Visualizing conversational data in virtual reality

ARVID SÄTTERKVIST
Visualizing conversational data in virtual reality

ABSTRACT
Since the first implementation of a simple chatbot was made in 1964, countless research and development have been made to make the fascinating idea of talking to a computer a reality. But not until recently, have chatbots started to make an appearance in everyday lives amongst a broader audience. As the popularity of chatbots increases, the demands and functionality of the chatbots rises which consequently expands the size and complexity of the chatbot. The conversational data from a chatbot can become very complex and hard to understand. Therefore, to ensure the continuous advancement of features in chatbots, the developer needs tools and instruments to compete in the growing market.

Through a prototype based design process, a problem amongst developers to visualize and understand the conversational data from a chatbot is first identified and addressed. A Conversational Data Visualization (CDV) prototype in virtual reality is then developed with the intention to help developers understand and explore the conversational data from the chatbot they are working on. The design of the CDV is based on theories about key features of visualizations in 3D and related work that study visualizations with similar data structures as the conversational data from chatbots. Furthermore, the features of the CDV is based on the identified problem of visualizing conversational data amongst developers. Due to the importance of participatory design in a design process, an exploratory usability test of the CDV prototype was conducted to further explore the design choices regarding the identified problem.

The conversational data is visualized with tree structures in a circular formation to allow for visualization of links between different conversations. Results from the explorative usability test indicates that the visualization gave the users of the CDV an understandable overview of the conversational data. However, finding specific stories and nodes in the conversational data was identified as a problem due to inadequate information in the overview of the visualization.

Visualisering av konversationsdata i virtuell verklighet

SAMMANFATTNING
Sedan den första implementationen av en chatbot år 1964, har en stor mängd forskning och utveckling skett för att göra den fascinerande idén att prata med en dator verklighet. Det är inte förens på senare tid som chatbot har spridit sig till det vardagliga livet hos den stora massan. Samtidigt som utspridningen av chatbotar ökar så höjs kraven på f"{a}nktionalitet vilket i sin tur ut"{o}"kar storleken och komplexiteten hos chatboten. Konversationsdata hos en chatbot kan bli v"{a}ldigt komplex och svår att f"{a}r"{a}. F"{o}r att s"{a}kerst"{a}lla den fortsatta utvecklingen av chatbotar beh"{o}vs d"{a}rf"{o}r verktyg och instrument utvecklas f"{o}r att hj"{a}lpa utvecklare av chatbotar.

Genom en prototyppasserad designprocess identifieras ett problem hos utvecklare att visualisera och f"{o}rst"{a} konversationsdata från en chatbot. En typ av en konversationdata-visuallisering (KDV) är sedan utvecklad med syftet att hj"{a}lpa utvecklare f"{o}rst"{a} och utforska konversationsdata från chatbotar de jobbar pa. Designen på KDV är baserad pa teorier ang"{a}ende nyckelomr"{a}den inom 3D-visuallisering och relaterade forskningsarbeten som studerar visualiseringar med data liknande konversationsdata fran chatbotar. Designen av KDV"{a}r också baserad pa problem som identifieras hos utvecklare av chatbotar. På grund av hur viktigt det är att inkludera användaren i designprocesser så utf"{o}rs en utforskanande användbarhetsstudie pa KDV f"{o}r att utforska implementeringarna av dem identifierade designbesluten ang"{a}ende det identifierade problemen hos utvecklare.

Konversationsdata"{a}r visualiserad med tr"{a}dstrukurer i en cirkul"{a}r formation f"{o}r att till"{a}ta visualisering av l"{a}nkar mellan olika konversationer. Resultat f"{o}r den utforskande användbarhetsstudien indikerar att KDV"{a}r en visualisering som f"{o}rst"{a}s av anv"{a}ndarna. Dock så identifierades ett problem med att hitta specifika noder i konversationsdata efter som oversikten av visualisering inte inneh"{o}ll tillr"{a}ckligt med information.
Visualizing conversational data in virtual reality

Arvid Sätterkvist
Royal Institute of Technology / CSC
+46 730 67 67 64
arvidsat@kth.se

ABSTRACT
Since the first implementation of a simple chatbot was made in 1964, countless research and development have been made to make the fascinating idea of talking to a computer a reality. But not until recently, have chatbots started to make an appearance in everyday lives amongst a broader audience. As the popularity of chatbots increases, the demands and functionality of the chatbots rises which consequently expands the size and complexity of the chatbot. The conversational data from a chatbot can become very complex and hard to understand. Therefore, to ensure the continuous advancement of features in chatbots, the developer needs tools and instruments to compete in the growing market.

Through a prototype based design process, a problem amongst developers to visualize and understand the conversational data from a chatbot is first identified and addressed. A Conversational Data Visualization (CDV) prototype in virtual reality is then developed with the intention to help developers understand and explore the conversational data from the chatbot they are working on. The design of the CDV is based on theories about key features of visualizations in 3D and related work that study visualizations with similar data structures as the conversational data from chatbots. Furthermore, the features of the CDV is based on the identified problem of visualizing conversational data amongst developers. Due to the importance of participatory design in a design process, an exploratory usability test of the CDV prototype was conducted to further explore the design choices regarding the identified problem.

The conversational data is visualized with tree structures in a circular formation to allow for visualization of links between different conversations. Results from the explorative usability test indicates that the visualization gave the users of the CDV an understandable overview of the conversational data. However, finding specific stories and nodes in the conversational data was identified as a problem due to inadequate information in the overview of the visualization.

Keywords
Virtual reality, Information visualization, Conversational user interface, Chatbot, Design

1. INTRODUCTION
The possibility to have conversation with a computer has been a fascinating idea for a long time. Conversations with computers have been featured in science fiction shows like Star Trek first aired in 1966, and movies like Her from 2013. Due to advancements in technology and AI the reality of speaking to the computer is closer than ever and conversational user interfaces are really up and coming in recent years [33].

