INTRODUCING ASSISTIVE ANALYTICAL AGENT FOR EXPLORATORY AND DECISION MAKING IN ANALYTICAL MULTI-USER SESSIONS

by

YASIN FINDIK

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University

July 2018

INTRODUCING ASSISTIVE ANALYTICAL AGENT FOR EXPLORATORY AND DECISION MAKING IN ANALYTICAL MULTI-USER SESSIONS

APPROVED BY:

Assoc. Prof. Dr. Selim Balcısoy
(Thesis Supervisor)

....

Assoc. Prof. Dr. Kemalettin Erbatur

....

Asst. Prof. Dr. Cengiz Uraz Türker

....

DATE OF APPROVAL: 31/07/2018

INTRODUCING ASSISTIVE ANALYTICAL AGENT FOR EXPLORATORY AND DECISION MAKING IN ANALYTICAL MULTI-USER SESSIONS

Yasin Fındık

Computer Science and Engineering, Master Thesis, 2018

Thesis Advisor: Assoc. Prof. Dr. Selim Balcısoy

Keywords: Data visualization, Tabletop, Assistive analytical agent,

Human-data interaction

Abstract

In multi-user analytical sessions, participants analyze data and collaborate toward a shared vision. These decision-making processes are challenging and generally prolonged. In this thesis, we introduce a system for facilitating decision-making in multi-user analytical sessions. Our system is comprised of a tabletop setup and an assistive analytical agent. The tabletop, as a medium for data visualization, makes data tangible, hence more comprehensible and natural to operate with. Simultaneously, while the participants interact with the data on the table, the assistant agent understands their ongoing conversations and offers extra information about the data. In addition, the agent answers the participants' questions either regarding the data or open-domain ones, supports or rejects their ideas by providing its reasoning, and preserves the productivity and the efficiency of the session by confirming that the participants do not deviate from the session's goal. Overall, by combining data visualization with the analytical agent our system accelerates the analytical process, fosters exploration through it, and guides next sessions using previous sessions' summarizations.

ÇOK KULLANICILI ANALİTİK OTURUMLARDA KEŞİF VE KARAR VERME İÇİN ANALİTİK YARDIMCININ TANITILMASI

Yasin Fındık

Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Tezi, 2018

Tez Danışmanı: Assoc. Prof. Dr. Selim Balcısoy

Anahtar Kelimeler: Veri görselleştirmesi, Masaüstü, Analitik yardımcı, İnsan-veri etkileşimi

Özet

Çok kullanıcılı analitik oturumlarda, katılımcılar verileri analiz eder ve paylaşılan bir vizyona doğru işbirliği yapar. Bu karar verme süreçleri zorludur ve genellikle çok fazla zaman alır. Bu tezde, çok kullanıcılı analitik oturumlarda karar vermeyi kolaylaştırmak için bir sistem sunuyoruz. Sistemimiz bir masa üstü kurulum ve analitik yardımcıdan oluşmaktadır. Masa üstü kurulum yardımıyla yapılan veri görselleştirme, verileri elle tutulur hale getirir, dolayısıyla daha anlaşılır ve doğal yöntemler ile çalışabilir. Katılımcılar masadaki verilerle eşzamanlı etkileşimde bulunurken, yardımcı aracı devam eden görüşmelerini anlar ve veriler hakkında ek bilgi sunar. Buna ek olarak, analitik yardımcı katılımcıların görselleştirilen veriyle alakalı ya da günlük hayata ait ucu açık soruları cevaplandırabilmekte, gerekçelerini sunarak katılımcıların fikirlerini destekleyebilmekte ya da reddebilmekte, katılımcıların toplantının amacından sapmasına engel olmaya çalısarak toplantının üretebilirliğini verimliliğini korayabilmektedir. Genel olarak, veri görselleştirmesini analitik aracıyla birleştirerek, sistemimiz analitik süreci hızlandırır, keşif sürecini geliştirir ve önceki oturumların özetlerini kullanarak sonraki oturumlara rehberlik eder.

to my beloved mother....

ACKNOWLEDGEMENTS

A paragraph does not suffice to express my gratitude towards my thesis advisor Assoc. Prof. Dr. Selim Balcısoy. I thank him for everything he has done for me, especially for his invaluable guidance, limitless support, and understanding. He constantly encouraged me to develop my own ideas and apply them, but steered me in the right direction whenever he thought I needed it.

My sincere thanks go to my fiancee, for all her love, patience, and support. I am grateful to her for keeping my sanity during the unpleasant times, for continuously inspiring me and for endlessly supporting me to chase my dreams. I also thank her for revising the drafts of this thesis.

I would also like to express my very profound gratitude to my family for the love, and faith. Without the inspiration, drive, and support that they have given throughout my life, I would not be the person I am today.

Part of my research would not have been realized without the industrious support of Erdem Kaya and Hasan Alp Boz. I am very thankful for their efforts during and after our collaboration. I also thank Erdem for his advice on the demo of this thesis.

To my lab-mates, thanks for making the time spent in the lab fun. I thank Gökhan Göktürk, Mehmet Çagri Çalpur, Ece Egemen and Emir Artar for the good times. I greatly look forward to having all of you as colleagues in the years ahead.

Finally, I acknowledge Flaticon for providing hand gesture icons used in some of the figures of this thesis.

TABLE OF CONTENTS

CHAPTER 1 1
Introduction..1
CHAPTER 2 3
Background Information..3
CHAPTER 3 7
Tabletop..7
3.1. Setups..7
3.1.1. Hardware Setup..7
3.1.2. Software Setup..9
3.2. Data Overview..10
3.3. Data Visualization..11
3.4. Properties of Tabletop System..12
3.5. Interactions..14
3.5.1. Human-Table Interaction..14
3.5.1.1. Touch Detection..15
3.5.1.2. Zoom detection..16
3.5.1.3. Panning Detection..18
3.5.2. Interaction with Analytical Agent..21
3.6. Response Time..21
CHAPTER 4 22
Analytical Agent..22
4.1. Setups..22
4.1.1. Hardware Setup..22
4.1.2. Software Setup..24
4.2. Visualization of the Analytical Agent..26
4.3. The Capability of Analytical Agent..28
4.3.1. Interactions..28
4.3.1.1. Human-Analytical Agent Interaction..28
4.3.1.2. Analytical Agent-Data Interaction..29
4.3.1.3. Analytical Agent-Table Interaction..29

4.3.3. User Identification....31
4.3.4. User Engagement....31
4.3.5. Session Coordination....31
4.3.6. Meeting Summarization....32
4.4. Response Time....33
CHAPTER 5....34
The System....34
5.1. System Design....34
5.2. Communication Between Table and Agent....35
5.3. Entire System Description....37
CHAPTER 6....47
Conclusion and Future Work....47
6.1. Conclusion....47
6.2. Future Work....48
REFERENCES....50

LIST OF FIGURES

Figure 3.1: Hardware Setup of Table Top........8

Figure 3.2: Table Surface........8

Figure 3.3: Showing Hardware Setup........9

Figure 3.4: Sample Data........11

Figure 3.5: View of the Projected Data on Table........12

Figure 3.6: An Example of Meeting Appearance........13

Figure 3.7: Giving Response to Touch Action........13

Figure 3.8: Simulation of Touching........16

Figure 3.9: Initiated Zoom-in Module and Zooming in........17

Figure 3.10: Completing the Zoom-in Action........17

Figure 3.11: Initiated Zoom-out module and Zooming out........18

Figure 3.12: Completing the Zoom-out Action........18

Figure 3.13: Initiated Panning Module........19

Figure 3.14: Panning Movement........20

Figure 3.15: Completing Panning Module........20

Figure 4.1: Hardware Setup for the Analytical Agent........23

Figure 4.2: Meeting Room........24

Figure 4.3: Demonstrating the relation between sub-programs........26

Figure 4.4: Initial Screen of First Monitor........27

Figure 4.5: Initial Screen of Second Monitor........27

Figure 4.6: Showing Desired Filter........30

Figure 4.7: Incoming Data on the Table and the Previous Button........30

Figure 5.1: System Design........34

Figure 5.2: Registration and Example Message to Register........35

Figure 5.3: Example of Sending and Receiving Message..36

Figure 5.4: Direction of the arrows shows the communication way..............................36

Figure 5.5: Starting Zoom-in Action by User...37

Figure 5.6: Zoomed-in on the Table's Map...37

Figure 5.7: Activating Zoom-out Action by User..38

Figure 5.8: After Zoom-out Action Done...38

Figure 5.9: Start Command to Pan by User...39

Figure 5.10: When the user is panning..39

Figure 5.11: Appearance of the map after a pan action is done.....................................40

Figure 5.12: Releasing Map...40

Figure 5.13: Showing branches from the selected filter which is only red and green color responses..41

Figure 5.14: After giving a zoom-in command to the analytical agent..........................41

Figure 5.15: Simple Filtered Data..42

Figure 5.16: Sophisticated Filter..42

Figure 5.17: Request to Bring Map..43

Figure 5.18: Requested Map on the Table..43

Figure 5.19: Properties of the Touched Area..44

Figure 5.20: Detailed Information about the Touched Area...44

Figure 5.21: Simulating the Current Speaker and Others around Table.........................45

Figure 5.22: Example Question and its Answer...46

Figure 5.23: Sample Conversation with the Analytical Agent.......................................46

CHAPTER 1

Introduction

When babies are born, they explore the world around them not by using a mouse or a keyboard but by touching objects with their hands. Hence, the most natural way to interact with data is by seeing and touching it. Nonetheless, traditional human-computer interaction techniques do not fully meet this necessity. Adaptations of data visualization techniques on digital screens aim to facilitate the analysis of complex data. The purpose of data visualization is to help people derive meaning from data by stimulating the sense of sight. While a large number of studies have focused on assisting the analytical process of complex data by visualization, considerable limitations still persist. These limitations are accentuated especially when collaborative work is involved in the analytical process.

Just as infants' innate instinct to ask caregivers about the surroundings, people learn more the data, not only touching and seeing it, but also with the help of a subject that has knowledge about the data. While this subject may be a human being, it can be a computer too. With current developments, computers are increasingly taking the role of assistants in our daily lives (e.g. Siri, Alexa, Cortana). Such assistant with domain knowledge is beneficial in collaborative analytical processes, as well. During these processes the assistant can answer the questions asked by the participants regarding the data, it can also support/reject their idea by giving an explanation after understanding their conversation. Moreover, it can be crucial in preserving the productivity and efficiency of the meeting, by making sure the participants do not deviate from the meeting goal.

In this thesis, we aimed to create a system that both visualizes the data on a tabletop setup and also assists the meeting participants during the analytical process. Participants using the platform interact with the data by touching the table and also communicate with an assistive analytical agent. In this way, they can understand the data in more detail and obtain analytical results faster.

The setup consists of three main parts: a tabletop visualization system, an assistive analytical agent part, and a communication channel between the systems. The tabletop system comprises a physical table to provide tangible interactions, a Kinect to perceive the environment, and a projector to visualize the data on the table. The analytical agent part contains two monitors to visualize agent's state, and a microphone and Kinect to obtain real-time audiovisual data of the meeting.

This thesis is structured as follows: in Chapter 2 we present background information about tabletop setups, assistive agents, and projects that include both of them. Tabletop part of the system is described in Chapter 3. Also, detailed properties of the tabletop and its construction are described in this chapter. Chapter 4 describes the assistive analytical agent part of the system. The implementation, technologies used, and the capabilities of the agent are specified in this chapter. Whereas, Chapter 5 gives more detail of the overall system setup – both hardware and software and how tabletop and assistive analytical agent operate together. Finally, Chapter 6 consists of the conclusion and future work.

CHAPTER 2

Background Information

The main concepts examined in this work are: data visualization, user interfaces, assistant agents, and their effects on perception and teamwork. A detailed literature review on these issues is presented below. As can be seen, studies in the field of physical visualization are still at the maturity level and many of them do not integrate with any data analysis system. On the other hand, research about assistive systems is also presented. In the field of perception and collective work, only important research on the field of this area is covered.

Physical visualizations put the data into a physical form instead of converting them into pixels in the virtual world. People have been using the small objects that they made from the clay as physical visualization since ancient times [1]. Nowadays, these visualizations have gained popularity as data sculptures [2]. These data sculptures are art oriented, hence are used to express emotions rather than to facilitate the comprehensibility of the data. These data sculptures come out in different varieties from 3D maps [3] to jewelry [4]. Both in industrial and academic worlds, we can see a few previous works that have come to the forefront. The reorderable matrices [5] Bertin created in 1970 can be an example of this.

The last examples below focus on the data visualization part rather than the art part. Data visualization, in contrast to artistic expression, aims to convey unbiased information about the data to people and to create a foresight. For a long time, physical data visualization remained in the shadow of virtual data visualization, especially due to the cost and time of production. This has recently changed with the development of two main research areas. The first area is the digital production which has flourished with the advent of a 3D printer. With the rapid growth of the field, it is predicted that soon human organs will be printed by 3D printers and will be transplanted to the patients [6]. Hence, research on physical data visualization using such digital production techniques is gaining momentum day by day [7]. As the second area, the increased attention in the area of touchable interfaces [8], [9] also paves the way for physical data visualization. The vast progress experienced in these two fields shows that physical data visualization will become easily reproducible in the coming periods.

Along with the developments described above, recently significant studies have been conducted in the field of physical data visualization. For the first time in 2013, Jansen and Dragicevic carried out user experiments on the success of the physical data visualization. These experiments demonstrate that the resulting physical data visualization outperform certain areas of virtual visualizations [10]. Moreover, another study in the same year showed that physical visualizations could be used for analytical tasks [11]. Jansen and Dragicevic's later works presented a pipeline for different visualization techniques beyond the screen and created a path map for the creation of such visualizations [12]. In 2014, Saiganesh has developed a program for a rapid production of physical visualizations and has taken an important step in using the developing technology for the production of physical visualizations [13]. In addition to these academic studies, some companies use physical visualizations as part of their daily work. General Motors monitors the faults in vehicle production by creating 3D visualizations as a toy block [5]. Finally, the use of physical visualizations by academicians like Hans Rosling in their presentations [14] has also boosted the popularity of physical visualizations.

Studies in educational sciences and developmental psychology have shown that physical objects enhance people's ability to learn and understand [15], [16]. One of the main goals of this work is to increase the use of this innate learning motivation of humans, to facilitate visualizations understanding and to foster interaction with it. Almost all of the visualization tools to this end require prior visual knowledge and non-direct interaction, such as a mouse and a keyboard. For these reasons, these tools can only be used by the end user after a long learning period. The visualization part of this work aims to make data visualization easier for the end user to comprehend and interact with it.

A study published in 2014 explains collaborative work on a reflective tabletop [17]. The main feature in this work is to prevent different users from seeing each other's personal work screens. The simplicity of such tables is now being used in recreational and educational applications. Reactable [18], a musical product, is an important work that demonstrates the development of this subject over the years.

On top of its visualization, this work aims to facilitate the interaction with the data by introducing an analytical assistive agent. The role of the agent in this scenario is to respond to the subjects' questions, either regarding the data or open-domain ones. It also tries to extract the meaning of the conversation among the subjects and present related schemes/diagrams or suggest different ideas. This way the agent complements the data visualization.

By using speech as a medium of communication, conversational agents are increasingly being used as daily assistants. The most famous ones being: Apple's Siri [19], Amazon's Alexa [20], Google's assistant and other systems used in smart homes [21]. The common usage of these systems is as reminders that facilitate daily life and daily chores of people.

An assistant can be physical (i.e. robots) or virtual (i.e. virtual agents). Some applications of robots as assistive technologies, mainly developed for physical support, are: smart wheelchairs, walking assistants, external skeletons and robotic butlers [22], [23], [24]. Communication devices and robotic pets [25], [26] are developed for social purposes. Whereas, reminder devices and sensor monitoring platforms provide cognitive assistance [27]. Studies [28], [29] show that robots can be developed to assist with getting out of bed, taking a shower, walking, going to the toilet and house maintenance. [30], [31] have demonstrated that robots advanced to converse with users are beneficial since they invoke human social behaviors and help communication.

Bickmore et al.'s study [32] experiments with a virtual agent. It evaluates the responses of elderly users to relational agent Laura. Laura's aim is to give health advice to the elderly. The main purpose of the agent's interaction is to establish a relationship with the users along with social dialogue, empathy, humor, and self-disclosure. Findings of the study demonstrate that the agent was considered by the participants as a speech partner on health and health behavior. It was also considered as reliable, friendly, and a good health consultant.

