EXPLORING DISCOVERABLE CONVERSATIONAL INTERFACES FOR MODEL STATE CONTROL

How can a conversational UI improve the control of model state information in holographic models?

A. SIDDIQUI
University of New South Wales, Australia
a.siddiqui@unsw.edu.au

Abstract. Communication in the Architectural, Engineering, and Construction industry is a key factor when explaining and understanding ideas between team members. Conversational discoverable user interfaces have become a mainstream technology in smart devices. Applying this method to controlling the model state of architectural 3D models can change the way data is perceived and consumed by the user. This research paper will explore the principles of discoverable user interfaces and how natural language processing can prove to be an intuitive and viable solution when interacting and viewing information of a 3D architectural model. This will be developed for a holographic environment using the HoloLens as the testing device.

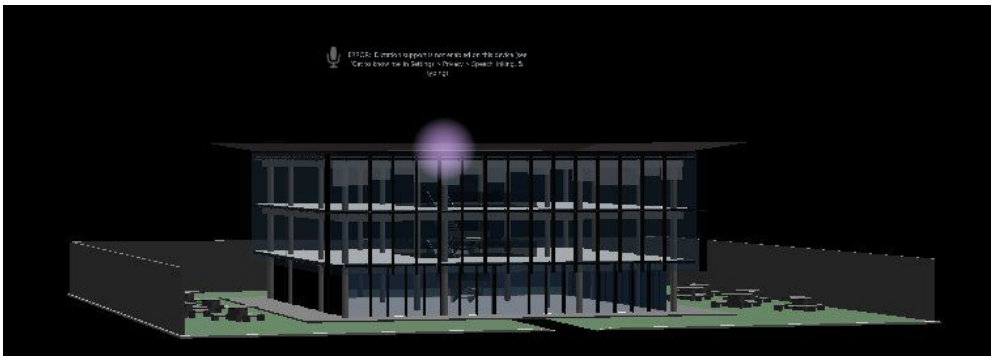
Keywords. Conversational UI, Discoverable UI, Augmented Reality, HoloLens.

1. Research Aims and Motivation

Communication and collaboration is a key aspect in the Architectural, Engineering, and Construction (AEC) industry, especially with the design review stages and within client meetings and presentations. Current primary collaboration methods use 2D monitors with the operator controlling what happens on screen who is directed by a project leader who issues verbal commands. The following research and methodology shows how Mixed Reality can prove to be a viable medium and allow all members of a team to clearly communicate and rapidly make crucial design decisions. More specifically, how conversational voice interactions can be used to control and view model state information in a Mixed Reality setting.

1.1. INTRODUCING HOLOSYNC

Mixed Reality (MR) opens new avenues of communication and collaboration in the AEC industry (Wang, Schnabel, 2008). HoloSync is a holographic model viewing application that uses the Microsoft HoloLens as its platform and will be developed in collaboration with ARUP. It tackles the issue of efficient collaboration during the design process and aims to improve the collaborative work space and understanding between peers. This MR environment allows users to view multiple design iterations through an interactive interface that is controlled through voice input and feedback. This research paper explores the viability of conversational voice input as the primary interactor and what effects it has in a collaborative setting.



1.2. DEFINING MODEL STATE

Model state refers to how a 3D model is currently being portrayed in an application. For example, if a user is only viewing the floors and columns of a model, that would be its current state. The aim of the conversational UI is to give the user control over these features through voice. The features will focus primarily on layer control and certain geometric data. Traditional architectural modelling programs tend to have a relatively static 3D model and access geometric data through other tools and menus. Model state aims to use a completely dynamic model where information is fed to the user through conversations with the UI as well as graphical overlays. The 'tools' are activated through keywords in the spoken sentences that dynamically manipulate the model. This draws more focus towards the model while also being accessible to users that don't have prior knowledge of using the application.

Removing the technical knowledge barrier allows for project leaders and clients to take control of the application without having to explain their desires to a 'middleman' whose specific role is only model coordination. Furthermore, a clear communication of ideas is exchanged between peers when the user has full model state control.

2. Research Aims and Motivation

Speech input and voice commands have been a significant method of human-computer interaction since Lenny Braum introduced the first voice recognition software in the 1970s. Subsequently, over the last decade, voice assistants have been heavily invested in by large corporations such as Google, Amazon and Apple. "Users can engage in cooperative conversations with a machine to complete a request or series of requests using a natural, intuitive, free form manner of expression." (Baldwin, Freeman, Tjalve, Ebersold, Weider, 2011). This principle is the primary motivation for creating a discoverable conversational user interface for an application that's purpose is to improve collaboration and the exchange of information between people.

3. Research Questions

Conversational UI's have already become a part of mainstream technology with the implementation of chatbots such as Apple's Siri and Amazon's Alexa. They have introduced a new method of interaction through chatting with a computer and automating the process of accessing desired

information. Current model viewing applications on the HoloLens (e.g. Google's SketchUp Viewer and Microsoft's 3D Viewer) use 3D virtual UI panels that host 2D buttons. These interaction methods are replicating similar applications that are available on desktop computers (Revit and Google SketchUp) and offer the same functionality. Voice is a natural human communication method and can prove to be an intuitive tool for controlling the state of a model in a holographic environment.