A conversational user interface, often in the form of a chatbot, is a way for humans to interact with computers in a more conversational way through spoken or written communication [20]. Chatbots are already integrated in many different applications today. There are chatbots that are very general and designed to understand and answer a broad list of commands such as Google Home1 and Siri2, also known as virtual personal assistants. There are also chatbots specialized for a specific task and topic like the Brisbot3 that helps and comforts children and Right Click4 that builds websites according to the users’ answers and requests. Chatbots are still not perfect and can mostly only perceive simple and accustomed sentences [34]. However, a lot has happened since the first chatbot was implemented by Joseph Weizenbaum in 1964 called ELIZA that could interpret simple sentences and ask counter-questions depending on identified words [28].

As datasets and data structures in chatbots are getting larger and more complex due to more demanding features, the ability to visualize and make sense of it all in an understandable way is important for the developer of chatbots. Raw data can become complex and difficult to follow for the observer without the help of a visualization system that lets the user of the visualization explore the data on simpler terms. Despite the need to create information visualization systems with good interaction [6], relatively little has been done to implement new interaction techniques in this area [15]. Therefore, research is needed to explore different ways to visualize the complex data and convoluted connections that many software projects struggle with today.

2. BACKGROUND
In this research project, a Conversational Data Visualization (CDV) prototype is designed for virtual reality and developed to help visualize conversational data from a chatbot to be used by developers of chatbots.

2.1 Chatbots
The possibilities for anyone to make a custom chatbot are many. Some examples of platforms and tools that let anyone with a computer and internet connection create a chatbot with your own rules are Wit5, Slack6 and Kik7. These services provide the developer with the underlying engine of the chatbot. Basically, the message from the user of the chatbot is interpreted and “understood” by the engine and data will be created accordingly. In theory, the only thing the developer of the chatbot must provide the engine with is all the potential things that the potential users of the chatbot could say and what the chatbot would answer. This will

1 https://madeby.google.com/home/
2 http://www.apple.com/ios/siri/
3 http://brisbot.com/
4 https://rightclick.io/
5 https://wit.ai/
6 https://api.slack.com/bot-users
7 https://dev.kik.com/#/home
from here on be called *conversational data*. In this research project, the chatbot engine used is the Wit engine. The structure of the conversational data in the Wit chatbot is made of different *stories*. A story can be thought about as a conversation between the chatbot and the user with a specific theme, for example the weather, places or music genres. The stories can at any time divide into branches and create paths down the story leading to different endings. At any point in the stories a link to anywhere in another or the same story can occur. A link with this identifier will from now on be called a *jump*.

If a chatbot developer specifies in the Wit engine that the user of the chatbot can start a conversation with “What's the weather tomorrow in Palo Alto?” the developer can tell the Wit engine what keywords to look for in that sentence. The word “weather” and the question mark at the end of the sentence tells us that the user of the chatbot is asking for information about the weather, therefore the developer wants the chatbot engine to set the intent of the user to “weather forecast”. After the intent of the chatbot user is known, the developer wants the chatbot engine to interpret in what area the user is interested to know the weather forecast. In this case “Palo Alto” is the area in question and therefore the place would be set to Palo Alto in California by the chatbot engine. This information will then be available to the developer to decide on and implement a suitable answer. In this example, an appropriate action would be to fetch weather data from Palo Alto and compile a response to the chatbot user presenting the weather forecast. Figure 1 shows a visualization of this process.

![Figure 1: Visualization of the process of how the chatbot identifies keywords and intention in a message from the user. Source: www.wit.ai, screenshot taken 2017-05-08.](https://example.com/figure1.png)

### 2.2 Virtual reality

According to Andries van Dam [8], the most common interaction setup for information visualization is the “point and click” WIMP GUIs (graphical user interface based on windows, icons, menus and a pointer) and it has been the leading way of interacting with computers since it became popularized by Apple and the Macintosh in 1984. This way of visualizing and interacting with data is not always suitable for more complex applications and visualizations since the desired data often hides behind multiple button presses and menus [15]. There are a lot of potential alternatives to interaction with WIMP GUIs in information visualization, for example tangible views [26] or sketch based interaction [5]. In this project, the interaction system studied will be Virtual Reality (VR).

VR is the use of multimodal technologies, such as displays, haptic feedback and motion sensors, to provide sensory information to the user of the VR system. Interaction with VR systems occurs in real time and lets the user manipulate, navigate and occupy the virtual environment [16].

Immersion is one phenomena often spoken about in relation to VR. According to M. Slater et al. [25], immersion in VR is defined as the number of modalities and the diversity in the sensory information that the user experiences. A higher amount of sensory information equals a higher immersion. Immersion in VR has been found to be beneficial in abstract information visualization due to faster completion times of tasks typically conducted in visualizations, such as single point search and trend determination [21]. Other potential benefits of immersion have been discussed before, for example, a decrease of information clutter in the 3D scene. As mentioned before, the WIMP GUI sometimes suffers from a cluttered view when visualizing complex data. A higher level of immersion can potentially reduce the clutter and consequently increase the comprehensibility of the virtual environment [2].

Traditionally, the only sense that has been used in conventional 2D monitors and the WIMP GUI is the vision. Due to the prioritization of immersion and higher amount of sensory inputs in VR systems it can be argued that VR can offer an increased intake of information [16]. A high number of modalities can potentially distribute the input to multiple senses, leading to an increased ability to process information.

VR technologies have been explored and been part of research projects for a long time. Ivan Sutherland did research on head mounted displays as early as 1968 [27]. Recently, consumer versions of VR systems like the Oculus Rift⁸, HTC vive⁹ and Playstation VR¹⁰ with relatively low price tags have reached the market. This, coupled with modern computer power, could open possibilities for individuals and companies to use and work with VR technologies at home or in the office environment. Therefore, it is in many companies’ interest to see what tools and visualizations can be made to help in the development of new products.