In the aforementioned studies, it was emphasized that the assistant agent could help people in various ways as a physical robot or as a virtual agent. The focus of the assistive analytical agent in this study is to monitor the analytical process, speed up it if necessary, and increase its productivity by analyzing the meeting environment, extracting meeting notes, and understanding users' conversations. Cognitive Assistant that Learns and Organizes (CALO) [33], [34] is an example of research on such analytical agents. The authors of CALO argue that automated summarization of the meeting and issuance of the decisions increased the efficiency and improved the productivity of the meeting for participants and non-participants. Whereas, [35] highlights the contribution of the assistant agent for the organization of the meetings where participants could attend remotely. Ehnes [36] provides an example of the system including both tabletop and a supporting agent. However, this study's aim is not to assist participants about the analytical process. Its purpose is to understand participants discussion and demonstrate related documents on top of the table.

Developing a cooperative platform is another motivation for this work. In the literature, data analysis is defined as a task where two or more people work together, rather than a single person task [17]. Traditional computer software usually does not support collaborative work, but in recent years a plethora of studies have focused on the collaborative work [37], [38], [39]. The success of such platforms is largely dependent on the participants seeing each other during the collaborative work [40].

CHAPTER 3

Tabletop

This chapter thoroughly covers the tabletop part of the setup. Section 3.1 explains system installations for both hardware and software. Section 3.2 and section 3.3 describe the data and the tools used for its. Afterwards, section 3.4 explains the tabletop system properties such as multi-user support, ease of use and its responsiveness. Whereas, section 3.5 describes the human-tabletop and tabletop-analytical agent interactions.

3.1. Setups

3.1.1. Hardware Setup

The hardware setup of the tabletop part is depicted in Figure 3.1. The setup is comprised of a desktop computer, Kinect for Xbox One [41] that serves as an RGB-D camera, an HD projector, and a wooden table without any sensor or electronic parts.

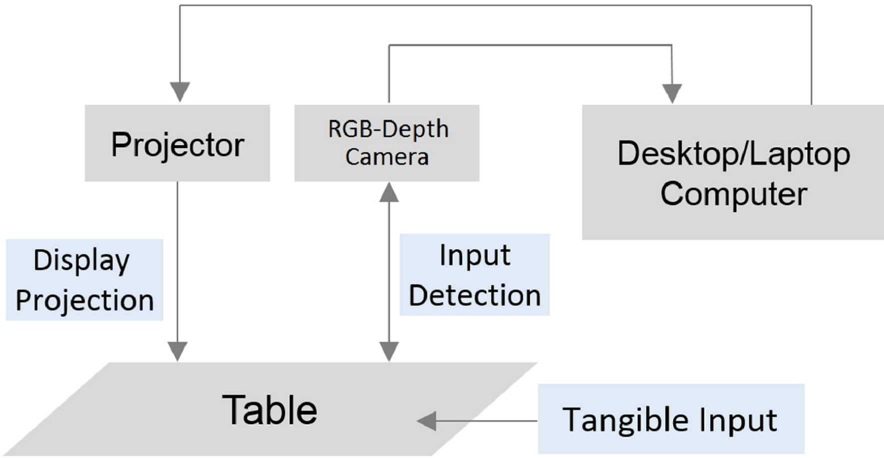

Figure 3.1: Hardware Setup for the Tabletop

The bare wooden surface of a table with no additional electronic appliances settles in the middle of the meeting room. The table height should be adjusted at the most comfortable position for the attendee users to touch. The table does not need to have any other additional property.


Figure 3.2: Table Surface

To establish the hardware setup of Kinect and projector, there are several important issues to consider. The most important one is that the Kinect and the projector must be parallel to each other and perpendicular to the table. The projector is positioned on top of the table. However, the distance between the table and projector depends on the resolution of the projector. Also, the projector should be calibrated so as to project its contents (a map in our case) onto the table. Kinect cannot be very far or extremely close to the table, approximately the distance between table and Kinect should be one and a half or two meters. The reason behind this is the limitation of the Kinect. When Kinect's distance exceeds the distance threshold, the noise of the depth sensor increases. Because of that noise, we cannot decide whether the user is touching the table or not. On the other hand, if Kinect is excessively close to the table, then the system cannot resolve who is touching table. Hence, the distance is a parameter to be tuned accordingly in this setup.

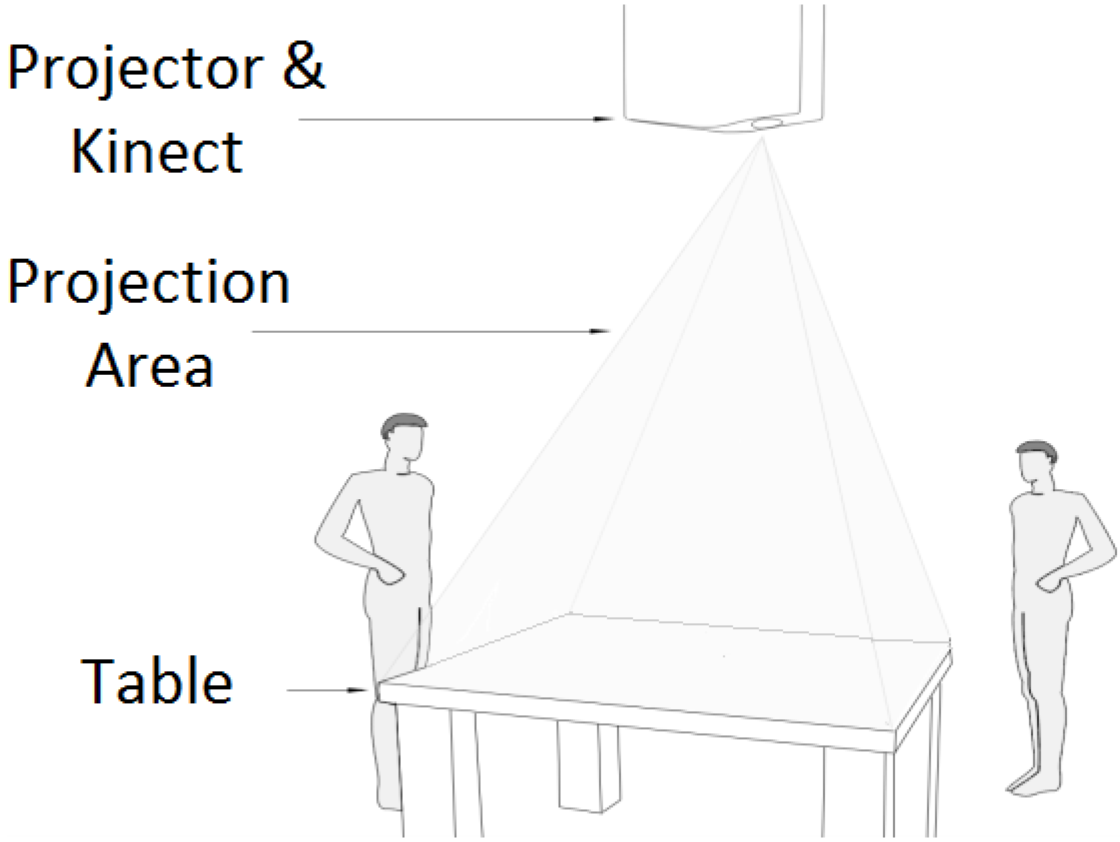

Figure 3.3: Showing Hardware Setup

3.1.2. Software Setup

A Processing [42] application processes the interaction for the tabletop part. All its user input comes via the RGB-D camera, then its output is sent to the projector. Processing is an open-source software and environment built for the media arts communities. Hence, with a little coding background, coding with Processing is straightforward. In addition, a vast number of image processing and Kinect libraries is available. These libraries facilitate extracting and processing depth and infrared images from Kinect.

Whereas to visualize the projected data, we use an open-source JavaScript library, Leaflet [43]. This piece of software is efficient and well-documented. We use Leaflet to display friendly interactive maps to users. Due to its efficiency, the latency is not noticeable even when multiple users interact with the application concurrently.

3.2. Data Overview

In this system, we use data from an insurance company. This data gives information about the branch offices of this particular insurance company, along with their location (latitude and longitude). Data are separated into several segments, namely governance, performance, and response.

Governance segment indicates the relation between the change in the total premium volume of the head office produced by the product group and the premium volume changes performed in the same outcome group at the branch office. Performance segment displays how close a branch office is to its estimated potential. The potential metric is calculated by the rate of actual production to the potential derived from the premiums of similar branches. Finally, the response segment is an indicator that shows how long branch office has reacted and accommodated itself to the change in sales strategies of the head office based on product groups.

Each segment has its own sub-segments (color segmentation) categorized as red, yellow and green. Red displays segment value that is lower than the average of the office, yellow color shows average and green specifies greater than average.

Branch_Code	Governance_Segment	Response_Segment	Performance__Segment
3452	Neutral	Neutral	Normal Performance
3589	Positive	Positive	Low Performance
3802	Positive	Positive	Low Performance
3890	Neutral	Neutral	Normal Performance
3901	Neutral	Positive	Normal Performance
5025	Positive	Neutral	Low Performance
5049	Positive	Neutral	Normal Performance
5118	Neutral	Neutral	Normal Performance
7146	Positive	Positive	High Performance
7227	Positive	Positive	Low Performance
7328	Neutral	Neutral	Low Performance
7413	Positive	Neutral	Low Performance
7513	Negative	Neutral	Normal Performance
7514	Positive	Neutral	Normal Performance
7883	Neutral	Neutral	Normal Performance
7885	Neutral	Neutral	Normal Performance
7895	Positive	Positive	Normal Performance
8071	Neutral	Neutral	Normal Performance
8312	Positive	Positive	Normal Performance
8505	Positive	Neutral	Normal Performance
8525	Positive	Positive	Normal Performance

Figure 3.4: Sample Data

While we use another table to map branch codes to their longitudes and latitudes, we cannot share it due to a privacy policy.

3.3. Data Visualization

Data provides longitude and latitude information of the insurance company's branch offices. Visualizing the location of the branches on a map projected on the tabletop rather than just showing a piece of an excel sheet is a more effective way to comprehend data. By providing users with the opportunity of seeing all branch offices on the map simultaneously, they can readily figure out the relation between any sub-segment values of the branch office and their environmental factors. In addition, users can form opinions on how branch offices interact with each other or they can explore other undiscovered relationships from the data.

Fundamentally, data is demonstrated on the bare wooden surface of a table with no additional electronic appliances with the help of the projector. Initially, governance segmentation is shown with entire color segmentation on the map. To change the foremost segment, three filter options according to their main segmentation (mentioned in the data overview section) are available top-right of the map. However, users can see only one filter at the same time, if there is no assistive analytical agent effect. Besides, users can decide which kind of color-segment(s) will be shown. Different from main segmentation, all color segmentation or each two of them or single color segment can be shown at the same time on the map. Visualized data is shown in Figure 3.5.

Figure 3.5: View of the Projected Data on Table

3.4. Properties of Tabletop System

The tabletop system can correspond to at most six users at the same time. This limitation not only depends on the table size but also the Kinect used by the analytical agent, which also can detect only six different bodies at the same time. The system is able to handle entire touch action independent from the user's settlement. In other words, users can interact with the table from every position. Example of the meeting modeling is demonstrated in Figure 3.6.

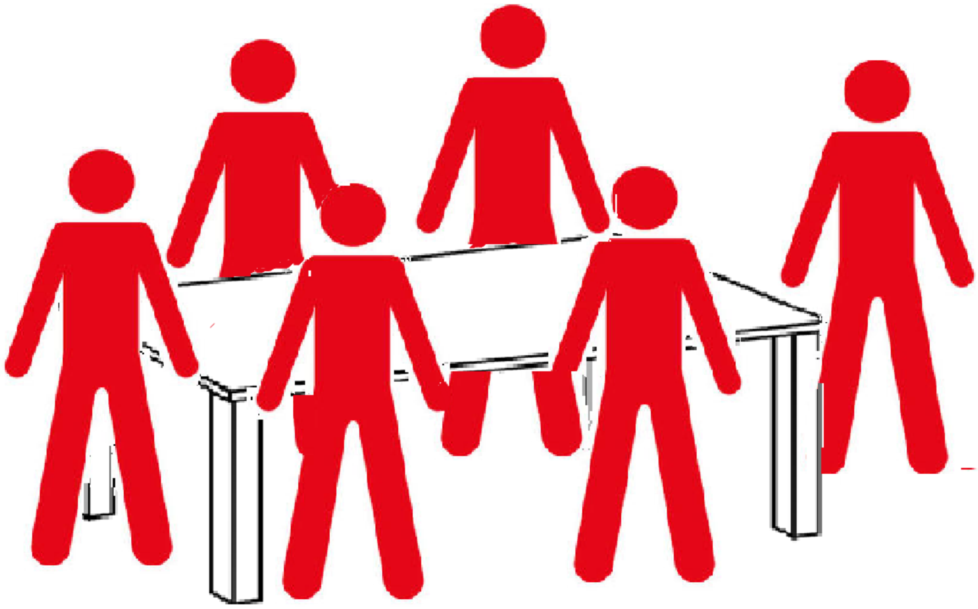

Figure 3.6: An Example of Meeting Appearance

If different users touch the table at the same time, the tool will show the properties of the touched locations for both users. To hide information shown by the system about the touched location, the user should touch the same location again. Thus, the application can deal with multiple user touch and also prevent information pollution from the table according to users' choice. Figure 3.7 shows an example of the touching table.

Figure 3.7: Giving Response to Touch Action

To use the tabletop system, attendees do not need to be instructed. They should only know that the data shown on the table is touchable and the hand gestures for zooming in/out and panning. Users do not need additional time to learn how to use application. This provides an important advantage to users that they can simply adapt to the tool and play with the data [44].

Research shows that interactions with the data by using touch and hand gestures make data easier for people to understand and explore rather than depending solely on conventional inputs, as mouse and keyboard. Touch and hand gestures address the users' perception naturally [9].

The responsiveness is another significant property of the tabletop system. After giving a command to the system such as touching or the other actions, the system responds instantly to the users' command(s). Latency is the imperative point for this kind of interactive systems. Users should not spend much time to get an answer or result from their actions [45].

3.5. Interactions

3.5.1. Human-Table Interaction

The RGB-depth camera has the capability to generate depth and infrared maps of a given view area with high frequency. A thread continuously scans and processes these infrared and depth maps in order to locate and track the changes happening in the view-port. For given two points in time t_0 and t_1, let us assume that the depth (D) and infrared (I) maps captured by Kinect are D_0, I_0, D_1, and I_1. The delta images of depth (D) and infrared (I) maps are calculated as $|D_0 - D_1|$ and $|I_0 - I_1|$, respectively. The delta images are binarized with respect to a threshold (τ) and merged into a composite delta image C with the $(D > \tau) \vee (I > \tau)$ operation. Composite delta image (C) is fed into the finger detection module developed by utilizing Processing FingerTracker library [46]. After elimination of small defected contours, a vector of the fingertip point locations and major contours are returned by the module. The index finger of users is considered as the one used for the interaction (e.g., pressing button) [47].

Note that we cannot use the hand detection function of the Kinect since to complete this Kinect needs to find a body in its field of view. Because of our Kinect's location, Kinect cannot detect bodies.

3.5.1.1. Touch Detection

Whenever the index finger's location (i.e., a pixel) is updated by the finger tracker module, the depth values of the neighbor pixels around the index finger location are aggregated into a single average depth value (df). Similarly, on the base depth image (Db), the average depth value of the pixels that are in the neighborhood of index finger location are averaged into a single value (db). If the difference $|df - db|$ is below a certain threshold, the index finger is considered to be touching the table. The aggregation process has been a necessity due to the relatively high level of noise of the data acquired from RGB-depth camera [47]. After user touch is detected, according to its location, the application gives a proper response.

Users can touch special points on the table. As specific locations, there are two different actions the application will take. One of them is filter locations. If the user wants to change the filter, he/she can touch corresponding check-box. The other one is the previous button which can be only activated by using the analytical agent. As the name of the button implies, it will reverse a previous step.

The simulation of the non-special points touched on the map is shown in Figure 3.8. This section also includes other simulation according to different user events. Moreover, actual responses of the events are given in chapter 5 in detail.

