

# Effective Natural Language Interfaces for Data Visualization Tools

Zur Erlangung des akademischen Grades eines  
Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

von der KIT-Fakultät für Wirtschaftswissenschaften  
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Marcel Ruoff, M.Sc.

---

Tag der mündlichen Prüfung: 27. März 2023

Referent: Prof. Dr. Alexander Mädche

Korreferent: Prof. Dr. Brad A. Myers

Karlsruhe

März 2023



# Acknowledgments

I'd like to thank Prof. Alexander Mädche, my mentor and PhD supervisor, for his advice, motivation, and comments throughout my studies. While I had no idea what to anticipate when I started my PhD, I quickly realized how fortunate I was with my supervisor. Alexander was always available to answer my inquiries, thoughts, and concerns. I am extremely appreciative of your assistance. On that topic, I'd additionally like to thank Prof. Brad A. Myers, Prof. Benjamin Scheibehenne, and Prof. Johannes Brumm for volunteering to serve on my PhD committee. I had a great time discussing my study with you! Brad deserves special recognition for welcoming me into the HCI community in Pittsburgh during my semester abroad and helping me improve both my research at hand and my approach to future research projects!

I'd like to thank my colleagues at the Institute of Information Systems and Marketing (IISM), particularly those in the Human-Centered Systems Lab. Thank you for constantly assisting with pre-tests, providing helpful comments during our research sessions, and for the frequent coffee breaks. I am grateful for that.

I am extremely grateful and delighted to have always had the support of my friends! I am grateful for the bonds that have remained strong from the early days of my university study, and I am grateful to those I met during my later studies. Thank you for the good times, your support, and your constant presence.

My heartfelt thanks also go to my parents and family. This dissertation is dedicated to my mother, Birgit Richter-Ruoff, and my father, Oliver Ruoff. I will be eternally grateful for your unending and unconditional love and support, as well as your great faith in my skills. Thank you for accompanying me on this trip.

Marcel Ruoff

Karlsruhe, Germany

March 2023

# Abstract

*How many Covid cases and deaths are there in my hometown? How much money was invested into renewable energy projects across states in the last 5 years? How large was the biggest investment in solar energy projects in the previous year?* These questions and others are of interest to users and can often be answered by data visualization tools (e.g., COVID-19 dashboards) provided by governmental organizations or other institutions. However, while users in organizations or private life with limited expertise with data visualization tools (hereafter referred to as end users) are also interested in these topics, they do not necessarily have knowledge of how to use these data visualization tools effectively to answer these questions. This challenge is highlighted by previous research that provided evidence suggesting that while business analysts and other experts can effectively use these data visualization tools, end users with limited expertise with data visualization tools are still impeded in their interactions.

One approach to tackle this problem is natural language interfaces (NLI) that provide end users with a more intuitive way of interacting with these data visualization tools. End users would be enabled to interact with the data visualization tool both by utilizing the graphical user interface (GUI) elements and by just typing or speaking a natural language (NL) input to the data visualization tool. While NLI for data visualization tools have been regarded as a promising approach to improving the interaction, two design challenges still remain. First, existing NLI for data visualization tools still target users who are familiar with the technology, such as business analysts. Consequently, the unique design required by end users that address their specific characteristics and that would enable the effective use of data visualization tools by them is not included in existing NLI for data visualization tools. Second, developers of NLI for data visualization tools are not able to foresee all NL inputs and tasks that end users want to perform with these NLI for data visualization tools. Consequently, errors still occur in current NLI for data visualization tools. End users need to be therefore enabled to continuously improve and personalize the NLI themselves by addressing these errors. However, only limited work exists that focus on enabling end users in teaching NLI for data visualization tools how to correctly respond to new NL inputs.

This thesis addresses these design challenges and provides insights into the related research questions. Furthermore, this thesis contributes prescriptive knowledge on how to design effective NLI for data visualization tools. Specifically, this thesis provides insights into



---

how data visualization tools can be extended through NLI to improve their effective use by end users and how to enable end users to effectively teach NLI how to respond to new NL inputs. Furthermore, this thesis provides high-level guidance that developers and providers of data visualization tools can utilize as a blueprint for developing data visualization tools with NLI for end users and outlines future research opportunities that are of interest in supporting end users to effectively use data visualization tools.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Abbreviations</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Research Questions . . . . .	5
1.3 Thesis Structure . . . . .	10
<b>2 Study I: Designing Conversational Dashboards for Effective Use in Crisis Response</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Theoretical Foundations and Related Work . . . . .	15
2.2.1 Conversational User Interfaces . . . . .	15
2.2.2 Dashboards . . . . .	16
2.2.3 Conversational Dashboards . . . . .	17
2.2.4 Theory of Effective Use . . . . .	19
2.3 Designing Conversational Dashboards for Crisis Response . . . . .	20
2.3.1 Design Process . . . . .	20
2.3.2 Problem Description and Meta-Requirements . . . . .	22
2.3.3 Design Principles . . . . .	23
2.3.4 Testable Design Propositions . . . . .	26
2.3.5 Artifact Description . . . . .	27
2.3.5.1 Dashboard and Data Visualization Component . . . . .	27
2.3.5.2 Interaction Management Component . . . . .	28
2.3.5.3 Natural Language Processing (NLP) Component . . . . .	29
2.3.5.4 Conversational Onboarding . . . . .	30


2.4	Evaluation . . . . .	31
2.4.1	Performance Evaluation of the Natural Language Processing (NLP) Component . . . . .	32
2.4.1.1	Performance Measures . . . . .	33
2.4.1.2	Results . . . . .	33
2.4.2	User Evaluation . . . . .	35
2.4.2.1	Method . . . . .	37
2.4.2.2	Results . . . . .	39
2.5	Discussion . . . . .	42
2.5.1	Theoretical Contributions . . . . .	43
2.5.2	Practical Implications . . . . .	45
2.5.3	Limitations and Future Research . . . . .	46
2.6	Conclusion . . . . .	47
<b>3</b>	<b>Study II: Designing Multimodal BI&amp;A Systems for Co-Located Team Interactions</b>	<b>49</b>
3.1	Introduction . . . . .	49
3.2	Related Work and Theoretical Foundations . . . . .	51
3.2.1	Business Intelligence & Analytics Systems for Teams . . . . .	51
3.2.2	Multimodal User Interfaces . . . . .	52
3.2.3	Theory of Effective Use . . . . .	52
3.3	Design Science Research Project . . . . .	53
3.4	Results . . . . .	57
3.4.1	Awareness of the Problem . . . . .	57
3.4.2	Suggestion . . . . .	59
3.4.3	Development . . . . .	61
3.4.4	Evaluation . . . . .	63
3.5	Discussion . . . . .	65
3.6	Conclusion . . . . .	67
<b>4</b>	<b>Study III: ONYX: Assisting Users in Teaching Natural Language Interfaces Through Multi-Modal Interactive Task Learning</b>	<b>68</b>
4.1	Introduction . . . . .	68
4.2	Related Work . . . . .	71
4.2.1	Learning Tasks through Demonstrations . . . . .	71
4.2.2	Natural Language Interfaces with Learning Capabilities . . . . .	72
4.2.3	Natural Language Interfaces for Data Visualization Tools . . . . .	73

4.3	Formative Study & Design Goals . . . . .	73
4.3.1	Understanding of the NLI's Existing Knowledge . . . . .	74
4.3.2	Ambiguous Direct Manipulation Demonstrations . . . . .	75
4.3.3	Design of Assistance . . . . .	75
4.4	ONYX . . . . .	76
4.4.1	Example Scenario . . . . .	76
4.4.2	Key Design Features . . . . .	79
4.4.2.1	Learning from Multi-Modal User Demonstrations . . . . .	79
4.4.2.2	Suggestions . . . . .	79
4.4.2.3	Follow-Up Questions . . . . .	81
4.4.2.4	Visual and Textual Aids . . . . .	82
4.4.2.5	Generalization. . . . .	83
4.4.3	System Architecture . . . . .	83
4.4.3.1	Natural Language Parser . . . . .	84
4.4.3.2	Interactive Task Learning Agent . . . . .	85
4.5	Evaluative User Studies . . . . .	85
4.5.1	Participants . . . . .	85
4.5.2	Procedure . . . . .	86
4.5.3	Tasks . . . . .	87
4.5.4	Results . . . . .	88
4.5.4.1	Suggestions and Follow-Up Questions . . . . .	88
4.5.4.2	Visual and Textual Aids . . . . .	91
4.5.4.3	Display and Adaptation of <i>ONYX</i> 's Understanding . . . . .	92
4.6	Discussion . . . . .	92
4.7	Limitations and Future Work . . . . .	94
4.8	Conclusion . . . . .	95
<b>5</b>	<b>Study IV: ContextIT - Interactively Contextualizing Natural Language Inputs in Data Visualization Tools</b>	<b>97</b>
5.1	Introduction . . . . .	97
5.2	Related Work . . . . .	100
5.2.1	Natural Language Interfaces with Learning Capabilities . . . . .	100
5.2.2	Natural Language Interfaces for Data Visualization Tools . . . . .	101
5.3	Formative Studies & Design Goals . . . . .	102
5.3.1	Clarifying Ambiguities in NL Inputs through Context . . . . .	103

5.3.2	Supporting Users in Contextualizing NL inputs . . . . .	104
5.3.3	Guiding Users in Contextualizing NL inputs . . . . .	105
5.4	ContexIT . . . . .	106
5.4.1	Example Scenario . . . . .	106
5.4.2	Key Features . . . . .	107
5.4.2.1	Specifying Contextual Conditions. . . . .	108
5.4.2.2	Guiding Users through Visual and Textual Cues. . . . .	109
5.4.2.3	Suggesting Contextual Conditions. . . . .	110
5.4.2.4	Refining Existing Conflicting Interpretations. . . . .	111
5.4.2.5	Generalization of Contextual Conditions. . . . .	112
5.4.3	System Architecture . . . . .	112
5.4.3.1	NL Parser . . . . .	112
5.4.3.2	ITL Agent . . . . .	113
5.5	Evaluative User Study . . . . .	113
5.5.1	Participants . . . . .	113
5.5.2	Procedure . . . . .	114
5.5.3	Tasks . . . . .	114
5.5.4	Results . . . . .	115
5.5.4.1	Overall Effectiveness of ContexIT. . . . .	115
5.5.4.2	Effectiveness of Support by <i>ContexIT</i> . . . . .	117
5.5.4.3	Effectiveness of Guidance by <i>ContexIT</i> . . . . .	119
5.6	Discussion . . . . .	119
5.7	Conclusion . . . . .	121
<b>6</b>	<b>Discussion</b>	<b>122</b>
6.1	Theoretical Contributions . . . . .	122
6.2	Practical Implications . . . . .	127
6.3	Limitations and Future Research . . . . .	128
6.3.1	Large-Scale Quantitative Evaluation & Field Studies . . . . .	128
6.3.2	Improving the NL Processing . . . . .	129
6.3.3	Improving the Interaction with the NLI . . . . .	129
6.3.4	Integrating Richer Context into the NLI's Interpretation . . . . .	130
6.3.5	Extending the Tasks Supported by the ITL-based NLI for Data Visualization Tools . . . . .	131
6.3.6	Switching more easily between Datasets . . . . .	131

6.3.7	ITL-based NLI beyond Data Visualization Tools . . . . .	132
6.3.8	Connecting Natural Language and Pointing Devices more tightly . .	132
6.3.9	Beyond the Combination of Natural Language and Pointing Devices	133
6.3.10	Combining ITL and Machine Learning-based Approaches for NLIs .	133
6.3.11	Learning Across Different Users . . . . .	134
<b>7</b>	<b>Conclusion</b>	<b>135</b>
<b>8</b>	<b>Appendix</b>	<b>136</b>
A	Study I: Additional Material for User Evaluation (Online Experiment) . . .	136
B	References to Code Repositories, Study Procedures, and Data sets . . . . .	138
B.1	Study I: . . . . .	138
B.2	Study II: . . . . .	138
B.3	Study III: . . . . .	138
B.4	Study IV: . . . . .	138
	<b>Bibliography</b>	<b>139</b>
	<b>List of Publications</b>	<b>160</b>

# List of Figures

1.1	Overview of Research Questions addressed in this Thesis. . . . .	5
1.2	Structure of the Thesis. . . . .	12
2.1	Overview of our DSR approach. . . . .	20
2.2	System Architecture of the Conversational Dashboard. . . . .	27
2.3	Screenshot of the Conversational Dashboard with DP1 and DP2. . . . .	28
2.4	Screenshot of the Conversational Onboarding (DP3). . . . .	32
2.5	Effect of Proportion of Interactions performed via Natural Language and Conversational Onboarding on Transparent Interaction (CDB-NLE). . . . .	41
3.1	Theory of Effective Use (adapted from Burton-Jones and Grange (2013)). . . . .	53
3.2	Design Science Research Project (adopted from Kuechler and Vaishnavi (2008)). . . . .	54
3.3	Multimodal BI&A System in Co-located Team Interactions at Industry Partner. . . . .	61
3.4	Instantiation of the second and third Design Principle. . . . .	62
4.1	Data Visualizations and NL Inputs utilized in the Scenario. . . . .	77
4.2	User Interface of the Data Visualization Tool with integrated ITL-based NLI during the Demonstration Process. <sup>(A)</sup> Visualization canvas, <sup>(B)</sup> Buttons to adapt chart type and encodings, <sup>(C)</sup> Filter pane to provide constraints, <sup>(D)</sup> NLI providing text input and feedback, <sup>(E1)</sup> Signalling indicator and buttons to finish the demonstration mode, <sup>(E2)</sup> NL input to be demonstrated, <sup>(E3)</sup> Visual Representation of the Script. . . . .	78
4.3	Progression of the User Interface, the Script, and the ITL Agent during the Training for the NL Input  . . . . .	78
4.4	Screenshot of the UI when <i>ONYX</i> detects a possible Goal Attainment during the Demonstration Process. . . . .	80
4.5	A Sample Sentence tagged and processed by <i>ONYX</i> to derive Suggestions. The Words of the Sentence are connected with directed Arrows based on their Dependency Structure. . . . .	81

4.6	Follow-Up Question, after <i>ONYX</i> detects a Direct Manipulation Ambiguity in the last User Action. . . . .	82
4.7	UI when Users hover over underlined Parameters during the Demonstration Process, with highlighted States Filter at <b>A</b> . . . . .	82
4.8	System Architecture Overview. . . . .	84
4.9	Target Visualizations after completing each Task A-C. . . . .	87
4.10	Boxplot of the Accuracy and Time for both Conditions for all 3 Tasks. For Accuracy, a higher Rating means better Accuracy. For Time, a lower Rating means better Time. . . . .	88
4.11	The Average Error in the Learned Scripts across Tasks A - C colored by the Reason of the Error. . . . .	89
4.12	Responses to Post-Study Likert-scale Questions about the Experience of Participants with <i>ONYX</i> 's Features. . . . .	89
4.13	The Number of Participants receiving and addressing Follow-up Questions at different Timings during the Task . . . . .	91
4.14	The Average Number of Visual and Textual Aids requested by Participants across Tasks A - C in the Baseline and Treatment Conditions. The Values are colored by the Concept explained by the Visual and Textual Aid. . . . .	91
5.1	User Interface of <i>ContexIT</i> during the Contextualization Process. . . . .	99
5.2	<i>ContexIT</i> 's User Interface depicted in its three Modes: (i) Interaction <b>A</b> , (ii) Demonstration <b>B</b> , and (iii) Contextualization <b>C</b> . . . . .	106
5.3	<i>ContexIT</i> 's User Interface during the Contextualization with the Visual and Textual Aids <b>B</b> & <b>B</b> and depicting the Contextual Conditions <b>D</b> when Users are hovering over a GUI Element <b>A</b> . . . . .	108
5.4	<i>ContexIT</i> 's Textual Aids when switching from Demonstration to Contextualization Mode <b>A</b> and when Users ask for Assistance during the Contextualization <b>B</b> . . . . .	110
5.5	<i>ContexIT</i> 's decision-making for deriving the Contextual Conditions for differentiating two conflicting Interpretations. . . . .	111
5.6	Accuracy in specifying Contextual Conditions across Tasks for each Participant. . . . .	116
5.7	Responses to Post-Study Likert-scale Questions about the Experience of Participants with <i>ContexIT</i> 's Features. . . . .	117
5.8	Size of the Groups of Participants utilizing the same Contextual Conditions in relation to all Participants across the five Tasks. . . . .	117



5.9	Number of Suggestions received per Participant and whether they have been selected for Contextualizing the NL Input for Specification Tasks (A.2, B.2, and B.3.) . . . . .	118
5.10	Number of Options for refining existing conflicting Interpretations received per Participant and whether they have been selected for differentiating the conflicting Interpretations for the Tasks A.1, B.1, and B.2. . . . .	119

# List of Tables

2.1	Intents, Entities, and Examples in the Language Model . . . . .	30
2.2	Natural Language Response Templates . . . . .	31
2.3	Performance Evaluation Results for Entity Extraction and Intent Mapping	34
2.4	Overview of Dashboard Types and Instantiated Design Principles. . . . .	36
2.5	Descriptive Statistics for Transparent Interaction. . . . .	40
2.6	Summary of the Results. . . . .	42
2.7	A Nascent Design Theory for Conversational Dashboards in Crisis Response .	48
3.1	Summary of the SWOT Analysis. . . . .	63
5.1	Tasks covered by the User Study, whether Users are required to Specify or Refine Contextual Conditions to correctly contextualize the Interpretation, and a short Description of the Interpretation. . . . .	115
6.1	Summary of the Theoretical Contributions of this Dissertation. . . . .	123
A.1	Demographic Information of Participants . . . . .	136
A.2	Experimental Tasks . . . . .	136
A.3	Calculation of Transparent Interaction based on Users' Navigation Path: Examples . . . . .	137

# List of Abbreviations

AI .....	Artificial Intelligence
BI .....	Business Intelligence
BI&A .....	Business Intelligence & Analytics
CUI .....	Conversational User Interface
DG .....	Design Goal
DP .....	Design Principle
DSR .....	Design Science Research
GDSS .....	Group Decision Support Systems
GUI .....	Graphical User Interface
HCI .....	Human-Computer Interaction
IS .....	Information System
ITL .....	Interactive Task Learning
LUIS .....	Language Understanding Intelligent Service
MR .....	Meta-Requirements
MUI .....	Multimodal User Interface
NL .....	Natural Language
NLI .....	Natural Language Interface
NLP .....	Natural Language Processing
NLU .....	Natural Language Understanding
RQ .....	Research Question
SWOT .....	Strength-Weakness-Opportunity-Threat
TEU .....	Theory of Effective Use

# 1. Introduction <sup>1</sup>

## 1.1 Motivation

People are increasingly provided with the ability to explore and analyze private life data on their own. They utilize this data among others to better understand their health (Y.-H. Kim et al., 2021), their smart homes (Castelli et al., 2017), and the course of current crises (e.g., COVID-19; Pietz et al., 2020). To satisfy this need they often rely on data visualization. Data visualizations are a visual representation of information and data, such as maps and plots, and can be understood as graphical aids that assemble thousands of data objects into pictures, revealing hidden patterns (Card et al., 1999).

In organizations, data visualization tools (systems supporting the interaction with data visualizations) are viewed as one of the most useful tools for analyzing data and deriving decisions from the gained insights (Yigitbasioglu & Velcu, 2012). Due to this importance, specific roles are established in organizations, such as business analysts, that focus on creating data visualizations utilizing various commercial applications (e.g., PowerBI, Tableau) that are later used by business users with dedicated domain expertise (Tory, Bartram, et al., 2021). To deliver comprehensive information, business analysts utilize a special form of data visualization tool, namely dashboards. Dashboards are a “visual display of the most important information needed to achieve one or more objectives; consolidated and organized on a single screen so the information can be monitored at a glance” (Few, 2006). Existing research has shown that business analysts are able to increase their task performance by effectively using dashboards in a single-user context (Nadj et al., 2020). Hence, dashboards are widely applied in organizations to support various tasks, such as IT architecture management (Widjaja & Gregory, 2020), supply chain management (Park et al., 2016), and production planning (Hu et al., 2012).