### 2.3 Use Case

Mobiento is an agency specializing in digital solutions and applications mostly focusing on mobile and web. They have developed a chatbot to answer customers’ most frequently asked questions about insurances. For this project, Mobiento used Wit as their chatbot engine. In Wit, the developers of the chatbot had to supply the chatbot engine with example phrases from the potential chatbot user and answers from the chatbot itself. A problem Mobiento had while developing their chatbot was that the amount and the structure of the data got unwieldy and it was difficult to get a clear overview of it.

A simplified reproduction of how the conversational data was visualized by Mobiento in the project can be seen in Table 1. A spreadsheet like this makes it difficult to track all the connections that are made to different questions and answers. Some answers from the user of the chatbot results in a jump to another part of the context. This created confusion and a lot of time was put in to get all the connections right. If the developers cannot structure and visualize the data in an easy way it might lead to bugs and undesired behavior in the software.

---

⁸ [https://www.oculus.com/](https://www.oculus.com/)
⁹ [https://www.vive.com/](https://www.vive.com/)
Table 1: A simplified reproduction of the spreadsheet used by Mobiento to understand the conversational data.

<table>
<thead>
<tr>
<th>Introduction and greetings</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>Questions/Phrases</td>
</tr>
<tr>
<td>1.1</td>
<td>Hello/Hi</td>
</tr>
<tr>
<td>1.2</td>
<td>Tell me about insurances.</td>
</tr>
<tr>
<td>1.3</td>
<td>Tell me about something else.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Yes, tell me about insurances.</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Tell me more about insurance X</td>
</tr>
<tr>
<td>2.1.1.1</td>
<td>Yes</td>
</tr>
<tr>
<td>2.1.1.2</td>
<td>No</td>
</tr>
</tbody>
</table>

3. THEORY

With the increasing amount of complex data, information visualization becomes important. Information visualization aims to help the user understand and visualize data. An information visualization can show patterns, trends and anomalies in complex data that can assist in finding the desired information [14]. S. Card et al. [6] defines information visualization as follows:

The use of computer-supported, interactive, visual representations of data to amplify cognition.

In simpler terms this means that the information visualization should aim to raise the knowledge of the user with the help of interaction and graphics. To be able to raise the knowledge of the user of the visualization, the minimum requirement on the visualization is that it conveys the data it visualizes in an understandable format to the target group.

Often when using information visualizations, the information that the user needs is not known beforehand. Therefore, it is important to let users freely explore the data and find patterns themselves to get the desired information. Shneiderman [23] has presented a three-step process when dealing with visual exploration of data, called “The visual information seeking mantra”, that is often used:

Overview first, zoom and filter, and then details-on-demand.

First, the user of the visualization should see the whole collection of data in a zoomed-out view without too much details of the data. From this overview, the user of the visualization can filter out irrelevant information and zoom in on areas of interest. After the collection of represented information is reduced to a smaller number, the details of each collection should be available to the user to further inspect the data.

3.1 Information visualization model

Van Wijk [29] proposes a way to assign value to different components in information visualization systems by looking at a model of the process of information visualization (Figure 2). The model simply states that data (D) and the specifications (S) for the visualization creates the visualization (V), which is represented as an image (I) to the user. The user perceives (P) and interprets the image, and therefore the knowledge (K) of the user rises. The user can then often interact with the data (E) and change the way the visualization is generated. From this model, multiple conclusions (though mostly obvious) can be drawn. For example, the specification on how the visualization of the data is generated changes how the user perceives the information which in its turn negatively or positively impacts at which rate the user gains knowledge (dK/dt).

![Figure 2: Van Wijk's model of information visualization.](image)

Van Wijks information visualization model enables a sense of quantification in the development of the visualization to compare and value the different components in the visualization system. This is important when developing a visualization to be able to make the right decisions when it comes to the implementation of these components in the information visualization system.

3.2 Information visualization in 3D

Young and Munro has identified six key areas that should be considered when designing software visualizations for 3D [32]. These key areas are explained in the next couple of sub sections and are extended with theory relevant to this project. They will be the basis in this research project to make a visualization that can convey the conversational data in an understandable way.

3.2.1 Representation

In visual environments, the representation decides how the data is converted and shown as a graphical object. This is one of the most important aspects of a visualization and can be the deciding factor between an intuitive and useful visualization tool or a complex and confusing one. Representations of components should be individual and have a distinctive appearance. Meaning that identical components should be visualized with the same representation but differing components should be visualized by a representation that is easily distinguishable from each other. The representation should also allow for a high amount of information while at the same time have a low visual complexity [32]. Compromises between showing a lot of information and lowering the visual complexity needs to be done when deciding on the type of representation since they are not mutually exclusive as can be seen in Van Wijk’s model (Figure 2).

3.2.2 Abstraction

When designing a visualization, it is important to consider what information to show and in how much detail. It is often necessary to abstract away finer details of the information to let the user get a better overview of the data. The density of the data decreases as the user zoom in and enables more details to be shown. According to
Van Wijk’s information visualization model (Figure 2), the representation and abstraction is part of the specification that decides how the data should be visualized. Therefore, it is important to get right in order for the user to be able to use the visualization to its fullest potential.

When deciding on how to abstract the information shown in the visualization, it is important to realize that there are two kinds of abstraction to think about, data abstraction and visual abstraction [7]. Data abstraction changes the data before visualization occurs, e.g. filtering or switching of data sources. Visual abstraction is the representation of data points, for example a snippet of a text that grows to its full length when the user zooms in on that particular data point. The data was there all the time and was not changed but the visualization of it was.