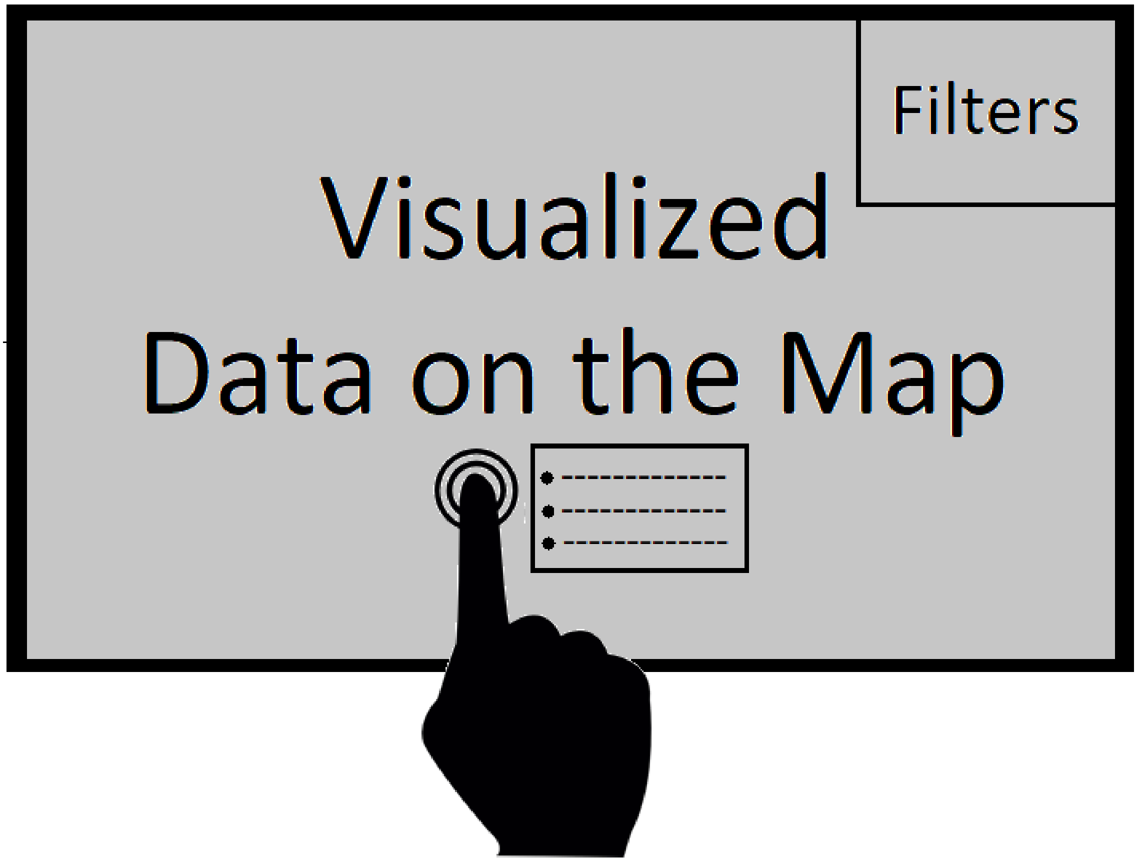

Figure 3.8: Simulation of Touching

3.5.1.2. Zoom detection

Whenever a user extends two arms over the table, finger tracker module finds these two arms as fingers. After depth calculation same with the touch detection part, if the difference $|df - db|$ is above a certain threshold (15-20 centimeters) zoom module will be activated. Changes in the distance between the arms (as a finger) in time determine zoom type; whether the user wants to zoom in or zoom out. Figure 3.9 and Figure 3.11 show activating zoom in and out modules.

If the distance increases over time, it refers to zoom-in. As the user starts to bring his/her hands closer together, the tool begins to zoom-in according to the distance between the hands.

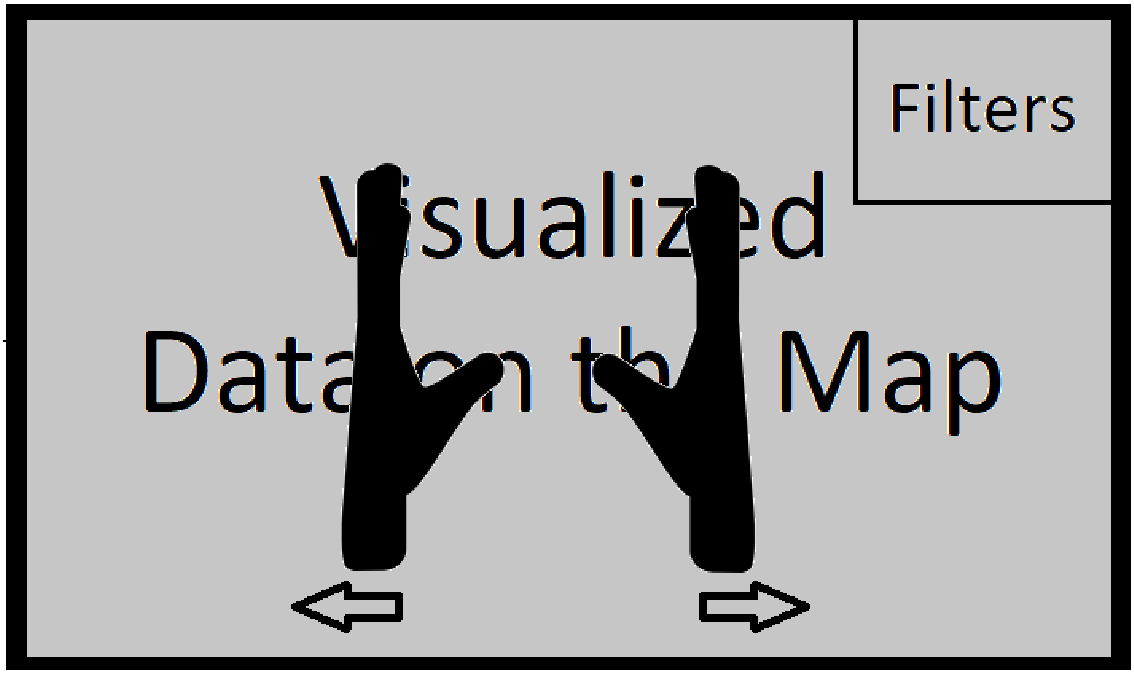

Figure 3.9: Initiated Zoom-in Module and Zooming in

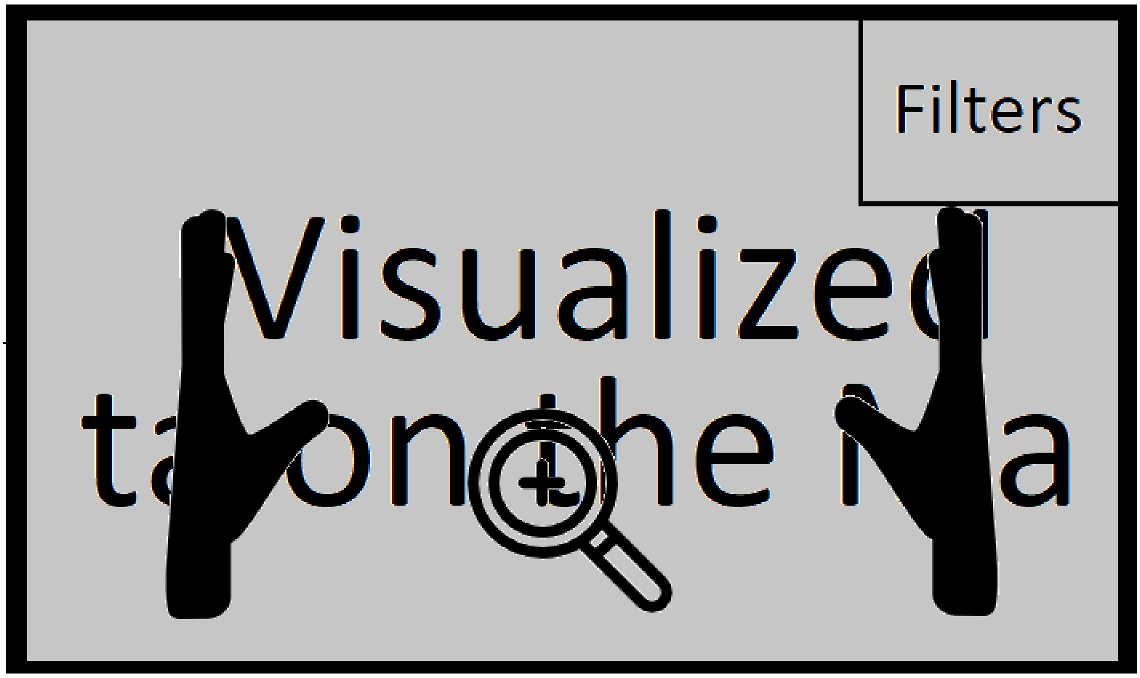

Figure 3.10: Completing the Zoom-in Action

If the distance decreases over time, it starts to the zoom-out module. As the user starts to move away his/her hands from each other, the module begins to zoom-out with respect to the distance between the hands.

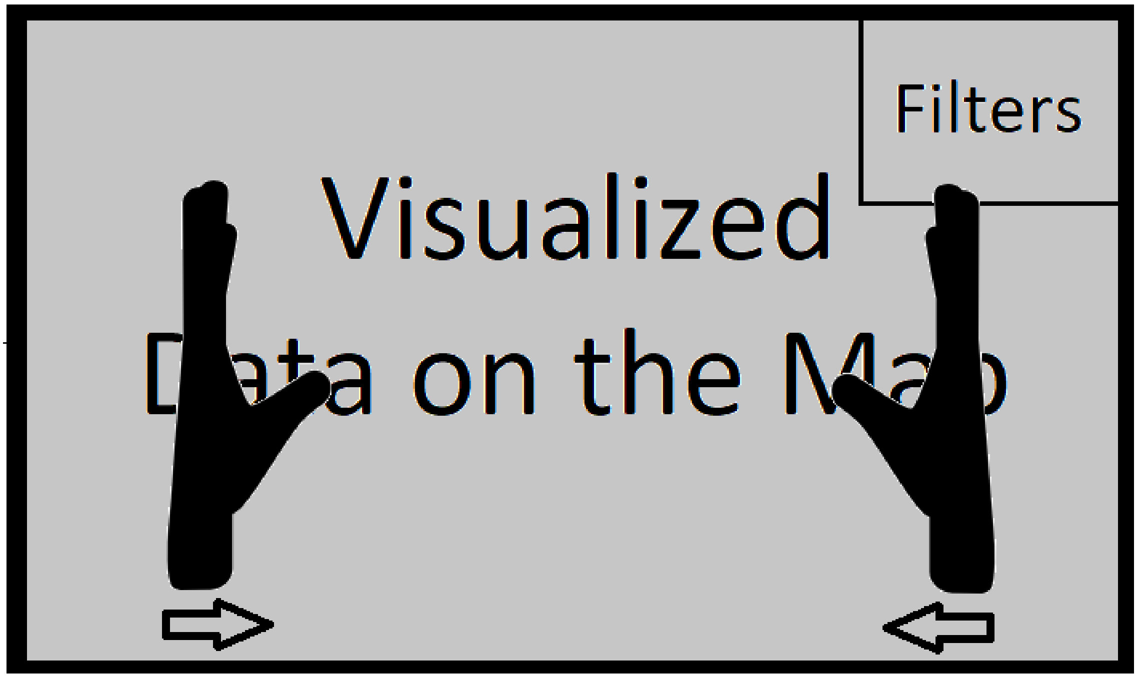

Figure 3.11: Initiated Zoom-out module and Zooming out

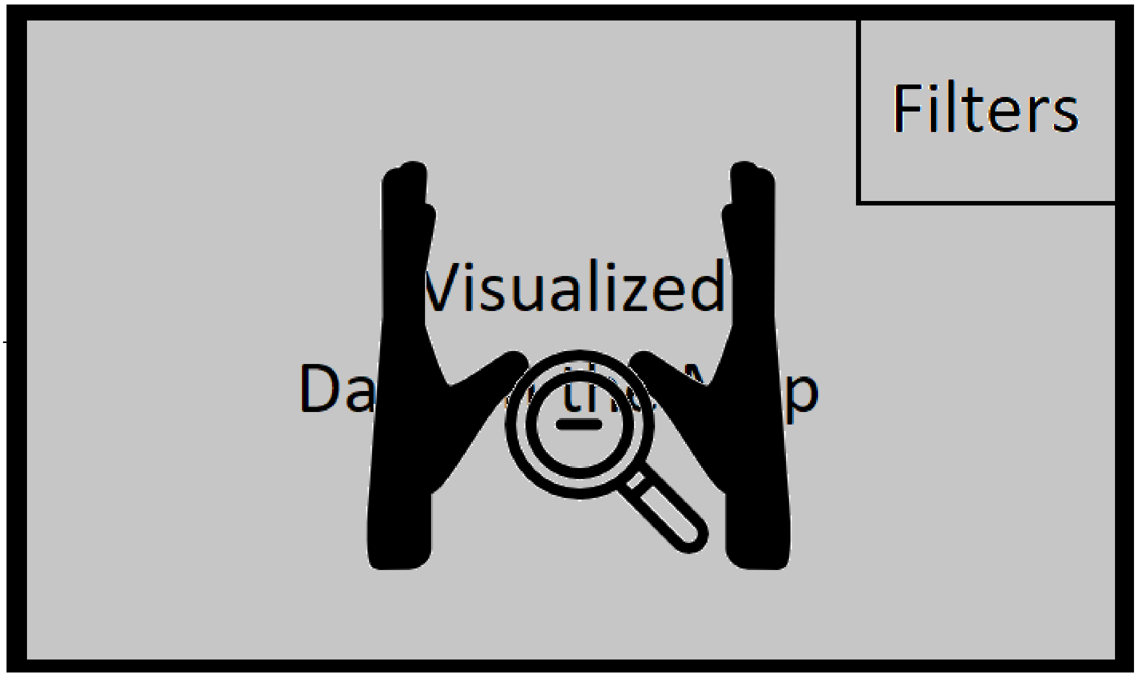

Figure 3.12: Completing the Zoom-out Action

3.5.1.3. Panning Detection

Whenever a user opens his/her hand fingers over the table, finger tracker module finds fingertip locations. After depth calculation same as with the touch detection part, if the difference |df − db| is above a certain threshold (15-20 centimeters) and the total amount of available fingers is five, panning module will be initiated. In other words, starting panning module's signal is seeing five fingers over the table with the defined distance from the table. After initiated panning module, when the user makes her/his hand a fist, panning module will grab the map. Since the user grabs the map, the focus point of the map will be rotated with respect to the user's fist movement. This rotation arrangement will be continued until the user will release to map. To release the map, the user should make the fist into an open finger position again. Consequently, map focus will be changed according to the user's hand movement.