4. Methodology

The UI will aim to provide the user with the following controls over the model:

(a) Toggle visibility of multiple model elements.
(b) View shadows at different times throughout the day.
(c) View geometric statistics such as floor area.

The prototyping and incremental improvements to this method of interaction will stem from basic level user input.

(a) First iteration - User gives voice commands to trigger the desired actions.
(b) Second iteration - The voice commands use keyword recognition allowing the user to use natural sentences rather than only saying the exact sentence that the UI is listening for. In addition to keyword recognition, adding Text To Speech functionality allow the UI to talk back to the user depending on the input.
(c) Third iteration - Attempt to add conversational complexity to the UI via questions that are fed to the user, resulting in a more natural human-computer interaction process. For example - User: "Only show me the floors." UI: "Would you also like to see the floor areas?"

Creating a prototype that performs basic natural language processing that automates specific tasks will convey how a conversational UI is intuitive and effective. "Our mouth is the most effective output device we have, because obviously most people can talk faster than they type, write or make signs" (Pasztor, 2017).

[FIGURE]

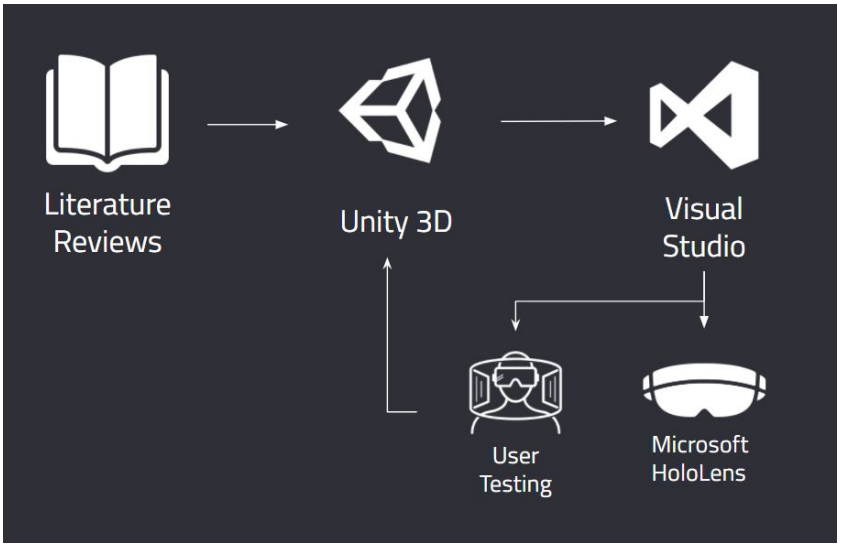


Figure 1. Prototyping Workflow Diagram

5. Background Research

5.1. HOLOLENS USER INTERFACES

5.1.1. Microsoft's '3D Viewer'

3D Viewer is a model viewing app developed by Microsoft. It allows the user to upload 3D models via their OneDrive account and gives them access to basic model manipulation tools within the application.

Their user interface uses a menu-based system that activates every time you select the model. All interaction methods revolve around virtual buttons and sliders that control the rotation and scale of the model as well as other functions. The direction of the user's head determines the position of the cursor, and the air tap gesture of the HoloLens is the equivalent of a left mouse click on a traditional computer operating system. However, the functionality and methods of interaction do not take advantage of either voice input or output. The menu consists of icons only and shows labels when you hover over the tools. It requires multiple air taps, and hence time, to perform a sequence of simple functions.

5.1.2. Google's 'SketchUp Viewer'

SketchUp Viewer is the most developed model viewing application on the HoloLens in terms of functionality. Google has linked the app with user's gmail accounts, allowing them to upload models from the desktop app. Essentially, they've made a light version of Google SketchUp that uses holograms and provides more detailed interactions such as selecting specific items within a model and accessing information about that item.

Similarly to '3D Viewer', SketchUp Viewer also uses a menu-based user interface. It has tools that allow for measurement between two points, toggle layers, selection of specific objects, and basic manipulation such as move, rotate, and scale. The menu uses virtual buttons that mimics the desktop version and does not use any voice control in the UI.

[FIGURE]