3.2.3 Navigation
Visualizations of a large amount of data often means that the user needs a way to navigate and move around the data without getting lost. A landmark or something immovable as a reference is recommended [9]. These aspects are important considerations when making a visualization because the navigation of the visualization should be made as simple as possible to not make the user of the visualization spend time on orientating instead of exploring the information.

To be able to create a visualization that is easy to navigate some attributes about the virtual world are important to analyze. Darken and Sibert defines three attributes in the virtual world to look at when developing a toolset for navigating a virtual world, size, density and activity [9]. Size refers to the size of the virtual world. A small world where the whole world can be seen from one viewpoint has the advantage of pointing the user’s focus to one object thus a landmark to navigate by is indirectly created. Density refers to the density of objects in the virtual world. Large open spaces in virtual environment can easily become hard to orientate due to the low number of visual cues. At the same time, a cluttered virtual world with a lot of objects and visual cues tends to obscure important objects. Activity refers to the amount of movements and change over time of objects in the virtual world. A static virtual world with small movements and changes of objects tends to be easier to navigate since the landmark and visual cues do not change.

A dynamic virtual world tends to increase the complexity of navigation since the landmarks and visual cues move. All these properties are subjective to the user of the visualization but can still give an indication of navigational problems that can occur in the environment.

3.2.4 Correlation
When a developer, for example, finds an error in a visualization of some source code, it is safe to assume that he or she probably would like to fix the error. Since the source code of the visualized data is where the developer will do the actual fix, the correlation between the representation and the part in the source code should be made clear. If for example a singular node in a hierarchical visualization is of interest, the source code for that node should be easy to locate and work with. Moreno et al. developed a visualization tool to help novice programmers learn Java where the visualization of the variables, methods, expressions etcetera was linked and highlighted in the Java code that was visible alongside the visualization at all time [19]. This was done to create the feeling of a holistic view with no visualization elements which seemed to appear from nowhere.

3.2.5 Automation
Another consideration when designing 3D visualizations is how much of the visualization should be created automatically and how much influence the user of the visualization should have during the creation of the visualization image. Sometimes it is better to let the user of the visualization create and build the visualization’s different parts to get a better understanding of the data. It is also important to carefully think about how and what information should be visualized if the visualization image is automatically created from a data source. A way for the user of the visualization to tweak and transform the data after the automatic creation could be a way to fix potential layout errors when the algorithm fails [17].

3.2.6 Interaction
Complex visualizations often need some sort of interaction from the user of the visualization to filter and sort through all the data. It is important to consider in what way the user of the visualization should interact and whether it is necessary at all. A high amount of interaction and transformation of the visualization can make the collected information subjective and misleading due to letting the user of the visualization fine tune the visualized data for different desired outcomes [29]. Therefore, it is important to have a default view of the visualization that conveys information important to the user without too much tweaking of the data.

Interaction in visualizations are traditionally done with the help of our sense of vision, for example buttons and sliders, but other modalities can be of help when interacting with visualizations. Haptic feedback can for example help people with visual impairments by providing another output channel that does not rely on the vision of the user of the visualization [12]. Haptic feedback can also increase the immersion in virtual reality by being another modality as described in section 2.2.

3.3 Related work
To not have to start from scratch designing the CDV a couple of earlier research projects were identified that addresses problems and solutions that emerges during the development of the visualization.

3.3.1 Cone Tree
Robertson et al. [22] made a hierarchical information visualization system in 3D called Cone Tree. They used a 3D environment where objects in the world overlapped each other to maximize the screen space for complex data structures. The user of Cone Tree could also grab and drag the nodes around to focus on the information that was of interest. A big focus in their work is the use of animations to transform and modify the data. When rotating the tree structures in the data, the animation shifts some of the cognitive load to the human perceptual system. The authors argue that the assimilation to a new image from the visualization could take several seconds to understand without animations the user of the visualization can follow perceptually during a transformation of the visualization. Therefore, the use of animations when transforming the visualization can be an effective way to minimize the time the user of the visualization needs to adapt to and grasp the new information.

3.3.2 TreeNetViz
Gou and Zhang [13] created an information visualization called the TreeNetViz that visualizes different tree structures with links between nodes in different trees. The visualization is based on a radial, space-filling technique to represent the trees with lines going through the middle of the circle to visualize links between trees (see Figure 3). A big focus that Gou and Zhang had in TreeNetViz was to reduce the visual complexity of the visualization. This was made by implementing a novel edge (line) bundling technique that bundles lines together to create one thicker line instead of multiple
3.4 Research Question

So far, different theories on how to make an understandable and explorative information visualization have been described which serve as a theoretical foundation for this project. This project also stems from a specific problem at a digital agency company where employees (programmers, designers and technical project managers) had problem visualizing conversational data from a chatbot they were working on. Therefore, the target group for this research project will be everyone involved in the process of developing chatbots. The environment that the information visualization will be developed for is VR since it is a research field in need of more research and the more traditional computer visualizations and interaction are often not adequate for more complex information visualizations. Therefore, the research question is:

What is a way of visualizing conversational data from a chatbot in virtual reality to make developers of chatbots understand and explore the conversational data?

4. Method

To create the visualization prototype of conversational data from a chatbot, a prototype based design process was conducted. First, semi-structured interviews with the developers, designers and technical project managers at Mobiento were performed to get a deeper understanding of how the team is working with the conversational data, and to identify critical points to visualize. Results from these initial interviews were then set as a basis for the development of the CDV prototype. Due to the importance of participatory design in a design process [3], an exploratory usability test of the CDV prototype were then conducted on employees from Mobiento. The intention of the exploratory usability test was to explore further research and implementation regarding the critical points identified in the initial interviews.