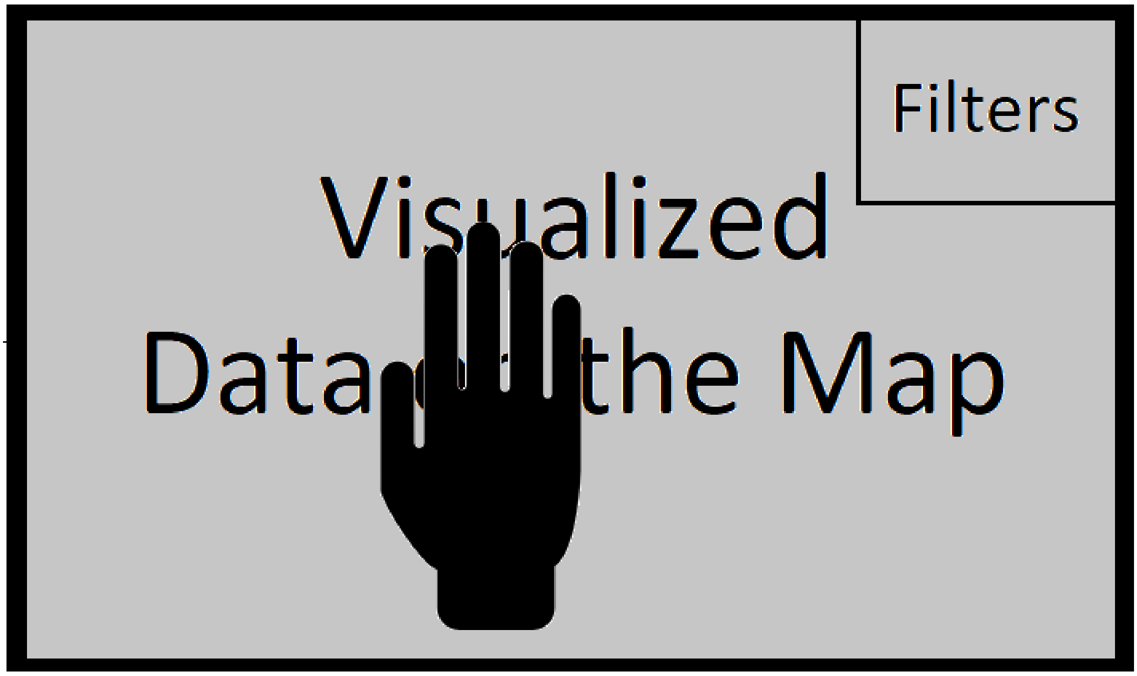

Figure 3.13: Initiated Panning Module

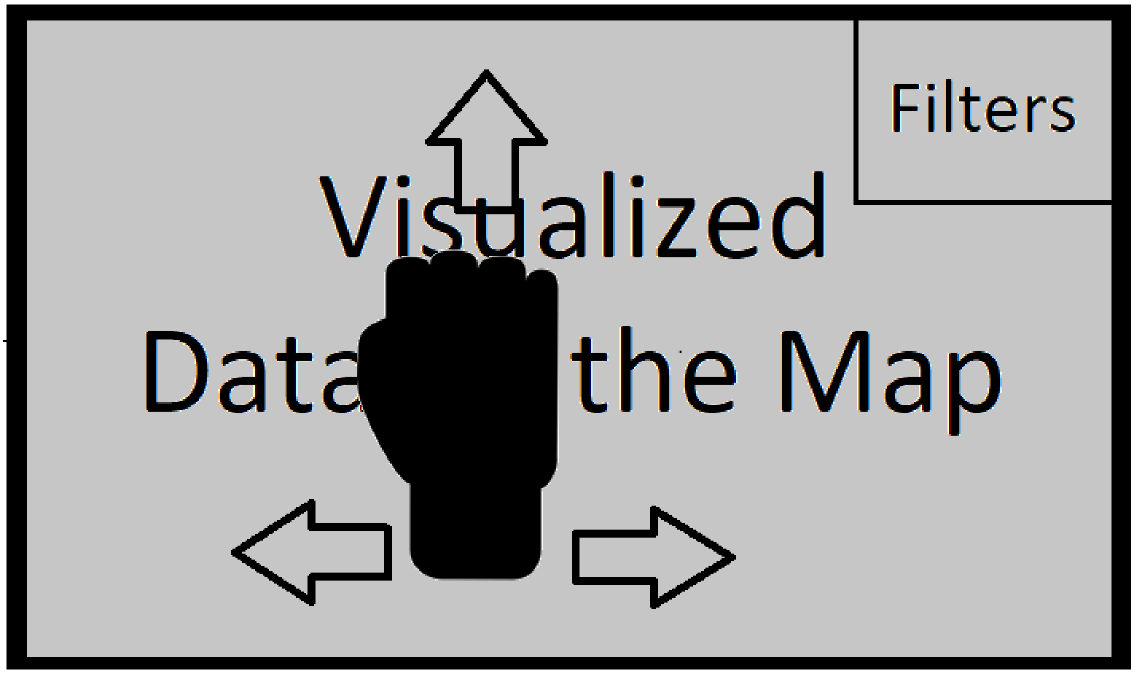

Figure 3.14: Panning Movement

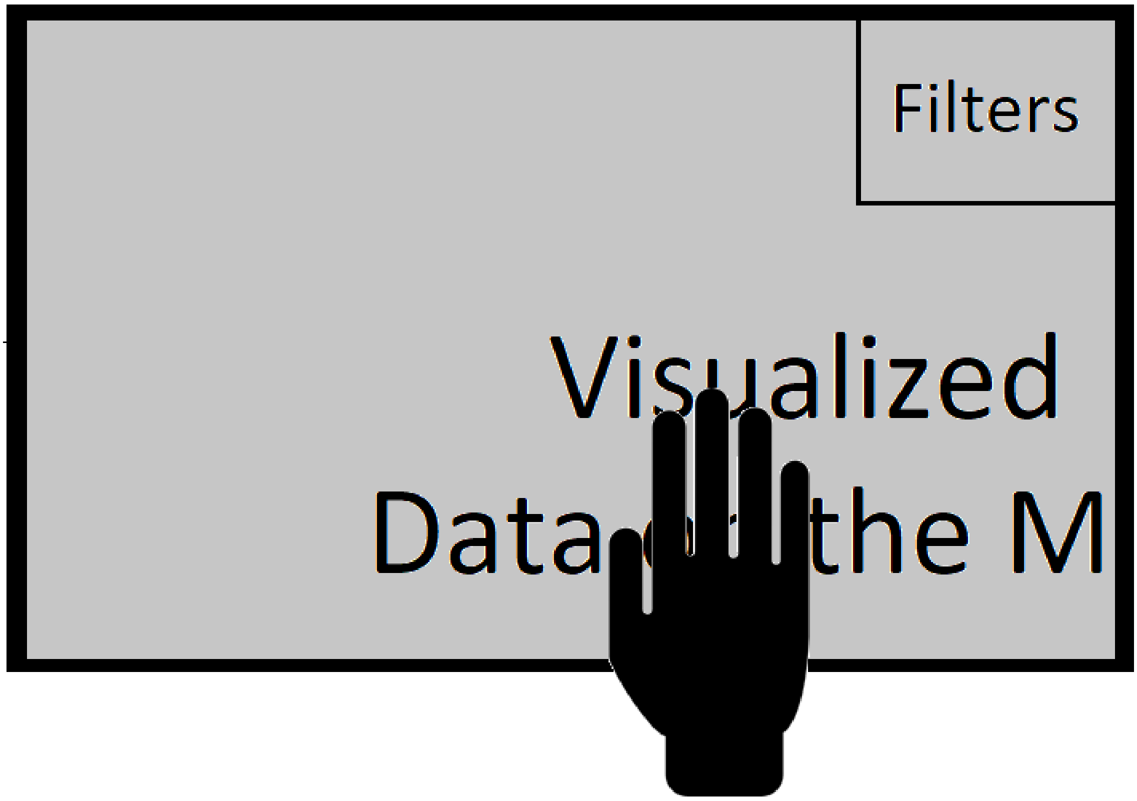

Figure 3.15: Completing Panning Module

3.5.2. Interaction with Analytical Agent

The tabletop system is also involved in the cooperation with the assistive analytical agent using an implemented socket server. Detailed information about the socket server implementation is provided in chapter 5. The touched position on the table, zoom in/zoom out actions, panning, and their timestamps are sent to the analytical agent using the socket server. The analytical agent can predict the user's point of interest using this information. Then, it can realize which user needs to be motivated or which user has deviated from the topic and slows down the analytical process. This way, it can motivate people or stop irrelevant speech in session. Technical details about analytical agent are given in chapter 4.

In addition, the analytical agent gives more detailed information about touched data on its screen such as comparing touched data with whole data from several points of view and explaining their meaning to guide the users.

3.6. Response Time

Kinect's frame rate ranges between the 15-30 FPS depending on the lighting conditions. When users initiate an action such as panning, to detect it, the system takes on average 200 ms. However, the total time of the action depends on the user's hand movements. The latency of the changing the map according to actions is less than a second. Nonetheless, we provide also a timing threshold for the hand gestures, in order to prevent unwilling changes from the users.

CHAPTER 4

Analytical Agent

This chapter explains assistive analytical agent part of the setup. Hardware and software installations of the system are described in section 4.1. Section 4.2 covers the visualization of the analytical agent, its ‘mind’. Whereas, section 4.3 explains the capabilities of the analytical agent in detail.

4.1. Setups

4.1.1. Hardware Setup

The hardware setup of the assistive analytical agent part of the application is depicted in Figure 4.1. The setup consists of a desktop computer, an additional Kinect for Xbox One [41] as an RGB-depth camera, a microphone to capture users speech, and two monitors.

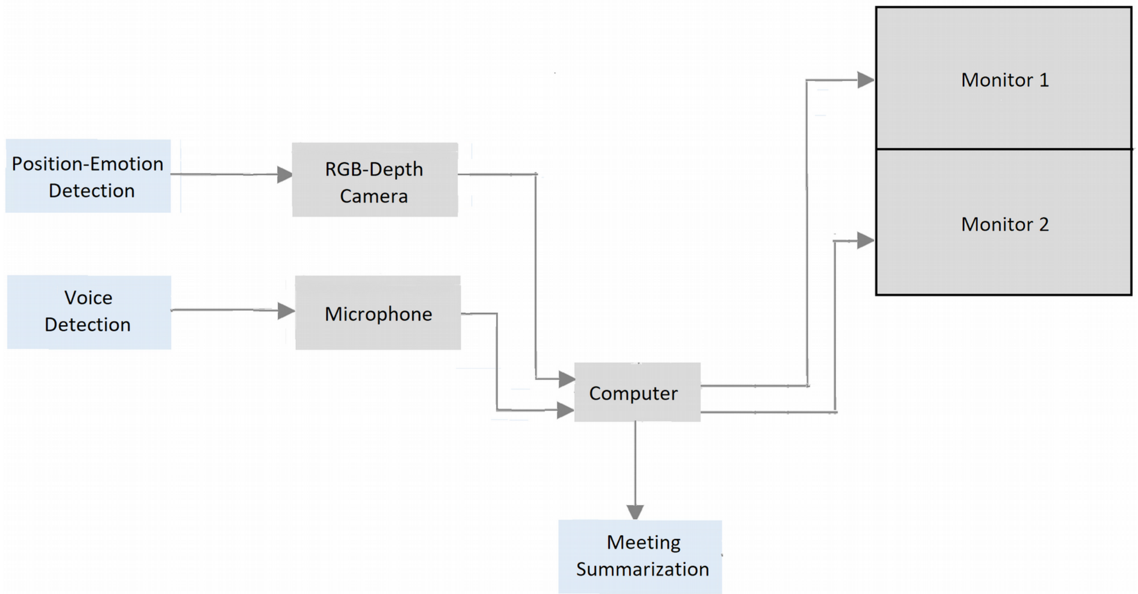

Figure 4.1: Hardware Setup for the Analytical Agent

Kinect's position should be arranged to see the entire meeting. The intended purpose of this Kinect is to detect attended users' real-position and their emotions. Kinect can detect real-position of each user in the meeting room if only the most part of the users' bodies are in the Kinect's field of view. After finding bodies, Kinect gives the emotion for each user according to user's mimic. Moreover, whether users wear glasses or if they change their gaze can also be extracted from the Kinect. All of this information can give the analytical agent clues to know each user better. We are not using entire information about the users within the scope of this project but using this knowledge about the users, the analytical agent can get a better grasp of the situated interaction.

To listen to users' conversations, the system is using a single microphone. At the beginning of the project, multiple microphones were used to get clearer speech. However, taking multiple sound streams from microphones and processing them causes the system a huge amount of the computation time. In other words, the analytical agent can either make related analysis more extensive about the current topic without any loss of speech or miss some analysis but give the users quick responses according to their actions. These actions can either be a question or just a situation that the agent can give a suggestion upon. Human-computer interaction systems are predominantly affected by the degree of responsiveness [48]. Providing responses without spending unnecessary time helps the system gain the users' trust and also gives the users the chance to spend their time more effectively. The meeting room is demonstrated in Figure 4.2.

Figure 4.2: Meeting Room.

4.1.2. Software Setup

Various programming languages are used in the implementation of the assistive analytical agent. The implementation involves four different subprograms. Two of them are for visualization of the agent's state and data, and the others are for the back-end of the application. Each subprogram is written different programing language based on the problem definition.

For visualizing to the analytical agent, Leaflet which is described software setup of the tabletop part and D3 [49] which is also JavaScript library are used. While the leaflet demonstrates the response of fulfilled users command on the map (A) and, the agent's mind is displayed with the help of D3 (B). D3's emphasis on web standards combines powerful visualization components and a data-driven approach to Document

Object Model (DOM) manipulation, providing with all the features of a modern browser without having to attach a custom framework.

To play with data and to recommend new ideas according to users concern, another subprogram (C) is coded using Python [50]. Python provides various libraries about the machine learning techniques to explore, analyze, and process text. Using these libraries the agent can learn about the structure of the data, and also make reasonable estimates from data.

Another subprogram (D) is generated to get users' position, emotion, and to detect the current speaker by using C++ [51]. C++ is the most compatible programming language to get data from Kinect because Kinect's libraries are also coded in C++. Using C++ minimizes the time required to receive the data. Latency is more important here, giving information lately is refers to give wrong information about who is talking, the position of the all users, and their emotion. If a wrong output is shown, it causes users to lose their confidence in the system.

To get users' speech and convert it to text yet another subprogram (E) is utilized. While microphones capture the speech of the users, subprogram obtains this audio as a stream using C# audio libraries [52]. Google Speech-To-Text [53] application programming interface(API) is used converting speech to text. The API can provide a text of the stream even if the user is speaking with significant accuracy.

The agent itself (F) is coded in Java [54]. This subprogram receives information from all other subprograms and decides what can be done with this information. After deciding what to say to the users or how to modify user interfaces, the required commands are sent to the related subprograms which can be (A) and (B). At the end of the meeting, it will also generate notes from the meeting. These notes are the users' speech transcriptions with timestamps, users position during the meeting, users actions like where he/or she touched table, decisions taken. This part of the program is described in more detail in Meeting Summarization part.

Subprograms can communicate with each other using the implemented socket server. If one of the subprograms sends a message to the socket with giving message and name of the subprogram that message needs to reach, the socket will steer the message to the respective subprogram. The relation between the subprograms can be seen in Figure 4.3.

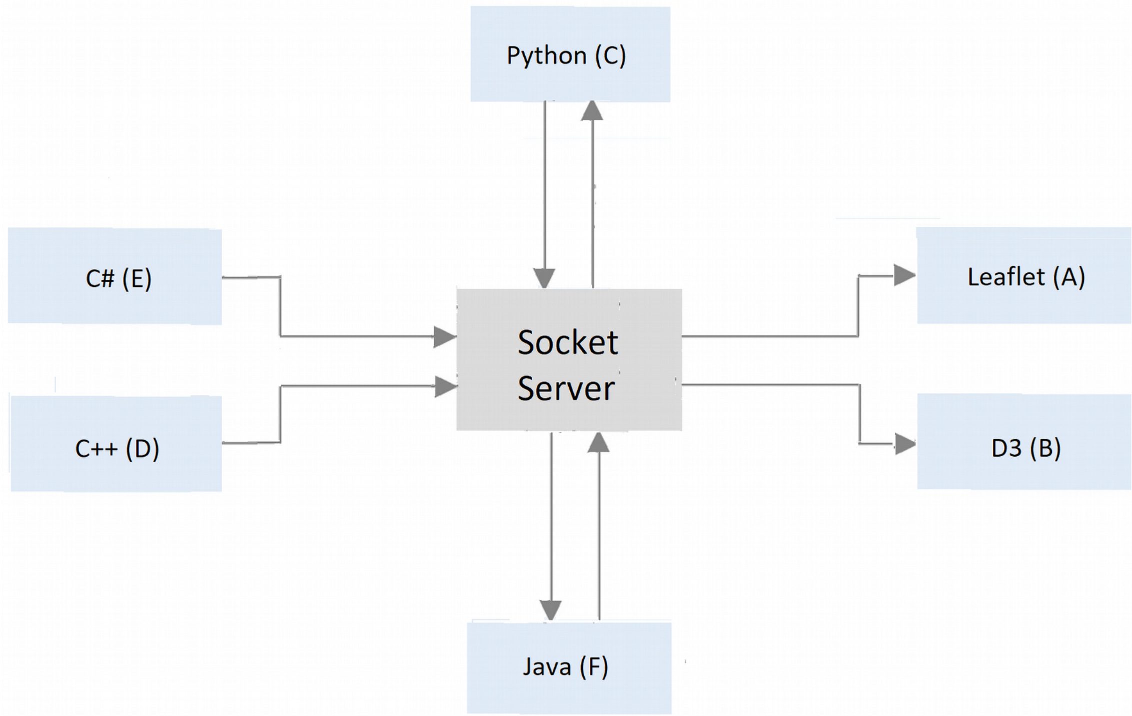

Figure 4.3: Demonstrating the relation between sub-programs

The actual purpose of the socket server is to make analytical agent communicate with the tabletop application. Detail of the socket server such as what they need to communicate with each other, what message format is described in Chapter 5.

4.2. Visualization of the Analytical Agent

Visualization of the analytical agent is also important to users. If they can see analytical agent state or understand how it sees their dialog, they will be elicited to talk to him for the initial process due to human curiosity.

To show data on the map modified by the analytical agent and analytical agent state user interfaces are created to show up on monitors. Thus, users can understand what is going on the agent side.

Monitor 1 shows the data on top of the map similar to the tabletop. Different from the tabletop, filter options do not exist in the screen because neither the monitor is touchable, nor users can use the mouse. When they need to filter data, telling the analytical agent the desired filter is enough. Additionally, analytical agent responses are demonstrated on the map.