While researchers and practitioners assumed that users with limited technical expertise (hereafter referred to as *end users*) would benefit from data visualization tools (e.g. in the form of dashboards) similarly to business analysts as long as the underlying data is accurate (Patino, 2021; Soper et al., 2021), these assumptions were shown to be incorrect (Cay et al., 2020; Momenipour et al., 2021; Smuts et al., 2015; Young, Kitchin, & Naji, 2021). Generally, users of data visualization tools want to answer their current questions by finding

---

<sup>1</sup>This chapter is based on the following studies which are published: Ruoff and Gnewuch (2021a), Ruoff and Gnewuch (2021b), Ruoff, Myers, et al. (2021), Ruoff, Gnewuch, Maedche, and Scheibehenne (2022) and Ruoff, Myers, et al. (2023).

the data visualization that best reflects the insights they require (Tory, Bartram, et al., 2021). To enable users in achieving this effectively, designers of data visualization tools have to find the appropriate trade-off between providing the right amount of information and functionality to derive the required insights, without overwhelming the users with excessive information or functionalities (Yigitbasioglu & Velcu, 2012). The appropriate trade-off greatly depends on the individual characteristics of users, their technical expertise with data visualization tools, and their knowledge about the domain (Cay et al., 2020; Smuts et al., 2015). Particularly, end users with their limited technical expertise “struggled to direct themselves around the [data visualization tools], being regularly confounded by unclear primary navigation options and indistinguishable secondary navigation schemes” (Young, Kitchin, & Naji, 2021, p. 12). Therefore, end users have issues using data visualization tools to get to the data visualization that answers their questions. Often, they fail with their task due to these challenges in deriving the required insights (Momenipour et al., 2021). For example, users can face challenges when trying to select points of interest, such as focussing on a specific location, as they have issues zooming in or filtering on these points of interest (Cay et al., 2020). Some end users even are overwhelmed by data visualization tools in general and do not know where to start with their data exploration and analysis (Cay et al., 2020; Young, Kitchin, & Naji, 2021). This impediment in translating the questions of end users to the correct data visualizations prevents end users from effectively using data visualization tools to fulfill their goals.

One promising way to address this challenge is to move beyond the traditional graphical user interface (GUI) and to provide users with a more natural way of interacting with a data visualization tool using natural language (NL) (Lee, Choe, et al., 2020). NL can make navigating the data visualization tools less complex and finding the required data visualizations less difficult because it allows users to formulate their information needs more naturally, similar to the way they would in an everyday conversation. For example, instead of using menus, filters, and sliders, users could simply ask the data visualization tool for any information they need (e.g., “*What was the number of COVID-19 cases in Pennsylvania last week?*”).

The idea of using NL “to enable non-technical people to access complex databases” through NL interfaces (NLI) has been around for some time (Turban & Watkins, 1986, p. 127). With NLI, users can communicate via NL with the data visualization tool. While users provide their input either through spoken or written form to the NLI, the NLI provides a response to users either through text, speech or by adapting the GUI. The first NLIs

date back to the 1960s with ELIZA, one of the first NLIs (Weizenbaum, 1966). Early on, NLIs have been proven to provide “reasonably good natural language access to specific data bases” and to be able to “answer direct questions” (Hendrix, 1982, p. 56). With the increasing advances in NL processing, researchers have been investigating NLIs in various contexts (Diederich, Brendel, Morana, et al., 2022). Building on these insights, NLIs are now available across various use cases to enable users to interact through NL to answer general questions (e.g., ChatGPT) and perform common tasks on smartphones or smart speakers (e.g., Siri, Google Assistant), to navigate the web (e.g., FireFox Voice; Cambre et al., 2021), and to ask for data visualizations in data visualization tools (e.g., Tableau AskData, PowerBI Q&A).

However, despite the promising capabilities of NLIs to enable users to directly ask data visualization tools the questions they seek to answer, existing NLIs for data visualization tools often do not meet users’ expectations (Tory & Setlur, 2019). Researchers and practitioners alike have realized that just enriching traditional data visualization tools by adding an NLI does not enable end users to effectively use this data visualization tool with NL capabilities on its own (Srinivasan & Stasko, 2018; Tory & Setlur, 2019). Particularly, new challenges and requirements for the design of data visualization tools arise when enabling users to interact through NL.

**First**, NLIs have mainly been regarded as an alternative to traditional GUI-based data visualization tools. And it was assumed that at some point NLIs might replace data visualization tools that are equipped solely with a GUI (Fast et al., 2018; Hearst & Tory, 2019). While it is easier for users to ask questions in NLIs as they do not have to translate their questions into adaptations of the GUI, it is difficult to convey the extensive information required during data exploration and analysis solely in an NLI (Hearst & Tory, 2019; Setlur & Tory, 2022). Therefore, instead of *replacing* the data visualization tools, it could be more valuable to *complement* its GUI with an NLI so that users can use *both* interfaces to access information during the data exploration and analysis. However, adding a new way of interaction to data visualization tools (e.g., NL) requires understanding how this changes the required design of data visualization tools and how this differs between business analysts and end users. For example, traditional data visualization tools for end users enable them to only utilize a few functionalities to interact with data visualizations to limit their information overload. However, through more intuitive ways of interaction, more possibilities to interact with data visualization tools could be provided to users without overwhelming them (Y.-H. Kim et al., 2021).

Early research on extending data visualization tools through NLI has predominantly focused on assessing the practical viability of this approach. The systems developed and investigated were restricted to simple research prototypes (Lee, Srinivasan, et al., 2021). Sun et al. (2010), for example, demonstrated in their Articulate system how an NLI integrated into a data visualization tool could be technologically accomplished. However, in their evaluation, they only analyzed the performance of the NLI in correctly classifying the NL inputs. In contrast, recent research highlights that the design of data visualization tools with NL capabilities should not only be technology-centric and focus on the performance of the NLI (e.g., accuracy in intent detection) but should also take the user perspective into consideration (Setlur & Tory, 2022). While business analysts are increasingly considered in small samples when designing data visualization tools with NL interfaces for single-user use cases (e.g., Setlur, Battersby, et al., 2016; Srinivasan and Stasko, 2020; Y. Wang et al., 2022), the design required by end users with limited technical expertise are not well understood.

**Second**, errors still occur in current NLIs when they fail to understand what users want them to perform based on the users' NL input, also known as *breakdowns* (Ashktorab et al., 2019). Current approaches to address these breakdowns only aim to elicit information to address the current breakdown or to adapt the behavior of users by teaching them how they should interact with the NLI integrated into the data visualization tool for it to work accurately. However, it would be beneficial for the effective use of the data visualization tool to also enable the NLI to learn from users how to handle new NL inputs after a breakdown or unexpected system behavior, such as an incorrect interpretation of the NL input. Through this approach, NLIs could reduce their number of overall breakdowns. And users would be able to use their personal linguistic style in future interactions and to better remember the NL inputs they are able to perform with the NLI (S. I. Wang et al., 2017). While NLIs integrated into data visualization tools are starting to enable users to teach them how to handle new NL inputs, the expressiveness of these approaches is currently limited, and only simple form-filling techniques are provided to users to specify synonyms (e.g., *price*  $\Leftrightarrow$  *fee*) and simple boolean concepts (e.g., important product  $\Leftrightarrow$  products with yearly profits over 1 million).

In this thesis, I investigate how to improve the design of NLIs for data visualization tools. Specifically, I address both design challenges discussed above: (i) to enable end users to effectively use data visualization tools integrating an NLI to fully leverage the potentials of the NL capabilities, and (ii) to enable them to effectively teach the NLIs if breakdowns occur.

To derive theory-driven designs to address the first challenge, I utilize the design science research (DSR) paradigm (Hevner et al., 2004) which is prominent in the information systems discipline. To cope with the novelty of the second challenge, I more strongly focus on methodologies and insights from the human-computer interaction (HCI) discipline. By building and evaluating novel artifacts to address both challenges, I contribute prescriptive knowledge in the form of design principles and goals as well as nascent design theories. Subsequently, I describe the research gaps in more detail and derive research questions (RQs) for the studies of my thesis.

## 1.2 Research Questions

This thesis explores data visualization tools that integrate NLI. Particularly, how to support end users in effectively using these data visualization tools for their data exploration and analysis and how to enable them to effectively teach the NLI how to perform new NL inputs. Therefore, I highlight two main RQs that I addressed with four studies presented in this thesis, as shown in Figure 1.1. I introduce these research questions in the following in more detail.

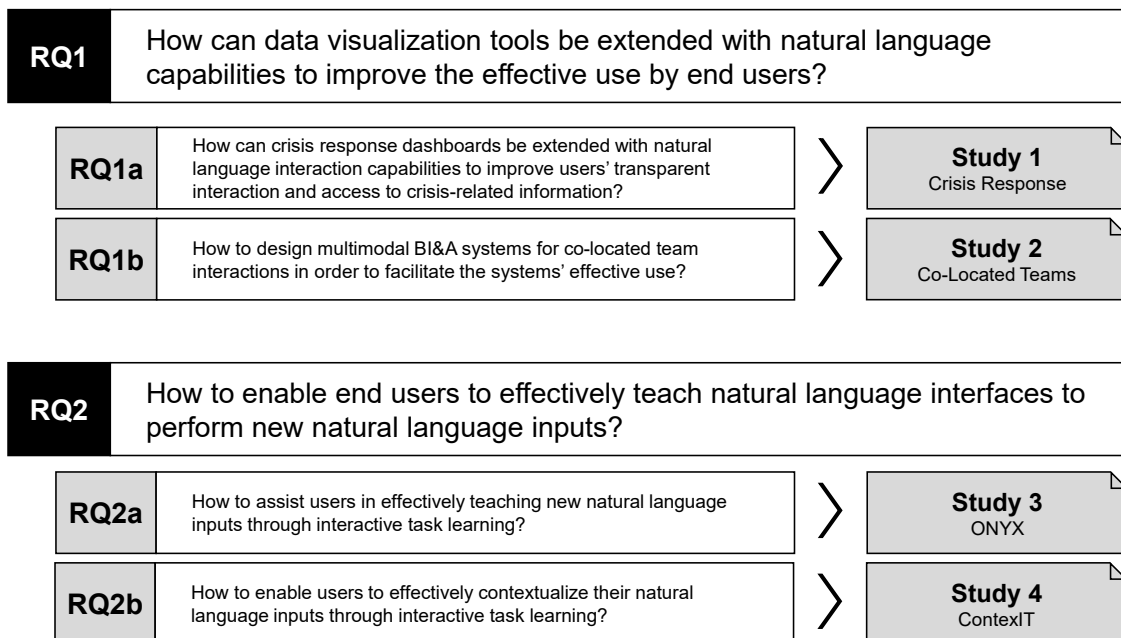


Figure 1.1: Overview of Research Questions addressed in this Thesis.

The **first RQ** deals with the design of data visualization tools integrating NLI for effective use by end users to address the first design challenge.

As end users have little experience with data visualization tools and how to properly interact with them, it is necessary to understand how NLI can be utilized to support these end users



in their data exploration and analysis. However, existing research on data visualization tools integrating NLI has not thoroughly considered the special requirements of end users for their design. After earlier research has demonstrated the technical requirements for integrating the GUI of data visualization tools with NLI, recent studies have highlighted how crucial it is to also consider the user perspective of the interaction (Setlur & Tory, 2022; Srinivasan, Lee, et al., 2020). While these studies are increasingly looking beyond the technological requirements, they are still primarily focussing on how to support data analysts or other expert users in supporting the usage of data visualization tools integrating NLI. However, due to the differences in how data analysts and experts use data visualization tools in contrast to end users, we know little about how to design data visualization tools integrating an NLI for effective use by end users. Therefore, I seek to answer the following main research question:

***RQ1:*** *How can data visualization tools be extended with natural language capabilities to improve the effective use by end users?*

To answer the first main research question, I investigate data visualization tools in two different scenarios in which end users are utilizing data visualization tools. First, I explore the design of crisis response dashboards (e.g., COVID-19 dashboards), a type of data visualization tool targeted at supporting end users in their decision-making, and how to extend them through natural language interaction capabilities. Second, I explore the design of multimodal BI&A systems in organizations that are utilized in co-located team meetings during decision-making, specifically dashboards that provide interaction through touch and speech for ad-hoc data analysis. Subsequently, I explain the associated research gaps in more detail to motivate the two sub-research questions.

**First**, crises are by nature unpredictable, sudden, and often chaotic situations. When a crisis occurs, people want to find accurate and up-to-date information quickly so that they can make the best decisions for themselves, their families, and their communities (C. Leong et al., 2015). Especially during the COVID-19 pandemic, end users have been increasingly provided with dashboards by organizations and governmental institutions to address this information need (Pietz et al., 2020). However, recent research has shown that end users have issues interacting with these crisis response dashboards due to their limited expertise with dashboards (Cay et al., 2020; Young, Kitchin, & Naji, 2021). However, when users are unable to interact with a dashboard transparently, their chance of making effective decisions based on the data provided in the dashboard is dramatically reduced (Burton-Jones & Grange, 2013). Therefore, I seek to answer the following sub-research question:

**RQ1a:** *How can crisis response dashboards be extended with natural language interaction capabilities to improve users' transparent interaction and access to crisis-related information?*

To answer this research question, I conducted a DSR project (Hevner et al., 2004) to propose a theory-driven design for dashboards integrating an NLI that can be effectively used by end users. Particularly, I investigated how end users can be supported in crisis response by a data visualization tool integrating an NLI, specifically in the case of COVID-19 dashboards. I proposed three design principles for data visualization tools integrating NLIs based on the theory of effective use. Subsequently, I evaluated the design in a large-scale online experiment and articulated a nascent design theory from the derived insights. In the large-scale online experiment, participants had significant differences in transparent interaction due to both the type of interaction provided to participants ( $F(2, 265) = 48.3$ ,  $p < .001$ ) and whether they were provided with onboarding ( $F(1, 265) = 7.38$ ,  $p = .007$ ). Furthermore, the results show that transparent interaction ultimately has a significant positive effect on the participants' efficiency ( $\beta = 0.58$ ,  $p < .001$ ) and effectiveness ( $\beta = 0.9$ ,  $p < .001$ ).

**Second**, decision-making in organizations is increasingly performed by teams consisting of multiple end users in co-located interactions (Dennis, 1996; Isenberg et al., 2012; Majchrzak et al., 2012) and supported by BI&A dashboards as part of BI&A systems. However, few BI&A dashboards for data-driven decision-making in organizations support the ad-hoc analysis of data in co-located team interactions (Berthold et al., 2010; Isenberg et al., 2012) and many teams are struggling in working together both equitable and flexible using contemporary BI&A systems (Dayal et al., 2008; Kaufmann & Chamoni, 2014). Furthermore, these BI&A systems are mainly targeted at individual users with technical expertise with the underlying system. To improve the interaction between the team and the BI&A dashboard, providing multiple modalities that compensate for each other's weaknesses could be promising to support teams in using these BI&A dashboards more effectively (Deng et al., 2004). Therefore, I seek to answer the following sub-research question:

**RQ1b:** *How to design multimodal BI&A systems for co-located team interactions in order to facilitate the systems' effective use?*

To answer this research question, I conducted a DSR project (Hevner et al., 2004) to propose a theory-driven design for a multimodal BI&A dashboard that can be effectively used by multiple end users in organizations. Particularly, I investigated how multiple end

users in a co-located scenario can be supported in effectively using a dashboard during a meeting by providing them the means to interact with the system through touch and speech. I again derived three design principles based on the theory of effective use. The design was subsequently evaluated in focus groups to investigate the effectiveness of the design. The results of the evaluation suggest that the effective use of multimodal BI&A systems in co-located team interactions can be increased by offering touch and speech modalities on a large interactive display.

The **second RQ** deals with supporting end users in teaching NLI to prevent future breakdowns of the NLI to address the second design challenge.

One of the major drawbacks of contemporary NLI is how they handle breakdowns when incorrectly interpreting the NL input of end users. The breakdowns can be classified into three categories (Yu & Silva, 2020): (1) The NL input is not supported, (2) the context is invalid or information is missing, or (3) unexpected system behavior. In existing NLI, the first error category is often addressed through generic prompts, such as “Unable to process that command. Please try a different one” (Srinivasan, Lee, et al., 2020, p. 7) or “Sorry, I couldn’t understand.” (Y.-H. Kim et al., 2021, p. 6). This enables users to understand the limitations of the NLI and to adapt their behavior by rephrasing their NL input or by switching to the traditional GUI of the data visualization tool. To address the second error category, NLI like Iris (Fast et al., 2018) use NL prompts to request missing information that is required for the task. Eviza (Setlur, Battersby, et al., 2016) additionally employs simple pragmatics to maintain information between subsequent NL inputs. DataTone (Gao et al., 2015), DIY (Narechania, Fourney, et al., 2021), and Sentifiers (Setlur & Kumar, 2020) use disambiguation widgets to extract explicit information with the help of users from ambiguous data attributes (Gao et al., 2015; Narechania, Fourney, et al., 2021) and vague modifiers (Setlur & Kumar, 2020). The third error category is mainly addressed by providing users the ability to adjust the provided data visualization which does not fit their expected result of the NL input using either direct manipulation or subsequent NL inputs. However, when users continue after a breakdown of the NLI either by rephrasing their NL input, by providing additional information, or by switching to the GUI, the NLI does not remember how to solve the breakdown in future interactions and needs to involve users again even if the same or similar situation occurs. A promising approach to allow NLI to learn from end users after a breakdown without requiring the end users to familiarize themselves with a programming language is *interactive task learning* (ITL) (T. J.-J. Li, Azaria, et al., 2017; S. I. Wang et al., 2017). ITL-based systems learn from the actions users

perform in the actual system after a breakdown to prevent future breakdowns in similar situations. However, in existing systems, end users can be overwhelmed by the information required by the ITL-based system to address ambiguities in their demonstrated actions or would be required to provide a plethora of examples. Hence, end users are currently unable to effectively teach these ITL-based systems how to perform new NL inputs. Therefore, I seek to answer the following main research question:

**RQ2:** *How to enable end users to effectively teach natural language interfaces to perform new natural language inputs?*

To investigate the second research question, I explore how users can be supported through ITL in two important steps of teaching NLIs to perform natural language inputs. First, I investigate how users can be supported during the demonstration of a new interpretation of an NL input to improve the effectiveness of the teaching outcome. Second, I investigate how end users can be enabled to contextualize the NL inputs by providing pre-conditions based on the current context for their interpretations to allow the NLI to have multiple possible interpretations for one NL input. Subsequently, I explain the associated research gaps in more detail to motivate the two sub-research questions.

**First**, existing ITL-based systems that are able to learn how to perform new NL inputs currently have two shortcomings. First, while demonstrations of interpretations in current ITL-based systems communicate what a user does, existing ITL-based systems are limited in deriving why or how to perform these actions in varying contexts. Hence, users are either required to explain their reasoning in lengthy textual descriptions of their intentions for each action (T. J.-J. Li, Labutov, et al., 2018) or by providing a plethora of examples. Second, existing ITL-based systems are limited in the support they provide to end users during the demonstration process based on existing knowledge of the NLI itself. Hence, what is currently missing are ITL-based systems that support users in teaching the NLI integrated into data visualization tools through active assistance. Therefore, I seek to answer the following sub-research question:

**RQ2a:** *How to assist users in effectively teaching new natural language inputs through interactive task learning?*

I address RQ2a by designing and developing *ONYX*, an ITL agent for NLIs integrated into data visualization tools with the ability to learn from end users. *ONYX* aims to support end users through suggestions during the demonstration process and supports them in addressing ambiguities in their demonstrated actions through follow-up questions.

The design was derived through participatory design and six design goals were derived. I instantiated these design goals in *ONYX* and showed in an online experiment that participants utilizing the design achieved significantly ( $p < 0.001$ ) higher accuracy in teaching new NL inputs (median: 93.3%) in contrast to those without (median: 73.3%). I further provide qualitative insights through additional think-aloud sessions to better understand how users utilized the design.

**Second**, while the context is crucial for deriving the correct interpretation for an NL input (Reinhart, 1981; Setlur, Battersby, et al., 2016), existing ITL-based systems are unable to learn the underlying contextual pre-conditions for the different interpretations of an NL input. Due to this shortcoming, existing ITL-based systems are only able to learn one interpretation for each NL input. However, NL inputs have often multiple correct interpretations that depend on the context, such as the current state of the system. To address this shortcoming, I seek to answer the following sub-research question:

***RQ2b:** How to enable users to effectively contextualize their natural language inputs through interactive task learning?*

I address RQ2b by designing and developing *ContexIT*. *ContexIT* aims to support users in teaching the NLI to understand the contextual pre-conditions of their interpretation of the NL input through suggestions and refinement of previous knowledge. The design goals were derived from an NL elicitation study, a design workshop, and participatory design studies. The design was subsequently evaluated in think-aloud studies to provide evidence for the effectiveness of the design and to receive insights into the RQ. The results demonstrate that participants were able to accurately contextualize the possible interpretations of their NL inputs with an accuracy of 92.5% with the help of *ContexIT*.

