, decreasing the time spent in uncomfortable working positions. However, it should be noted that some tasks are preferred to be performed manually, and the amount of surrounding noise might also impact the extent to which voice control is used. Thus, manual work might never be entirely excluded, but the voice control feature as a whole still has great potential to contribute ergonomically.

6.4 Selection of keywords

The keyword-based solution required that each function was connected to one or several keywords. Some functions had apparent options while others were considerably more difficult to determine. In some cases the keyword had different meaning to different persons, such as "reset" and "return" as discussed in "Insights" in section 4.4.1. In other cases, some keywords were applicable for several different functions. For example, "go to" was used in combination with almost every other function, such as "go to camera 1" or "go to 14:25". The frequent usage of "go to" made it complicated to set it as a keyword for time, since it likely would be used in other situations as well. The same issue was applied to "o'clock" which could be used to set the time during both scrubbing and exporting. These issues could be resolved in different ways. We generated a large amount of different keywords so that there were numerous options for the same function. By doing this we allowed a wider range of formulations, increasing the flexibility of the system. In context based dilemmas such as "go to" and "o'clock" we weighted the risk against the flexibility. When there was a high risk of someone giving a command that would result in the wrong action, we decided to exclude the keyword. If the risk was lower and would happen only in case of exceptional formulations, the keyword was retained. We also worked with rules of priority when several keywords were mentioned in the same command. For example, "timestamp today" contains two keywords, "timestamp" and "today". The first keyword was related to a time, while the other was related to a date. We would then prioritize "today" and change the date instead of sending an error message with a warning of an invalid time.

A technical limitation which affected the choice of keywords was the restricted communication with ACS. The voice control solution could send directions to ACS through deep links, but the information ACS could send back was limited. See section 4.3.1 for a more detailed description of the structure. This meant that the voice control solution had no information about the status of ACS. Hence, the keywords had to work independently of the state and could therefore not make use of the context in which the command was given. This might otherwise have been used to decide whether an action is reasonable to perform or not. This knowledge in combination with rules of priority could make it possible to provide a wider range of keywords, that perform different actions in different contexts.

6.5 Functionality

A central part of the solution was which functions that should be manageable through voice control. As previously stated, we chose to focus on the most common workflows. Navigating between cameras and views, scrubbing and the export make up the standard operations performed in the system. Hence, we chose to implement functions related to these activities, see table 5.1 in section 5.1.2. There were some functions that were not possible for us to implement into our solution. While some functions lacked deep links, other functions were possible to implement, but the restricted time frame made us prioritize other aspects in the solution. However, some of these functions were requested during testing and should hence be considered during real implementation. These are presented in table 5.2 in section 5.1.2. Some of them were an important part of the most common workflows, such as "play/pause" and "skip to next event". These two functions would considerably facilitate the scrubbing by voice. Another function which also could benefit the scrubbing was the function "back x" which would rewind the video with the specified time x. However, there was a risk of collision with the command "return" which would undo the most recent action. One of the insights during testing in section 4.4.1 was that the relation between a keywords and an action was not always uniformly perceived. There are different impressions of which action a certain keyword should lead to, and keywords as "back" and "reset" were at risk of confusion.

The functions that were implemented varied in convenience. The navigation between different cameras and views was well executed. However its usefulness depended on the amount of available cameras. During testing we had access to one camera, which made it easy to find and simple to click on. However, this might not be the case in a real user environment. As discovered during the observation in section 4.1.2, there might be a considerable amount of cameras and views, which makes it hard to find a certain option in the list. Then it would be easier to say the name of it instead. However, this might be more difficult if there are several cameras and views to remember.

Functions relating to scrubbing were at higher risk for mistakes. A common problem was different mental models regarding the timeline. The voice control solution separated dates from the timeline while some users thought that the timeline were made up of both date and time, as discussed in "Insights and Suggestions" in section 4.6.1. This issue could be resolved with different approaches. A vital problem was that the speech recognizer performed poorly on lengthy input. Therefore it was necessary to keep date and time divided in two separate commands. However, the selection of keywords could be further refined, making the functions more clearly related to their keywords. For example, the keyword "time" might better reflect the action of changing the time, than "rewind" or "timestamp". Removing these keywords would make the solution less flexible, but it would communicate a more clear division of the commands. Another option would be to let "rewind" act as keyword for both time and date. This would be a more flexible solution, but it might result in users trying to give date and time in the same command which is at high risk of failing. Hence, the commands still need to be kept separate.