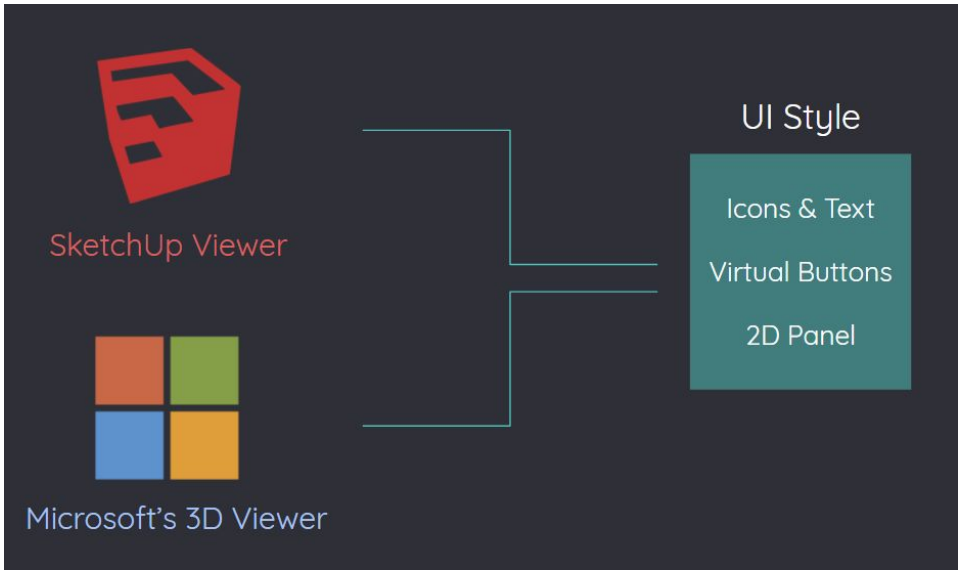


Figure 2.1. Precedents Interaction Method

5.3 CONVERSATIONAL USER INTERFACES & PRINCIPLES

Speech driven user interfaces refers to commands and lines of dialogue being exchanged between the program and the user. “Voice input is often the most appropriate mode of interaction, especially on small devices where the physical limitations of the real estate of the device make typing and tapping more difficult” (McTear, 2016). It is a natural way to combine different modes of input (e.g. multimodal interaction) to form a more cohesive and natural interface (Laakso, 2011). Conversational interfaces seem to be the future of human-computer interaction as large companies such as Apple, Microsoft, Google, and Amazon have invested heavily into speech API’s and personal voice assistants.

As computers were becoming more prevalent in the 80s, their operating systems (OS) would use menu-based user interfaces that would require a competent level of knowledge and tech savvy-ness to access desired information. (Weber. 2002). The Desktop was designed to imitate an office where the user could point (with the mouse) and choose what he/she wanted to access (Moggridge, 2006). This same system is also carried over to the Windows HoloLens OS and model viewing applications where all tools and functions are grouped in sub-menus hidden within the main menu. The air tap gesture on the HoloLens is the equivalent of clicking a mouse on a computer; the typing is done through a virtual keyboard where the user needs to airtap each individual letter. This results in much slower UI navigation as the point and click/type method is designed for hardware such as the keyboard and mouse.

5.3.1 Chatbots

Chatbots are today’s most common examples of discoverable conversational interfaces. AI such as Amazon’s ‘Alexa’ allows users to communicate with it through voice and text while also learning patterns of the owner and presents information that is related. “We are hardwired to communicate and converse with each other. Chat-based interfaces take advantage of this fact.” (Wisniewski, Fichter, 2017). It is inevitable that this method of interaction is intuitive and will take over traditional UI as technology improves, catering for all users of technology allowing them to interact with applications even if they’ve never used them.

[FIGURE]

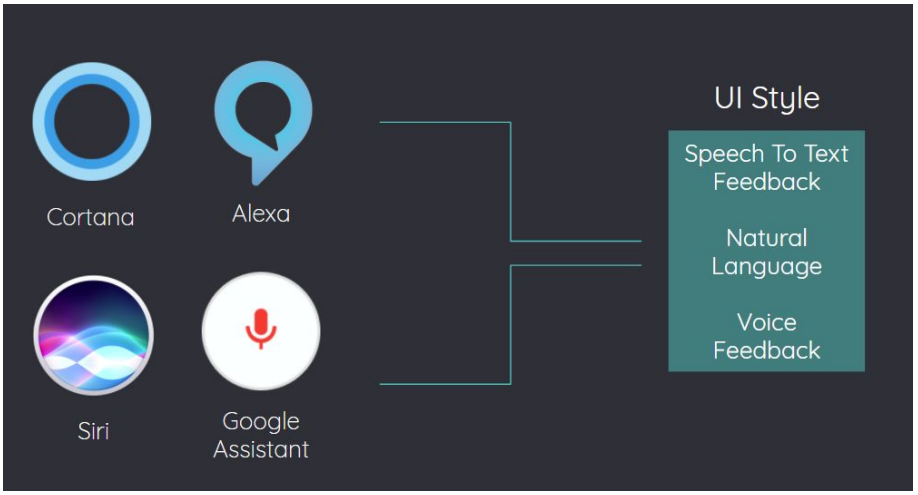


Figure 2.2. Chatbot Interaction Methods

5.3.2 Delegation Cycle

Communication in any project is key when creating an efficient work flow. Most of the time the teams that carry out these projects will consist of the project managers explaining to the technicians what they're aiming for and the specifics of the objective. Primarily in coordination and client presentation meetings, there will be a member that is designated to controlling the software that hosts the model solely because they know how to use the UI. This can cause friction in a team as time needs to be dedicated towards coordination and ensuring the ideas are understood between all parties involved. With voice implementation, the coordinator of a meeting could control the model in a seamless manner without having to explain their exact requests to a 'middle man'. "Effective delegation and the sharing of authority are vital prerequisites to the successful management of a project" (Stickney, F. A. & Johnston, W. R. 1983). By lowering the amount of people that make up the delegation cycle, less information is lost and ideas are more clearly represented to anyone involved.