### 1.3 Thesis Structure

In Figure 1.2 the structure of the cumulative thesis is illustrated. **Chapter 1** describes the motivation of the thesis and its design challenges, describes the derived research questions, and outlines the structure of this thesis. The research questions are addressed in four studies, which are described in **Chapters 2, 3, 4, and 5**. Specifically, **Chapter 2** focuses on the nascent design theory for data visualization tools integrating NLIs in the context of crisis response for end users. **Chapter 3** investigates the design of data visualization tools integrating NLIs in the context of co-located teams of end users. **Chapter 4** describes the design for the *ONYX* systems and the insights derived through its evaluation. **Chapter 5** investigates through *ContexIT* how to design NLIs in data visualization tools that enable

end users to teach the contextual pre-conditions of the end users' interpretation of an NL input. **Chapter 6** summarizes the overall findings of this thesis and discusses its theoretical and practical implications. Furthermore, it highlights the limitations of the four studies and outlines opportunities for future research. Finally, **Chapter 7** concludes this thesis.

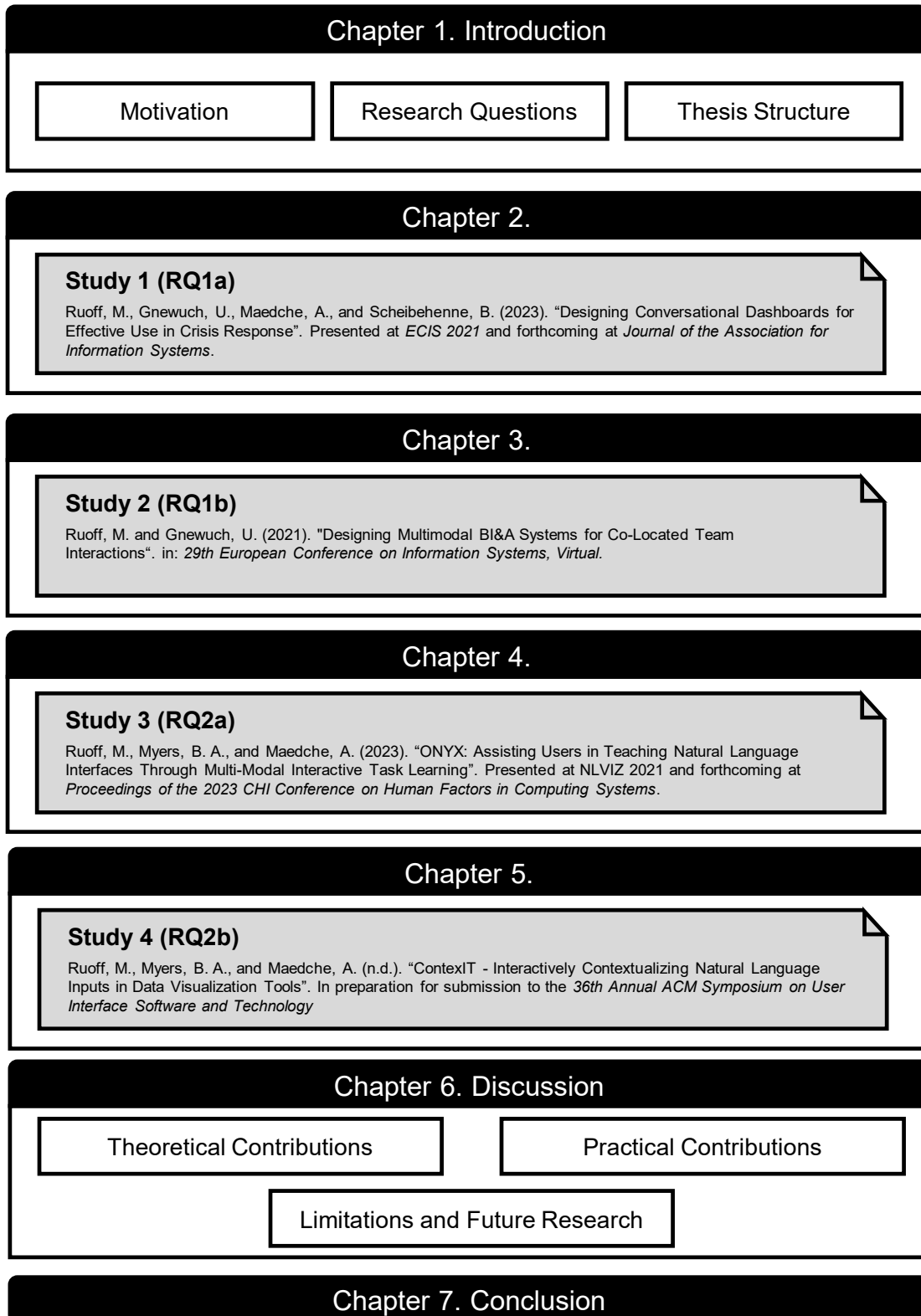


Figure 1.2: Structure of the Thesis.

## 2. Study 1: Designing Conversational Dashboards for Effective Use in Crisis Response <sup>2</sup>

### 2.1 Introduction

Crises are by nature unpredictable, sudden, and often chaotic situations. When a crisis occurs, people want to find accurate and up-to-date information quickly so that they can make the best decisions for themselves, their families, and their communities (C. Leong et al., 2015). To satisfy information needs, governments and health organizations increasingly rely on crisis response dashboards. Similar to business intelligence (BI) dashboards designed to support decision makers in organizations (Abbasi et al., 2016), crisis response dashboards are designed to provide citizens with key information about the current state of a crisis. As such, these data science artifacts primarily aim to democratize data science by making complex data accessible to the general public (Koch, 2021; Matheus et al., 2020). While crisis response dashboards had been developed earlier for earthquakes (Zook et al., 2010), wildfires (S. B. Liu & Palen, 2010), and virus outbreaks (Cheng et al., 2011), they took center stage during the COVID-19 pandemic (Pietz et al., 2020). For example, the dashboard provided by Johns Hopkins University received more than a billion hits per day during the height of the pandemic (Gardner et al., 2021). COVID-19 dashboards not only became a primary source of information about cases, deaths, and other key metrics for the general public, but were also frequently used to guide everyday decision making (e.g., about visiting a friend or getting a haircut) (Flowers, 2020). Both researchers and practitioners seem to share the assumption that COVID-19 dashboards were highly effective in helping billions of users to find the information they needed quickly, as long as the underlying data was accurate and the visualizations were interactive (Patino, 2021; Soper et al., 2021). However, while the reported numbers of daily dashboard users certainly look impressive, we know from the literature that people must use information systems (IS) effectively—rather than just using them—to achieve their goals (Burton-Jones & Grange, 2013). The fundamental dimension of effective use is transparent interaction, which describes how well users can access information from an IS unimpeded by its physical and surface structures (e.g., the user interface) (Burton-Jones & Grange, 2013). If users are unable to interact with a

---

<sup>2</sup>This chapter is based on the following studies which are published: Ruoff and Gnewuch (2021b) and Ruoff, Gnewuch, Maedche, and Scheibehenne (2022).



dashboard transparently, they are unlikely to find the information they need and make good decisions (e.g., about wearing a mask in regions with increasing case numbers). Against this backdrop, it is important to highlight that achieving transparent interaction with dashboards in general, and crisis response dashboards in particular, can be more difficult than expected, especially for users who are not familiar with the technology and/or have limited domain knowledge. These users often struggle to find their way around the dashboard’s interface, deal with its complexity, and obtain the information they are interested in (Young & Kitchin, 2020; Young, Kitchin, & Naji, 2021). Additionally, anecdotal evidence from a review of 52 state-level COVID-19 dashboards in the United States shows that many of them “were overly complex to navigate, and even experienced health researchers had difficulty finding key information” (Prevent Epidemics, 2020, p. 17). These findings suggest that users could face difficulties in interacting with a dashboard transparently, so that finding the information they need quickly might not be as easy as designers intend.

Given that crisis response dashboards, such as the ones developed for the COVID-19 pandemic, are designed to inform the general public, it is imperative that they enable a wide range of users—regardless of their socio-demographic backgrounds and technical expertise—to achieve transparent interaction. A promising way to address this challenge is to move beyond the traditional graphical user interface (GUI) and provide users with a more natural way of interacting with a dashboard using natural language. With recent technological advances in artificial intelligence (AI), natural language could make navigating the dashboard and finding information less difficult because it allows users to articulate their information needs more naturally, as they would in everyday conversation (Lee, Choe, et al., 2020). However, despite the technological advances, we know little about how to design a crisis response dashboard with natural language interaction capabilities (hereafter referred to as a conversational dashboard) and whether natural language actually enables users to interact with the dashboard more transparently. Therefore, we seek to answer the following research question:

*How can crisis response dashboards be extended with natural language interaction capabilities to improve users’ transparent interaction and access to crisis-related information?*

To address this question, we follow the design science research (DSR) approach (Hevner et al., 2004). Drawing on Burton-Jones and Grange (2013) theory of effective use (TEU), we propose a theory-driven design for conversational dashboards in crisis response and instantiate our proposed design in a novel data science artifact: a conversational dashboard

for the COVID-19 pandemic that enables natural language interaction in spoken or written form and helps users familiarize themselves with the use of natural language through conversational onboarding. The evaluation of our artifact shows that the ability to use natural language improves users' transparent interaction with the dashboard and ultimately increases their efficiency and effectiveness in finding the information they need. These findings suggest that our artifact contributes to the democratization of data science in the context of crisis response by making the information dashboards provide more accessible to broader audiences, thereby narrowing the gap between data and insights. Our work also contributes to research on dashboard design and use, both in general and in the specific context of crisis response, by providing prescriptive knowledge for extending dashboards with natural language interaction capabilities. In addition, our findings shed light on potential design trade-offs when users are provided with multiple ways to interact with a dashboard, and suggest an approach for addressing these trade-offs using conversational onboarding. With our findings, we provide actionable guidance to data scientists and dashboard providers on how to design crisis response dashboards that are more accessible to broader audiences.

## **2.2 Theoretical Foundations and Related Work**

Our work is situated at the intersection of two research streams: conversational user interfaces (CUIs) and dashboards. Here, we first provide an overview of related work in these streams from both an IS and a human-computer interaction (HCI) perspective. Next, we describe existing research at the intersection of CUIs and dashboards (i.e., on conversational dashboards), which has emerged as a prominent research area in the HCI field. Finally, we introduce our kernel theory (i.e., TEU) and explain its key constructs.

### **2.2.1 Conversational User Interfaces**

Conversational user interfaces (CUIs) enable people to interact with IS using spoken or written language in a natural way. The term conversational specifically emphasizes that these interfaces support the use of spontaneous natural language, in contrast to earlier applications (e.g., interactive voice response systems) that required a more restricted form of user input (e.g., "Press or Say 1 for English") (McTear et al., 2016). In recent years, CUIs in the form of chatbots and conversational agents have received considerable interest from IS researchers (Diederich, Brendel, Morana, et al., 2022). A key focus of this research has been to empirically investigate how the human-like design of CUIs influences user

perceptions and behaviors (e.g., Schanke et al., 2021; Seeger et al., 2021). Further, prior IS studies have focused on designing CUIs for specific contexts, such as for border screening (Nunamaker, Derrick, et al., 2011), in job interviews (Diederich, Brendel, & Kolbe, 2020), or in mental health care (Ahmad et al., 2022). Additionally, the HCI field has a long tradition of investigating CUI design, dating back to the 1960s when the first chatbot, ELIZA, was developed (Weizenbaum, 1966). A key focus in this research stream is to examine users' expectations of and interactions with CUIs in real-life settings in order to identify design challenges (e.g., Luger and Sellen, 2016; Porcheron et al., 2018). For example, Luger and Sellen (2016) found that users often do not understand the limitations of CUIs and therefore need to be given feedback about the actual capabilities. A related line of research seeks to address the challenges related to ambiguity and complexity in natural language interaction. For example, existing studies have suggested design principles for handling conversational breakdowns (Ashktorab et al., 2019) and for providing conversational context to help users interact with CUIs (Jain et al., 2018). Another, more technical set of studies in this stream focuses on the development of new system architectures and the application of advanced machine learning techniques to improve the technical components underlying a CUI (e.g., Huang et al., 2018; Xu et al., 2017). Finally, a growing number of studies investigate the design of CUIs for specific contexts (e.g., virtual team collaboration; Benke et al., 2020) and specific target groups (e.g., children; Z. Zhang et al., 2022). A great deal of research in both IS and HCI has regarded CUIs as an alternative to graphical user interfaces (GUIs). Several tech companies have even claimed that it is only a matter of time before CUIs replace apps and websites equipped with GUIs (McTear et al., 2016). However, it is difficult to convey the amount of visual information rich GUIs provide, as with data visualization in a dashboard, using natural language. This suggests that more could be achieved by complementing rather than replacing a GUI with a CUI. Against this backdrop, we next introduce related work on dashboards that typically feature rich GUIs, and subsequently present prior research on conversational dashboards that aim to combine both types of user interface.

### **2.2.2 Dashboards**

Dashboards are “visual displays of the most important information needed to achieve one or more objectives; consolidated and organized on a single screen so the information can be monitored at a glance” (Few, 2006). Many organizations use BI dashboards to provide decision makers with a comprehensive overview of key performance indicators, thereby supporting their decision making (Abbasi et al., 2016; H. Chen et al., 2012). Against this

backdrop, most IS studies focus on dashboards designed for domain experts in organizations. Examples include business users in areas such as supply chain management (Park et al., 2016) and health professionals such as physicians (L. Chen et al., 2016). While the specific contexts and dashboard designs may differ, these target users have in common that they are familiar with the application domain, which helps them understand the data underlying the dashboard, and that they are likely to use the dashboard on a regular basis as part of their job. In contrast, very little IS research has been devoted to dashboards designed for broader audiences outside of organizational structures who might be less familiar with dashboard technology. Thus, existing dashboard designs rarely include additional integrated learning features besides help buttons (e.g., Nadj et al., 2020; A. Nguyen et al., 2021) or tooltips (e.g., Vallurupalli and Bose, 2018), which would particularly benefit such audiences. Recker (2021) study is the only one that focuses on the general public as target users of a dashboard, and it is also one of the few IS studies that investigate dashboards in the context of crisis response. Overall, this dearth of research is surprising given the increasing pervasiveness of dashboards designed for broader audiences, particularly in the crisis response context (Koch, 2021; Matheus et al., 2020). Further, existing dashboards found in the IS literature almost exclusively rely on GUIs to display data visualizations, ranging from simple line charts (e.g., A. Nguyen et al., 2021) to more complex network graphs (e.g., Lu et al., 2021). These dashboards typically provide additional features, such as filters and drill-downs, to enable users to interact with visualizations and navigate the GUI. While GUIs are well suited to display complex data visualizations, research suggests that users who are not familiar with dashboards and have limited domain knowledge can struggle to interact with them (Young, Kitchin, & Naji, 2021). Therefore, other types of user interfaces (e.g., CUIs) might be more suitable for less tech-savvy audiences (Lee, Choe, et al., 2020). However, so far no IS study has investigated a dashboard with a CUI.

### **2.2.3 Conversational Dashboards**

In contrast to the IS literature that has focused on investigating dashboards equipped with traditional GUIs, HCI research has considered CUIs as a promising extension to make dashboards more accessible (Lee, Choe, et al., 2020). A key focus of this research is to provide and improve the technical foundations that enable natural language interaction with data visualizations in a conversational dashboard. For example, several studies address the challenges of ambiguity in natural language by proposing design features for disambiguating unclear user input (e.g., Gao et al., 2015; Setlur, Battersby, et al., 2016). In addition, an emerging body of work explores how users interact with conversational dashboards using

speech, touch, and keyboard (e.g., Saktheeswaran et al., 2020). However, similar to IS research, the majority of HCI studies have focused on dashboards designed for domain experts and tech-savvy groups of users (e.g., data analysts or computer science students; Gao et al., 2015; Setlur, Battersby, et al., 2016). The only study that specifically targets the general public is one that developed a smartphone app for exploring personal health data captured by a Fitbit tracker (Y.-H. Kim et al., 2021). Further, prior HCI research has predominantly focused on assessing the practical viability of conversational dashboards using relatively small samples (Srinivasan, Lee, et al., 2020) rather than conducting rigorous evaluations of the underlying design principles. For example, Setlur, Battersby, et al. (2016) compared their conversational dashboard to a traditional dashboard without CUI in a user study with twelve domain experts from a software company. Based on our review of the IS and HCI literature, we made three major observations about the current state of research on dashboard design. First, as the literature has primarily focused on dashboards designed for domain experts within organizational settings (e.g., managers, physicians) or for tech-savvy user groups (e.g., data analysts), research on the design of crisis response dashboards for broader audiences is scarce. This gap in the literature needs attention because previous studies indicate that novice and less tech-savvy users particularly can find interacting with a dashboard difficult (Young & Kitchin, 2020; Young, Kitchin, & Naji, 2021), suggesting that a different dashboard design is needed to accommodate broader audiences. Second, although HCI research identified CUIs as a promising way of making dashboards more accessible, existing designs of conversational dashboards have not been derived from a solid theoretical foundation and are often not rigorously evaluated to ensure their utility. As a result, a theory-driven design for conversational dashboards, particularly for the crisis response context, is lacking. Finally, while research on the development of advanced dashboard features (e.g., new visualizations, better analytical capabilities) has prospered, much less has been advanced on integrated learning features that would particularly benefit the average user who is less familiar with dashboards and how to use natural language to interact with them. This is another critical research gap since users of crisis response dashboards might not have received any dedicated training, and do not have an IT department for assistance. Our work addresses these gaps in the literature by proposing, instantiating, and rigorously evaluating a theory-driven design for conversational dashboards in crisis response that improves users' interaction and access to crisis-related information.

### 2.2.4 Theory of Effective Use

Drawing on representation theory, Burton-Jones and Grange (2013) proposed TEU based on the premise that rather than just being used, IS must be used effectively to obtain maximum benefits from them. They defined effective use as “using a system in a way that helps attain the goals for using the system” (p. 633) and conceptualized it as an aggregate construct with three hierarchical dimensions: (1) transparent interaction, (2) representational fidelity, and (3) informed action. This paper focuses on the first dimension of effective use—transparent interaction. According to Burton-Jones and Grange (2013), each lower-level dimension is necessary but not sufficient for the higher-level dimension. Therefore, when users are unable to interact with an IS transparently (transparent interaction), their chances of obtaining faithful representations (representational fidelity) and eventually acting upon these representations in an informed way (informed action) are dramatically reduced, if not eliminated. Transparent interaction is formally defined as “the extent to which a user is accessing the system’s representations unimpeded by its surface [e.g., user interface] and physical structure [e.g., computer, input/output devices]” (p. 642). For example, the surface structure of a traditional dashboard is a GUI, which typically consists of menus, sliders, and additional interactive features that can be used to navigate the GUI and change the data visualizations. TEU also identifies two major factors that act as drivers of effective use: adaptation and learning (Burton-Jones & Grange, 2013). Adaptations are users’ actions to improve the representations in a system or the way they can be accessed (e.g., through the surface structure). Learning involves users’ actions to learn the system’s components (e.g., representations, surface structure), the fidelity of its representations, and how to leverage representations toward taking more informed action. Given our emphasis on transparent interaction, we focus on two specific adaptation and learning actions that can increase users’ ability to interact with a system transparently, namely adapting surface structure and learning surface structure. Typically, users can engage in adapting a system’s surface structure by personalizing the user interface themselves or by suggesting improvements to system designers who then adapt the interface for them (Barki et al., 2007). In addition, organizations that introduce new IS usually offer training sessions and provide system manuals to facilitate users’ learning of the system’s surface structure (Lauterbach et al., 2020). However, in the context of crisis response dashboards, such strategies would be difficult to implement because these dashboards are often used in an ad-hoc manner and, unlike in an organization, there is no clearly defined group of users. Consequently, TEU as a kernel theory provides convincing theoretical arguments on why adaptation and learning

should improve transparent interaction, but it does not offer prescriptive guidance on what should be done through design to address users’ lack of transparent interaction, nor on how to achieve this. Therefore, design knowledge on how to adapt the surface structure of a crisis response dashboard and facilitate users’ learning to improve transparent interaction is scarce.

## 2.3 Designing Conversational Dashboards for Crisis Response

Our research project follows the DSR approach (Hevner et al., 2004) to design a conversational dashboard for crisis response that improves users’ transparent interaction and access to crisis-related information. The DSR approach is well-suited to guide our research as it aims to generate design knowledge through innovative solutions for real-world problems (Hevner et al., 2004). In this section, we first describe our design process and then elaborate on the design outcomes, that is, our meta-requirement (MRs), design principles (DPs), and software artifact.