4.1 Interviews

As background, interviews with four employees at Mobiento who recently worked on the chatbot described in section 2.3 were conducted. The main purpose of these interviews was to get an understanding of how professional developers and designers worked with the conversational data during the development of a chatbot and where the critical points were. Furthermore, these interviews also functioned as a method to get an understanding of how the conversational data is structured to later decide on an appropriate data visualization technique to implement. The interviews were conducted in a semi-structured format, meaning that a set of predetermined questions were prepared for the interview, but other questions emerged and were discussed according to the dialogue between the interviewer and the interviewee [10]. The semi-structured interview technique was chosen to gather qualitative data and dive deep into the problem to find the necessary design requirements that are unknown beforehand.

4.1.2 Development of the CDV Prototype

Before the development of the CDV prototype, a chatbot with dummy data was made in order to retrieve conversational data that could be used for testing during the development of the CDV. The chatbot was made in the chatbot engine Wit to match the structure of the conversational data from the use case described in section 2.3. The dummy chatbot was created in the online interface provided by Wit and the data was then downloaded in the form of a JSON-file.

The dummy conversational data consisted of seven different stories with a maximum story length of five different information exchanges between the user and the chatbot. The number of branches in any spot of the stories ranged from one to four. The user input, and answers from the chatbot, consisted mostly of short answers with a maximum character length of 98.

For the development of the CDV, the HTC Vive was chosen as the VR platform. The reason for this is partly that it was available at Mobiento where this master thesis was written, but also because it is one of the state of the art devices in VR and is similar to other popular state of the art devices [1]. The HTC Vive also lets the user interact with the virtual environment with a set of hand controls mapped one to one between the virtual world and the real world. The development of the visualization prototype of the conversational chatbot data was done in Unity\textsuperscript{11} since it is an engine that supports the HTC Vive [1].

4.1.3 Exploratory Usability Test

An exploratory usability test involving the CDV was conducted on employees at Mobiento. Each test took place in a quiet and undisturbed place and took around 30 minutes to complete. The test started with five minutes of explanations of the different aspects of the CDV, like how the conversational data is represented and how to manipulate the visualization. After ensuring the participant

\textsuperscript{11} https://unity3d.com
understood how to use the visualization and offering further explanation and time to adapt to the VR, the participant had to perform nine tasks intended to test the understanding and explorative capabilities of the CDV. These tasks were also constructed to explore issues and further implementation with the critical points gathered from the initial interviews to this research project. To collect qualitative data from the exploratory usability test, the participants were sound recorded and encouraged to "think aloud", meaning they should talk about and convey their thought process when they perform the task [11]. The nine tasks were also screen captured from the participants' point of view to gather qualitative data about how the participants navigate and interact with the CDV prototype. After the nine tasks were completed a semi-structured interview was conducted to get reactions and thoughts from the participants about the CDV prototype for further discussions.

After the exploratory usability test, the recordings were transcribed and analyzed with a thematic analysis method [4]. The qualitative data was divided into themes based on the critical points identified in the initial interview step.

5. RESULTS
The results are divided into three sections according to the steps taken in the design process method. First the results from the initial interview will be presented to identify the critical points in the visualization. Then a review of the CDV prototype will be given according to the critical points and theory. Finally, the results from the exploratory usability test will be presented to identify issues and further implementation of the CDV prototype.

It is also worth noticing that the interviews and exploratory usability studies where done in either English or Swedish depending on language preference from the participant. Consequently, Swedish quotes are translated to English with the original quote in fine print below.

5.1 Interviews
During the initial interviews, everyone expressed that the hardest part about working with the conversational data of the chatbot was to visualize the structure and get a good overview of the data. One employee said that the data…

...was kind of like a hierarchical structure, but any part of a story could point to any other part of another story at all times. It was really hard to structure the conversation. The jumps to different location was really hard to visualize.

They used a spreadsheet (Table 1) to visualize all the different stories in their chatbot. However, as the chatbot grew it was hard to keep the spreadsheet updated and in sync with the conversational data of the chatbot. Every change that was made to the chatbot had to also manually be changed in the spreadsheet. There was no automatic update of the spreadsheet to keep it in sync with the chatbot.

When asked about the most important information to visualize in their opinion, the interviewees answered that the connection between the stories are very important to clearly see. According to one interviewee, the most important information that needed to be visualized was:

How everything fits together. The whole conversation. All the jumps between stories. Follow a conversation from start to finish. Then you can see were the errors are, if there are any. It is only possible to see one conversation at a time in Wit.

Another thought from the interviewees about important information to visualize was that every path in the story should lead to an endpoint to avoid situations where the chatbot stops responding due to reaching the end of the story. Endpoint in this context means a phrase that resets the conversation, e.g. “Do you need anything else?”. Therefore, it is important to clearly visualize if there are any story paths that does not lead anywhere so that the developer of the chatbot can take appropriate actions. For example, the last answer from the chatbot in a story should almost always have a connection to another story to continue the path, otherwise the conversation between the user and the chatbot would halt.

Being able to identify what are the path. Every possible conversation should lead to one of the endpoints. You don’t want to create frustration, every need should be catered for, even if the need is “call a human”.

To shortly summarize the initial interviews there are mainly two identified areas of the conversational data that seems to be of importance in the visualization. The connections between stories and end part of stories with no further connections, so called leafs in a tree structure. These areas were decided to be in focus in the design of the CDV.

5.2 Conversational Data Visualization
The foundations underlying the CDV stem from the archetypes of Cone Trees visualization as well as TreeNetViz. To visualize the conversational data structure of a chatbot, the conversation in the stories are divided into nodes. Every node represents one information exchange between the chatbot user and the chatbot. In other words, a node represents when the user says something that the chatbot answers. For example, a node could be the exchange where the user says “Tell me a joke” to which the chatbot answers “What kind of joke?”. To this node there would then probably be a couple of child nodes linked with the different alternatives for a joke theme. A node often has a parent node and one or multiple child nodes connected to it. The adjacent nodes connected to a node are called parent nodes if they are higher up in the hierarchical order and child nodes if they are lower in the hierarchical order. Every story has a root node that is the first node in the story and therefore starts the conversation. This root node would then have child nodes connected to it with their continuation of the conversation.