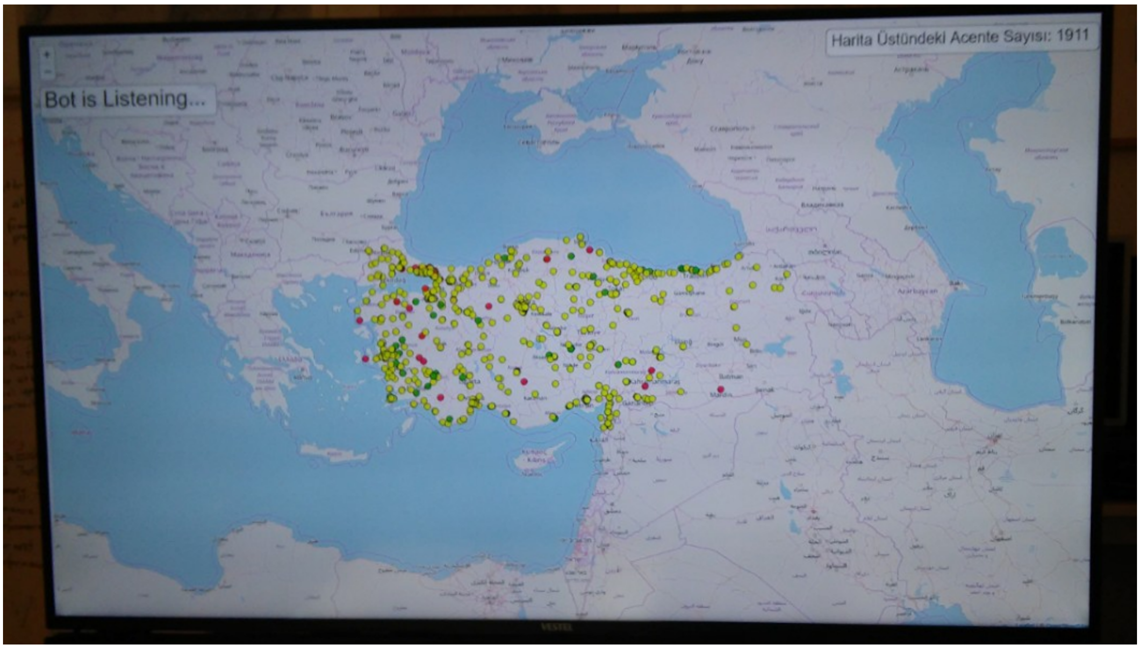

Figure 4.4: Initial Screen of First Monitor

The second monitor displays the analytical agent inputs like the users' speech and the simulates their position around the table. On the left side of the screen, it also displays detailed information about the spoken or touched location of the branch offices and the relation between other branch offices. Top left part of the screen depicts users' location. Dialogs with the speaker id are displayed on the bottom left side of the screen.

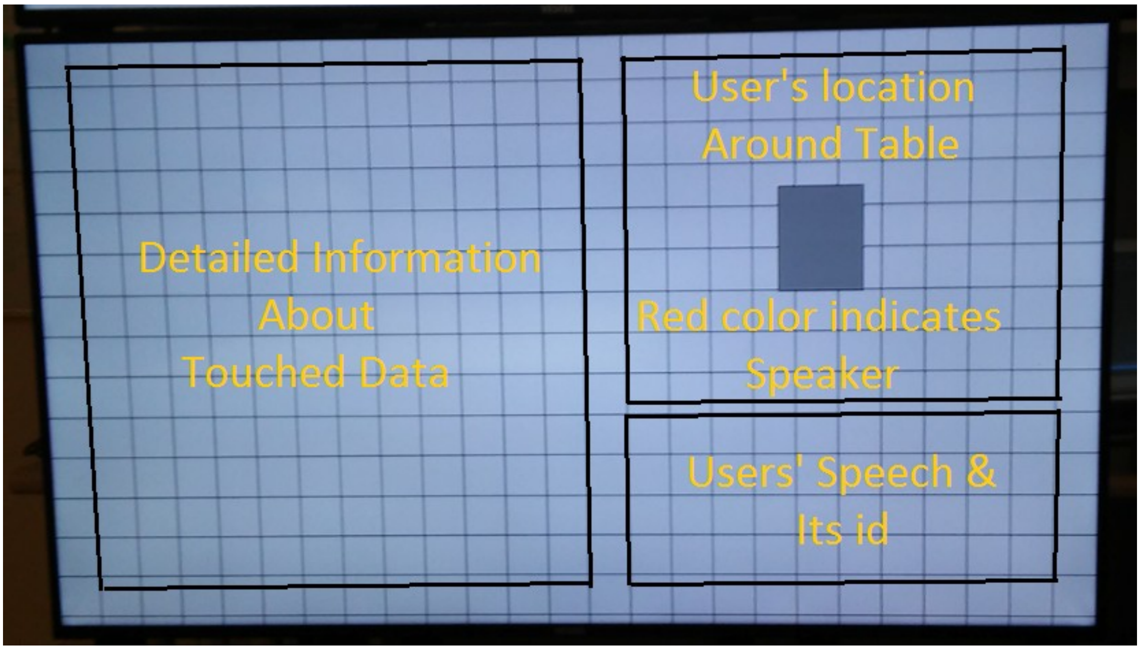

Figure 4.5: Initial Screen of Second Monitor

4.3. The Capability of Analytical Agent

4.3.1. Interactions

To provide interactions between user-analytical agent, analytical agent-data, and analytical agent-table, a socket server is created. Detailed information (i.e. how the messages are sent or received) is described in chapter 6.

Analytical agent-data and analytical agent-table communications depend on users' utterances. According to the user choices, the analytical agent takes action on data or changes what tabletop shows to the users.

To understand what the users say, microphone takes speech and sends this stream Google Speech-To-Text API returns text format of the user(s) speech. For good speech recognition accuracy, the user should speak on a slightly slower pace.

Moreover, another important point is speaker recognition. Although, Kinect has its own microphone, for better speech recognition accuracy we use another separate microphone. However, Kinect's microphone provides the angle of the voice with respect to Kinect's center. Additionally, users' angles with Kinect is calculated by using the users' the real-world location extracted from Kinect and the real-world location of the Kinect's center. By using this information, the speaker is recognized as the closest angle between the detected voice and the users' angles. This way, we know which of the users is speaking at a given time.

4.3.1.1. Human-Analytical Agent Interaction

The analytical agent has a key set which helps to understand user requests. The key set is obtained by examining meetings of the concerned parties about the data. The analytical agent provides responses searching these keywords in the users' dialogs.

Users directly ask a question to the analytical agent either about the topic of the meeting or open-domain questions. If user question is related to the topic such as: "Can you show me agencies which have yellow color performance close to green?", analytical agent analyzes the sentence using keywords and gives the proper response to the user's request. If user question an open-domain one such as: "How are you today?", the analytical agent directs these kinds of question to the online daily conversation bot, takes the answer from them, and shows it to the users. Currently, Cleverbot [55] is being used to answer these questions.

The analytical agent also listens to the attendees during the meeting and if it can deduce reasonable result from their speech, it calculates the result of deduction and shows in the monitor 2. Moreover, if the analytical agent decides that the users talk out of topic for a time interval longer than the threshold, then it politely warns the users to talk about the related topics.

4.3.1.2. Analytical Agent-Data Interaction

If the analytical agent decides or users want the analytical agent to make some operation on data, it interacts with the data using its Python subprogram for specific operations. Besides, when users touch the table, the analytical agent compares branch offices in the touched area with whole data with respect to governance, performance, and response metric. After the calculation is done, it shows these results using charts on the second monitor.

The analytical agent sends a message to Python script using socket server. This message includes keywords that describe what should be done to the data. After Python script calculates results, they are sent to visualization tool D3 to be shown to the users.

4.3.1.3. Analytical Agent-Table Interaction

If users want to see the first monitor's version of the data which is modified by the analytical agent on the table, analytical agent changes the table's data according to monitor's data. Figure 4.6 shows a given command and change on the data. This process gives more opportunity to the user that apply more detail and specific filter to the data or examine data with different ways. The details of these filters are explained in chapter 6. After the users are done with the data on the table, they can return to previous work on the table by clicking the previous button.

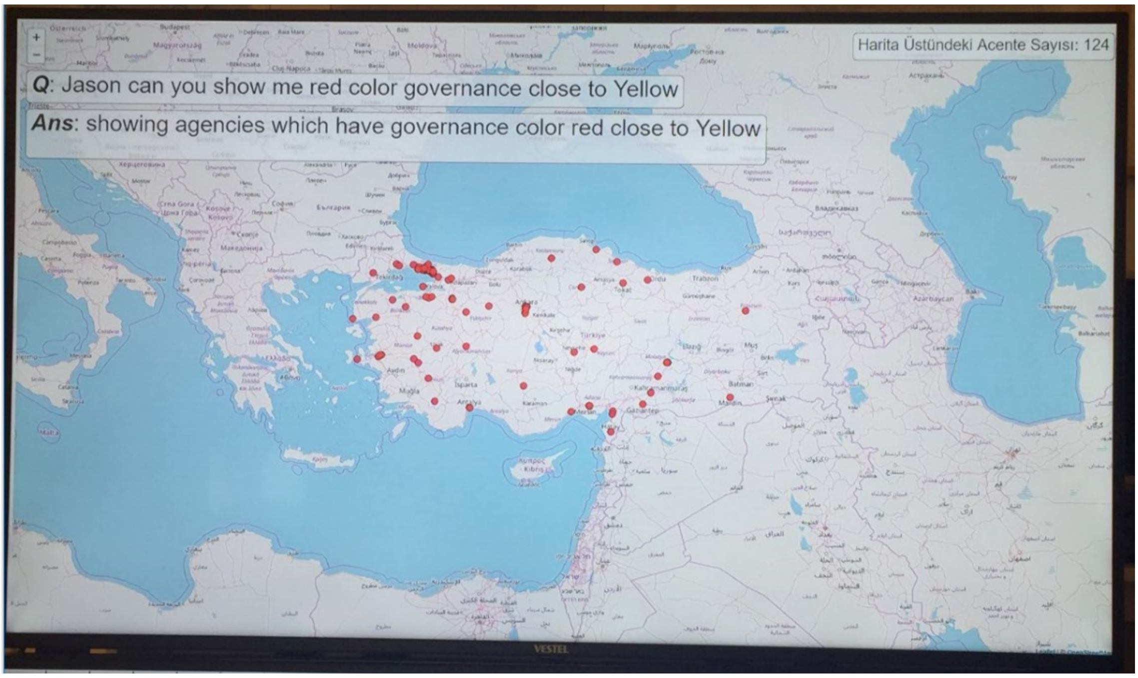

Figure 4.6: Showing Desired Filter

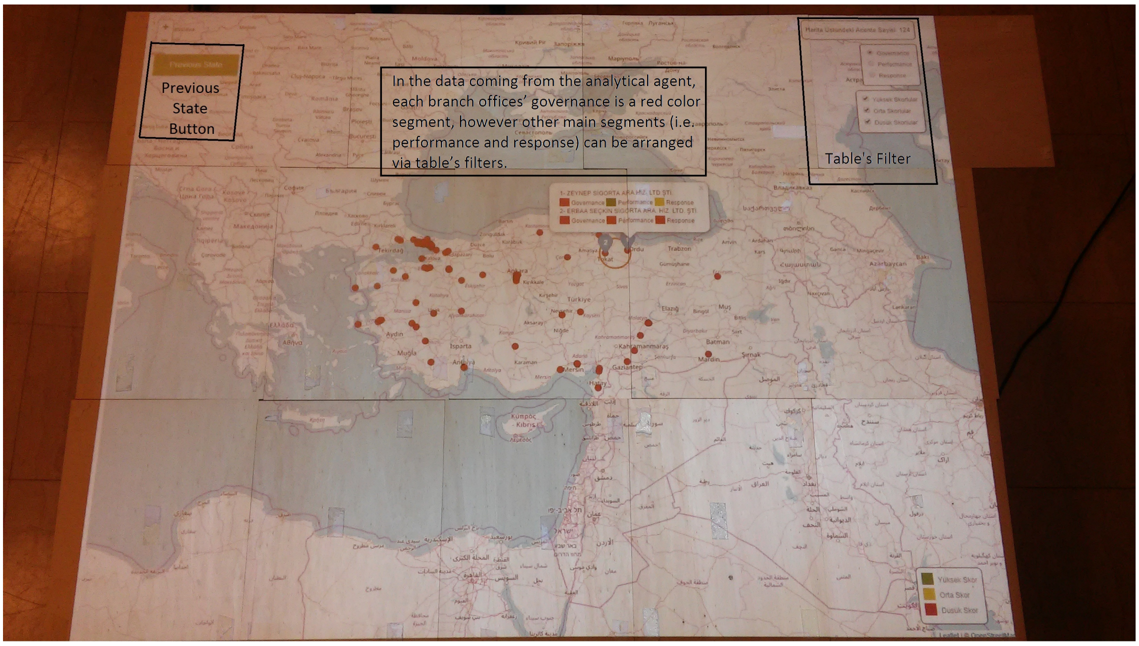

Figure 4.7: Incoming Data on the Table and the Previous Button

4.3.3. User Identification

When Kinect detect the bodies on its field view, it assigns ids to each body. Since the system can detect the speaker and knows its id, if the users tell the analytical agent their name, the analytical agent will match the ids with their name. In this way, the analytical agent can address the users by their name and also it can take notes for each user by using their name rather than the numeric ids. However, it depends only on users, they can choose to be anonymous.

4.3.4. User Engagement

The analytical agent can get ideas about which user needs engagement with the help of the user speech and table touching frequency of the users. Using previous actions or previous notes of the users, it finds a way to engage them.

Since the analytical agent knows how much users talked, it can analyze the users' motivation. After detecting less interested users, if past meeting notes exist about these users, it can ask questions to these users about their point of interest. If past meeting notes do not exist about less interested users, the analytical agent asks generic questions to these users like "Have you looked at this?" or "Do you want to see this?" to get their attention.

Keeping users engaged on the tabletop is also agent responsibility. The analytical agent gets information from the tabletop part about which users touched the table more. After identifying the less interested users about the table, the analytical agent can ask questions to these users such that to give an answer they must touch the table such as "User x, can you say which most important branch office in city A is the closest one to the city center?"

4.3.5. Session Coordination

Meeting management is the key factor to make an effective meeting. Meeting leader has specific duties during the meeting. He/she helps the attendees to stay focused and productive. Additionally, he/she makes a follow-up plan for the analytical process. What meeting leader should do after the meeting is as important as duties during the meeting [56]. Analytical agent's mission is not to be the meeting leader, its mission is to alleviate the difficulties of leadership by helping the leader of the meeting, to make leader focused on more important subjects. Therefore, the analytical agent helps the attendees to stay focused and effective.

However, the hardest part of the helping meeting leader is the session coordination. Directly interfering users' speech and stopping them causes attendees to feel offended. To prevent that, the analytical agent just provides the information about the meeting status on the screen without taking much attention and meeting leader can easily follow-up this information.

To give information to meeting leader about the session coordination, the analytical agent can figure out the meeting process and understand users in each step. There are obvious steps to follow in analytical studies. In this case, users can define problems, produce solutions to the problem, and apply the solution to data. Until results are satisfactory the users repeat these steps. The key set is also produced by examining meetings of the concerned parties about the current data for these three steps.

In a session, users can introduce more than one problem and work toward their solution. After finding solutions for a problem or getting command form users, the analytical agent will note the problem and its solutions in a file to summarize meeting.

4.3.6. Meeting Summarization

Analytical agent listens to each user during the meeting and also takes note all conversation with user id, timestamps, and user's emotion, if it exists. Moreover, if the users emphasize specific ideas to the agent to note them, the analytical agent logs these as a decision taken during the meeting. Besides from dialogs, the analytical agent can note interested users, and also takes note about users' interesting areas considering where they take part in the conversation about which topics. According to the users' actions, the analytical agent can infer which part of the data is attracted and is able to produce new ideas. Also, looking at the user's actions, those trying to deviate from the topic are noted.

Outsiders can access information about the meeting status such as decisions taken or more interested region of the data, and they can also obtain information about its attendants like which user tries to disrupt the meeting or which users are more willing to contribute using these notes. Since the analytical agent will know more about the users and data using these notes, the productivity of the next meetings is increased.

4.4. Response Time

As mentioned previously, the second monitor shows the position of the users around the table. The delay of showing users location is approximately 400 ms. This delay is due to communication time between the subprogram that detects user movement from Kinect and the subprogram responsible for drawing the locations.

Providing the response of the open-domain questions takes on average 1.5 seconds depending on the question meaning. One of the slowest parts is getting the response from the daily conversation bot, which takes time to generate it. The other bottleneck is converting audio to text using Google API. The time of the giving response of the question also depends on the sentence length. Speech to text API gets slower based on the sentence size.