### 2.3.1 Design Process

We adopted the DSR framework proposed by Kuechler and Vaishnavi (2008) and divided our project into two iterative build-evaluation cycles. Here, we briefly summarize our activities in each cycle. As illustrated in Figure 2.1, the work presented in this paper primarily focuses on the outcomes of the second and final design cycle.

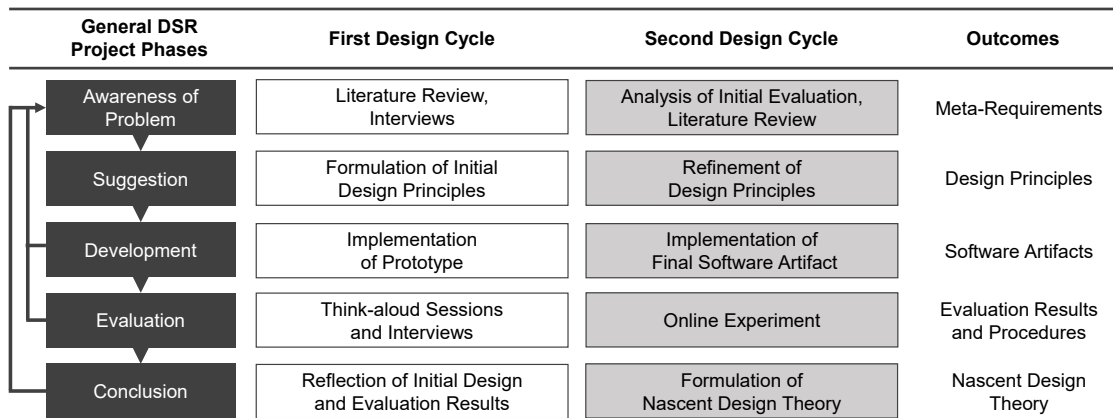


Figure 2.1: Overview of our DSR approach.

We started the first cycle by gaining an in-depth understanding of the problem space in order to identify barriers and design challenges that make it difficult for broader audiences to interact with crisis response dashboards. In this step, we first conducted a review of the IS and HCI literature on the design and use of dashboards in several application areas

including, but not limited to, crisis response. To supplement what we found in the literature, we conducted interviews with six actual and potential dashboard users (three females, three males) with an average age of 53.2 years ( $SD = 23.2$ ) and diverse backgrounds (e.g., seniors, students, professionals). Our goal was not to obtain as representative a sample as possible, but rather to invite less tech-savvy participants who do not use dashboards on a regular basis. In the interviews, we encouraged them to interact with the COVID-19 dashboard by Johns Hopkins University (Dong et al., 2020) and then asked them about the challenges they faced during the interaction. The findings from the interviews and the literature review revealed that transparent interaction with a dashboard is particularly important for effective use but achieving it can be more difficult than expected. Drawing on TEU (Burton-Jones & Grange, 2013) as our overarching kernel theory, we then derived two MRs. Subsequently, we proposed three initial DPs for conversational dashboards to address these MRs based on the idea that natural language interaction can help users achieve higher levels of transparent interaction with a dashboard. Finally, we instantiated our initial DPs in a first prototype that had natural language interaction capabilities but, in contrast to our final artifact, did not yet offer conversational onboarding. Instead, we implemented both a help button and a help message in the chat to provide instructions on how to interact with the dashboard. We evaluated our prototype with fifteen participants (7 females, 8 males) with an average age of 43.1 years ( $SD = 22.7$ ) and different levels of IT experience using think-aloud sessions combined with interviews. Overall, we found that all participants appreciated being able to use natural language to interact with the dashboard. Less tech-savvy participants reported that it allowed them to directly formulate their information needs in natural language and navigate the dashboard without dealing with interactive features such as sliders or filters. Conversely, more tech-savvy participants highlighted that using natural language improved their efficiency in the interaction and allowed faster access to the information in the dashboard. Nonetheless, most participants stated that they would prefer natural language as an addition to rather than a replacement of mouse interaction. Additionally, we found that most participants did not use or even recognize the help button or help message. Consequently, one of the key challenges participants mentioned was their lack of familiarity with and confidence in using natural language to interact with the conversational dashboard. This finding showed that our initial design was unable to provide sufficient support for users in learning how to interact with the dashboard, thus highlighting the need to refine our DPs to better address the MRs in the final artifact. Consequently, this reflection served as the entry point to the second cycle and eventually led to the development of the conversational onboarding. The second cycle started with a



refinement of the initial problem definition, MRs, and DPs. Based on the results of the first evaluation, we realized that users need a more systematic, “hands-on” approach to learn how to use natural language to interact with the dashboard. Therefore, we extended our review of dashboard studies and specifically analyzed the design of integrated learning features in current dashboards. Since the results showed that most dashboards rely on help buttons and tooltips (similar to our first prototype), we took inspiration from research on technology-mediated learning that has proposed the concept of enactive learning for enhanced learning outcomes (Gupta & Bostrom, 2009; Gupta, Bostrom, & Huber, 2010). Drawing on this concept, we then refined our third DP based on the idea of conversational onboarding. Subsequently, we developed a fully functional version of our artifact that instantiated the refined DPs. To rigorously evaluate the DPs, we conducted a large-scale online experiment with 271 participants and measured their level of transparent interaction. Finally, we abstracted and synthesized the design and evaluation results into a nascent design theory for conversational dashboards in crisis response.

### **2.3.2 Problem Description and Meta-Requirements**

Crisis response dashboards, such as the ones developed for the COVID-19 pandemic, are designed to give the general public access to important information during a crisis (Ivanković et al., 2021; Recker, 2021). However, our review of the dashboard literature in IS and HCI and our interviews with less tech-savvy individuals suggest that the average user is likely to have a hard time interacting with a crisis response dashboard and ultimately finding the information he or she needs. For example, one interviewee mentioned that she had to “search the dashboard extensively before even knowing how to get to the needed information”. Another interviewee explained that he “did not know what changed in the visualization based on [his] interaction”. Viewed through the lens of our kernel theory, there often appears to be a lack of transparent interaction with crisis response dashboards. To derive meta-requirements (MRs) for addressing this problem, we draw on TEU (Burton-Jones & Grange, 2013). As outlined in Section 2.2.4, TEU proposes two important factors that can improve transparent interaction: adaptation and learning. Given that the design problem we address relates to the difficulties in interacting with the user interface of crisis response dashboards, we specifically focused on TEU’s adaptation and learning actions related to surface structure. Drawing on these theoretical underpinnings, we derive two MRs on how the dashboards’ surface structure might be adapted and how learning it could be better supported. In line with TEU, adapting the dashboard’s surface structure (i.e., its user interface) is one approach to improving users’ transparent interaction and

ultimately their access to information. The surface structure of current crisis response dashboards consists of a GUI that can primarily be navigated using a mouse, keyboard, or touchscreen. Therefore, a promising way to adapt the surface structure is to move beyond the traditional GUI and provide users with a more natural way of interacting with the dashboard, for example, using natural language (Lee, Choe, et al., 2020). Natural language could simplify navigating the dashboard navigation and so make finding information less difficult by allowing users to formulate their information needs more naturally, as they would in everyday conversation. One interviewee hinted at this possibility in wondering “why [he] could not just ask the dashboard and talk to it”. Following this line of thought, we propose our first MR:

**MR1:** *The surface structure of a crisis response dashboard should be adapted to allow for a more natural way of interaction in order to improve transparent interaction.*

A second, complementary approach to improve users’ transparent interaction would be to support users in learning how to interact with the surface structure of a crisis response dashboard (Burton-Jones & Grange, 2013). In contrast to dashboard users in organizations (e.g., managers, health professionals), the average user of a crisis response dashboard would possibly not have received any dedicated training and not be able to call an IT department for assistance. Since current crisis response dashboards primarily offer integrated learning features in the form of passive help buttons and tooltips, a promising way to facilitate users’ learning of its surface structure would be to enable the dashboard to actively familiarize users with possible ways of interaction, particularly when it offers novel ways with which users might not be familiar (e.g., using natural language). Based on these considerations, we propose our second MR:

**MR2:** *A crisis response dashboard should actively support users in independently learning its surface structure in order to improve transparent interaction.*

### 2.3.3 Design Principles

To address the two identified MRs, we derive three DPs by building on existing theory and the current body of prescriptive knowledge for dashboards. Regarding our first MR, namely adapting the dashboard’s surface structure to enable a more natural way of interaction, we draw on the concept of affordances (Gibson, 1977), which is linked to TEU in several ways (Burton-Jones & Volkoff, 2017). Affordances are a key concept in the HCI and IS fields to describe and understand how users interact with an IS, thereby providing a solid theoretical grounding for our first and second DPs. Affordances are defined as action possibilities

that the environment provides to an actor (Gibson, 1977). According to Burton-Jones and Grange (2013), the surface structure of an IS relates particularly to physical affordances. Physical affordances are design features, such as buttons, that help users to perform a physical action in the user interface (Hartson, 2003). For example, dashboards offer interactive features, such as menus, sliders, and filters, that enable users to directly change the data visualizations. However, actualizing these physical affordances is difficult for some users, for example, because they do not know how and when to use the interactive features that enable navigating the dashboard. To address this challenge and offer users a more natural way of finding the information they need, we propose using natural language, which is the primary means of communication between humans (Knote et al., 2021). In contrast to clicking buttons, scrolling, and setting filters, natural language can provide a more natural way of performing actions in the interface and therefore “make affordances easy to actualize” (Knote et al., 2021, p. 434). It might also require less effort because users could directly use natural language input instead of translating their information need into a series of actions in the interface (e.g., setting filters). While the possibility of having a natural conversation with a dashboard might have seemed far-fetched in the past, recent technological advances, particularly in the area of large generative language models (e.g., GPT-3), suggest that in the future developers can make this scenario a reality with minimal manual effort or domain knowledge. Consequently, we propose enabling users to seamlessly navigate the dashboard using natural language. Thus, based on MR1, we formulate our first DP using the schema suggested by Gregor, Chandra Kruse, et al. (2020):

***DP1:** To enable the general public to seamlessly navigate a dashboard for crisis response, provide users with the ability to use spoken or written language in a natural way because articulating an information need in natural language is easier than translating it into a series of actions in the graphical user interface.*

While the first DP postulates that a crisis response dashboard should allow natural language interaction, it does not specify whether the ability to use natural language should complement or replace existing ways of interacting with a dashboard (e.g., using a mouse). At first glance, it could seem better to restrict users to natural language interaction, thereby removing the need for them to understand how and when to use interactive features, such as menus, sliders, and filters, to navigate the dashboard. However, according to TEU (Burton-Jones & Grange, 2013), transparent interaction involves not only the system itself (e.g., a dashboard) but also the user and task. This clarification is particularly important for crisis response dashboards because they need to accommodate a wide range of users,

ranging from novices who have never seen a dashboard before to tech-savvy groups of individuals (Ivanković et al., 2021). Therefore, different users could prefer different ways of interaction to achieve the same goal; then, restricting them to only natural language could backfire. Additionally, characteristics of the task at hand, such as its complexity, can also influence the suitability of using natural language or the mouse for a particular task. Considering this, we argue that users should be able to use both natural language and mouse, and need the freedom to choose between them in their interaction with the dashboard. Thus, based on MR1, we formulate our second DP:

*DP2: To enable the general public to seamlessly navigate a dashboard for crisis response, provide users with the ability to choose between natural language and mouse because it gives them flexibility for the task at hand and takes their individual preferences into account.*

Our second MR focuses on supporting users in independently learning the surface structure of a crisis response dashboard. To formulate our third DP based on MR2, we draw on the concept of enactive learning (Gupta & Bostrom, 2009). Enactive learning has proven to be a feasible approach for web-based training and therefore provides a good theoretical foundation for addressing MR2, particularly because formal training approaches are difficult, if not impossible, to implement in the context of crisis response dashboards designed for the general public. As enactive learning “involves learning from the consequences of one’s actions” (Gupta, Bostrom, & Huber, 2010, p. 16), it is an effective approach to onboard users to a new IS. Based on the idea of providing “a guided simulated environment with rich feedback to [enable users to] evaluate their actions” (Gupta, Bostrom, & Huber, 2010, p. 18), we propose integrating conversational onboarding that allows users to familiarize themselves with using natural language to interact with the crisis response dashboard. Given the relative novelty of natural language interaction, particularly in the context of crisis response dashboards, conversational onboarding should provide users with the opportunity to try out interacting with the dashboard using natural language in a step-by-step manner, observe the consequences of their actions (e.g., how and why data visualizations change), and receive feedback when something goes wrong. Then, before actually using the dashboard to find the information they are looking for, users can learn how to use natural language to navigate the dashboard. Taken together, based on MR2, we formulate our third DP as follows:

*DP3: To enable the general public to seamlessly navigate a conversational dashboard for crisis response, provide users with conversational onboarding that takes them step-by-step through the natural language interaction with the dashboard because this helps users*

*familiarize themselves with how to interact with the dashboard using spoken or written language.*

### 2.3.4 Testable Design Propositions

Testable design propositions are a core component of a design theory (Gregor & Jones, 2007). Through the lens of our kernel theory, we therefore derived two testable propositions from the presented DPs. The primary outcome of interest and core construct from TEU is transparent interaction, which can be understood as the extent to which users can access information from an IS unimpeded by its user interface (Burton-Jones & Grange, 2013). As noted earlier, users can struggle to navigate the rich GUI of current crisis response dashboards to the information they need, due to difficulties with dashboards' sliders, filters, and other interactive features. Considering these challenges, we argue that users can interact with a crisis response dashboard more transparently they have the ability to choose spoken or written language in their navigation of the dashboard. Instead of users having to translate an information need into a series of actions in the GUI (e.g., button clicks), which requires knowing and being able to use its features, formulating it in natural language is much easier (Lee, Choe, et al., 2020). Consequently, users should be able to achieve higher levels of transparent interaction with a conversational dashboard built according to our DP1 and DP2. Hence, we propose:

***Proposition 1:*** *A crisis response dashboard equipped with a conversational user interface allowing users to interact with the dashboard using natural language enables them to achieve higher levels of transparent interaction.*

TEU also posits that learning how to interact with the user interface of an IS can improve transparent interaction. As described earlier, current crisis response dashboards offer little help beyond a few tooltips and help buttons in teaching users how to navigate the interface and access information. Moreover, natural language is a rather new form of interaction with a dashboard that users might still need to learn. Therefore, we argue that providing users with conversational onboarding that can walk them through the natural language interaction with the dashboard (DP3) should facilitate their learning by helping users familiarize themselves with using spoken or written language in navigating the dashboard. Consequently, users should be able to achieve higher levels of transparent interaction if the conversational dashboard offers conversational onboarding built according to DP3. Hence, we propose:

***Proposition 2:*** *A conversational crisis response dashboard equipped with conversational*

onboarding walking users through the natural language interaction with the dashboard enables them to achieve higher levels of transparent interaction.

### 2.3.5 Artifact Description

To instantiate our DPs in an artifact, we developed a system architecture and implemented four key components. To ensure replicability and provide practitioners with actionable guidance on how to translate our DPs into appropriate features (Lukyanenko et al., 2020), we leveraged existing open-source frameworks and libraries rather than developing components from scratch. Next, we present a detailed description of the overall system architecture (see Figure 2.2), its four key components, and its conversational onboarding.

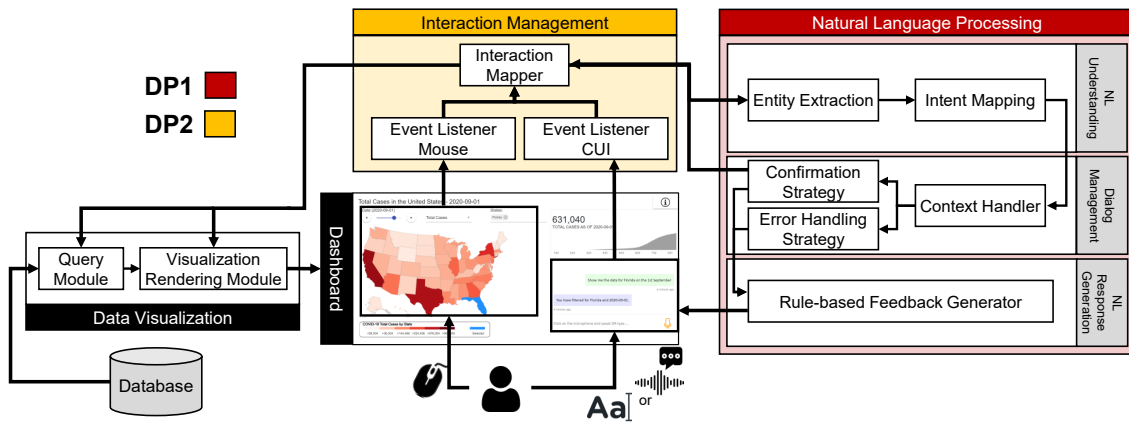


Figure 2.2: System Architecture of the Conversational Dashboard.

#### 2.3.5.1 Dashboard and Data Visualization Component

The core component of our artifact is the conversational dashboard itself, which provides information about the COVID-19 pandemic through several data visualizations (e.g., charts, KPIs, maps) and offers users two ways of interacting with these visualizations: using natural language and a mouse. We identified common interaction types in current crisis response dashboards and decided to provide users with the ability to filter the data displayed in a visualization, to roll-up (abstract), and to drill-down (elaborate) the data on the state level. We used D3.js to create interactive data visualizations (Bostock et al., 2011) based on publicly available data from Johns Hopkins University’s COVID-19 data repository (Dong et al., 2020). To change visualizations in real-time, the data visualization component retrieves the required data from the database through a query module and provides it to the visualization rendering module, which then updates the data visualizations in the dashboard.

### 2.3.5.2 Interaction Management Component

The interaction management component is responsible for managing the communication between the event listeners that capture user interactions (e.g., button clicks, natural language input) and the corresponding functionality of the dashboard. For example, when a user selects a state in the drop-down menu, the event listener captures the interaction type (i.e., filtering) and the selected state so that the interaction management component can decide what dashboard functionality to invoke. In line with DP1, we connected this component to the NLP component that provides users with the ability to use spoken or written language. While natural language input in written form is directly sent to the NLP component for further processing, spoken user input is first translated into written text by the speech-to-text feature provided by Microsoft Cognitive Services. After the user input has been processed successfully, the results are returned to the interaction management component, which then adjusts the data visualizations accordingly. The mapping between the results provided by the NLP component and the dashboard functionality is implemented as a rule-based approach due to its finite nature. In line with DP2, the interaction mapper allows users to choose and switch between natural language and mouse interaction at any point in time depending on their preferences. As Figure 2.3 illustrates, users can set a filter for Florida, for example, either using natural language (e.g., “Show me the data for Florida”) or by selecting Florida in the drop-down menu using their mouse.

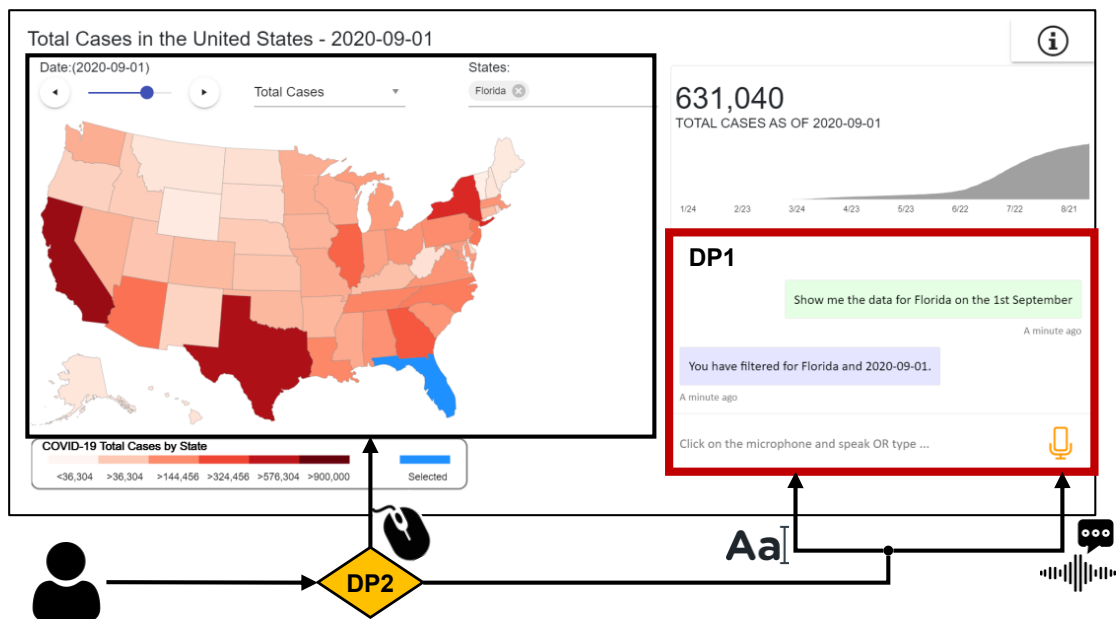


Figure 2.3: Screenshot of the Conversational Dashboard with DP1 and DP2.