5.3.3 State Ontology

Architectural modelling applications such as Revit, SketchUp, Rhino, and 3DS Max all separate features in the UI as tools. The categorisation of geometry is all within the layers tool, the geometric properties are in the data panels, and the solar analysis would be conducted through an environmental tool/plugin. This method is known as the categorisation of features for context-aware applications (Dey, 2000). It was popularised in early CAD programs and has been used up until now as it became the only way users knew how to navigate through these applications. The prototype developments in this paper explore how model state control can be more effective through voice and break the trend of this industry standard of feature categorisation, allowing inexperienced users to navigate the model and access information for collaboration purposes.

5.3.4 Voice UI Principles

Effective conversational UIs such as Siri combine voice, audio and visual elements to engage the human sensory system. “Humans have different input and output devices, just like computers. Our eyes and ears are our main input sensors...Our mouth is the most effective output device we have” (Pasztor, 2017). The voice/audio aspects are beneficial when accessing tools through commands and asking simple questions that require relatively simple answers. “...because obviously most people can talk faster than they type, write or make signs” (Pasztor, 2017). Visual elements are used for displaying data, status information (informing the user that the UI is listening while he/she is talking), and lists of items. “We are very good at pattern recognition and at processing images. This means we can process complex information faster visually” (Pasztor, 2017).The prototype development will involve incorporating both the visual and audio feedback from the UI.

5.4 VOICE COMMANDS VS. CONVERSATION

A conversational UI is not the same as an application that understands voice commands. The basic principle of a conversation is the bi-directional flow of information between two parties, in this case the user and the UI. In contrast, voice commands only requires the user to tell the application what specific event should be activated. A conversation allows for new features to unintentionally be discovered through questions being asked by both parties. For example, User: "Only show me the floor areas." UI: "Do you also want to see the floor areas?" After this short exchange, the user now knows that the floor areas can be viewed. This ability to learn from one another shows how discoverable conversational UIs can be very beneficial to creating intuitive virtual experiences.

[FIGURE]

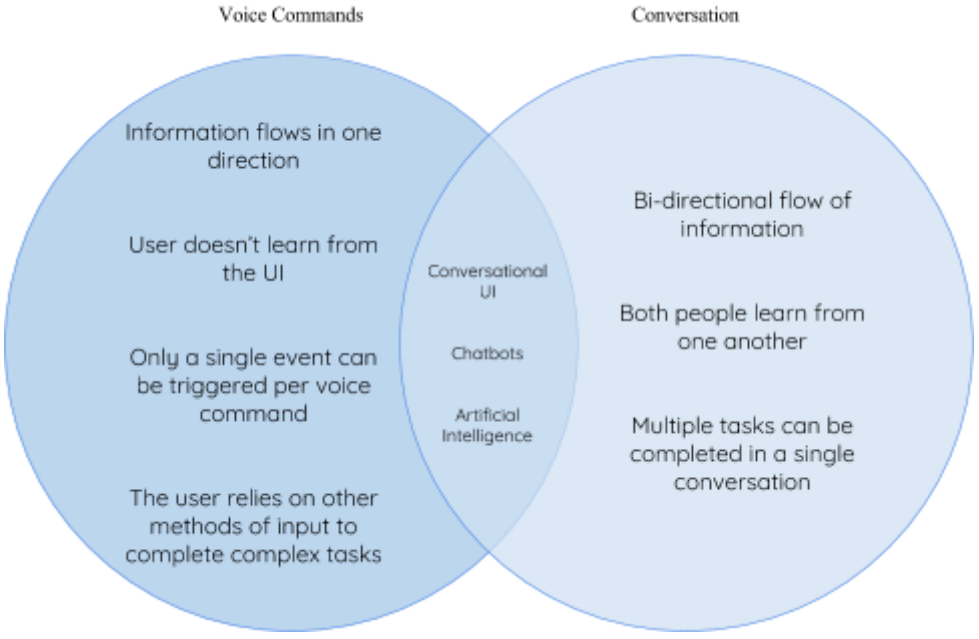


Figure 2.3. Combining human conversation with UI.

6. Case Study

When developing a conversational UI there needs to be a flow of information between the user and the application. This exchange of information also needs to advance through information, making it a productive conversation where the user learns more and can access desired model states. This paper will discuss the specifics of event triggers through voice and how other variables react to these events. All the scripts used are from the MixedRealityToolkit on GitHub and from the HoloLens Academy example projects. They have been modified to achieve what was aimed for in this project.

6.1 USING VOICE COMMANDS TO TRIGGER EVENTS

There are two primary scripts that control the recognition of voice and using that recognition to trigger an event. The recognition script is the Keyword Recognizer which converts the voice input into a format that can be understood by Unity. The Speech Input Source handles the activation of the Keyword Recognizer which can be toggled on or off through a function.

[FIGURE]

```
public void StartKeywordRecognizer()
{
    if (keywordRecognizer != null && !keywordRecognizer.IsRunning)
    {
        keywordRecognizer.Start();
    }
}
```

```
public void StopKeywordRecognizer()
{
    if (keywordRecognizer != null && keywordRecognizer.IsRunning)
    {
        keywordRecognizer.Stop();
    }
}
```

Figure 3.1. Functions that start and stop the KeywordRecognizer.