Displaying the answer of the users' questions or taking action on the map in response to the users' commands is faster comparing with the open-domains questions. These actions take on average 750 ms according to the sophistication of the filters requested, the latency of the action increase

CHAPTER 5

The System

This chapter thoroughly explains the entire system that consists of the tabletop and the assistive analytical agent. Section 5.1 shows the system design of the integration of tabletop to the analytical agent. Section 5.2 covers the implementation of the socket server and how sub-systems use it to communicate with one-another. On top of that, section 5.3 depicts the entire system with images taken from a running session, showing users' hand gestures for tabletop and the communication with the analytical agent.

5.1. System Design

As seen in Figure 5.1, the system is formed by combining a tabletop and an analytical agent.

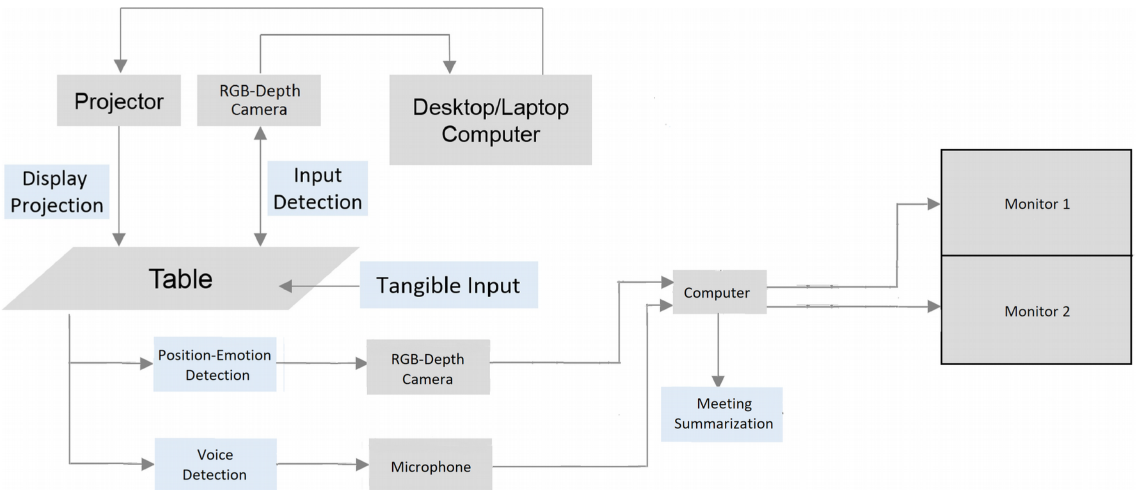

Figure 5.1: System Design

5.2. Communication Between Table and Agent

To establish the communication between the tabletop and the analytical agent, a socket server is implemented. The socket server also provides subprograms' communication. When each of the subprograms starts, they send their identification to the server. Figure 5.2 depicts the registration of the subprograms. After each of the subprograms is registered to the server, they can communicate with each other. Detailed information about the tabletop of the system, its subprograms, and their implementation are described in chapter 3. Chapter 4 explains the technicality of the analytical agent's subprograms.

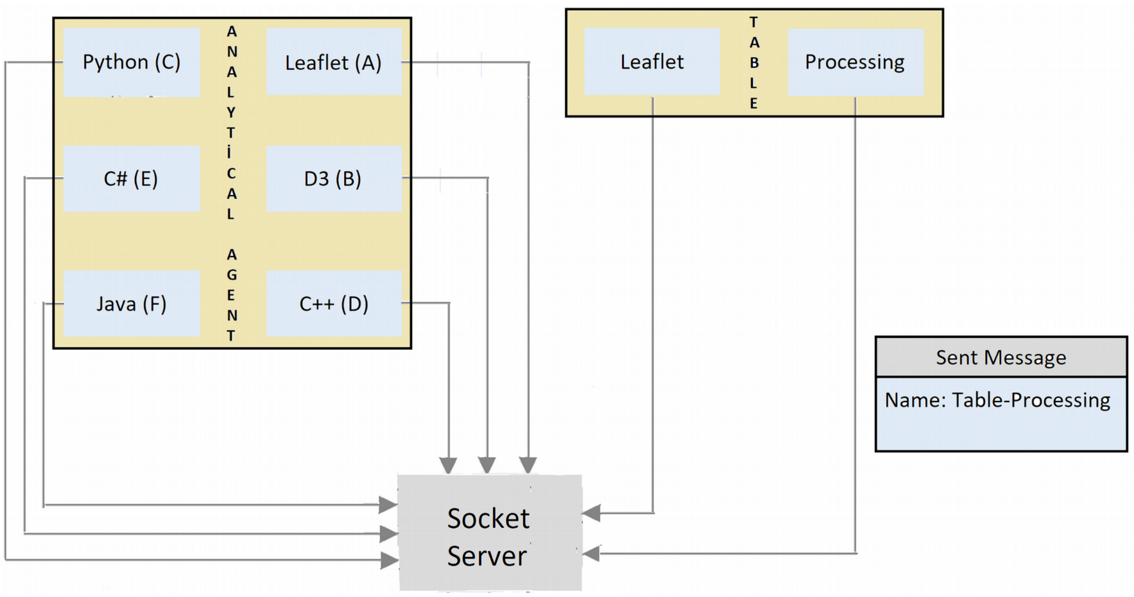

Figure 5.2: Registration and Example Message to Register

Each subprogram has a message handler. Whenever a message is received by a subprogram, its handle is triggered. In other words, the socket server uses push notification rather than the pull notification. An example of sending a message is shown in Figure 5.3. If a subprogram needs to send a message to another subprogram, the message should contain the receiver subprogram's name, sender subprogram's name, and the message itself.

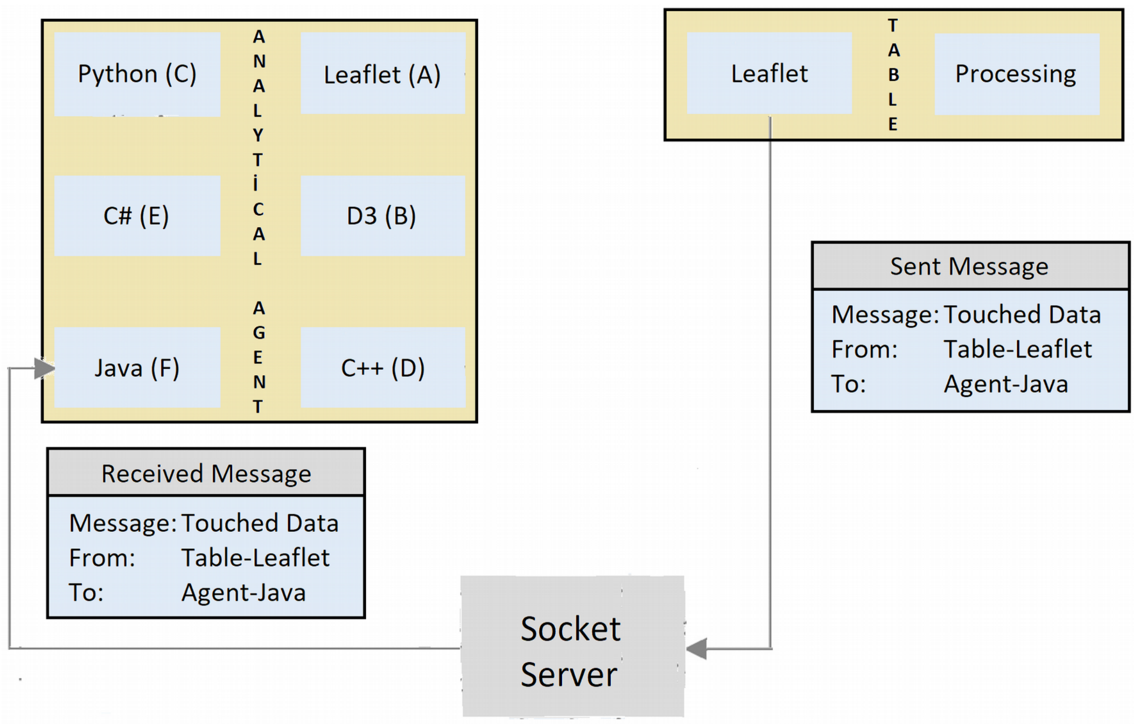

Figure 5.3: Example of Sending and Receiving Message

According to subprograms' functionality, each subprogram can be receiver, sender, or both of them. Figure 5.4 demonstrates the relation between subprograms and socket server.

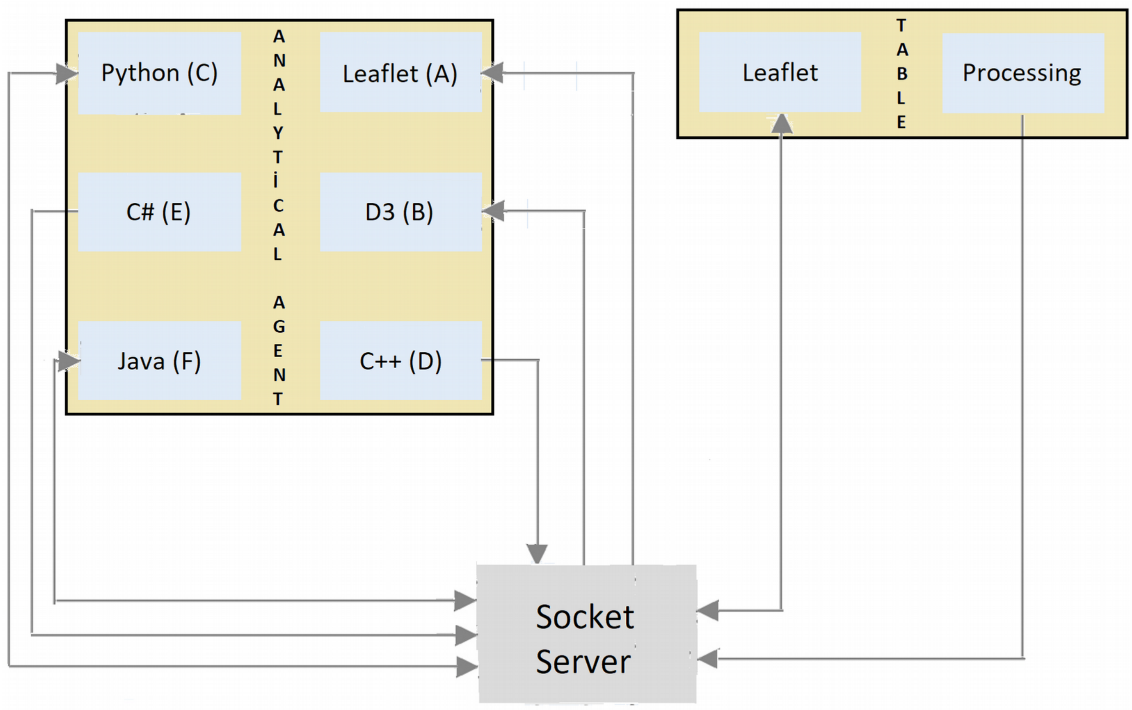

Figure 5.4: Direction of the arrows shows the communication way

5.3. Entire System Description

The tabletop and the analytical agent are described separately in chapter 3 and 4, respectively. In this section, a detailed usage of the entire system will be demonstrated with images taken from the running system to demonstrate possible use cases.

Figure 5.5: Starting Zoom-in Action by User

Figure 5.6: Zoomed-in on the Table's Map

Figure 5.7: Activating Zoom-out Action by User

Figure 5.8: After Zoom-out Action Done

Figure 5.9: Start Command to Pan by User

Figure 5.10: When the user is panning

Figure 5.11: Appearance of the map after a pan action is done

Figure 5.12: Releasing Map

As mentioned before, users can filter the map on the table. They can observe only one main segmentation at the same time, but multiple/single choice can be shown for the color-segmentation with respect to the selected main segment.

Figure 5.13: Showing branches from the selected filter which is only red and green color responses

When using a map on the computer, users can zoom-in/out, pan, and touch filters using mouse or keyboard. Hand gestures are assigned to specific tasks for the map onto the table. Figure 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, show users hands action and the result of these actions. However, users cannot use these hand gestures for the analytical agent's map showed by the first monitor. To zoom in/out, pan and filter, the analytical agent needs users to say the intended action.

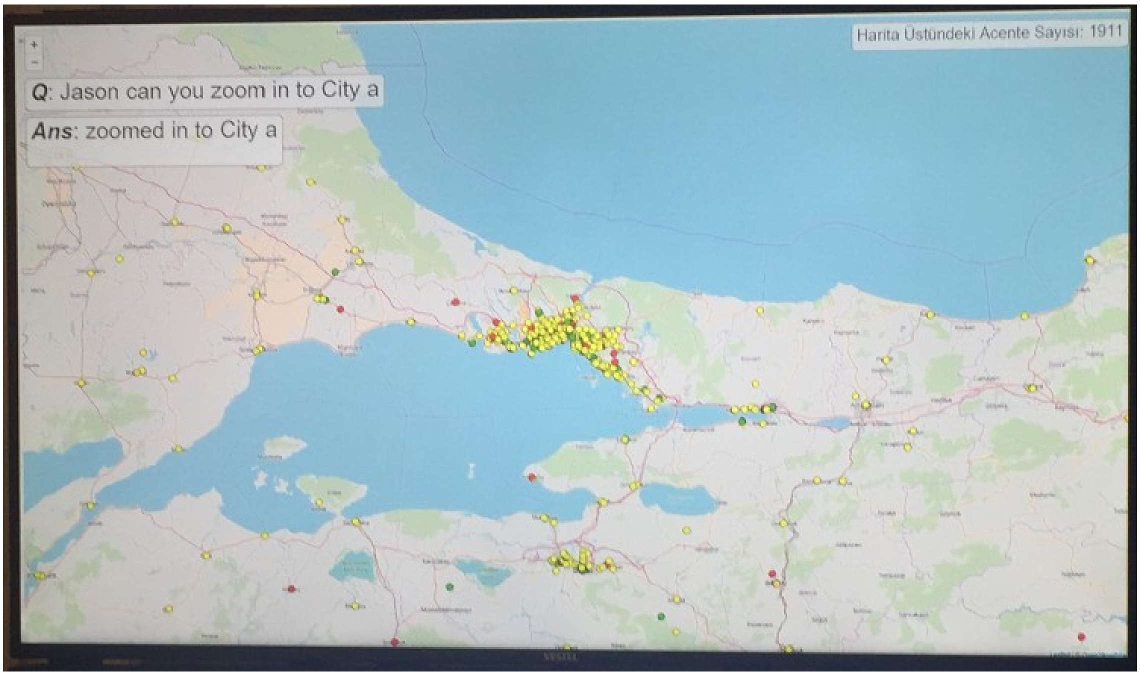

Figure 5.14: After giving a zoom-in command to the analytical agent

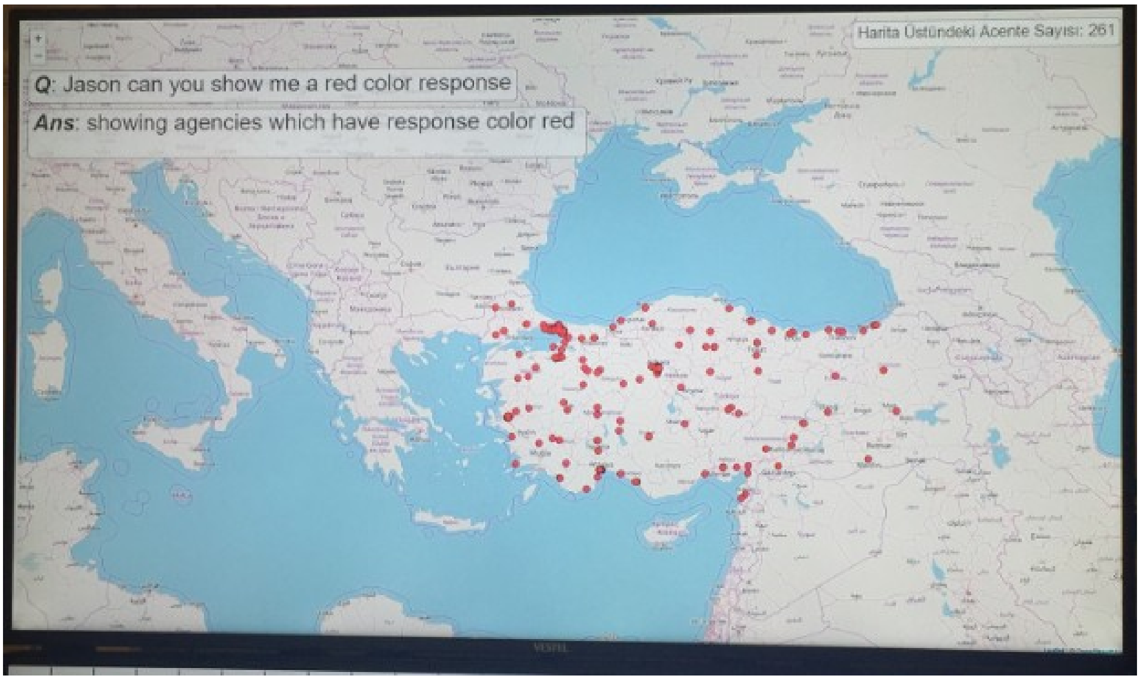

Figure 5.15: Simple Filtered Data

One of the most important benefits of the analytical agent understanding users command is that users can request sophisticated filters from the analytical agent. An example is shown in Figure 5.16.