### 2.3.5.3 Natural Language Processing (NLP) Component

To develop the NLP component, we used Microsoft’s Bot Framework (Microsoft, 2021), a comprehensive open-source framework for building conversational AI systems, which enables developers to create and manage conversation flows. In the following, we explain our implementation along the three subcomponents of (1) natural language understanding, (2) dialog management, and (3) natural language response generation (McTear et al., 2016).

**Natural Language Understanding (NLU).** In contrast to traditional mouse interaction where a click directly triggers an action in the dashboard, a user’s natural language input (e.g., “Show me the data for Florida”) first needs to be analyzed to identify the user’s goal (e.g., filtering for Florida). For the development of the NLU subcomponent, we used Microsoft’s Language Understanding and Intelligent Service (LUIS). LUIS enables developers to create and train custom, purpose-specific language models by leveraging pre-existing and pre-built language models (Microsoft, 2017). Using LUIS, we created a custom language model to extract relevant entities (e.g., dates, state names, metrics) and to recognize the users’ intent (e.g., filter, drill-down, roll-up) from their spoken or written input. To create and train our language model, we performed two steps: First, since the language model had to be capable of extracting relevant entities, we derived an entity hierarchy with state names, dates, and metrics as entities together with their possible values from our database (e.g., all state names for the entity “state”). Subsequently, we integrated the entity hierarchy into LUIS to perform the entity extraction task through keyword matching (i.e., for state names, metrics) and prebuilt entities provided by LUIS (i.e., for dates). Second, the language model had to contain intents for each possible interaction type in the dashboard (i.e., filter, drill-down, roll-up), which can be mapped to the users’ natural language input. Thus, we created three intents with a set of training examples and identified entities that had to be included in a user input together with each intent. Since user input might not map to any of the possible interaction types, we also created the fallback intent for unspecific input such as “Hey” or “What can I do?”. Table 2.1 provides an overview of intents, entities, and examples. Finally, we refined our language model using training data collected from 27 Amazon Mechanical Turk workers who were asked to provide different formulations for each possible interaction type in the dashboard. The final model included 23 unique training examples for the filter intent, 17 for drill-down, and 9 for roll-up.

**Dialog Management.** The dialog management subcomponent maintains the dialog state, tracks the state of the dashboard, and generates a system action based on the previously extracted intent and entities. Using Microsoft’s Bot Builder SDK for .NET V4 (Microsoft,



Table 2.1: Intents, Entities, and Examples in the Language Model

Intent	Example	Required Entities	Possible Entities
Filter	“Show me the data for <b>Florida</b> on the <b>1st of September</b> ”, “Show me <b>deaths</b> ”	At least one possible entity	Metric; Date; State
Drill-Down	“Go to <b>Texas</b> ”	State	Metric; Date
Roll-up	“Go to overview”	-	-
Fallback	“Hey”, “Blue”, “What can I do?”		

2021), we implemented the following three key features: context handler, confirmation strategy, and error handling strategy. The context handler is primarily responsible for determining whether an action can be carried out in the dashboard based on its current state. For this, the context handler uses a rule-based approach to first check whether the entities extracted from the user’s input satisfy the requirements of the recognized intent (see Table 2.1). Additionally, it continuously tracks the dialog and dashboard state at runtime in a local storage object. Based on the dashboard’s current state, the context handler then checks whether the action type mapped to the intent is valid or whether constraints apply. If the context handler deems an action to be valid, it invokes the confirmation strategy, updates its current state, and forwards the recognized intent with the extracted entities to the interaction management component. However, if the context handler deems an action to be invalid, for example when user input with zero entities is mapped to the filter or drill-down intent or if the fallback is triggered, it invokes the error handling strategy to inform the user that their desired action could not be performed in the dashboard.

**Natural Language Response Generation.** Regardless of whether the confirmation or error handling strategy is invoked, the dashboard responds to users after they have provided input, giving explicit feedback about what actions were performed. Thus, the natural language response generation subcomponent, which is a crucial component of any CUI (McTear et al., 2016), enables turn-by-turn conversations between the dashboard and its users, consistent with our objective of designing a conversational dashboard. Its key feature is a rule-based feedback generator that uses pre-defined response templates (see Table 2.2) to provide informative feedback when the confirmation strategy is invoked and suggestive feedback when the error handling strategy is invoked or the fallback intent is triggered.

#### 2.3.5.4 Conversational Onboarding

To instantiate DP3, we implemented step-by-step conversational onboarding through which users can familiarize themselves with using natural language to interact with the dashboard (see Figure 2.4). When users access the conversational dashboard for the first time, they

Table 2.2: Natural Language Response Templates

Strategy	Intent	Response Table	Example
Con- firmation Strategy	Filter	You have filtered for <b>State</b> , <b>Metric</b> , and <b>Date</b> .	You have filtered for Idaho, Cases, and 2020-05-15.
	Drill-down	You have selected <b>State</b> {for <b>Date</b> and <b>Metric</b> }.	You have selected Florida for 2020-08-29.
	Roll-up	You are back at the overview.	-
Error Handling Strategy	Fallback	You can use the following commands to interact with the dashboard: <ul style="list-style-type: none"> <li>• <b>Filter</b>: “Show me Florida for June 1st.”</li> <li>• <b>Zoom In</b>: “Go to New York”</li> <li>• <b>Back to Overview</b>: “Go to overview”</li> </ul>	-

are asked to complete the onboarding before they can start interacting with the dashboard. Following the suggestions of Gupta and Bostrom (2009), we implemented the following features in our conversational onboarding that correspond to high levels of the enactive learning dimensions (e.g., structuredness and restrictiveness of practice, feedback). To help users practice the essential skills for interacting with the dashboard using natural language, we focused their practice on how to formulate natural language input for the core dashboard functionalities such as filtering, drill-down, and roll-up. Further, we restricted the practice flow to a predefined sequence so that at first users are introduced to the basic actions with exemplary input, and then gradually learn more complex actions that combine several basic ones. After each demonstration of an action in the dashboard, users are prompted to immediately reproduce it in order to minimize the lag between the demonstration and users’ practice. Finally, users receive immediate feedback on their natural language input. For example, if relevant entities were missing in their natural language input, users are informed that not all entities were included in order to reproduce the action in the dashboard.

## 2.4 Evaluation

We performed two evaluations of our artifact. First, we conducted a performance evaluation of its key technical component to assess whether it enables effective natural language interaction in spoken and written form. Second, we carried out an experimental study to test whether our proposed design can improve users’ transparent interaction with a crisis response dashboard.

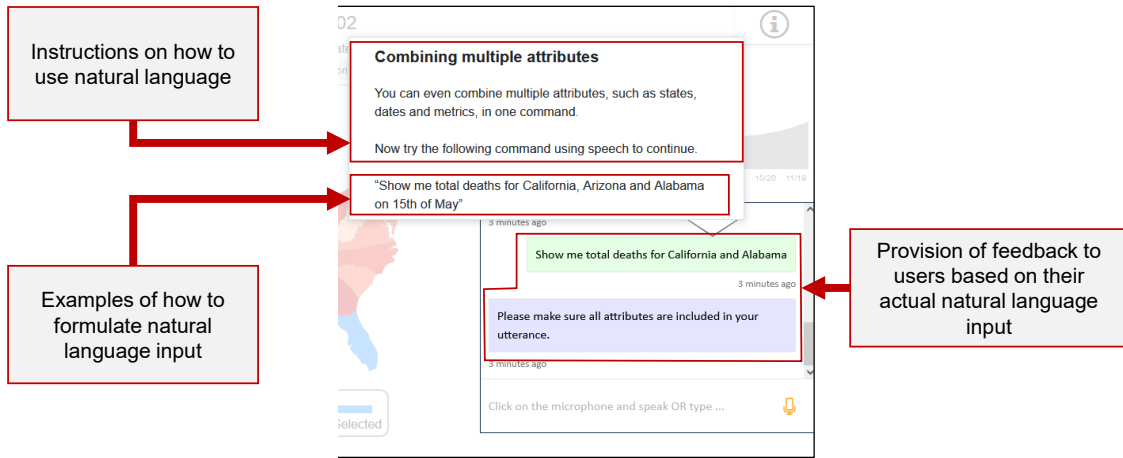


Figure 2.4: Screenshot of the Conversational Onboarding (DP3).

### 2.4.1 Performance Evaluation of the Natural Language Processing (NLP) Component

At the heart of our artifact is an NLP component that allows the conversational dashboard to understand and act on the user’s natural language input. To assess the NLP component’s quality, we conducted a performance evaluation that specifically focused on speech-to-text translation, entity extraction, and intent mapping. For the evaluation, we used the dataset of 3119 natural language inputs collected in our user evaluation (see Section 2.4.2 for details). Initially, this dataset only included natural language input used to navigate the dashboard (e.g., “Show me Idaho August 31st”) and the corresponding results provided by the NLP component (e.g., intent = “filter”, entities = “Idaho” and “August 31”). Since no ground truth was available in the dataset for the evaluation, we recruited 264 crowd workers on Amazon Mechanical Turk to obtain ground truth labels for each natural language input. Additionally, we instructed workers to highlight if they recognized an input as syntactically incorrect or if there were misunderstandings (e.g., “soon out” instead of “zoom out”). Each input was labeled by two workers who had a moderate level of agreement (Cohen’s Kappa = .52). To break ties in cases of disagreement, a research assistant who received the same instructions and explanations reviewed each input with a disagreement between the workers and assigned a final label. The final dataset included 3119 natural language inputs, results of the NLP component, and human ground-truth labels for speech-to-text, entities, and intents.

### 2.4.1.1 Performance Measures

We used established measures to evaluate the performance of the NLP component. First, to verify the speech-to-text translation quality for all spoken input, we used the binary label that specified whether a particular input was syntactically correct. Based on our labeled dataset, we calculated the accuracy of speech-to-text translation as the ratio of correctly translated spoken inputs to the total number of spoken inputs. Second, to evaluate the entity extraction and intent mapping performance, we compared the results provided by our NLP component against the intent and entity labels human workers provided. We used standard classification measures that have been used in similar work (e.g., Siering et al., 2021)—that is, precision, recall, and F1-score—and calculated them through micro-averaging the classes (i.e., intent or entity). Precision measures the percentage of correctly classified instances (i.e., intents and entities) to the total number of instances for that class of instances retrieved by the intent mapping or entity extraction ( $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$ ). Recall measures the percentage of correctly classified instances among all true positive cases of that class of instances ( $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$ ). The F1-score is calculated as the weighted average of precision and recall ( $\text{F1} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$ ). Further, to benchmark our intent mapping against related systems (Srinivasan & Stasko, 2018), we additionally calculated the accuracy as the number of correctly classified inputs divided by the total number of inputs ( $\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FP} + \text{FN} + \text{TN})$ ).

### 2.4.1.2 Results

**Speech-to-Text.** Out of the total 3119 natural language inputs, 2499 (80.1%) were performed using spoken language. For these spoken inputs, we calculated an overall speech-to-text accuracy of 90.2%. This means that 9.8% of the spoken input included syntactical errors that could have negatively affected the subsequent entity extraction and intent mapping steps. For example, the word “cases” was incorrectly translated to “kisses” several times, resulting in user input that missed the metric “cases”. However, the overall accuracy of more than 90% indicates that our speech-to-text translation was able to achieve a good performance. **Entity Extraction.** For the analysis of the entity extraction performance, we used all 3119 spoken and written inputs made by users, including those labeled as syntactically incorrect. Based on the F1-scores shown in Table 2.3, entity extraction worked the best for dates (95%) and states (94%). In contrast, entity extraction yielded a lower F1-score of 83% for the entity “metric”. One reason for the lower performance of this entity is that the phrase “people had died” was used to describe “deaths”, which was not included in the training data for this entity. Overall, the entity extraction step achieved an F1-score

Table 2.3: Performance Evaluation Results for Entity Extraction and Intent Mapping

		<b>Precision</b>	<b>Recall</b>	<b>F1</b>	<b>N</b>
Entities	State	96%	92%	94%	2,251
	Date	96%	95%	95%	1,579
	Metric	80%	87%	83%	1,176
	<b>Overall</b>	<b>92%</b>	<b>92%</b>	<b>92%</b>	
<hr/>					
Intents	Filter	89%	92%	90%	2,437
	Drill-down	67%	49%	57%	472
	Roll-up	61%	54%	57%	112
	Fallback	20%	29%	24%	98
	<b>Overall</b>	<b>82%</b>	<b>82%</b>	<b>82%</b>	<b>3,119</b>

of 92%. Moreover, the results show that our entity extraction performed equally well for spoken (91%) and written input (93%). Table 2.3 provides the overall and entity-level precision and recall measures.

**Intent Mapping.** The final step of the analysis was to evaluate the intent mapping performance. Again, we used all 3119 natural language inputs, including those labeled as syntactically incorrect. Only 98 inputs (3.1%) were labeled as fallback (i.e., not supported input). Overall, our intent mapping achieved a high accuracy of 82%, demonstrating comparable or better performance than related systems that offer natural language interaction with data visualizations (e.g., Srinivasan and Stasko, 2018). Additionally, the results show that our intent mapping performed equally well for spoken (83%) and written input (79%). The slight differences can be partially explained by spelling mistakes, such as “Select ketncuky”, which mainly occurred in written input. In sum, our intent mapping achieved a high overall F1-score of 82% on the unbalanced dataset. While our main intents performed well, the fallback intent achieved a lower precision level (20%) because it included user input with errors from incorrect speech-to-text translation. Further, the recall for the fallback intent was only 29% since the intent mapping learned in the training phase that the phrase “Show me...” is strongly associated with the filter intent. Therefore, it also recognized inputs, such as “Show me this”, as a filter intent and not as a fallback, which our artifact consequently needed to deal with since a target entity was missing. Table 2.3 presents all overall and intent-level results.

**Response Time.** Finally, to evaluate the NLP component’s performance in terms of speed, we analyzed the overall response time for valid inputs starting from the time a user provided spoken or written input and ending with the NLP component sending the results back to the interaction management component. The results of this analysis show that it took the NLP component only 0.9 seconds on average to fully process natural language input

and update the visualizations in the dashboard accordingly. Taken together, the results of our performance evaluation based on a dataset of 3119 manually labeled natural language inputs show that the NLP component performed well in terms of accuracy and speed, suggesting that it provides a robust technical basis to enable natural language interaction with our conversational dashboard in spoken or written form. More specifically, the NLP component achieved a satisfactory performance on all tasks (i.e., speech-to-text translation, entity extraction, and intent mapping), indicating that it was able to effectively understand what users were looking for in the dashboard and to feed this information back to the other components of our conversational dashboard.

### 2.4.2 User Evaluation

To evaluate whether our proposed design can improve users' transparent interaction with a crisis response dashboard, we conducted a large-scale online experiment. Following the approach of Morana et al. (2019), we developed six versions of our artifact with different combinations of instantiated DPs to examine their effect on transparent interaction. More specifically, we compared a traditional dashboard (TDB) with two types of conversational dashboards: natural language-only (CDB-NLO) and natural language-enhanced (CDB-NLE). As Table 2.4 shows, the CDB-NLO instantiated only DP1, whereas the CDB-NLE instantiated both DP1 and DP2. TDB did not instantiate these DPs to establish a baseline condition representing the current design of crisis response dashboards. Further, we developed two different versions of each dashboard with and without conversational and/or traditional onboarding (DP3) depending on the respective dashboard type, resulting in a total of six different dashboards.

Against the backdrop of these different artifact instantiations, we translated our previously derived design propositions (see Section 2.3.4) into four specific hypotheses that we empirically tested in the experiment. According to our first proposition, a crisis response dashboard equipped with a CUI should improve transparent interaction because it allows users to interact with the dashboard using spoken or written language in a natural way. Based on this proposition, we argue that users will achieve higher levels of transparent interaction with a conversational dashboard than with a traditional dashboard, regardless of whether natural language interaction replaces existing ways of interacting with a dashboard using the mouse (CDB-NLO) or whether it complements them (CDB-NLE). Hence, we hypothesize:

*Users who interact with a natural language-only conversational dashboard (CDB-NLO; **H1**)*

Table 2.4: Overview of Dashboard Types and Instantiated Design Principles.

	Dashboard Type	Design Principles Instantiated*	Description
<b>Traditional Dashboard</b>	Traditional dashboard (TDB)	-	Participants were restricted to interact with the dashboard using a mouse.
<b>Conversational Dashboard</b>	Natural language-only conversational dashboard (CDB-NLO)	DP1	Participants were restricted to interact with the dashboard using natural language.
	Natural language-enhanced conversational dashboard (CDB-NLE)	DP1 DP2	Participants were able to interact with the dashboard using both natural language and mouse.
Note. *For each dashboard type, we developed two versions: one with onboarding (DP3 instantiated) and one without (DP3 not instantiated), resulting in six different dashboards used in the experiment.			

or a natural language-enhanced conversational dashboard (CDB-NLE; **H2**) achieve higher levels of transparent interaction than those interacting with a traditional dashboard (TDB).

According to our second proposition, a conversational crisis response dashboard equipped with conversational onboarding should improve transparent interaction because it facilitates users' learning by walking them through the natural language interaction with the dashboard. Therefore, based on our second proposition, we argue that users of conversational dashboards, regardless of whether natural language interaction replaces existing ways of interacting with a dashboard using the mouse (CDB-NLO) or whether it complements them (CDB-NLE), will particularly benefit from completing the conversational onboarding before interacting with the dashboard. Hence, we hypothesize:

*Users who complete the conversational onboarding of a natural language-only conversational dashboard (CDB-NLO; **H3**) or a natural language-enhanced conversational dashboard (CDB-NLE; **H4**) achieve higher levels of transparent interaction than those who do not.*

Finally, we draw on TEU to formulate two additional hypotheses on the effects of transparent interaction on efficiency and effectiveness. TEU proposes that transparent interaction increases users' efficiency by saving them time when they navigate the system and improves their effectiveness by helping them stay focused on the task rather than getting distracted by the difficulties of finding their way around the system's interface (Burton-Jones & Grange, 2013). Based on this reasoning, we argue that higher levels of transparent interaction with

a crisis response dashboard will increase users' efficiency and effectiveness in finding the information they need. Users who navigate the dashboard more quickly are able to access information in less time. In addition, they are more effective because they make fewer mistakes in their interaction and thus are less likely to give up on a task or end up with incorrect information. Hence, we hypothesize:

*Users' transparent interaction with a crisis response dashboard increases their efficiency (H5) and effectiveness (H6) in finding the information they need.*

#### 2.4.2.1 Method

To test our hypotheses, we conducted an online experiment in which participants interacted with one of the six dashboards to perform four information finding tasks. The experiment used a 3 (dashboard type: TDB vs. CDB-NLO vs. CDB-NLE) x 2 (onboarding: absent vs. present) between-subjects design, resulting in six experimental conditions.

**Experimental Procedure.** Participants accessed the experiment via a link provided on Amazon Mechanical Turk (MTurk). After reading a short description and providing informed consent, participants were randomly assigned to one of the six experimental conditions. In the first step, participants were instructed to test their microphones to ensure that they would be able to use natural language in spoken form during the experiment and that there was only minimal background noise. Only if the system was able to understand them correctly, could they continue with the experiment. Next, participants watched a 50-second video that provided an overview of the dashboard and its COVID-19 data visualizations. After watching the video, participants in the three conditions without onboarding immediately entered the main part of the experiment. In contrast, participants in the other three conditions first completed the onboarding of their dashboard. The onboarding was designed to match the specific experimental condition so that participants only familiarized themselves with the ways of interaction that they would be able to use later. For example, the onboarding in the TDB condition did not include an introduction to natural language interaction and resembled an interactive guided tour through the GUI. In the main part of the experiment, participants were instructed to perform four different information finding tasks using the dashboard. The task order was randomized and the dashboard was reset after each task. The tasks were designed to represent realistic information needs based on our discussions with actual and potential dashboard users. For a fair comparison between different dashboard types, we designed the tasks in such a way that participants could not simply "copy and paste" the task description into the chat



window and solve the task; rather, they needed to reframe it and/or break it down into multiple steps. For each task, participants could enter their solution in an input field below the dashboard or skip the task if they were not able to come up with a solution. Finally, after completing the main part of the experiment, participants filled out a survey in which they could provide feedback and report on technical problems. On average, the experiment took 25 minutes to complete.