In addition, the speech input source is also how the user sets the desired keywords in the Inspector panel.

[FIGURE]

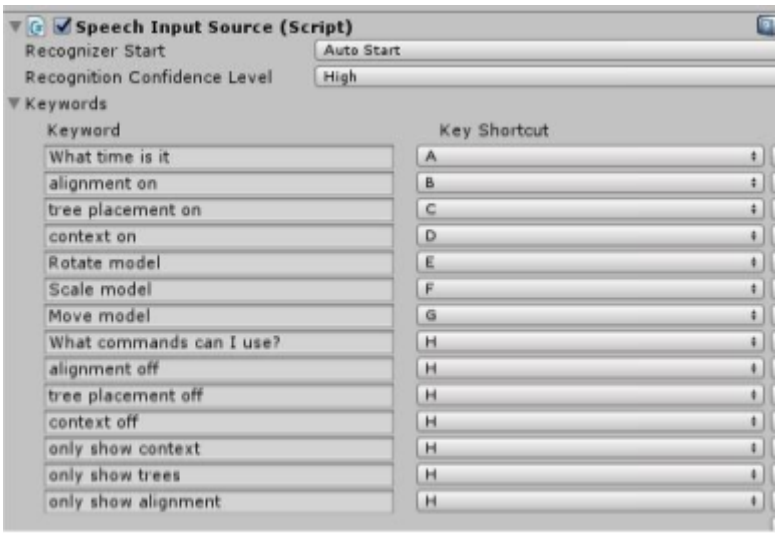


Figure 3.2. Assigning keywords in the inspector panel

The Speech Input Handler is the script that uses the recognised words to trigger an event. It is listening for voice commands set by the user to trigger an event that is also set by the user.

[FIGURE]

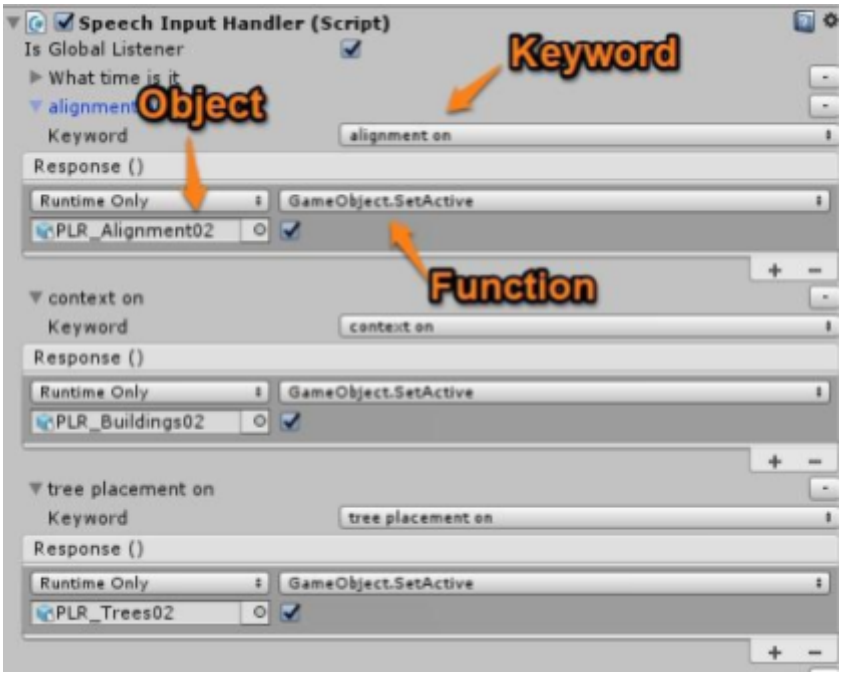


Figure 3.3. Keyword triggering a function

Figure 1.4 displays the process of assigning the keyword “alignment on” to toggle on the visibility of the alignment object. First a keyword needs to be chosen. Then the object that needs to be effected is inserted into the event dispatcher and the desired function is chosen, in this case “GameObject.SetActive” which is set to ‘true’ indicated by the filled checkbox.

Although this method works, it doesn’t allow for natural speech input as the user needs to say the keywords word for word, meaning the commands must be learned and remembered by the user while using this UI.

6.2 USING DICTATION MANAGER FOR NATURAL SPEECH INPUT

Moving on to the second iteration of this UI development, the focus is achieving natural speech input as this is the key to creating an intuitive interface that can prove to be a viable solution. Ideally, the user would be able to give the UI any sentence that includes keywords which would trigger a function. Instead of using the Speech Input Source and Handler scripts, a Dictation script will be used. The TextToSpeechManager scene in the MixedRealityToolkit includes a MicrophoneManager script that handles all the words recognised from the user.

[FIGURE]

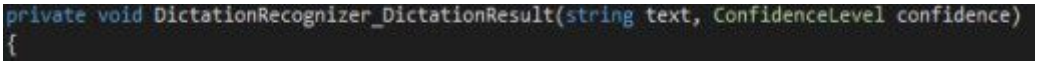


Figure 4.1. Resulting string after each sentence.