The workflow related to the export was somewhat troublesome during development. It was hard to achieve an intuitive workflow were the user understood the different steps in the process. However, the resulting workflow was manageable for most users during testing, with support from the quickguide and the contextual guidance, seen in figure 4.17 in section 4.6. Although, it still needs to be further improved. The information provided in the contextual guidance in figure 4.17 was important in order to visualize the different steps and components in the workflow. The quickguide was also crucial since all users needed this support in order to finish the export related task. However, once the learning curve had been overcome, the export worked relatively well depending on the user's export technique. Some users preferred scrubbing their way to the beginning and end of the video sequence. This fine-tuned scrubbing was not easy to do via voice control. Other users preferred to coarsely rewind to the approximate time of the sequence, and let the video play until the beginning was reached. They would then pause the video and set the start marker by voice. Next they would let the video continue playing until the end of the sequence was reached, which was followed by setting the end marker. This export technique worked well with voice control, but could be made more efficient if the playback speed could also be easily altered by voice. The export also worked well on entire events, since the user in this case did not have to place any of the time markers, but just confirm the predefined sequence immediately.

During testing, several users tried to export a video sequence with one single command instead of doing it step by step as it was designed. It might feel more natural to give the command cohesively, for example "export from 15.50.53 to 15.51.25" but in practice this did not work very well. This type of command was prone to errors, due to its length and many numbers. The speech recognition performed poorly on lengthy input and there was a high risk that some parts had to be repeated multiple times.

6.6 User support

As stated in section 5.1.3, it was necessary to provide the user with information of how the voice control feature works, as well as information about its current state. Table 5.3 gave an overview of the information needed and how it was presented in the user interface. This section will use the table as foundation for an extended discussion around the visual tools that were created to communicate necessary information.

System ready for speech input. Visually indicating that the system is ready to use is a way of being transparent about its state to the user. The increased transparency makes it easier for the user to understand when commands can be given. When a command has been processed, the text goes from displaying "...Listening..." to displaying the recognized words. These words will remain until a new command is given, and indicate that the processing of the previous command has been completed. There was a discussion around whether the recognized words should disappear after a specified amount of time had passed. This would have made it possible to display a message that directly states that the system is ready for new input. However, there had been no issues surrounding this in testing. Furthermore, making sure that the recognized words were always visible to the user was higher prioritized, because of the importance of that feedback.

System listens. Knowing when the system is listening is an important part of understanding the state of the system. It also partially improves the sense of privacy, since it is clear to the user when audio input is being collected. When the key is held down and the speech recognizer starts listening, the text box displays the message "...Listening..." as feedback. In the final evaluation, there was a remark concerning the visibility of this feedback, where the wish was to highlight it more. However, this was only requested by one participant. It should also be noted that the graphical support tools overall need to be designed with careful consideration to how they will be perceived. Feedback and guidance tools need to be apparent to the user, but not at the cost of being perceived as disruptive and annoying. The aim is to find a balance between the two, especially since most of these tools are a permanent part of the user interface. When the user has learnt to use the system, which has been found to happen quickly, frequent visual effects become redundant and disruptive. The long-term perspective is therefore of high importance when designing these visual tools.

Recognized words. In terms of feedback, displaying the recognized words was the most frequently used and appreciated feature throughout the evaluation steps. It helps with isolating where the error occurred, and indirectly encourages the user to try again in cases where they were misheard. Since this was a part of the voice control solution from the start, we had the opportunity to confirm the positive results several times. Thus, we recommend this as a vital part of the feedback to users.