**Participants.** We recruited 292 participants via MTurk. Researchers increasingly use MTurk because the participant pool is more diverse than typical university participant pools (Buhrmester et al., 2011), which supported our objective of reaching a wide range of users from different backgrounds. We excluded 21 participants who failed an attention check question, leaving 271 participants for analysis (45–46 participants per condition). Of these participants, 121 were female (44.6%) and 150 were male (55.4%). The mean age was 38.33 years (SD = 11.1). Participants were paid \$4.50 for their participation. Further, they could earn a bonus payment of \$0.20 for each correctly solved task and an additional bonus of \$0.20 if they were among the 20% fastest participants for this specific task. Therefore, the maximum payment was \$6.10 ( $\$4.50 + 4 \times \$0.20 + 4 \times \$0.20$ ).

**Variables and Operationalization.** Transparent interaction can be assessed using self-reported measures and behavioral measures (Burton-Jones & Grange, 2013). Since self-reported measures can be subject to a range of biases and demand effects (Dimoka et al., 2011), we used a behavioral measure of transparent interaction. Following the suggestions of Burton-Jones and Grange (2013), we operationalized transparent interaction based on “the extent to which a user’s navigation path [...] approaches the quickest path that can be taken” (p. 655). For each task in the experiment, we identified the quickest path by determining the minimum number of steps (e.g., button clicks, natural language inputs) required to navigate the dashboard to access the information needed to complete a particular task. Since this number depends on which ways of interaction a dashboard offers its users (e.g., natural language and/or mouse), we calculated separate values for each dashboard type. For each participant, we then calculated the level of transparent interaction as the average ratio of the minimum number of navigation steps required for accessing the needed information to the number of navigation steps a participant actually took to correctly solve a task. In contrast to transparent interaction (a dimension of effective use), effectiveness and efficiency are dimensions of (task) performance. Effectiveness, which is defined as the “extent to which a user has attained the goals of the task for which the system was used” (p. 654), was operationalized as the number of correctly solved tasks. Efficiency, which

is defined as “the extent of goal attainment for a given level of input (such as effort or time)” (p. 654), was calculated as the average time needed to complete all tasks that were correctly solved. Thus, effectiveness and efficiency correspond to users’ higher-level goal of accessing a particular piece of information to answer a specific question (the desired end), while transparent interaction relates to users’ lower-level goal of navigating the dashboard in a transparent way (the means) (cf. Burton-Jones and Grange, 2013, p. 641). Finally, we examined users’ demographics (i.e., age, gender, education) and prior experience with computers, dashboards, and natural language interaction as control variables.

#### 2.4.2.2 Results

**Manipulation and Randomization Checks.** We conducted two manipulation checks to ensure that participants used the different versions of the dashboard as intended. First, we asked participants to identify how they were able to interact with the dashboard (i.e., with a mouse, spoken and written language) and found that 98 percent of the participants correctly identified their condition, which indicates that the dashboard type manipulation was successful. Second, to examine whether the onboarding successfully manipulated users’ perceived ability to navigate the dashboard, we asked participants in the respective conditions before and after the onboarding to indicate their level of self-efficacy in using the dashboard on a 7-point Likert scale (Hsieh et al., 2008). The results of a paired-samples t-test show that participants’ self-efficacy was significantly higher after completing the onboarding ( $M = 6.40$ ,  $SD = 0.94$ ) than before ( $M = 6.13$ ,  $SD = 1.02$ ;  $t(134) = 3.58$ ,  $p < .001$ ). Moreover, participants in the conditions with onboarding rated their self-efficacy significantly higher after familiarizing themselves ( $M = 6.40$ ,  $SD = 0.94$ ) compared to participants in conditions without it ( $M = 5.90$ ,  $SD = 1.04$ ;  $t(266.76) = 4.15$ ,  $p < .001$ ). Taken together, these results suggest that the onboarding also successfully manipulated users’ perceived ability to interact with the dashboard. Finally, we assessed the efficacy of our randomization procedure by comparing the six experimental conditions on several control variables. There were no significant differences in age ( $F(5, 265) = 0.77$ ,  $p = .57$ ), gender ( $\chi^2(10) = 7.38$ ,  $p = .69$ ), education ( $\chi^2(20) = 18.6$ ,  $p = .55$ ), prior experience with computers ( $F(5, 265) = 0.61$ ,  $p = .69$ ), prior experience with dashboards ( $\chi^2(20) = 15.2$ ,  $p = .77$ ), and prior experience with natural language interaction ( $\chi^2(20) = 10.2$ ,  $p = .96$ ). This suggests that the randomization in our experiment was also successful.

**Hypothesis Testing.** The descriptive statistics for transparent interaction across the experimental conditions are shown in Table 2.5. To test our hypotheses on the effects of dashboard type and onboarding on users’ transparent interaction with the dashboard

Table 2.5: Descriptive Statistics for Transparent Interaction.

		<b>Onboarding</b>	
		Absent	Present
<b>Dashboard Type</b>	TDP	0.30 (0.14)	0.33 (0.15)
	CDB-NLO	0.57 (0.26)	0.63 (0.24)
	CDB-NLE	0.31 (0.19)	0.42 (0.25)
Note. Means with standard deviations in parentheses.			

(H1-H4), we conducted a two-way ANOVA. The results show significant effects of both dashboard type ( $F(2, 265) = 48.3, p < .001$ ) and onboarding on transparent interaction ( $F(1, 265) = 7.38, p = .007$ ). The interaction effect was not significant ( $F(2, 265) = 0.97, p = .38$ ). Subsequently, we used planned contrasts to test our hypotheses. First, consistent with H1, the results show that participants in the CDB-NLO condition ( $M = 0.60, SD = 0.25$ ) achieved a significantly higher level of transparent interaction than participants in the TDB condition ( $M = 0.31, SD = 0.14; t(265) = 9.2, p < .001$ ; H1 supported). However, we find no significant difference in transparent interaction between participants in the CDB-NLE ( $M = 0.36, SD = 0.23$ ) and TDB condition ( $M = 0.31, SD = 0.14; t(265) = 1.64, p = .10$ ; H2 rejected). Further, in the CDB-NLO condition, transparent interaction shows no significant difference between participants who completed the onboarding ( $M = 0.63, SD = 0.24$ ) and those who did not ( $M = 0.57, SD = 0.26; t(265) = 1.31, p = .18$ ; H3 rejected). In contrast, in the CDB-NLE condition, participants who completed the onboarding ( $M = 0.42, SD = 0.25$ ) achieved a significantly higher level of transparent interaction than those who did not ( $M = 0.31, SD = 0.19; t(265) = 2.65, p = .008$ ; H4 supported). Overall, these results suggest that compared to traditional dashboards, conversational dashboards improve transparent interaction, particularly if participants can use only natural language and mouse interaction is removed (i.e., CDB-NLO). However, if users can choose between natural language and mouse as in CDB-NLE, only participants who have completed the onboarding achieve a higher level of transparent interaction with the dashboard. Finally, to test the remaining hypotheses on the effects of transparent interaction on efficiency (H5) and effectiveness (H6), we ran a multivariate regression with transparent interaction as the independent variable and efficiency and effectiveness as the two dependent variables. Consistent with our hypotheses and in line with TEU, the results show that transparent interaction has a significant positive effect on efficiency ( $\beta = 0.58, p < .001$ ; H5 supported) and effectiveness ( $\beta = 0.9, p < .001$ ; H6 supported).

**Post-hoc Analysis.** Contrary to our expectations, participants in the CDB-NLE condition did not achieve significantly higher levels of transparent interaction than participants in

the TDB condition. Since participants in the CDB-NLE (vs. CDB-NLO) condition could choose whether or not to interact with the dashboard using natural language, a possible explanation could be that some of them used only the “traditional” way of interacting with the dashboard using the mouse, which might have negatively affected their level of transparent interaction. To investigate this further, we conducted a post-hoc analysis of user behavior in the CDB-NLE condition. For each participant, we calculated the proportion of navigation steps they took using natural language (in both spoken and written form). This resulted in a continuous variable ranging from 0 to 1, where a value of zero indicates that natural language was not used at all. Subsequently, we ran a linear regression model with transparent interaction as the dependent variable and the proportion of navigation steps via natural language as our independent variable. The results in Figure 2.5 show that the proportion of navigation steps via natural language significantly influenced transparent interaction ( $\beta = 0.54$ ,  $p < .001$ ), suggesting that the more users interact with the dashboard using natural language, the higher their level of transparent interaction.

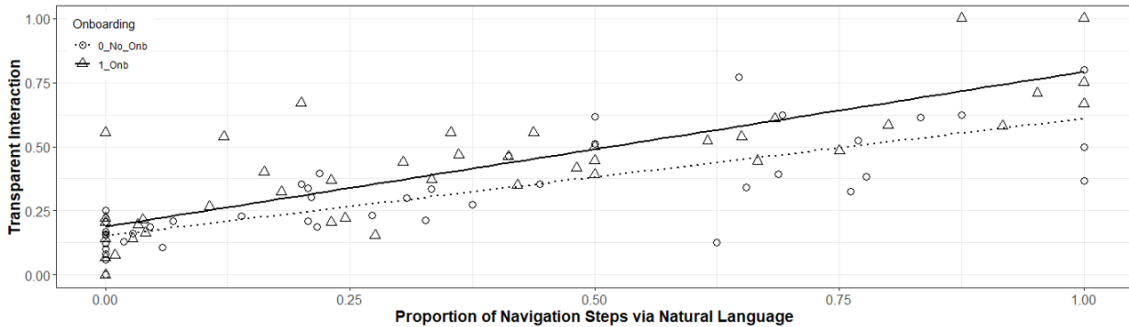


Figure 2.5: Effect of Proportion of Interactions performed via Natural Language and Conversational Onboarding on Transparent Interaction (CDB-NLE).

Since half of the participants in the CDB-NLE condition completed the conversational onboarding to familiarize themselves with how to interact with the conversational dashboard, it is conceivable that those participants also more frequently used natural language than participants who did not receive the onboarding. Therefore, we conducted a mediation analysis using the bootstrapping approach with 5000 samples (Hayes, 2017). We estimated a simple mediation model (Model 4) with onboarding as the independent variable, proportion of navigation steps via natural language as the mediator, and transparent interaction as the dependent variable. The results show that the direct effects of both onboarding ( $\beta = 0.089$ ,  $p = .002$ ) and proportion of navigation steps via natural language ( $\beta = 0.53$ ,  $p < .001$ ) are significant. However, the effect of onboarding on proportion of navigation steps via natural language ( $p = .46$ ), as well as the indirect effect of onboarding on transparent interaction

through proportion of navigation steps via natural language, are not significant ( $CI = [-0.03, 0.04]$ ;  $p = .47$ ). In summary, these results suggest that although completing the onboarding did not result in a significant increase in the use of natural language to interact with the dashboard, it helped participants to achieve higher levels of transparent interaction. A possible explanation could be that participants had learned when to choose which way of interaction and how to formulate natural language input more effectively to navigate the dashboard. The results also provide further evidence that since participants could choose between natural language and mouse, some of them did not harness the potential benefits of natural language interaction, which ultimately resulted in lower levels of transparent interaction. Put differently, on average participants in the CDB-NLE condition did not perform better than those in the TDB condition because some of them did not leverage our new functionality but used only their mouse to interact with the dashboard. Table 2.6 summarizes the results of the hypothesis testing.

Table 2.6: Summary of the Results.

Hypothesis	Result	Findings
H1	Supported	Users who can interact with a dashboard only using natural language (CDB-NLO) achieve higher levels of transparent interaction than users who can interact only using a mouse (TDB).
H2	Not supported	Users who can interact with a dashboard using both natural language and mouse (CDB-NLE) do not achieve higher levels of transparent interaction than users who can interact using only the mouse (TDB). However, a post-hoc analysis shows that transparent interaction with a CDB-NLE depends on whether and how often users use natural language in their interaction.
H3	Not supported	Completing the conversational onboarding of a CDB-NLO does not improve users' transparent interaction with a CDB-NLO.
H4	Supported	Completing the conversational onboarding of a CDB-NLE improves users' transparent interaction with a CDB-NLE.
H5	Supported	Transparent interaction increases efficiency.
H6	Supported	Transparent interaction increases effectiveness.

## 2.5 Discussion

Providing information to protect the public's health and safety is an important task in crisis response. In recent crises, such as the COVID-19 pandemic, many governments and health organizations developed dashboards that organize complex crisis-related data in an easy-to-digest visual format. Although these crisis response dashboards target the

general public, research suggests that the average user could face difficulties in interacting with a dashboard and finding the information needed to make everyday decisions. To address this challenge, we proposed a theory-driven design for conversational dashboards in crisis response and developed a conversational dashboard for the COVID-19 pandemic following the DSR approach. In contrast to current crisis response dashboards, our artifact enables users to use natural language in spoken or written form to interact with the dashboard. In addition, our artifact includes conversational onboarding that helps users familiarize themselves with how to interact with the dashboard using natural language. To rigorously evaluate our proposed design, we conducted a large-scale online experiment with six different versions of our dashboard. The evaluation results show that compared to a traditional dashboard, users achieve higher levels of transparent interaction with our dashboard, ultimately increasing their efficiency and effectiveness in finding the information they need. Moreover, the results demonstrate that the conversational onboarding supports users in learning how to interact with the dashboard, particularly when they can use both natural language and mouse, which further improves their transparent interaction. Following the guidelines of Gregor and Jones (2007), we have synthesized our findings into a nascent design theory for conversational dashboards in crisis response (see Table 2.7).

### **2.5.1 Theoretical Contributions**

This paper makes three important theoretical contributions. Our core contribution is a nascent design theory that offers explicit prescriptions on how to extend crisis response dashboards with natural language interaction capabilities in order to improve users' transparent interaction and access to information. While previous research and current crisis response dashboards have focused heavily on GUIs, we propose an innovative, theory-driven design for conversational dashboards and instantiate it in a novel data science artifact: a conversational dashboard for the COVID-19 pandemic equipped with a CUI to allow natural language interaction in spoken or written form. With these findings, we contribute to research on dashboard design, both in general and in the specific context of crisis response, by delivering prescriptive knowledge for designing conversational crisis response dashboards that enable faster and easier access to important crisis-related information. More broadly, our findings also add to the data science literature by providing novel insights on how natural language can narrow the gap between the creation and consumption of insights provided by data science artifacts, particularly when they are designed for broader audiences. While prior research has emphasized key activities (e.g., data analysis, model development) in the earlier stages of the information value chain (Abbasi et al., 2016), the

ultimate goal of data science is to offer actionable insights that support decision making (Grover et al., 2018). With our focus on the latter stages of the information value chain, we therefore complement existing data science research by providing design guidelines for helping users access information in dashboards so that they can extract insights required for improved decision making and ultimately take full advantage of such data science artifacts. Second, our findings shed light on potential design trade-offs that arise in providing users with multiple ways of interacting with a crisis response dashboard. As predicted, we find that users achieve higher levels of transparent interaction when they can use only natural language instead of only their mouse to interact with the dashboard, thus confirming our expectations that navigating a dashboard by articulating an information need in natural language is generally easier than translating it into a series of actions in the GUI. However, our results also suggest that when given the opportunity to use both natural language and mouse, a number of users prefer not to use natural language at all in interacting with the dashboard. Instead, they rely solely on the more familiar mouse interaction, which unfortunately often leads to lower levels of transparent interaction. This finding is consistent with TEU (Burton-Jones & Grange, 2013), which posits that transparent interaction is not a property of the system, but rather involves a user, system, and task. In other words, different users use the same dashboard for the same task but achieve different levels of transparent interaction because one leverages natural language while another uses the mouse. Consequently, it could be argued that natural language and mouse interaction should not be implemented together; rather, one must be chosen over the other (preferably natural language interaction). However, our results suggest that this dilemma can be addressed through conversational onboarding, which allows users not only to familiarize themselves with how to interact with the dashboard using natural language, but also to learn when and where to choose which way of interacting. This might also explain why onboarding has a weaker impact on transparent interaction when users have only mouse or only natural language available instead of both. In such contexts, users do not have the possibility of deciding for themselves and therefore inevitably have to deal with the benefits and challenges that come with one particular way of interacting with the dashboard. Taken together, our findings suggest that conversational onboarding is a valuable addition to conversational dashboards, even if it requires users to take an additional step before they can actually use the dashboard. In summary, these findings contribute to the emerging stream of research on novel interaction modes (e.g., Y. Liu et al., 2021) by uncovering and addressing design trade-offs in crisis response dashboards that can be navigated using both natural language and mouse.

Third, our research offers a methodological contribution to the IS use literature by demonstrating a novel approach for measuring transparent interaction—a key dimension of effective use—based on user interaction data. Although Burton-Jones and Grange (2013) have noted that “self-report measures alone may prove insufficient” to measure effective use objectively (p. 653), existing research has mostly relied on self-reported data (Trieu et al., 2022). Other, more objective approaches, such as the observation of users in their workplace setting (e.g., Burton-Jones and Volkoff, 2017), are often time-consuming and labor-intensive. In contrast, we use log data of user interactions with the dashboard (e.g., mouse clicks, natural language input) to provide a more objective assessment of users’ level of transparent interaction by comparing their actual navigation path to the minimum number of navigation steps that are required to access a particular piece of information in the dashboard. Therefore, researchers can use our approach as a blueprint for a viable, less time-consuming alternative or supplement to existing measurement approaches of effective IS use.

### **2.5.2 Practical Implications**

The outcomes of our DSR project have important implications for data science practitioners who build models, create visualizations, and develop dashboards for crisis response. Industry-standard data science processes, such as CRISP-DM (Shearer, 2000) and Microsoft’s Team Data Science Process (Microsoft, 2022b), emphasize that the successful deployment of data science artifacts (e.g., dashboards) and their use by the target audience is a crucial step in any data science project. Only if users are able to access and extract insights from a data science artifact, can its value be realized (Davenport & Malone, 2021). Against this backdrop, our work can help data scientists realize the potential of natural language interaction to make their artifacts in general and dashboards in particular more accessible to broader audiences. To this end, the design principles, system architecture, and in-depth description of our artifact—a conversational COVID-19 dashboard—provide actionable guidance on how to leverage existing open-source frameworks and cloud services (e.g., Microsoft’s Bot Framework) to develop conversational dashboards that enable users to easily access information using natural language.

Additionally, our work offers practical implications for governments, health organizations, and other institutions that provide crisis response dashboards with the aim of informing the general public. As our findings suggest that traditional dashboard designs could fail to accommodate the average user, we recommend practitioners to explore alternative ways of providing access to the information in a crisis response dashboard, for example, using natural



language. While our evaluation shows that natural language interaction could possibly replace traditional ways of interacting with a dashboard (e.g., using a mouse), several users reported that they would still prefer to have the option to revert to using their mouse or touchscreen if, for example, they are in a public space. Therefore, practitioners could first implement natural language interaction to complement rather than replace existing ways of interacting with their dashboard and, importantly, combine it with conversational onboarding to familiarize users with how and when best to use natural language. Following these guidelines, practitioners could make their crisis response dashboards more accessible to broader audiences and ultimately disseminate important information more effectively during a crisis.

### **2.5.3 Limitations and Future Research**

Our work is subject to some limitations. First, although we provide design knowledge for a class of artifacts (i.e., crisis response dashboards), the instantiation and evaluation of DPs focus on one particular instance of this class, namely a COVID-19 dashboard. Since dashboards for other crises, such as natural disasters, might produce different kinds of data and require different data visualizations, one limitation of this DSR project is its focus on the COVID-19 pandemic. However, since many crisis response dashboards build on the same underlying technology and provide similar user interfaces, our design theory should be generalizable to dashboard implementations for other crises. More specifically, the central idea of our nascent design theory—enabling users to interact with a dashboard using natural language—is independent of the underlying data and types of visualizations in a dashboard. However, future research is needed to test our design theory in the context of other crises.

Second, our DSR project focuses on transparent interaction as one key dimension of effective use. However, the conceptualization of effective use in TEU comprises two additional dimensions—representational fidelity and informed action—that were not included in our research. Although kernel theories are rarely used as-is in DSR “due to a mismatch in terms of scope and granularity between the theoretical frameworks and the design problem” (Arazy et al., 2010, p. 457), investigating the other two dimensions of effective use in the context of crisis response dashboards would be a fruitful future research direction. Further, future work could explore other parts of TEU by, for example, providing design knowledge for physical structures (e.g., microphones, screens). Finally, there are other important challenges for the design of these dashboards, such as their faithful representation

of real-world states (Recker, 2021) and data quality (Torres & Sidorova, 2019), which also warrant further research.