The 'DictationRecognizer_DictationResult' function is the location of a finished sentence once the user stops talking or pauses. This sentence is printed as a string and is called a 'text' variable. The sentence is then printed on the screen to give visual feedback to the user that their sentence is being recognised.

[FIGURE]

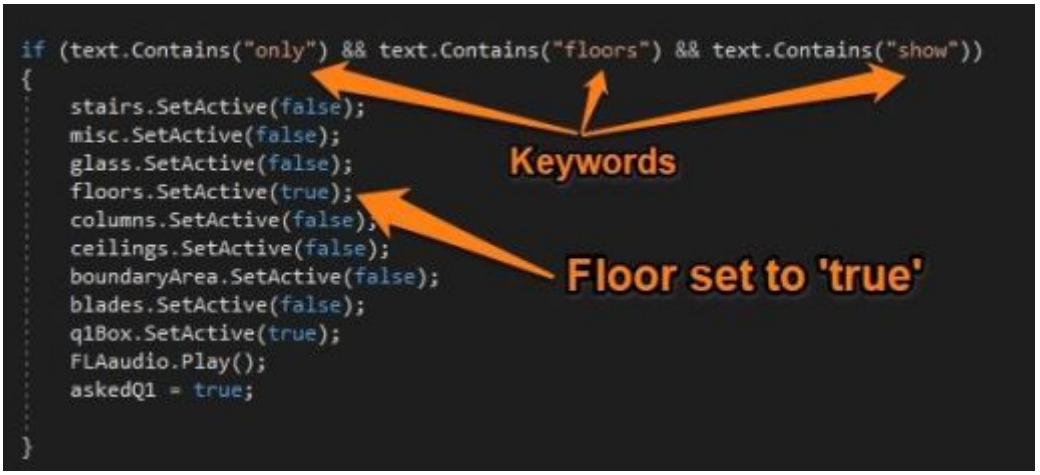


Figure 4.2. If statement triggered by keywords

The ‘if’ statement shown in figure 4.2 is within the dictation result function. If the ‘text’ variable contains any of the keywords, it will trigger the ‘if’ statement. In this case it will set the floors to be visible while hiding all the other objects. This means the user needs these three keywords in their sentence in any order causing the ‘if’ statement to be triggered. Giving the user the freedom to construct their own sentences allows for natural language input to be recognised and used to achieve the same result as the first iteration.

6.3 VOICE OUTPUT FROM THE UI

To create a conversation, there needs to be at least two output sources of information. Since this UI is conversation through a voice there needs to be an audio source in the scene that speaks back to the user depending on what the user has said.

[FIGURE]

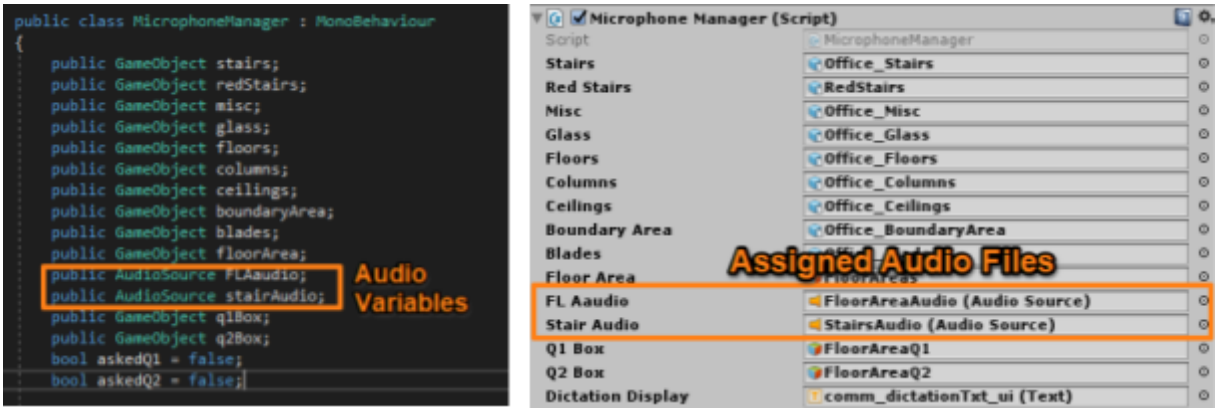


Figure 4.3. Creating an audio variable and assigning it to an audio file.

It is possible for the UI to use a Text To Speech script that can be customised for the user, however for this prototype a pre-recorded audio file has been used (shown in figure 4.3) as this is being tested in a controlled environment.

6.4 CREATING A CONVERSATION BETWEEN THE USER AND THE UI

A bi-directional flow of information can be created when combining the voice input from the user with the voice output from the UI.

[FIGURE]

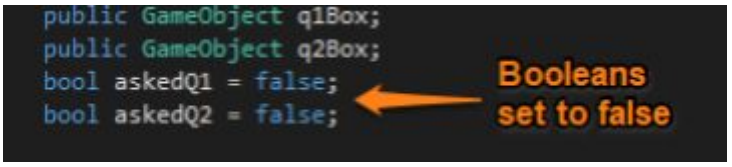


Figure 5.1. Question Booleans

Two booleans created and set to false at the beginning of the script. These booleans will determine which question is asked and what response the UI is listening for.