Occurrence of errors. When an error occurs, a warning symbol followed by an error message appears in the text box. For the first few seconds, the text is highlighted by blinking, blue light. This made the message noticeable, but not enough to cause disruptions to the workflow. As a concept, error messages did not resonate well with test participant's attitudes. Although they were often noticed, participants tended to want to solve the issues themselves. An interesting aspect to consider is that the tests were conducted with people that had an above-average level of technical knowledge. As a group, they were solution-oriented and confident in trying new things, meaning that they would rather explore and test their way forward than to ask for help or read error messages. The latter was reserved for situations where continuous testing had not lead to any success. This approach can be connected to the one that Hanna would likely use. Thus, it would have been interesting to conduct the same type of observations with groups that are more similar to Thomas and Berit. When the technical confidence of Hanna is lacking, it could be possible that the attitude towards error messages is different.

How to give commands. Knowing how to give a command is a vital prerequisite for getting started with voice control. This is why we chose to present this information in a pop-up window as soon as voice control is turned on. The instruction combined an animation with text, which worked very well in conveying the message. Furthermore, the information in the pop-up window was kept at a minimum. The written instruction consisted of one simple and direct sentence, only containing what was absolutely necessary. This was found to be a very effective way of learning the basic functionality without overwhelming the user.

Formulating commands. As well as knowing how to give a command, the user has to know what to say. The suggestions proposed include three ways of communicating this to the user. It is possible to hover over single functions to find what voice command is connected to it, the quickguide can be open for a summary, or the voice control tab can be opened to find a comprehensive list of all possible keywords to use. The way in which these differs from each other is the level of detail. They complement each other by offering either quick access, extensive information, or a combination. While the voice control tab mainly exists to let users learn more about specifics, the hovering support and the quickguide provide easy access and are appropriate as direct work tools. As explained in section 2.3.1, it can be hard to remember how to give voice commands. This has been confirmed by testing, where participants often tried to hover or open the quickguide for reference, which demonstrates their intended role as a work tools.

Existing quickguide. The quickguide was implemented in the second iteration, and was available during the evaluation that followed. When found, it was rated as very helpful. However, the participants never found it independently. When this information was moved from the text box to the pop-up window, every participant immediately opened and used it. This demonstrates that the value of a feature is directly dependent on whether it is noticed by users or not, which highlights the importance of visibility.

Supported functions. From the test connected with the first iteration, see section 4.4.1, we were able to observe what happened when no specific tasks were given and no information about supported functions existed. It was found to be very difficult to explore the voice control feature in such an open way, and there was a need for guidance to communicate the conceptual model of how it worked. This was communicated through the discovery mode. By highlighting supported functions directly in the user interface, the discoverability was found to be increased. However, this tool had only been possible to test individually, since there were technical difficulties with integrating it into the solution. More refinements regarding colours and how the mode should be turned on/off could be explored. The aspect that we have found to be most important is to ensure that the colour of highlighted features matches the colour of the voice control button, as seen in figure 4.13. This made the test participants connect them to each other.

Collected information about voice control. The user should be able to find all documentation relevant to voice control gathered in one place. Our suggestion is to create a new page for this, which opens as a tab in ACS when voice control is activated. This tab should not steal focus when opened, but should be available to the user if they are interested in learning more. The focus should instead remain on the cameras and the pop-up window instructions, as we suggest to avoid dramatic effects when starting voice control. Several new windows and being redirected might be perceived as deterrent, since the user might sense that they are losing control of the system.

As presented in figure 4.14, the page could contain an instructional video, how a command is given and comprehensive information about actions and keywords. Furthermore, any customization options should be possible to set here. If a self-defined key is to be used as key control, this could be an example. The tab has not been evaluated during this project, but it was argued that there needs to be a place where all information is collected. We have made a proposal for the concept of it, but the details of this are left open to explore.

6.7 Further research and development

Many areas of the voice control solution are subject to further improvement. This project can be seen as a pilot study that enlightens possibilities and difficulties, but more extensive research and testing is needed to draw more certain conclusions. The approach in this project was to work in relatively small iterations, hence the testing was also kept moderate. This means that the insights gained during this project were based on a relatively small amount of data. In addition, a majority of the test persons were colleagues working with ACS themselves, and they might have a more technical point of view than a real user. However, this moderate testing made it possible to build a conceptual prototype that addressed many different aspects. It was also considered reasonable given the scope of the project.

This project did not focus on speech recognition, however there might be an interest to develop a speech recognizer that is specifically adapted to to the type of commands given in this area of usage. This might lead to an improvement of the performance, but it might also make it possible to give longer commands where the currently used speech recognizer performed inadequately. Supporting longer commands also enables formulations of more complex commands. This would in turn require a more comprehensive speech interpretation.