Third, we used MTurk to recruit participants for our final evaluation. Although studies show that the demographics of MTurk workers are similar to that of the general U.S. population and more diverse than many other samples (Buhrmester et al., 2011), the MTurk sample might limit the generalizability of our findings. To address this limitation, we used the parameters MTurk provides to recruit participants with a wide range of socio-demographic backgrounds and experience levels (Steelman et al., 2014). However, future research should validate our findings with a nationally representative sample.

Fourth, our final evaluation was conducted in a laptop or desktop environment. Therefore, the traditional crisis response dashboard, which we compared to our conversational dashboard, only supported conventional mouse interaction. However, mobile devices, such as smartphones and tablets, might also offer users additional ways of interacting with a traditional dashboard using touch (e.g., swiping, pinching). Although touch and mouse interaction exhibit similar characteristics and limitations in the context of dashboards (Srinivasan & Stasko, 2018), future research should investigate how touch interaction affects users' level of transparent interaction.

Finally, we used behavioral data to measure users' transparent interaction, as well as their effectiveness and efficiency in finding information. Although we followed Burton-Jones and Grange (2013) suggestions to compare users' actual navigation steps against the "quickest navigation path" using log data, there could be other ways of calculating transparent interaction based on this data. Therefore, more research is needed to examine and compare our approach against other measurement approaches based on self-reported data.

## **2.6 Conclusion**

Dashboards are important data science artifacts designed to inform the general public during a crisis. During the COVID-19 pandemic, they attracted more public attention than ever before. Although IS and HCI research have dealt with the design and use of dashboards for decades, most research has focused on dashboards for decision makers in organizations, suggesting that previous findings might not generalize well to the class of crisis response dashboards that need to be designed for broader audiences. With our research, we show how IS theories and methods can be used to improve real-world data science artifacts and, more broadly, demonstrate that the IS community in general and DSR scholars in particular, can help the world to be better prepared for future crises.

Table 2.7: A Nascent Design Theory for Conversational Dashboards in Crisis Response .

Component	Description
Purpose and scope	The purpose of the design theory is to provide prescriptive knowledge on how to design conversational dashboards for crisis response.
Constructs	The design theory builds on the following construct from TEU (Burton-Jones & Grange, 2013): transparent interaction, efficiency, effectiveness, and the two drivers of effective use (i.e., adaptation and learning).
Principles of form and function	We propose three DPs for the design of conversational dashboards in crisis response: <ul style="list-style-type: none"> <li>- DP1: To enable the general public to seamlessly navigate a dashboard for crisis response, provide users with the ability to use spoken or written language in a natural way because articulating an information need in natural language is easier than translating it into a series of actions in the graphical user interface.</li> <li>- DP2: To enable the general public to seamlessly navigate a dashboard for crisis response, provide users with the ability to choose between natural language and mouse because it gives them flexibility for the task at hand and takes their individual preferences into account.</li> <li>- DP3: To enable the general public to seamlessly navigate a conversational dashboard for crisis response, provide users with conversational onboarding that takes them step-by-step through the natural language interaction with the dashboard because this helps users familiarize themselves with how to interact with the dashboard using spoken or written language.</li> </ul>
Justificatory knowledge	The three MRs were derived from TEU, our kernel theory. In addition, our DPs were informed by research on affordances (DP1-2) and enactive learning (DP3).
Testable propositions	We derived two testable propositions to evaluate our proposed design: <ul style="list-style-type: none"> <li>- Proposition 1: A crisis response dashboard equipped with a conversational user interface allowing users to interact with the dashboard using natural language enables them to achieve higher levels of transparent interaction.</li> <li>- Proposition 2: A conversational crisis response dashboard equipped with conversational onboarding walking users through the natural language interaction with the dashboard enables them to achieve higher levels of transparent interaction.</li> </ul>
Artifact mutability	The conversational dashboard is mutable, specifically with respect to the underlying data. While updates to the existing data can be handled without major changes, more adaptation is required for integrating new metrics (e.g., number of people vaccinated), providing new data visualizations, or supporting additional languages. With more substantive changes, the artifact could also be adapted for use in other crises (e.g., other pandemics or natural disasters).
Principles of implementation	To instantiate the DPs in our artifact, we developed a system architecture based on existing open-source frameworks and libraries, which can serve as a blueprint for implementing similar artifacts.
Expository instantiation	The design theory was instantiated in an artifact: a conversational dashboard for the COVID-19 pandemic.

# 3. Study II: Designing Multimodal BI&A Systems for Co-Located Team Interactions<sup>3</sup>

## 3.1 Introduction

The increasing importance of data-driven decision making in organizations reshapes work practices of employees at any level (H. Chen et al., 2012). To support employees' data understanding and decision making, most organizations have implemented business intelligence & analytics (BI&A) systems. These systems process and present data to a broad spectrum of users, for example, in the form of reports or dashboards. Given their widespread availability, BI&A systems are now used in all areas of business to facilitate decision making. However, the success of BI&A systems will be determined by how effectively they are used (Burton-Jones & Grange, 2013).

Today, decisions based on BI&A systems are not only made by individuals alone but increasingly also by teams. Due to this trend, teams are crucial for organizations in making data-driven decisions (Majchrzak et al., 2012). For example, before deciding on a new customer retention strategy, employees from sales, controlling, and management departments meet and analyze churn data from the past. These insights and informed actions are derived in co-located team interactions (Dennis, 1996; Isenberg et al., 2012; Schmidt et al., 2001). Yet, surprisingly few BI&A systems support co-located team interactions (Berthold et al., 2010; Isenberg et al., 2012) and many teams struggle with working together equitable and flexible using current BI&A systems (Dayal et al., 2008; Kaufmann & Chamoni, 2014). For example, with current BI&A systems, only one person in a team meeting would interact with the system and carry out the analysis, while the other meeting participants can only observe the activities or comment on the results. Consequently, achieving effective use of BI&A systems in co-located team interactions remains a challenge.

According to Burton-Jones and Grange (2013), effective use of information systems (IS) involves three core elements: transparent interaction, representational fidelity, and informed action. Teams need to unimpededly interact with a BI&A system in order to obtain faithful representations (e.g., data analyses), which ultimately enables them to take informed actions (e.g., make business decisions). Therefore, at the most fundamental level, BI&A systems need to be designed in a way that facilitates transparent interaction because otherwise

---

<sup>3</sup>This chapter is based on the following studies which are published or in work: Ruoff and Gnewuch (2021a).

achieving effective use is likely not possible. One approach to facilitate transparent interaction with BI&A systems in co-located team interactions could be to supplement the established interaction modalities of BI&A systems (i.e., mouse, keyboard, and touch) with speech interaction. In recent years, the capabilities of conversational user interfaces (CUI) have greatly improved and they are increasingly used to enable users to access information and interact with a system in a more natural and intuitive way (McTear, 2017). Hence, combining existing interaction modalities with speech interaction provided through a CUI may compensate for the disadvantages of each modality and, therefore, facilitating effective use of BI&A systems. Consequently, BI&A systems that support multiple modalities (hereafter referred to as multimodal BI&A systems) could enable teams to interact with a BI&A system in a flexible and effective manner and more actively support involving all team members in the decision making process (Deng et al., 2004; Oviatt, 1999).

However, while there is a large body of design knowledge on BI&A systems for individual use contexts, research on the effective use of BI&A systems for team interaction is scarce. Furthermore, multimodal BI&A systems have been predominantly studied from a technology-centric perspective (Turk, 2014). Thus, there is a lack of prescriptive knowledge on how to design multimodal BI&A systems for co-located team interactions. Moreover, it is not well understood whether and how multimodal BI&A systems can facilitate effective use and support decision making in co-located team interactions. Hence, we address the following research question:

*How to design multimodal BI&A systems for co-located team interactions in order to facilitate the systems' effective use?*

To address this question, we conduct a Design Science Research (DSR) project (Kuechler & Vaishnavi, 2008). Drawing on the theory of effective use (Burton-Jones & Grange, 2013) and existing design knowledge for multimodal user interfaces (MUI) (Deng et al., 2004; Reeves et al., 2004), we designed, implemented, and evaluated a multimodal BI&A system that combines touch and speech interaction. We developed and evaluated our software artifact using a confirmatory focus group in cooperation with the finance & accounting department of a large European energy provider.

This paper presents the results of our first design cycle. Overall, our DSR project contributes to the body of design knowledge for BI&A systems by demonstrating how the combination of touch and speech increases transparent interaction and representational fidelity in order to achieve effective use in co-located team interactions. Furthermore, our proposed design principles advance existing guidelines for MUIs and ground them in the theory of

effective use. In particular, we contribute with three design principles for multimodal BI&A systems for teams. Overall, our work represents an improvement in the DSR knowledge contribution framework (Gregor & Hevner, 2013), as it represents a more efficient and effective solution for a known problem. For practitioners, we provide applicable guidelines for the implementation of multimodal BI&A systems (Gregor & Jones, 2007).

## **3.2 Related Work and Theoretical Foundations**

### **3.2.1 Business Intelligence & Analytics Systems for Teams**

Business intelligence & analytics (BI&A) is often described as “techniques, technologies, systems, practices, methodologies, and applications that analyze critical business data to help an enterprise better understand its business and market and make timely business decisions” (H. Chen et al., 2012, p. 1166). BI&A reinforces human cognition as well as capitalize on human perceptual capabilities by integrating data analysis systems with decision support systems (Yigitbasioglu & Velcu, 2012). In order to accomplish this, tools, applications, and technologies focussing on decision making are required (Larson & Chang, 2016).

In the process of deriving knowledge from the data using BI&A systems and making decisions, additionally, tools are required that support teams in collaborating (Abbasi et al., 2016). Different approaches have been used to support teams during decision making and data understanding. Group decision support systems (GDSS), for example, have been researched for a long time in order to increase team effectiveness, efficiency, and satisfaction in decision making (Burstein et al., 2008; Nunamaker & Deokar, 2008). Key insights from these research streams are, that cross-functional teams can lead to an increase in effectiveness due to synergies. However, they can also lead to incomplete access to and use of information needed for successful decision making (Nunamaker & Deokar, 2008). These insights are crucial to data-driven decision making in organizations and, therefore, the collaborative aspect of decision making receives increasing relevance in BI&A system research (Abelló et al., 2013; Berthold et al., 2010). Suggesting that during the transfer from individual to team level, especially, the functional and technical aspects need to be mapped to the requirements teams pose to BI&A systems (Kaufmann & Chamoni, 2014). However, research on BI&A systems for co-located team interaction and their requirements is crucial but scarce (Berthold et al., 2010; Ruoff, Gnewuch, & Maedche, 2020).

### 3.2.2 Multimodal User Interfaces

Multimodal user interfaces (MUI) enable processing two or more input modalities from users, such as speech, touch, or gaze (Oviatt, 2003). Their fundamental idea is to remove existing constraints on human-computer interaction by leveraging the full communication and interaction capabilities of humans in order to provide a natural interaction between the user and the system (Turk, 2014). The first MUI was Bolt's "Put-that-there" system (1980) integrating speech and gesture to increase the ease of use of the system. Since then, many MUIs have been developed (e.g., Turk, 2014). Particularly, speech input has been often used in combination with other modalities, since speech has powerful complementary capabilities, such as providing complex interactions in contrast to the simple interactions of touch (Deng et al., 2004; Saktheeswaran et al., 2020). Several guidelines have been published by research describing the general requirements for MUIs (Reeves et al., 2004) and by practice describing requirements for the combination of specific modalities (Deng et al., 2004). Integrating insights from different research streams, such as research on CUI (Gnewuch et al., 2018; McTear, 2017) as well as interaction preferences (Pitt et al., 2011). Today, MUIs are attributed a high degree of relevance for BI&A systems as they can provide fluid interactions during decision making (Dayal et al., 2008; Roberts et al., 2014; Saktheeswaran et al., 2020). However, there is still a lack of research on multimodal BI&A systems, even though this could enhance the interaction between users and BI&A systems and could lead to improved effectiveness and efficiency (Dayal et al., 2008).

### 3.2.3 Theory of Effective Use

IS should be used effectively since the shallow use of them alone is not sufficient to ensure that the organization's objectives are met (Seddon, 1997). According to Burton-Jones and Grange (2013), effective use can be defined as "using a system in a way that helps attain the goals for using the system" (p. 4). Based on their conceptualization, effective use is an aggregated construct comprising three hierarchical dimensions: (1) transparent interaction, (2) representational fidelity, and the outcome dimension (3) informed action (Burton-Jones & Grange, 2013). As illustrated in Figure 3.1, the three dimensions of effective use are influencing each other. Initially, the unimpeded access to the system's representations (transparent interaction) improves the ability to obtaining representations that faithfully reflect the domain (representational fidelity). The representational fidelity in turn aims to improve informed action, which is the extent to which a user acts on faithful representations. Therefore, a user's overall level of effective use is determined by the aggregated levels of the

three dimensions (Burton-Jones & Grange, 2013). For example, users of a BI&A system need to access accurate business information (transparent interaction), such as which products had lower revenue than expected based on the purchase history (representational fidelity), to be able to make decisions for future business endeavors (informed action).

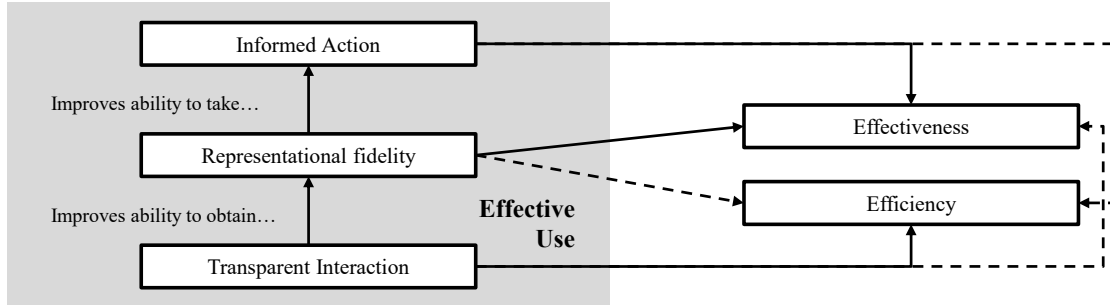


Figure 3.1: Theory of Effective Use (adapted from Burton-Jones and Grange (2013)).

In order to positively influence effective use during the interaction between users and IS, Burton-Jones and Grange (2013) identified two major drivers: adaptation actions and learning actions. In our paper, we focus on adaptation actions, which are defined as any action a user takes to improve (1) a system’s representation of the domain of interest; or (2) his or her access to them, through a system’s surface or physical structure. Therefore, researchers in the context of BI&A systems need to expand their focus from organizational aspects and data quality (Surbakti et al., 2020) to include also the interaction between users and the system. Especially, when designing multimodal BI&A systems, researchers should consider how users are able to adapt their interaction with multimodal BI&A systems according to the task and context.

### 3.3 Design Science Research Project

To design a multimodal BI&A system that can be effectively used in co-located team interactions, we follow the DSR approach as described by Kuechler and Vaishnavi (2008). We argue that this research approach is particularly suited to address our research question because it allows us to integrate existing design knowledge (Deng et al., 2004; Reeves et al., 2004), descriptive knowledge from the theory of effective use (Burton-Jones & Grange, 2013), and empirical results from our evaluation phases to incrementally improve our artifact. These foundations provide a rigorous grounding and allow us to contribute to the existing knowledge base. To further provide relevance to our rigorous approach (Hevner, 2007) in understanding multimodal BI&A systems, we collaborate with an industry partner serving as our research case. Our industry partner is the finance & accounting department



of a large European energy provider. The joint research project is conducted because the company is aware of the need to establish new forms of interaction with data. The access to practitioners enables us to sharpen our awareness of the problem as well as to perform evaluations with practitioners.

General Design Science Cycle	Cycle 1 Understanding	Cycle 2 Lab Experiment Evaluation	Cycle 3 Application to practice
Awareness of Problem	Literature review Interaction-elicitation study	Reflection of focus group analysis	Reflection of experiment analysis
Suggestion	Synthesis of design principles based on empirical findings and theory	Adapt design principles based on evaluation results and insights from focus groups	Adapt design principles based on evaluation results and insights from lab experiment
Development	Instantiation of design principles as a software artifact	Modification of software artifact	Modification of software artifact
Evaluation	Qualitative evaluation of software artifact (confirmatory focus group)	Quantitative evaluation of software artifact (lab experiment)	Quantitative evaluation of software artifact (field experiment)
Conclusion			Deliver nascent design theory

Figure 3.2: Design Science Research Project (adopted from Kuechler and Vaishnavi (2008)).

In our first design cycle, we focus on the fundamental dimension of the theory of effective use, the transparent interaction with multimodal BI&A systems, and the impact of the systems' design on their effective use.

**Awareness of Problem:** In order to better understand issues of data-driven decisions in co-located teams and potential issues in the design of multimodal BI&A systems, we started our research by conducting a literature review on multimodal BI&A systems for co-located team interactions. This literature review provided us with potential issues in the design of multimodal BI&A systems for co-located team interactions and allowed us to extract approaches on how to tackle these issues from various disciplines, such as the discipline of computer-supported cooperative work and information visualization.

Subsequently, we conducted an interaction-elicitation study following the approach by Morris (2012) to derive data on how people would want to interact with a multimodal BI&A system to compare the proposed guidelines to feedback from potential users. Overall, 30 participants with an average age of 22.8 years ( $SD = 1.9$ ) took part in the study. There were 8 female and 22 male participants, mostly students with a background in economics and engineering. In accordance with Badam and Elmqvist (2019), we motivate the choice of using students as the representative population as the focus of this study was to extract interactions with multimodal BI&A systems, and therefore, no specific expertise except the experience of using touch and speech interfaces was needed.

The interaction-elicitation study consisted of two parts (Ruoff & Maedche, 2020). First, the participants were shown 14 randomized core functionalities of BI&A systems, such as filtering, selecting, and obtaining details, which we extracted based on the framework of Yi et al. (2007). After each demonstration of a functionality, the participant was asked to propose an interaction on how s/he would invoke the functionality using speech, touch, and the combination of these modalities. For each modality, the participant stated in which context s/he would use this interaction. Furthermore, the participant rated for each functionality which modality s/he would prefer and stated why s/he rated the modalities in this order. Finally, after proposing interactions for each functionality, a semi-structured post-study interview was conducted with a focus on the use of multimodal BI&A systems as well as on how they provide assistance to users in order to interact properly. With the consent of the participants, audio and video were recorded for the whole interaction-elicitation study.

In order to analyze our results, we coded the post-study interviews to derive common issues from the users' perspective and the user-defined interactions for the core functionalities. To calculate the agreement for the interaction of each modality and core functionality, we derived the percentage of participants proposing the most popular interaction (Morris, 2012). For example, 17 participants proposed the interaction "Filter for <Entity>" as a speech interaction for the functionality "filtering". Therefore, the interaction for filtering using the modality speech has an agreement of 57%. Furthermore, based on the ranking of the modalities for each functionality, we were able to derive the modalities preferred for the functionalities.

Suggestion: To address the issues identified in the problem awareness phase, we proposed three design principles for multimodal BI&A systems. These design principles were derived based on our literature review, the results of our interaction-elicitation study, and the theory of effective use as our kernel theory.

Development: To demonstrate how these design principles can be implemented, we instantiated them in a software artifact using state-of-the-art technologies for the recognition of speech and touch input.

Evaluation: In the evaluation phase, we opted for confirmatory focus groups as they provide a collective view on a topic of interest from a group of experienced participants and to establish the utility of the software artifact in field use (Tremblay et al., 2010). We invited thirteen employees from the finance & accounting department with a focus on controlling, customer processes, data science, as well as general management in the context of finance

(9 males, Mage = 34.6 years, MWorkExp = 10.1 years). Therefore, all practitioners have experience using BI&A systems in co-located team interactions and can provide insights into the topic of interest. The guiding thought of these confirmatory focus groups were issues related to the use of multimodal BI&A systems of practitioners in co-located team interactions and possible strengths, weaknesses, opportunities, and threads of facilitating the interactions of the multimodal BI&A system through touch and speech.