[FIGURE]

Figure 5.2. UI asking question and boolean toggle

When the user asks to only view the floors, the UI will play the 'FLAduio' file. The FLAudio file is a voice asking a question: "do you want to see the floor areas?" In addition, the 'askedQ1' boolean is set to true.

[FIGURE]

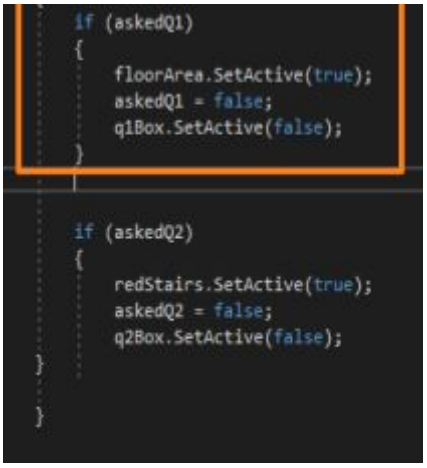


Figure 5.3. Functions if the condition is satisfied

If the user responds with "yes" and if 'askedQ1' is set to true, then the floor areas will be set to active. These nested conditional statements convey the basic concept of discoverable conversational interfaces.

6.5 VISUAL FEEDBACK

Informing the user that an application is reacting to their input is a crucial part of creating an intuitive UI. This is especially important in a conversational UI as there needs to be a natural 'human' feeling of question and answer rather than commanding a program to do something. The dictation script used from the Holo Toolkit allows for a live text preview of the voice input from the user.

[FIGURE]

```
private void DictationRecognizer_DictationHypothesis(string text)
{
    DictationDisplay.text = textSoFar.ToString() + " " + text + "...";
}
```

Figure 6.1. Visual feedback during voice input.

As the user talks, an ellipsis is displayed on the screen next to the last spoken word. The sequential construction of the sentence being printed symbolises the computer is ‘thinking’ and emulates the ebb and flow of a conversation.

Furthermore, as the UI speaks back to the user a text box of their question appears on the screen after the user says a command.

[FIGURE]

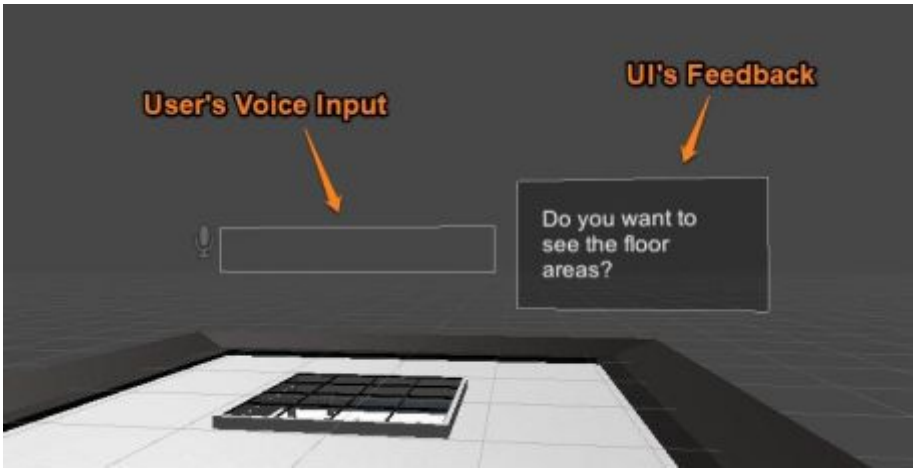


Figure 6.2. Question asked by the UI.

When the user asks to “only show the floors”, the UI replies and asks if the user wants to view more information about the element they’re interested in. This conveys the basic process of creating a discoverable UI that informs through conversation.

7. Significance of Research

The resulting research and prototype development displays the complexities and limitations of discoverable conversational UI. Although the testing has been done in a controlled environment, the prototype shows how advantageous natural language processing is in terms of ease of use and completing a series of task in one spoken sentence. This method of interaction removes the barrier of technical knowledge between the user and the application. Conversing with others through voice is a natural

communication method that is being mimicked to enable the possibility of discoverability as the user is using the application. They are given the opportunity to learn through questions rather than learning from instructions or aimlessly searching through menus and panels.

The concept of model state challenges the traditional CAD method of feature categorisation. When modelstate is combined with a conversational UI, it uses the 3D model as a dynamic source of information that can be customised; rather than a relatively static 3D model where information is accessed through other tools. Communication in the AEC industry is a key component when understanding and conveying ideas between people. This interaction method can save time and money for projects as less specialists are required to navigate the application, while also allowing clients to access unexpected or unknown information through the possibility of discoverability and questions that are asked by the UI.

Although all these factors are possible, it can only be achieved through further development of this project. The scope of this project has revealed the high amount of considerations that needs to be taken when creating a conversational UI. Human intuition and speech patterns cause for many unexpected results that can be accounted for with further user testing. With the unprecedented amount of possible input commands from the user, the path dependency would require highly advanced artificial intelligence to decrypt sentences in a way the program can understand and then react to. The user can learn from the UI, however the UI (in this project) does not collect and use data from the user allowing for a constantly improving application. This project sets the foundation for a conversational discoverable UI and displays the considerations and advantages it can have even at an early level of development.