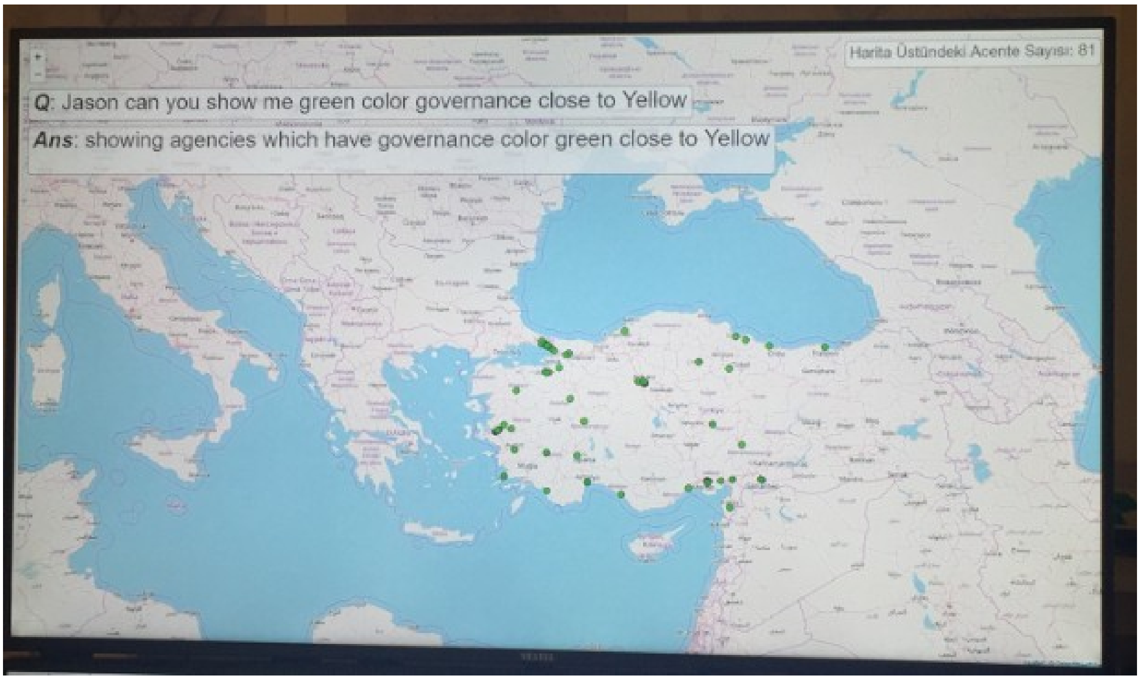

Figure 5.16: Sophisticated Filter

Users can play with these sophisticated filters on the table such as using also table's filter. Thus, they can use double filtering. It helps to overcome a drawback of the table’s filter which is showing only one main segment at the same time. An example of a request for bringing the version of the map from monitor 1 to the table is demonstrated in Figure 5.17. Also, Figure 5.18 shows the intended map on the table and the previous button. The previous button helps users to go back to the latest version of the work on the table.

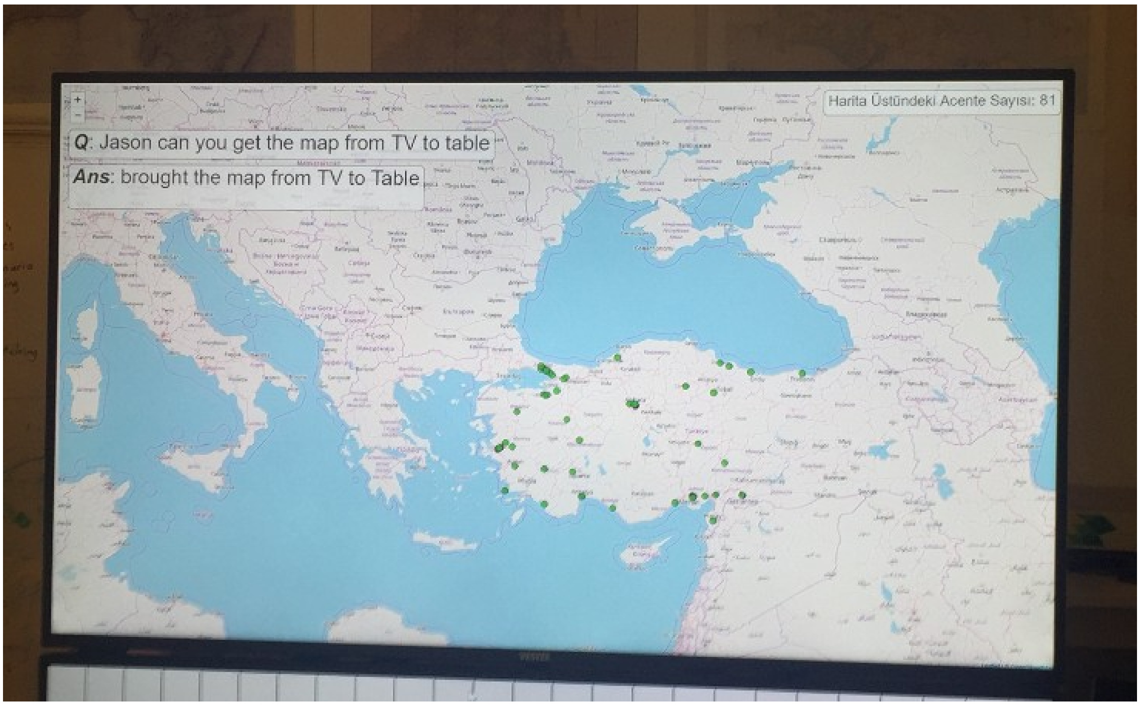

Figure 5.17: Request to Bring Map

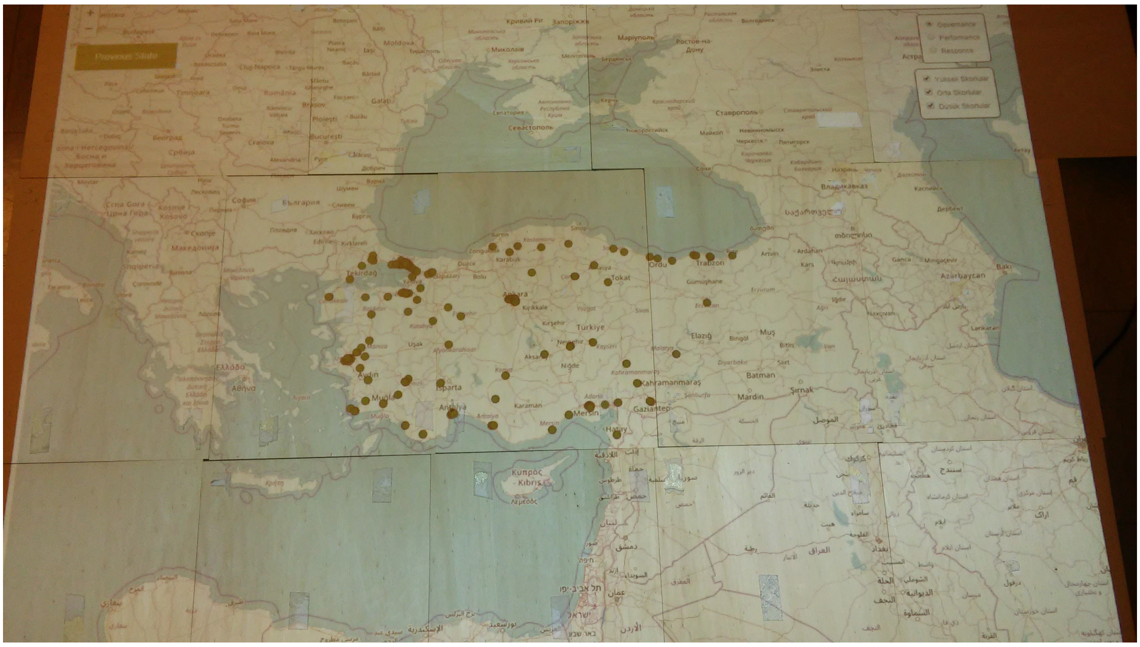

Figure 5.18: Requested Map on the Table

When users touch the data on the map, properties of branch offices in the touched area are shown. Besides, more detailed information about the touched data is demonstrated by using the second monitor. If the touched area includes branch offices more than a certain threshold, the map is automatically zoomed to this area.

Figure 5.19: Properties of the Touched Area

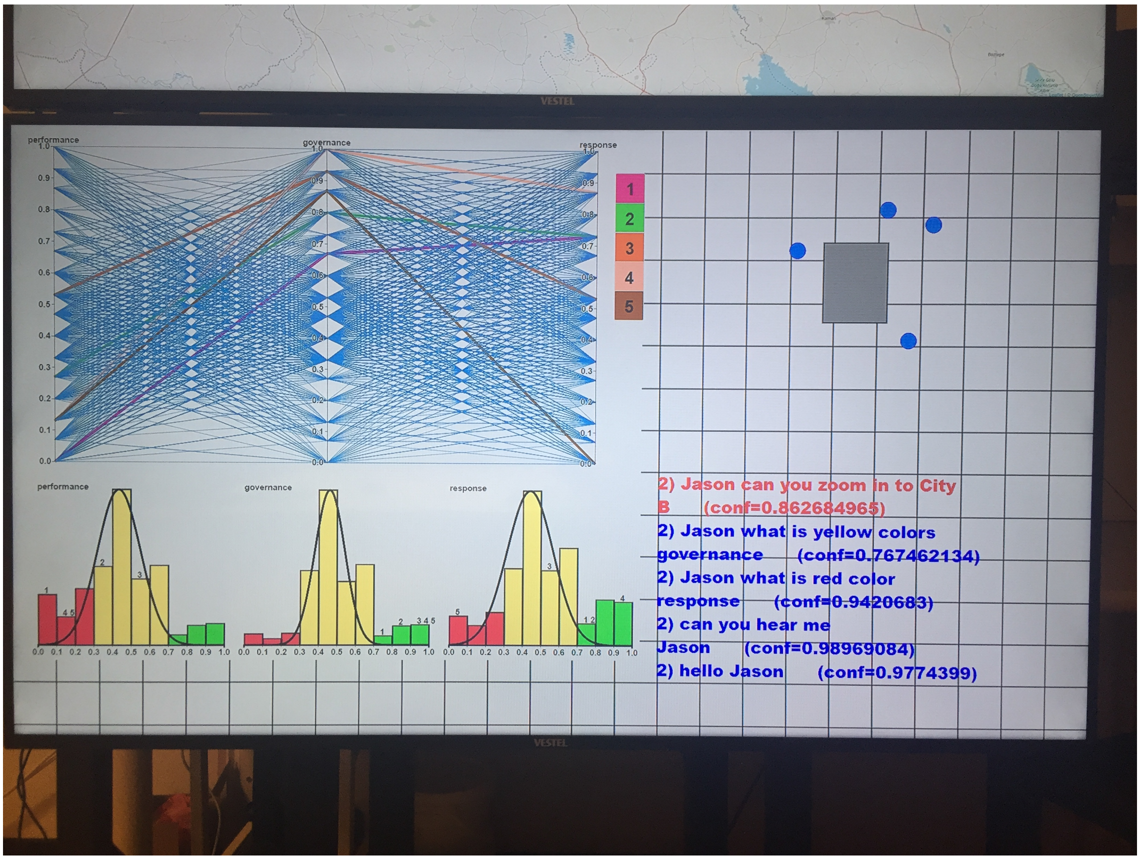

Figure 5.20: Detailed Information about the Touched Area

Figure 5.21 indicates the users' positions around the table and their transcribed speech with users' ids. In this way, users can notice if any misunderstanding occurs by the analytical agent and correct it. Current speech and speaker's color are demonstrated in red, others are blue.

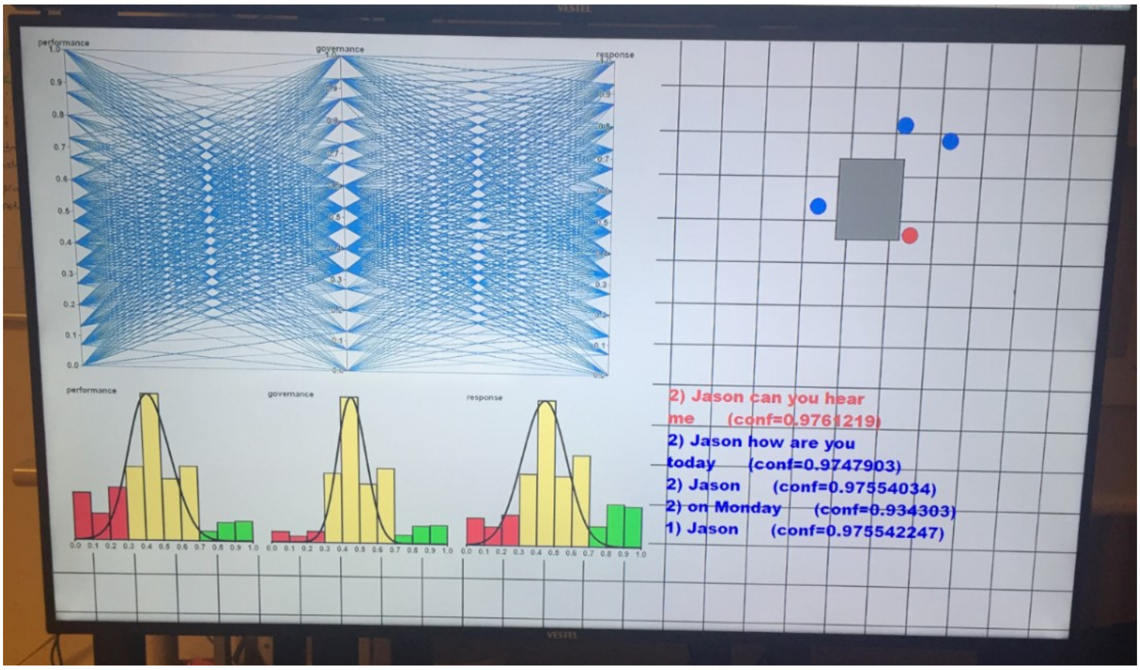

Figure 5.21: Simulating the Current Speaker and Others around Table

Users questions do not have to be related to the current topic. Users can ask an open-domain question such as "what day is today". Additionally, users can have a conversation with the analytical agent. As mentioned previously, these open-domain questions or sentences are directed to an online daily chatbot. (The answer of the online-bot may not always be meaningful.)