5.2.1 Layout
The CDV deploys all the root nodes from the stories in a circular formation around the Y-axis (pointing up) in Unity3D. From these root nodes, the rest of the story tree is built layer by layer downwards. Between each parent and child node there is a visualized link in the form of a line. The jumps are visualized in a green color and goes in a straight line between the nodes. The jumps in the data are the main reason for the circular formation of the visualization. It enables the lines to be visualized by a straight line
without risking intersecting any other stories in between. See Figure 4 for an overview of the visualization layout.

Figure 4: An overview of the layout of the CDV from the perspective of the user. The red boxes represent leaf nodes (loose ends) and the green lines are jumps between stories and their nodes.

Every node in all the stories are represented by a box that has both the input from the potential user of the chatbot and the answer from the chatbot written on it. The input from the potential user comes first and is written in bold for the user of the CDV to be able to quickly follow the conversational story of the inputs down the tree structure. Below the input from the potential user of the chatbot, the answer from the chatbot is written. Each node in the visualization therefore consists of both the input and the output of every possible information exchange that the potential user and the chatbot can make. See Figure 5 for an example of the representation of one of the nodes.

Figure 5: An expanded root node of a story about ordering pizza. The text in bold is what the user says and the text under “answer” is the answer from the chatbot.

5.2.2 Navigation

In the CDV, the text written on the nodes’ representations are often too long to show the full length of the text. Therefore, it is by default truncated with a trailing ellipsis indicating that the rest of the text is hidden. To show the full length of the string, there is a mechanic to grow the node for all the text to fit. By placing the virtual controller inside of the visualized node, it will grow to a size that can fit all the text bound to that node. Figure 5 shows an expanded representation of a node and the virtual controller used by the user of the visualization. By not showing the full length of the string at all time the visual clutter in the visualization is minimized. At the same time, the user of the CDV will still be able to identify the different nodes and stories by seeing the first part of the string of letters. For example, the sentence “I don’t feel well today” would be truncated to “I don’t fe…” and then expand to the full sentence when the user of the CDV hovers the controller inside the node.

Stories or subsets of stories can be highlighted in the CDV by hovering inside a node with the controller, just the same as expanding a node to fit all the text. All the child nodes of the selected node and its child nodes until the end of the tree structure gets highlighted in a blue color. Potential jumps to other nodes are also highlighted in the same blue color. However, the child node of the node jumped to are not highlighted since if there was no stop anywhere the highlighted connections could go on forever. The highlights are meant to serve as a filter for the user of the CDV to be able to focus on only the relevant stories. By clicking on a node, the highlight will stay activated even when the user removes the controller. This enables the option to highlight multiple stories and sub stories at the same time. See Figure 6 for an example of a highlighted story.

Figure 6: A node in the story is highlighted by placing the controller inside the box. Children and jumps of the node are also highlighted in the blue color to indicate where the conversation can continue from the selected node.

Since the layout of the visualization is in the form of a circle, the front of all nodes cannot be seen at the same time. It is possible in
the virtual environment to just physically walk around the environment to see all sides, but to make the interaction easier, there is a torus placed above all the stories that act as an anchor for the visualization. By grabbing this torus and move the controller up and down, the user can adjust the visualization in height to focus on different layers of the stories. By grabbing the torus and move the controller left or right, the user can rotate all the stories effectively functioning like a horizontal scroll.

5.3 Exploratory Usability Test
An exploratory usability test on the CDV was conducted on seven employees at Mobiento. One user experience designer, two programmers, two technical project managers, and two media technology students. The group consisted of three female and four male participants but no difference in behavior or understanding of the visualization could be identified between these groups in this exploratory usability test.

5.3.1 Understanding the data
Overall there does not seem to be any problem for the participants to understand how the stories are built up and how nodes connect to each other. No participant mentioned or was observed in the screen capture to have any problems understanding the tree structure that stories are made of. When asked about how many stories build up the conversational data in total every participant used the same technique by counting the root nodes of every story. All participants except one counted the right number of stories. The reason for not counting the right number of stories was not due to inadequate understanding of the visualization, but a navigational problem that is further explained in section 5.3.2.

All seven participants had no problem following the story of an example conversation between the potential user of the chatbot and the chatbot when there were no jumps to nodes in other stories involved. The flow of a conversation is always moving forward and therefore the flow in the stories is always moving down the tree structure from the root node that started the story. Jumps are also going in one direction from one node to the other. The representation of jumps between nodes are according to the exploratory usability test not conveying which direction they are going or if they are only going one way or both ways. Three participants felt that the jumps were confusing and not clear on which way they were going.

The links (referring to the jumps) when they go up the hierarchy was initially confusing me, and the directions of those lines were confusing.

Three participants also said that the jumps were easy to miss due to not being visible enough. When asked to follow a potential story path that involved jumps to other stories some participants got it wrong. Two participants thought that when the path reached its end in one story there would be no answer from the chatbot even though the last node jumped to another story. By looking at the screen captures from these participants, they did not follow the jumps to other nodes and evidently thought the story had reached its end.

Only three of the participants got all the answers to the tasks about following a certain path in a story right. Though, observations could be made during these tasks that the participants were not certain on the answer. Confusion was noticed in the form of hesitation in the motions and checking previous nodes several times to make sure they were on the right path.

5.3.2 Navigation
In general, the participants did not seem to have any substantial problems navigating around and interact with the visualization when performing the tasks. All seven participants said that the overall experience of the navigation and interaction felt good and natural. One participant thought that the motions were too wide and big, and that a secondary way of interacting with the visualization would be nice to prevent fatigue. For example, use the touchpad on the controller to rotate the torus instead of moving the whole arm in a wide motion above your head. During the interview after the test in VR one participant said about navigating with the controller:

It is a motion that you have to do all the time. If I would have to do it more than this it would have been annoying/exhausting.