8. Evaluation of Work

The aim of this project was to explore and create a discoverable conversational interface in the HoloLens that gave the user model state control. This was done through multiple stages of prototyping: Firstly getting voice commands to trigger events in the model. Next, adding natural language processing through keyword recognition. Lastly, creating basic conversation through discoverable audio feedback from the UI.

Although the prototypes were quite successful at achieving what they were aimed for, there were limitations that occurred throughout the process due to time constraints of the project. The user commands are limited to the keywords assigned in the script, so if the user does not mention any of the keywords, nothing will happen and they will have to ask the UI what commands it can use. This is relevant to the idea of making the entire script adaptable to multiple models, this would include automatic layer recognition and assigning keywords based on those layers. Improved keyword recognition would also include synonym library integration where the UI would search for synonyms of keywords to allow for almost any command inputs.

Furthermore, intelligent voice feedback from the UI could be achieved with further work on the project. Currently the responses are hard coded based on possible questions the user could ask. If the user asks a questions that hasn't been assigned in the script, there won't be any response. Ideally, the UI would recognise keywords such as layer names without manually scripting these details in. Thus always having a response if the user didn't know how to achieve a certain model state.

9. Conclusion

Reflecting on the research question, how can a voice controlled UI improve the control of model state information in holographic models? The resulting application of this paper allowed the user to have basic conversational interaction with the UI. Providing the user with model state control such as layers, areas and points of significance. This method achieves the same result as current HoloLens apps but in a more intuitive and effective manner. Users do not need any prerequisite skills to use this UI as voice is a natural human communication method. With more time and further development, a viable interactive method can be made that extends past the boundaries of just model state control. Ultimately, this research project has shown that discoverable conversational UIs can improve model state control in a holographic environment.

Acknowledgements

This project was done in collaboration with UNSW and ARUP Engineering.

References

2017. *Spoken Dialogue Technology: Toward the Conversational User Interface - Michael F. McTear - Google Books*. [ONLINE] Available at: https://books.google.com.au/books?hl=en&lr=&id=0kNmwa30o5IC&oi=fnd&pg=PR5&dq=conversational+user+interface&ots=zsDEMNKnfB&sig=rYpXcEmIz2ENo4wSp31Nlu-vKfo#v=onepage&q=conversational%20user%20interface&f=false. [Accessed 22 September 2017].

Michael McTear. 2016. *The Conversational Interface Talking to Smart Devices*. [ONLINE] Available at: https://link-springer-com.wwwproxy1.library.unsw.edu.au/content/pdf/10.1007%2F978-3-319-32967-3.pdf. [Accessed 18 September 2017].

Mikko Laakso. 2017. *Practical Navigation in Virtual Architectural Environments*. [ONLINE] Available at: http://papers.cumincad.org/data/works/att/9de9.content.pdf. [Accessed 17 September 2017].

Darlene Fichter & Jeff Wisniewski. 2017. *Chatbots Introduce Conversational User Interfaces*. [ONLINE] Available at: http://papers.cumincad.org/data/works/att/9de9.content.pdf. [Accessed 17 September 2017].

Frontiers. 2017. Frontiers | A truly human interface: interacting face-to-face with someone whose words are determined by a computer program | Psychology. [ONLINE] Available at: http://journal.frontiersin.org/article/10.3389/fpsyg.2015.00634/full. [Accessed 26 October 2017].

Delegation and sharing of authority by the project manager . 2017. *Delegation and sharing of authority by the project manager* . [ONLINE] Available at: https://www.pmi.org/learning/library/delegation-sharing-authority-matrix-organizations-1806. [Accessed 06 November 2017].

Project and portfolio planning cycle: Project-based management for the multiproject challenge - ScienceDirect. 2017. *Project and portfolio planning cycle: Project-based management for the multiproject challenge - ScienceDirect*. [ONLINE] Available at: http://www.sciencedirect.com/science/article/pii/0263786394900167. [Accessed 06 November 2017].

Anind K. Dey. 2017. *Providing Architectural Support for Building Context-Aware Applications*. [ONLINE] Available at: https://pdfs.semanticscholar.org/2ace/6a2f594be03a2b8f248bd95967cde6fa4784.pdf . [Accessed 31 October 2017].

Smashing Magazine. 2017. *Combining Graphical And Voice Interfaces For A Better User Experience – Smashing Magazine*. [ONLINE] Available at: https://www.smashingmagazine.com/2017/10/combining-graphical-voice-interfaces/ . [Accessed 06 November 2017].

Dean Weber. 2017. *Object Interactive User Interface Using Speech Recognition And Natural Language Processing*. [ONLINE] Available at: https://docs.google.com/viewer?url=patentimages.storage.googleapis.com/pdfs/US6434524.pdf. [Accessed 11 October 2017].

https://developer.microsoft.com/en-us/windows/mixed-reality/academy - Example Projects

https://github.com/Microsoft/MixedRealityToolkit-Unity - Example Projects