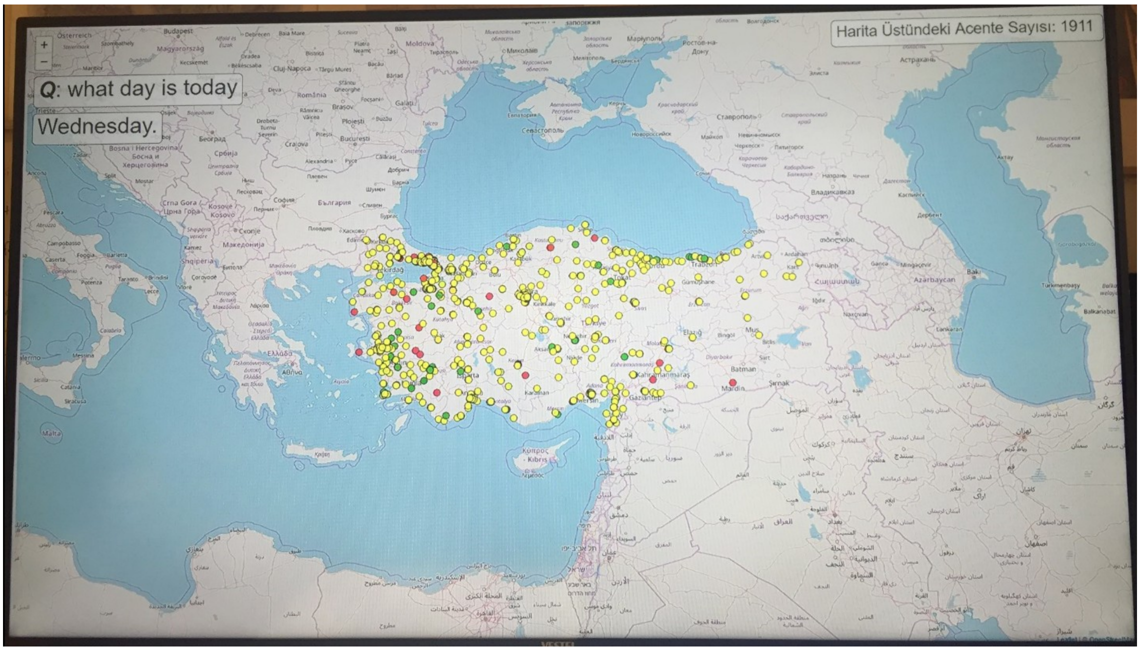

Figure 5.22: Example Question and its Answer

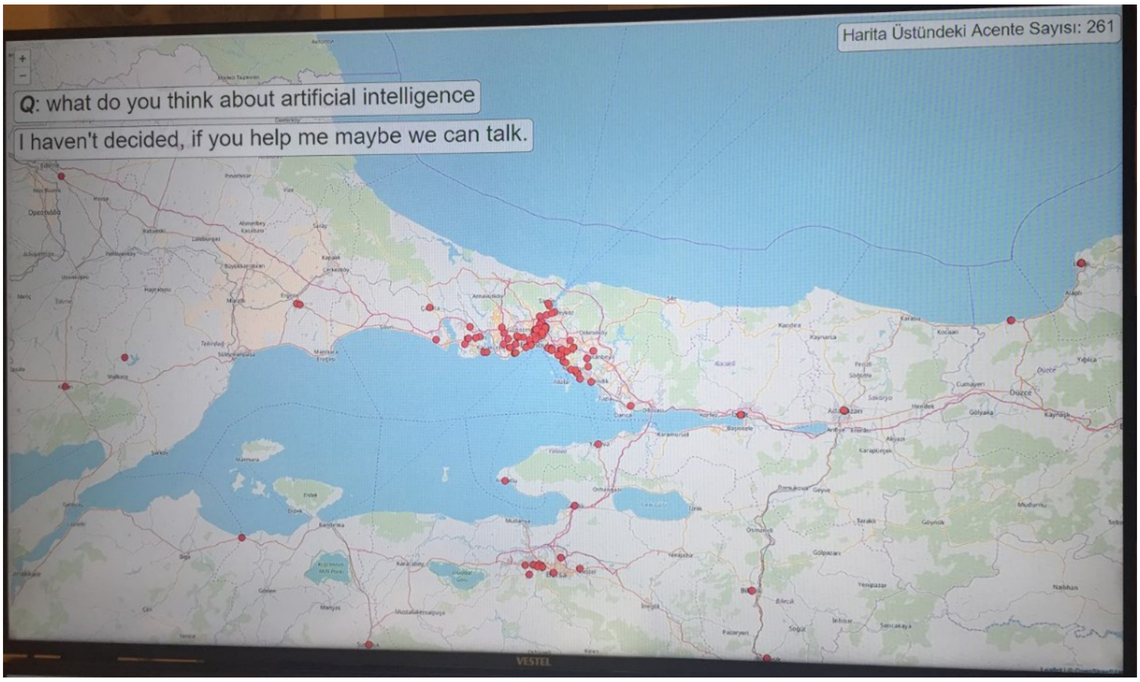

Figure 5.23: Sample Conversation with the Analytical Agent

CHAPTER 6

Conclusion and Future Work

6.1. Conclusion

In this thesis, we presented a tabletop system coupled with an assistive analytical agent to support meetings. Our aim was to establish an environment suited for analytical work. We built a platform that can visualize data on the map and project it onto the table. Our system also follows users' actions, predicts what the next actions of the users will be and fulfills users' commands. It attempts to engage inactive users during the session to increase productivity. In addition, the system helps the meeting leader with session coordination, meeting summarization, etc.

The tabletop part of the platform covers the data visualization. The data shown on top of the table is taken from an insurance company. We cannot share the name nor the data because of the privacy policy. In our previous work [47], we observed that using tabletop systems increases the collaboration among the users and the productivity of the sessions. Hence in this work as well, the data is demonstrated by using the tabletop system with the proper visualization framework. (In addition, it provides the users with the opportunity to touch the data in the physical environment.)

The assistive analytical agent part of the work aims to answer attendees' requests, have conversations with them, help meeting leader on organizing and managing meetings. It also creates notes on meeting results and other important issues to be used in the next meetings. Besides, we visualized the analytical agent's state and its actions too. Thus, users can understand the reason behind analytical agent's actions.

Finally, combining these two parts, we introduced an assistive analytical agent for exploratory and decision making in analytical multi-users sessions. Providing users an environment for collaboration, meeting organization, and other properties explained previously can help attendees accomplish more productive and engaging meetings.

6.2. Future Work

The current system contains minor limitations. These limitations can be internal or external. One of the internal limitations is too much memory consumption prevents the program from running for a long time. It can be fixed by limiting Kinect and analytical agent memory access. The other internal bug is when users touch the table, the extracted touched location is not exactly the same as the user's touched location because of the Kinect and projector calibration mismatch. We produced a temporal solution for this problem. We are drawing circles at the touched location and human eyes cannot notice the difference. An example of the external limitation is the quota provided by the free version of Google-Text-To-Speech API. After the limited time, API does not give any answer for a day to our requests.

To evaluate our system, real scenarios about data should be analyzed by concerning parties with and without our system. Afterwards, the result of these two experiments in terms of meeting productivity and user engagement should be reported. Looking at the difference between the results, we can comprehend the extent of the benefits of the system such as increased work efficiency and productivity.

As discussed previously, the entire system consists of nine different subprograms including a socket server. To run the whole system, each subprogram should be run in specific order manually. It can be automated by implementing one script to run each of subprogram with an order.

In addition, our system cannot recognize attendees using face recognition algorithms. If the users say their name, then the analytical agent matches the names with bodies id. However, when the user exits from the Kinect field of view, his/her body and hence the id will be lost. Afterwards, if the user reenters the Kinect field of view, a new id will be assigned to this user and he/she has to say his/her own name again. To avoid this, a face recognition algorithm can be implemented to the analytical agent. Using old session photos with tagged users' names, the analytical agent can predict users' identification for the further sessions.

REFERENCES

[1] D. Schmandt-Besserat, How writing came about. University of Texas Press Austin, 1996.

[2] A. Vande Moere, “Beyond the Tyranny of the Pixel: Exploring the Physicality of Information Visualization,” 2008 12th Int. Conf. Inf. Vis., pp. 469–474, Jul. 2008.

[3] Edward R. Tufte, The Visual Display of Quantitative Information, 2nd editio., no. v. 37, nos. 2–4. Graphics Press, 2001.

[4] J. Heer, S. Card, and J. Landay, “Prefuse: a toolkit for interactive information visualization,” Proc. SIGCHI Conf. …, 2005.

[5] Y. Dragicevic, P., Jansen, “List of Physical Visualizations,” 2013. [Online]. Available: http://www.aviz.fr/Research/PassivePhysicalVisualizations. [Accessed: 01-Mar-2014].

[6] “3D Tissue and Organ Printing Project - Sabancı University,” 2014. [Online]. Available: http://www.sabanciuniv.edu/en/3-boyutlu-doku-ve-organ-basım. [Accessed: 01-Mar-2014].

[7] A. Vande Moere and S. Patel, “Analyzing the design approaches of physical data sculptures in a design education context,” Vis. Inf. Commun. Int., 2009.

[8] M. K. Rasmussen, E. W. Pedersen, M. G. Petersen, and K. Hornbæk, “Shape-Changing Interfaces : A Review of the Design Space and Open Research Questions,” pp. 735–744, 2012.

[9] H. Ishii, D. Lakatos, L. Bonanni, and J. Labrune, “Radical atoms: beyond tangible bits, toward transformable materials,” interactions, vol. XIX, no. february, 2012.

[10] Y. Jansen, P. Dragicevic, and J.-D. Fekete, “Evaluating the efficiency of physical visualizations,” Proc. SIGCHI Conf. Hum. Factors Comput. Syst. - CHI ’13, p. 2593, 2013.

[11] S. Stusak and A. Butz, “Can Physical Visualizations Support Analytical Tasks ?,” pp. 5–6.

[12] Y. Jansen and P. Dragicevic, “An interaction model for visualizations beyond the desktop.,” IEEE Trans. Vis. Comput. Graph., vol. 19, no. 12, pp. 2396–405, Dec. 2013.

[13] S. Swaminathan, C. Shi, Y. Jansen, P. Dragicevic, L. Oehlberg, J.-D. Fekete, and others, “Supporting the Design and Fabrication of Physical Visualizations,” in CHI2014-Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2014.

[14] C. Stolte, D. Tang, and P. Hanrahan, “Polaris: a system for query, analysis, and visualization of multidimensional relational databases,” IEEE Trans. Vis. Comput. Graph., vol. 8, no. 1, pp. 52– 65, 2002.

[15] F. Bara, E. Gentaz, P. Colé, and L. Sprenger-Charolles, “The visuo-haptic and haptic exploration of letters increases the kindergarten-children’s understanding of the alphabetic principle,” Cogn. Dev., vol. 19, no. 3, pp. 433–449, 2004.

[16] M. Wilson, “Six views of embodied cognition,” Psychon. Bull. Rev., vol. 9, no. 4, pp. 625–636, 2002.

[17] R. Lisserman, J. Huber, M. Schmitz, J. Steimle, M. Mühlhauser, "Permulin: Mixed-Focus Collaboration on Multi-View Tabletops." CHI'14, ACM

[18] http://reactable.com/; 2018. Accessed: 2018-06-05.

[19] https://www.apple.com/ios/siri/; 2018. Accessed: 2018-06-05.

[20] https://en.wikipedia.org/wiki/Amazon_Alexa; 2018. Accessed: 2018-06-05.

[21] https://internetofthingsagenda.techtarget.com/definition/smart-home-or-building

[22] Guizzo E, Goldstein H (2005) The rise of the body bots [robotic exoskeletons]. IEEE Spectr 42(10):50–56

[23] Yanco HA (2001) Development and testing of a robotic wheelchair system for outdoor navigation. In: Proceedings of the 2001 conference of the rehabilitation engineering and assistive technology society of North America

[24] Graf B, Hans M, Schraft RD (2004) Care-O-bot II development of a next generation robotic home assistant. Auton Robots 16(2):193–205

[25] Beck A, Edwards N, Friedman B, Khan P (2003) Robotic pets and the elderly. Available from: http://www.ischool.washington. edu/robotpets/elderly/

[26] DiSalvo C, Gemperle F, Forlizzi J, Montgomery E, Yonkers W, Divine J (2003) The hug: an exploration of robotic form for intimate communication. In: RO-MAN 03

[27] Scanaill CN, Carew S, Barralon P, Noury N, Lyons D, Lyons GM (2006) A review of approaches to mobility telemonitoring of the elderly in their living environment. Ann Biomed Eng 34(4):547– 563

[28] Cesta A, Cortellessa G, Pecora F, Rasconi R (2007) Supporting interaction in the robo care intelligent assistive environment. In: Proceedings of AAAI spring symposium on interaction challenges for intelligent assistants, pp 18–25

[29] Jung JW, Do JH, Kim YM, Suh KS, Kim DJ, Bien ZZ (2005) Advanced robotic residence for the elderly/the handicapped: realization and user evaluation. In: 9th international conference on rehabilitation robotics (ICORR)

[30] Pollack ME, Brown L, Colbry D, Orosz C, Peintner B, Ramakrishnan S, Engberg S, Matthews JT, Dunbar-Jacob J, McCarthy CE, Thrun S, Montemerlo M, Pineau J, Pearl NR (2002) A mobile robotic assistant for the elderly. In: AAAI workshop on automation as eldercare

[31] Shibata T, Wada K, Tanie K (2003) Statistical analysis and comparison of questionnaire results of subjective evaluations of seal robot in Japan and UK. In: Proceedings IROS 2003

[32] Bickmore T, Caruso L, Clough-Gorr K (2005) Acceptance and usability of a relational agent interface by urban older adults. In: Proceedings of the ACM SIGCHI conference on human factors in computing systems, Portland, USA

[33] G. Tur, A. Stolcke, L. Voss, S. Peters, D. Hakkani-Tur, J. Dowding, B. Favre, R. Fernandez, M. Frampton, M. Frandsen, C. Frederickson, M. Graciarena, D. Kintzing, K. Leveque, S. Mason, J. Niekrasz, M. Purver, K. Riedhammer, E. Shriberg, Jing Tien; D. Vergyri, Fan Yang; "The CALO Meeting Assistant System," Audio, Speech, and Language Processing, IEEE Transactions on , vol.18, no.6, pp.1601-1611, Aug. 2010.

[34] “DARPA cognitive agent that learns and organizes (CALO) project.” [Online]. Available: http://www.ai.sri.com/project/CALO

[35] P. Thompson, A. James and R. Iqbal, "Agent based facilitator assistant for virtual meetings," Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Lausanne, 2011, pp. 335-341.

[36] J. Ehnes, "A tangible interface for the AMI Content Linking Device — the automated meeting assistant,"2009 2nd Conference on Human System Interactions, Catania, 2009, pp. 306-313.

[37] Li M, Gao S, Wang CC. Real-time collaborative design with heterogeneous cad systems based on neutral modeling commands. J Comput Inf Sci Eng 2007;7(2):113–25

[38] Hou J, Su C, Su Y, Wang W. A methodology of knowledge management based on ontology in collaborative design. In: Proceedings of the second international symposium on intelligent information technology application (IITA’08), 2. IEEE; 2008. p. 409–13.

[39] Bortolaso C, Oskamp M, Phillips G, Gutwin C, Graham T. The effect of view techniques on collaboration and awareness in tabletop map-based tasks. In: Proceedings of the ninth ACM international conference on interactive tabletops and surfaces. ACM; 2014. p. 79–88

[40] Fan M, Antle AN, Neustaedter C, Wise AF. Exploring how a co-dependent tangible tool design supports collaboration in a tabletop activity. In: Proceedings of the 18th international conference on supporting group work. ACM; 2014. p. 81–90.

[41] Kinect for Xbox one. https://en.wikipedia.org/wiki/Kinect; 2018. Accessed: 2018-06-14.

[42] Reas C, Fry B. Processing: programming for the media arts. AI & SOCIETY 2006;20(4):526–38.

[43] https://leafletjs.com/; 2018. Accessed: 2018-06-15.

[44] Coombs, S. The Psychology of User-friendliness: the use of information technology as a reflective learning medium. Korean Journal of Thinking & Problem Solving, 2000; 10(2):19-31.

[45] Card SK, Robertson GG, Mackinlay JD. The information visualizer, an information workspace. In: Proceedings of the SIGCHI conference on human factors in computing systems. CHI '91. New York, NY, USA: ACM; 1991. p. 181–6. ISBN 0-89791-383-3.

[46] Fingertracker library for processing. http://makematics.com/code/FingerTracker/; 2017. Accessed: 2017-06-16.

[47] E. Kaya, S. Alacam, Y. Findik and S. Balcisoy, "Low-fidelity prototyping with simple collaborative tabletop computer-aided design systems" Computers & Graphics, vol. 70, pp. 307-315, 2018.

[48] R. Ushirobira, D. Efimov, G. Casiez, N. Roussel and W. Perruquetti, "A forecasting algorithm for latency compensation in indirect human-computer interactions,"2016 European Control Conference (ECC), Aalborg, 2016, pp. 1081-1086.

[49] https://d3js.org/; 2018. Accessed: 2018-06-19.

[50] https://en.wikipedia.org/wiki/Python_(programming_language); 2018. Accessed: 2018-06-19.

[51] https://en.wikipedia.org/wiki/C%2B%2B; 2018. Accessed: 2018-06-19.

[52] https://en.wikipedia.org/wiki/C_Sharp_(programming_language); 2018. Accessed: 2018-06-19.

[53] https://cloud.google.com/speech-to-text/; 2018. Accessed: 2018-06-19.

[54] https://en.wikipedia.org/wiki/Java; 2018. Accessed: 2018-06-19.

[55] http://www.cleverbot.com/; 2018. Accessed: 2018-06-19.

[56] http://www.thwink.org/sustain/glossary/AnalyticalMethod.htm; 2018. Accessed: 2018-06-19.

[57] Heerink M, Kröse B, Evers V, Wielinga B (2010) Assessing acceptance of assistive social agent technology by older adults: the almere model. Int J Soc Robot 2(4):361–375