6.7.1 A step towards accessibility

This solution was developed as a complement to the current interface of ACS and aimed to be used in combination with manual actions rather than replacing them. The solution has not been developed in consideration to persons with motor impairments, and further research and development would be needed to meet these users' needs. However, our solution might facilitate for users that experience some limited mobility in hands and arms, or find it difficult to perform tasks that require small, stable movements.

6.8 Conclusion

If voice control is to be implemented as a complementary interaction form in a graphical user interface, it needs to contribute with added value to the existing interface. Added value could imply increased efficiency or comfort, or a decrease in cognitive load. It has been found that all of these aspects could be achieved with a keyword-based approach. Furthermore, when using a graphical interface as foundation, it can be assumed that not all actions are preferably performed by voice. Therefore it is important to allow for a seamless integration between manual interaction and voice commands, which a keyword-based approach is well-suited for. Finally, since voice interaction remains an invisible interaction form, it inevitably comes with challenges regarding learnability. Graphical support has been proven to meet these challenges, hence serving as a vital complement to the solution.

References

[1] Adjust. What is deep linking? https://www.adjust.com/glossary/deep-linking/, [Accessed 2023-03-09].

[2] A. Alliance. Manifesto for agile software development. http://www.Agilemanifesto.org, [Accessed 2023-02-01].

[3] John Armitage. Are agile methods good for design? *Interactions*, 11:14–23, 2004.

[4] Mattias Arvola. *Interaktionsdesign och UX*. Studentlitteratur, 2014.

[5] Axis. AXIS Camera Station Client. https://www.axis.com/products/axis-camera-station, [Version 0.0230.02].

[6] Axis. Axis camera station. https://www.axis.com/products/axis-camera-station, [Accessed 2023-02-01].

[7] Axis. We are axis. https://www.axis.com/about-axis, [Accessed 2023-02-01].

[8] Stephanie Chamberlain, Helen Sharp, and Neil Maiden. Towards a framework for integrating agile development and user-centred design. In *Extreme Programming and Agile Processes in Software Engineering*, pages 143–153, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[9] Eric Corbett and Astrid Weber. What can i say? addressing user experience challenges of a mobile voice user interface for accessibility. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16, page 72–82, New York, NY, USA, 2016. Association for Computing Machinery.

[10] Nazanin Firoozeh, Adeline Nazarenko, Fabrice Alizon, and Béatrice Daille. Keyword extraction: Issues and methods. *Natural Language Engineering*, 26:259–291, 2020.

[11] International Organization for Standardization. Ergonomics of human-system interaction. Standard ISO 9241-210:2010, Geneva, 2010.

[12] Interaction Design Foundation. Make it easy on the user: Designing for discoverability within mobile apps. `https://www.interaction-design.org/literature/article/make-it-easy-on-the-user-designing-for-discoverability-within-mobile-apps`, [Accessed 2023-05-11].

[13] Anushay Furqan, Chelsea M. Myers, and Jichen Zhu. Learnability through adaptive discovery tools in voice user interfaces. pages 1617–1623, 05 2017.

[14] Amy Gallo. A refresher on a/b testing. `https://hbr.org/2017/06/a-refresher-on-ab-testing`, [Accessed 2023-05-10].

[15] Microsoft. Speech Service. `https://learn.microsoft.com/sv-se/azure/cognitive-services/speech-service/`, [Version 1.26.0].

[16] Microsoft. Desktop guide (windows forms .net). `https://learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-7.0`.

[17] Christine Murad, Cosmin Munteanu, Leigh Clark, and Benjamin R. Cowan. Design guidelines for hands-free speech interaction. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, page 269–276, New York, NY, USA, 2018.

[18] Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, 7:19143-19165, 2019.

[19] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann, 1994.

[20] Don A. Norman. *The design of everyday things*. Basic Books, 2013.

[21] Martin Porcheron, Joel E. Fischer, Stuart Reeves, and Sarah Sharples. Voice interfaces in everyday life. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, CHI '18, pages 1–12, New York, NY, USA, 2018. Association for Computing Machinery.