Det är en rörelse man måste göra hela tiden och mer än så här hade jag tyckt varit jobbig.

Another participant thought the rotation of the torus should have some sort of momentum when you let go of the trigger. It should keep spinning and gradually stop by “friction” to make it feel more natural.

By observing the screen captures from the exploratory usability test, some instances of using the interaction methods that the VR device offers were identified. For example, a participant used the head tracking mechanism to look behind nodes when counting the number of outgoing jumps from that node (see Figure 7). Some participants also used the head tracking to get closer to nodes by physically moving the head up and down instead of moving the visualization.

When rotating the visualization some of the participants had a hard time following and keeping track of the nodes. This could be seen clearly when one participant tried to answer how many stories the data consists of by counting the root nodes and spinning the visualization at the same time. The participant counted one more node than what the correct number of nodes were. The confusion was identified to depend on the difficulty of keeping track of the node first counted while rotating the visualization, resulting in the participant to count it twice. Other instances where the same type of confusion occurred was identified in the thematic analysis of the screen captures. For example, when looking for a specific story, participants could come back to nodes that they already looked at several times before finding the right one. Participants that used the highlight function of the CDV had noticeably less problem with keeping track of the different nodes.

Figure 7: A participant uses the head tracking of the VR device to look behind a node.
None of the participants had any problem finding the loose ends in the visualization. However, three of them had forgotten that the red nodes are loose ends, but after a reminder they had no problem finding the red nodes quickly. When asked about if they think the loose ends should jump to another node and if so where, the answers were somewhat uniformed. All participants agreed that a loose end that just said “Goodbye” should not jump to any other node since the conversation is over. On another loose end where the potential user of the chatbot asked about hearing a joke, every participant except one answered that it should jump to the root node of the joke story.

“Do you want to hear a joke?” and then if I say “yes” there is no joke coming... so I would like the “yes” [the loose end node] to connect to the joke story.

When the participants were asked about where a loose end with no clear continuation could potentially jump to, the answers got vague. In this case, the chatbot asks “How are you?”, the user answers “Good” and then the chatbot answers with “That’s good to hear” with no advancement to other conversations. The participants in the test all said that the loose end should lead somewhere since the sudden stop does not make sense. Two of the participants wanted it to lead to a story that asked about further help, but most of them were not sure about where it should lead.

In fact, it [the loose end node] is open to let the user continue with any other conversation, so it probably does not have to lead anywhere, but at the same time it is not over.

6. DISCUSSION

In this research project, an attempt at making an information visualization prototype (CDV) for visualizing the conversational data to help in the development of chatbot has been made using a design process. Based on the theory and results several observations and tentative conclusions can be made about the CDV prototype and its design.

6.1 Layout

The layout of the CDV is partly based on the visualizations Cone Tree and TreeNetViz. Both these designs visualize tree structures by arranging them in a circular shape. The use of a circular shape in 3D-space allows to fit more data (nodes) on the screen at the same time. This is important when visualizing a large number of nodes and an overview of all the data is necessary. However, the design of Cone Tree is based on the notion that the visualization will be displayed on a regular 2D monitor which is not the case when working in VR. The head mounted display (HMD) of the HTC Vive does not have the same problem with screen space limitations like a regular 2D monitor since the information shown in the HMD depends on where the user looks, similarly to how it works in real life. Therefore, the amount of screen space available for data visualization is not the problem, instead the space area of the virtual environment is the limiting factor. The circular shape of the visualization does not help minimizing the area occupied by the visualization but it does help with how far away from the user the different data points visualizations must be. For example, putting the tree structure next to each other in a straight line would mean that the tree structure furthest away from the user would be much further away than if the same amounts of trees would be visualized in a circle formation. The distance to the different parts of the visualization matters since one of the requirements gathered from the initial interviews before the development of the CDV was to have a good overview of the whole conversational data set. If some data points of the set are very far away from each other in the virtual world they could be hard to spot and compare with each other without a lot of scrolling or manipulation of the visualization.

To minimize the visual clutter of the representation of the different nodes in the conversational data the text was heavily truncated when not selected in the visualization. According to the exploratory usability test this worked fine when no specific story or node had to be found and only more general tasks about the overall structure of the data set had to be answered. However, when the participants of the exploratory usability test had to find a specific story or node, the absence of more information without selecting the node raised problems. The participants could often not find the right node without individually selecting them one by one. Furthermore, it was sometimes hard to distinguish nodes between each other without first selecting them and read the details. For example, two of the stories had the truncated string “Can I…” visualized on the root node, which meant that there was no way to know before selecting the node and the full text appears which story it belongs to. This could indicate that the abstraction of the representation of the data is too high since it negatively impacts the navigation and exploration of the visualization. To reduce the amount of time the user spends looking for nodes other design choices to the layout can be made. For example, when looking for a specific node in the data it could help to first be able to easily find the story where the node in question exists. A potential solution could be to use color, similar to how Shneiderman distinguish between different file types in his visualization of file systems [24] the stories in the CDV could be of different color or grey shading to differentiate themselves from each other.

6.2 Navigation

From the exploratory usability test, it could be seen that when rotating the visualization by grabbing the torus with the controller some confusion emerges. The confusion seems to arise because of the lack of distinguishable land marks or features that helps the user of the visualization to orientate the virtual environment. Increasing the activity of the virtual world by rotating the visualization coupled with the stories visual similarity to each other makes the transformation hard to follow. The results from the exploratory usability test shows us that this can be prevented by highlighting a node and therefore creating a land mark in the visualization to navigate from. However, a more stable feature that helps with navigation should be implemented to erase the dependency of user interaction. The marker should be something internal that follows the rotation of the visualization to create a mark in the circle formation that otherwise can give the impression of having no start or ending.