[22] Jennifer Preece, Helen Sharp, and Yvonne Roger. *Interaction Design: Beyond Human-Computer Interaction, 4th Edition*. John Wiley & Sons, 2015.

[23] Jennifer Preece, Helen Sharp, and Yvonne Roger. *Interaction Design: Beyond Human-Computer Interaction, 5th Edition*. John Wiley & Sons, 2019.

[24] Francis Rakotomalala, Hasindraibe Randriatsarafara, Aimé Hajalalaina, and Ravonimanantsoa Ndaohialy Manda Vy. Voice user interface: Literature review, challenges and future directions. *System Theory, Control and Computing Journal*, 1:65–89, 2021.

[25] Lou Schwartz. Agile-user experience design: an agile and user-centered process? In *Proceedings of The Eighth International Conference on Software Engineering Advances*, Venice, Italy, 2013.

[26] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson, 2009.

[27] Ron Van Buskirk and Mary LaLomia. A comparison of speech and mouse/keyboard gui navigation. In *Conference Companion on Human Factors in Computing Systems*, CHI '95, page 96, New York, NY, USA, 1995. Association for Computing Machinery.

Appendices

INSTITUTIONEN FÖR DESIGNVETENSKAPER | LUNDS TEKNISKA HÖGSKOLA | PRESENTERAD 2023-05-23

EXAMENSARBETE Introducing Voice Control in a Graphical User Interface Using a Keyword-Based Approach
STUDENTER Ellinor Nelderup, Matilda Jansson
HANDLEDARE Kirsten Rassmus-Gröhn (LTH), Felix Kaaman (Axis Communications)
EXAMINATOR Johanna Persson (LTH)

Röststyrning för videoövervakning

POPULÄRVETENSKAPLIG SAMMANFATTNING **Ellinor Nelderup, Matilda Jansson**

Kan röststyrning förbättra användarupplevelsen i ett befintligt grafiskt användargränssnitt? Här presenteras ett förslag för hur röststyrning kan utformas för att uppnå ökad effektivitet och bekvämlighet.

Röststyrning blir ett allt vanligare fenomen, och ett flertal röststyrda produkter har redan kommit ut på marknaden. Denna relativt nya interaktionsform har stor potential inom flera olika områden. Dock saknas tydliga designriktlinjer, vilket introducerar en hel del utmaningar. Flera företag vill därmed undersöka hur röststyrning kan utformas i deras produkter för att det ska bidra med sina positiva effekter. En aktuell produkt är AXIS Camera Station (ACS), som är ett system för videoövervakning. Frågan är, kan röststyrning bidra till en positiv användarupplevelse i ACS?

De vanligaste arbetsflödena i ACS bestod av navigering i systemet, granskning av övervakningsmaterial och export av utvalda videosekvenser. Detta framkom efter en användarundersökning. Dessa flöden blev prioriterade för att effektivisera det vanligaste arbetet. Användarundersökningen följdes av implementering som varvades med utvärdering och testning för att komma fram till en välfungerande prototyp. Resultatet blev en lösning baserad på nyckelord som gjorde det möjligt att ge korta, snabba kommandon. Denna lösning tillät en kombination av röststyrning och kommandon för hand. Det kunde nämligen konstateras att bäst resultat uppnås när användarna själva får välja vilka delar de vill styra med rösten. Vissa saker var helt enkelt lättare att styra för hand, medan röststyrning var användbart i situationer som krävde flera klick eller upprepade musrörelser, t.ex. val av datum eller tid.

Voice control
...Listening...
Views
Cameras
G
"recordings"

Eftersom denna röststyrningsfunktion utgår från ett grafiskt användargränssnitt var det nödvändigt att även komplettera med vissa grafiska komponenter. Röststyrning är till sin natur en osynlig interaktionstyp och kan därför medföra vissa problem vad gäller "learnability". Genom olika grafiska hjälpmedel, som exempelvis en snabbguide, kunde detta problem tacklas. Det medförde att användarna helt självständigt kunde slutföra olika uppgifter i systemet via röststyrning. Vår slutsats blev därmed att röststyrning kan vara ett värdefullt komplement till ett grafiskt användargränssnitt.