In the CDV, there is no correlation between the visualization and the data source. For example, if a user of the CDV finds a loose end that should lead to another node, there is no way right now to quickly fix this with the CDV only. The only option is to manually find the right node in the data source. A link from a node in the CDV to the node in the data source could make it easier for the developer to fix bugs and faults in the data in a more effective way. To edit text directly in VR is not a simple task due to not being able to see your hands, but research and successful implementations have been made before [30,31,35]. Another way of directly interacting with the source code from the virtual world in the CDV would be to implement drag and drop features similar to visual
programming languages such as Scratch [18]. For example, to create a link between to nodes the user of the CDV could grab one node and drag it to the other to automatically create a jump in the visualization and the source code. The manipulation of the data source in real time requires the visualization to update automatically to match the data source at all time. If not, there is a risk that the visualization and the data can become out of sync which was a problem with the spreadsheet used by Mobiento.

6.3 Use Cases
Like the research question suggests the CDV is aimed at any developers of chatbots, meaning that it could be someone doing it as a hobby or working professionally for a company. A requirement for developing a chatbot is to have a computer of some sort, but a VR device is not a requirement. VR devices at home and work is an upcoming trend but is still far away from something that is available to every person. This becomes a problem if developers of chatbots wants to use the CDV but does not have access to a VR device of high enough immersion to run it. The physical space of the real world is another limiting factor of using a VR system for information visualization. For example, the HTC Vive requires a minimum area of 1.5 by 2.0 meters which could become a problem for people in a crowded working area with a lot of people moving around. These two limitations alone could potentially make the CDV not worth it for many companies and developers.

The possibility to partially or fully disregard and ignore the outside world due to the immersion from the VR device could raise the focus of the user of the visualization. Without impressions from the real world to process, the user could have an easier time focusing on the task of finding and perceiving the information in the visualization. A system like this, that in theory does not allow for any other activities than what it is made for, could help people to concentrate on the task at hand.

6.4 Future Research
Some participants in the exploratory usability test voiced concerns about how the visualization would work with a larger set of data which would be much easier to get lost in. Looking at other commercially ready chatbots, it is safe to say that chatbots can have a lot more conversational data implemented than the dummy data used in this research project. How the CDV would handle a larger set of data has not been tested in this project and will be left for future research. A concern for visualizing a larger data set in the CDV is the ability to find the right nodes the user is after in the visualization. As seen in the result there is already some problems arising when trying to find the right story and node and a larger set of data will probably not make that easier.

The data visualized in the CDV reflects the conversational data straight up without any regards for the underlying chatbot engine. When using the chatbot in a real-life situation the paths taken in the conversation does not have to follow the specific story and jumps defined by the developer. For example, when the user of the chatbot writes something that is not the same as something that was predefined in one of the stories by the developer, the Wit chatbot engine must interpret the message and decide where to start based on internal algorithms. Also, the chatbot engine can decide to continue a conversation in another story if the user says something that the engine interprets to be better suited continued in another story. A part of being a developer of a chatbot is to understand these underlying decisions and interpretations made by the chatbot engine and develop with them in mind. This can be important for the structure of the conversational data and to improve the CDV this integration with the chatbot engine should be a part of it.

Further research needs to be done on how this integration can be implemented.

Overall the design process method chosen in this research project worked well for the intended purpose of finding a novel way to visualize conversational data. However, none of the discussed issues found in the exploratory usability test have been fixed in the CDV as of the end of this research project. For future research, to get an iterative design process going, the discussed fixes should be implement in the CDV and another exploratory test conducted. Furthermore, to be able to evaluate the usability and understanding of the CDV according to the target group, a usability test on the CDV with criteria and measurable goals should be conducted. It is not possible to draw any definitive conclusions about the CDV regarding these issues from the exploratory usability test since it is only designed to further advance the implementation of the critical points identified in the initial interviews. A more structured usability test is therefore necessary.

7. CONCLUSION
In this research project, two main critical points were determined to be of importance when visualizing the conversational data from a chatbot, the jumps between stories and the loose ends. Furthermore, a need for an overview of the data was identified to see the conversational data in its entirety.

The stories in the conversational data has a hierarchical structure and was therefore visualized as tree structures in the CDV. To create a node representation of the information exchange between the chatbot user and the chatbot, the stories were divided into smaller data points. Every node in the visualization represents one information exchange between the potential chatbot user and the chatbot. Loose ends in the stories are represented with red nodes that are easy to distinguish and find amongst the regular nodes in the CDV.

To be able to visualize jumps, the tree structures in the CDV are arranged in a circular formation in 3D space. This makes the visualized jumps not overlap nodes or each other. The third dimension also add the possibility to visualize all the jumps at the same time, unlike TreeNetWiz that due to being in 2D only visualizes jumps between the current leaf nodes.

Results from the explorative usability test indicates that the visualized tree structure gave the users of the CDV an understandable overview of the conversational data. However, finding specific stories and nodes in the conversational data was identified as a problem due to inadequate information in the overview of the visualization.

The CDV is in its current state a quick prototype and a suggested start on how to visualize conversational data in VR. More iterations in the design process is needed to implement findings from the exploratory usability test and further advance the CDV.

8. ACKNOWLEDGEMENTS
I want to give a special thanks to my supervisor Rebekah Cupitt at the Royal Institute of Technology for her support and guidance throughout this research project. I also want to thank Zaidin Amiot for his valuable feedback. Lastly, a big thank you to Mobiento for supplying me with the necessary tools and facilities for this project and everyone that participated in the interviews and exploratory usability test. Thank you!
9. REFERENCES


27. Ivan E. Sutherland. 1968. A head-mounted three dimensional display. Proceedings of the AFIPS '68 (Fall,
part I)