After a short introduction into the goal and procedure of the confirmatory focus group, we separated them into two groups of six and seven practitioners. The confirmatory focus group with both groups followed the same procedure. First, the use case of leveraging the multimodal BI&A system in co-located team interactions was presented to the practitioners. The moderating researcher guided the practitioners through questions that are of interest in a typical decision making task (e.g., whether the price for an energy product should be increased in the future). During the demonstration of the use case, the moderating researcher was supported by our multimodal BI&A system and used various possible interactions with the multimodal BI&A system, such as speech for filtering or touch to select data of interest. The practitioners were included in the interaction with the system and could also use the multimodal BI&A system during the demonstration. After the demonstration, questions regarding the use case and the multimodal BI&A system were discussed. Following a 20 minute discussion, we explained the Strength-Weakness-Opportunity-Threat (SWOT) analysis method to the practitioners which was used to structure the confirmatory focus group. Subsequently, the practitioners were given time to write down their perceived strengths, weaknesses, opportunities, and threats of multimodal BI&A systems in co-located team interactions on index cards. Finally, the index cards were read out loud and explained by the respective practitioner, providing the researchers with the possibility to ask follow-up questions on recurring points. Both sessions were recorded with the consent of the practitioners and transcribed after the workshop.

Following the confirmatory focus groups, all audio recordings were transcribed using MAXQDA 2018. Similar to previous evaluation studies that used recorded verbalization, our “coding scheme consisted of a series of categories about the behavior to be studied” (Vitalari, 1985, p. 226). More specifically, our coding scheme included the concepts of effective use (e.g., transparent interaction and representation fidelity) and the relationships between them. In the first step, we combined similar index cards with overlapping explanations by the respective practitioner based on the results of the initial coding. In a second step, we derived first-order concepts from these groups (X. Zhang, 2017). For

example, “no tool knowledge needed” and “makes it easier to find options that can otherwise only be reached with many clicks” were combined with other similar statements to a group and the first-order concept “Limited knowledge about the functionality of the system necessary” was derived and mapped to the corresponding design principle.

As depicted in Figure 3.2, we plan to conduct two additional cycles to further refine our design and evaluate it in a lab and field experiment. In the second design cycle, we plan to refine the design principles based on the evaluation results of the first cycle. Furthermore, we will focus on how to adapt the multimodal BI&A system to team characteristics and context. We plan to experimentally evaluate how the adaptation of the transparent interaction and representation fidelity affects the effective use of the BI&A system. The final and third design cycle aims to fine-tune our design principles using the results of the previous evaluations. This will provide us the opportunity to introduce the multimodal BI&A system to various teams in the finance & accounting department and to better understand the impact of the design principles on effective use. Our ultimate goal is to deliver a nascent design theory for multimodal BI&A systems as described by Gregor and Jones (2007).

## 3.4 Results

### 3.4.1 Awareness of the Problem

In the following, we present the results of the problem awareness phase along the two main dimensions of effective use as a lens: (1) transparent interaction and (2) representational fidelity. Specifically, we raise three major issues (I) with regards to current BI&A systems.

**Transparent Interaction:** Researchers aim to facilitate effective use by providing unimpeded access to current BI&A systems through additional input modalities. Multiple studies have explored how the combination of different modalities in multimodal BI&A systems can assist teams during co-located team interactions (Badam, Amini, et al., 2016; Langner, Horak, et al., 2018; Lee, Smith, et al., 2015; H. Nguyen et al., 2017). The combination of modalities used in these studies varies between touch and speech, mid-air hand gestures and touch, mid-air hand gestures and speech as well as touch and pen. Therefore, it is difficult to generalize the results of these studies. However, the general conclusion of these studies is that only providing additional modalities to users does not automatically increase effective use (H. Nguyen et al., 2017). Therefore, it is unclear which and how multiple modalities in BI&A systems should be combined in order to facilitate transparent interaction (I1).

A common modality used for multimodal BI&A systems is touch since it conveys the team member’s “intention quickly and unambiguous to the system” (Badam, Amini, et al., 2016)

and is in line with the affordance of displays to be touched (Norman, 2016). However, teams are still unable to convey complex information to the multimodal BI&A systems without help from menus. To tackle the limitations of touch and to fulfill the requirements of the adaptivity of MUIs (Reeves et al., 2004), researchers combine touch with additional modalities. To augment touch as a modality, guidelines for MUIs and the results of our interaction-elicitation study indicate that speech could be beneficial to convey complex information (Deng et al., 2004; Saktheeswaran et al., 2020). Especially since the team can “easily manipulate the visualized data in a natural and intuitive approach” (H. Nguyen et al., 2017, p. 7) through speech. However, in most multimodal BI&A systems, speech is still a hidden affordance as the microphone is subtly integrated into the display and the interaction provides no physical feedback. Therefore, individuals and teams struggle to use modalities, such as speech, because they are less “visible” (I2).

**Representational Fidelity:** In many studies, achieving representational fidelity is supported by providing either a dashboard (Badam, Amini, et al., 2016; Langner, Kister, et al., 2019; Lee, Smith, et al., 2015) or a single information visualization (H. Nguyen et al., 2017). In order to maintain representational fidelity during decision making, teams need to be able to adapt the visual representations using transparent interaction (Srinivasan, Lee, et al., 2020), by altering queries to the data (Jetter et al., 2011), or by enhancing or changing the underlying data (Chung et al., 2014). These adaptation actions can be performed using different modalities. For example, users could click on a filter (touch) or ask the system to select a specific year (speech). However, in the context of MUIs, researchers currently design the mapping between interaction techniques, which users can utilize to maintain the representational fidelity, and the system functionality bottom-up based on their specific system. As a result, a guiding paradigm or design principle is missing to guide this process. Therefore, it is unclear how to map fundamental dashboard interaction techniques to multimodal system functionalities (I3).

In summary, there are several issues in the design of multimodal BI&A systems for co-located team interactions. Based on the results of our literature review and interaction-elicitation study, we determined that existing research is missing an understanding how to facilitate effective use of BI&A systems in co-located team interactions. Therefore, we subsequently focus on the gap in how multimodal BI&A systems need to be designed to facilitate effective use and how teams can be assisted during their interaction.

### 3.4.2 Suggestion

To address the identified issues of multimodal BI&A systems, we suggest designing a system that facilitates effective use by providing a MUI. Building on the theory of effective use, we argue that a multimodal BI&A system that provides unimpeded access to the system’s representation (transparent interaction) and enables users to obtain faithful representations (representational fidelity) will positively influence informed actions and, therefore, facilitate effective use. Consequently, we formulate two meta-requirements (MR) based on the dimensions of effective use: Multimodal BI&A systems should provide a high level of transparent interaction (MR1) and representational fidelity (MR2). To increase transparent interaction (MR1) and to tackle I1 & I2, the theory of effective use suggests adapting the physical structure and the surface structure. It further indicates, that “the sole purpose of these structures is to support access to representations” (Burton-Jones & Grange, 2013, p. 646). In the context of the physical structure, the core strength of providing multiple input modalities is that multimodal systems decrease the distance between intent and interaction (Lee, Isenberg, et al., 2012) and, therefore, support the access to the representations of the system, which is based upon the use of different modalities complementing each other (Sundar et al., 2015). By providing the possibility to choose between modalities, the multimodal BI&A system is robust to varying contexts, such as noise, and team member preferences. This addresses the guidelines for error prevention and adaptivity in the “Guidelines for multimodal user interface design” by Reeves et al. (2004). Furthermore, unimpeded access to the system’s representation in the context of co-located team interaction is only possible if the whole team can view the multimodal BI&A system. Particularly during decision-making, perspectives of all team members need to be considered in the analysis and thus systems are required to support all team members in their transparent interaction (Dennis, 1996; Dennis et al., 2001). Therefore, we articulate the first DP:

**DP1:** *To improve team members’ transparent interaction with a BI&A system in co-located team interactions, integrate multimodal interaction capabilities on large interactive displays.*

In the context of adapting the surface structure, the most critical mechanisms are the affordances and the feedback the system provides. In order to address the issues of hidden affordances (I2), we propose to implement signifiers for the affordances of multimodal BI&A systems, in accordance with the theory of affordances (Norman, 2016). The crucial affordances of multimodal BI&A systems providing touch and speech as modalities are touching the system and speaking to the system. However, even though most of the displays

used in co-located team interactions integrate microphones, in conformance with I2, the affordance to speak to the system is not visible to the team members and lacks signifiers. Therefore, an approach to make speech perceptible is to provide signifiers to the team members. These signifiers create awareness for team members on what modalities are available for interacting with the multimodal BI&A system. Furthermore, teams need to understand how to properly interact with the multimodal BI&A system in order to increase transparent interaction. Therefore, the multimodal BI&A system should provide perceptual information on the basis of which teams can reinforce and, if necessary, modify their behavior. Deng et al. (2004) proposed to implement reactive feedback in CUI in order to assist users during the interaction. Using reactive feedback, the system's interpretation of the team members' speech interaction can be visualized for confirmation and missing information can be requested by the multimodal BI&A system. For example, after a complex speech interaction, team members should be able to understand whether the system invoked the correct functionalities or if the team members need to undo the last step and try again in a different way. Therefore, we articulate the second DP:

***DP2:** To improve team members' transparent interaction with a BI&A system in co-located team interactions, employ feedback and signaling affordances that clarify its interaction capabilities.*

To increase the representational fidelity (MR2) and to tackle Issue 3, the theory of effective use suggests adapting the representations of the system. In this cycle, we focus on the visual representations of the system and not on adapting the mapping of the database or the functionalities of the system, which is also part of representational fidelity. In order to achieve higher representational fidelity by adapting the visual representations using transparent interaction, direct manipulation of the visual representations is crucial. Direct manipulation has been shown to simplify the mapping between goals and actions by reducing the semantic and articulatory distance (Frohlich, 1993). Furthermore, Yi et al. (2007) proposed a set of interaction techniques for visual representations, which are independent of the modality used for facilitation. Combining these two concepts enables the user to utilize transparent interaction to adapt the representation of the system in order to maintain representational fidelity during decision making. Even when the problem statement or the information need shifts. Therefore, we articulate the third DP:

***DP3:** To support team members in obtaining faithful representations while using a multimodal BI&A system in co-located team interactions, enable direct manipulation of visual representations using common interaction techniques (e.g., selecting, filtering).*

### 3.4.3 Development

For our first design principle, we chose a Microsoft Surface Hub 2S to provide the touch and speech modality as well as the visualization of the system (Figure 3.3), as it provides a large interactive display to the team. Our multimodal BI&A system should be independent of specific BI&A systems used in teams. Therefore, we used a two-layer architecture. The first layer is responsible for the integration of the BI&A system and its corresponding data into the system. We used the SDK of Microsoft Power BI, which is the platform for BI&A mainly used in the case organization. However, the focus of our system is on the second layer, which is responsible for the interaction between teams and the BI&A system. To provide a CUI and to implement speech interaction into our BI&A system we used Microsoft’s Cognitive Services. This provides us the capability to perform speech-to-text analysis and the identification of intentions. The touch interactions were facilitated by JavaScript.



Figure 3.3: Multimodal BI&A System in Co-located Team Interactions at Industry Partner.

To instantiate the second design principle, we provide a signifier for the affordance of speaking to the system. Signifiers in the digital context can consist of but are not limited to buttons, labels, and sounds coming out of a speaker, or haptic vibration. We provide a signifier, which is constantly available to the team. Furthermore, in the post-study interviews from our interaction-elicitation study, participants stated that they would prefer a visual representation, indicating the availability of speech to the team. Therefore, we opted for a visual representation of the affordance that provides a visible signifier to the team at all times during the interaction. A microphone symbol on the large interactive display indicates the ability to speak to the system and tapping the symbol initiates speech



interaction.

Additionally, to provide reactive feedback to the team members (DP2), the system displays the interpretation of the speech input and explains the changes that were made based on that interpretation in the CUI. Figure 3.4 shows the feedback that the team receives after filtering the dashboard via speech. It includes the functionality invoked (“filter”) and what parameters were changed (i.e., “Planning Status”). This provides team members the ability to check whether the system understood them correctly or if they need to undo the last step and try again in a different way.

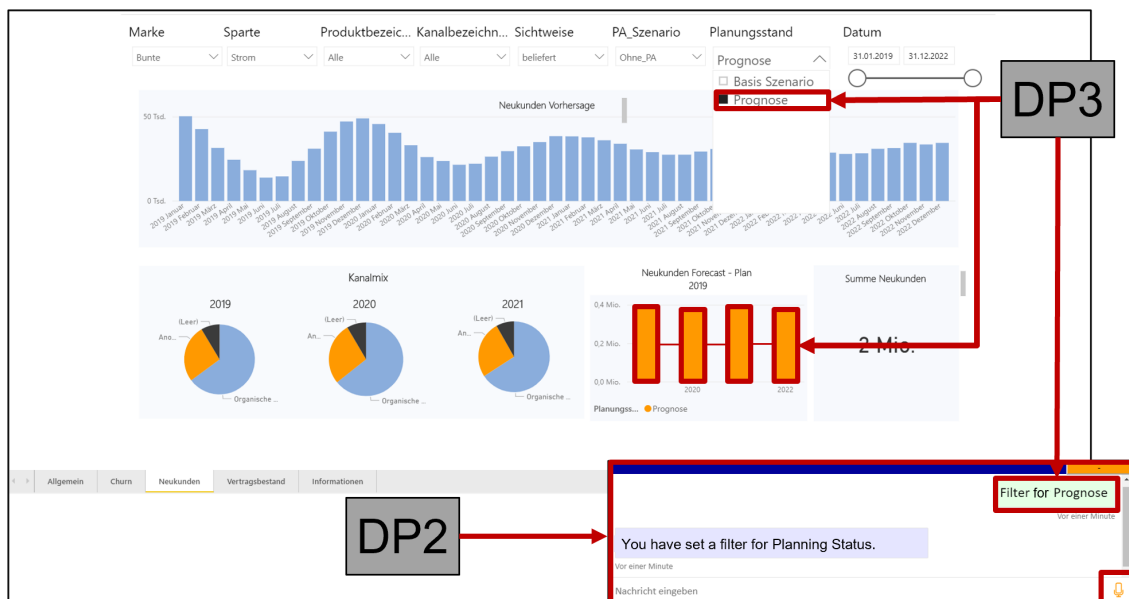


Figure 3.4: Instantiation of the second and third Design Principle.

Finally, to instantiate the third design principle, we used the results of the interaction-elicitation study to understand how users would like to perform the interaction techniques provided by Yi et al. (2007) with BI&A systems using touch and speech. To demonstrate the capabilities of a multimodal BI&A system and the implementation of our third design principle, we opted for filtering, selecting, reconfiguring visualizations, interacting with bookmarks, asking questions to the data (ex. What is the Product with the highest return in 2019?) as well as switching tabs as core functionalities provided by multimodal BI&A systems.

We selected for each modality and functionality the interaction that was proposed by most participants of the interaction-elicitation study. However, if multiple interactions had a high agreement for a modality and functionality and did not have a conflict, we integrated all. For example, for filtering and touch the integration of a drop-down menu has an agreement rate of 53%, and tapping on the depiction of a variable in a visualization has an

agreement rate of 40%. By providing both possibilities, we are able to provide interactions independent of team member preferences. Furthermore, we provide the possibility to choose between speech and touch at any step of the interaction. To continue the example of filter, as depicted in Figure 3.4, the team members are able to use speech (“Filter for Prognose”) or touch (Drop-Down Menu OR Tap on Variable in a Visualization) based on their current context and preferences.

### 3.4.4 Evaluation

In order to evaluate our multimodal BI&A system for co-located team interactions, we conducted confirmatory focus groups (Tremblay et al., 2010) with thirteen employees of our industry partner. The recorded focus group discussions were analyzed using a SWOT analysis. The results of the SWOT analysis in the context of each design principle are explained in more detail below.

DP	Strengths	DP	Weaknesses
1	S1. Modality can be selected based on context, team member characteristics, and task S2. Increased interactivity of co-located team interactions and involvement of all team members	1	W1. Missing trust in the reliability of speech and its adaptivity to the context and team member characteristics W2. Speech is seen to reduce the privacy of its users
2	S3. The team can concentrate on the communication and the task at hand		
3	S4. Limited knowledge about the functionality of the system necessary S5. Increased effectiveness of co-located team interactions due to ad-hoc analysis	3	W3. Onboarding needed to provide teams the ability to interact properly with the system
DP	Opportunities	DP	Threats
2	O1. Shifting the role of the BI&A system from an information provision platform towards becoming a key tool for teamwork	1	T1. Every team member can interact with the system which limits the control of a presenter and may lead to inefficient teamwork
3	O2. Increased effectiveness of co-located team interactions as additional information can be acquired based on more complex interactions with and drill-down into the data	3	T2. Simplification and automation of the functionality through more intuitive modalities can lead to unnoticed mistakes

Table 3.1: Summary of the SWOT Analysis.

First, participants stated that integrating multimodal interaction capabilities on large interactive displays (DP1) would help them in more effectively using the BI&A system. Particularly the interactivity and involvement of all team members in co-located team interactions was regarded as a major benefit. One participant stated: “When working with people who are experts in their field, everyone can interact from their standpoint and provide insights to the discussion” and that “the modalities in the system assist the interactivity of the meeting”. Furthermore, the first design principle was regarded as a

key strength of the multimodal BI&A system, “as it offers more possibilities in contrast to current systems and, therefore, enables us to choose the fitting modality. For example, if the noise in the room is too loud, the team members can switch to touch.” Moreover, the participants confirmed the insights from existing literature that “the combination of touch and speech is beneficial, as they are able to use speech for complex interactions and touch for simple and fast interactions.” However, one major weakness of the multimodal BI&A system, hindering effective use, is the missing trust in the reliability of speech processing and its adaptivity to the context and team member characteristics. The participants fear, that “the system would require an unnatural syntax for speech interaction” and that it cannot be adapted to the respective team members. Finally, participants mentioned that speech “decreases privacy, as everyone hears what you are working on.”

In general, the participants also liked the fact that the multimodal BI&A system employs feedback and signaling affordances that clarify its multimodal interaction capabilities (DP2). Especially, since in the context of decision making using BI&A systems, they fear that “through the ability to invoke complex functionalities with simple interactions, multimodal BI&A systems may misinterpret the intentions and provide the wrong information for the following discussion.”. Therefore, the reactive feedback would help them spot mistakes in the system’s interpretation of the interaction. However, during the discussion, team members may still miss the feedback provided by the system and use the information provided by an unfaithful representation to derive wrong insights. The participants additionally mentioned that enabling direct manipulation of visual representations using common multimodal interaction techniques helps them to “derive insights and configurations that else would be hard to find” and enables “ad-hoc analysis to answer questions arising in the discussion”, which supports the third design principle. They further stated that this would help them to improve their informed actions and would, therefore, facilitate the effective use of the multimodal BI&A system. As the system already provides transparent interaction (DP1 & DP2), in order to easily invoke complex functionalities of the system, the participants imagine the third design principle could provide “additional insights that would be overlooked in current meetings and would currently require the team to reschedule the meeting.” Moreover, “meetings and analysis, in general, could get faster.” However, according to the participants, providing the direct manipulation of the visual representations using speech might require “the user to learn the syntax beforehand.”

### 3.5 Discussion

While important decisions based on data are often made by cross-functional teams, current BI&A systems are primarily designed to support individual decision makers. To address this problem, we conduct a DSR project to design multimodal BI&A systems for co-located team interactions. Drawing on the theory of effective use, we examined how the combination of touch and speech modalities can facilitate the effective use of multimodal BI&A systems. In the first cycle of our DSR project, we proposed three design principles and instantiated them in our artifact. Subsequently, we conducted a confirmatory focus group evaluation with our industry partner. The results of our evaluation suggest that the combination of touch and speech for multimodal BI&A systems provides teams with additional possibilities to interact properly based on the team characteristics and context. However, the results also illustrate that the adaptivity of the speech interaction and an onboarding phase might further increase transparent interaction. Therefore, our DSR project provides valuable theoretical contributions and practical implications that we discuss in the following.

First, our research contributes to the body of design knowledge for multimodal BI&A systems in particular, and MUIs in general. The results of our evaluation suggest that the effective use of multimodal BI&A systems in co-located team interactions can be increased by offering touch and speech modalities on a large interactive display (DP1). This design principle enables team members to select modalities depending on their preferences and their current tasks, but they also have the ability to choose another modality if the context changes. Furthermore, the system creates awareness of possible modalities and provides reactive feedback (DP2), which allows team members to understand how to properly interact with the system and to spot mistakes in the system's interpretation (e.g., of their speech input). This reduces team member's worry to overlook possible mistakes of the system and using the wrong information to make decisions. Moreover, all design principles are key to provide the possibility to conduct ad-hoc analysis during co-located team interactions and to derive insights that would otherwise be overlooked. Therefore, these design principles can facilitate effective use of multimodal BI&A systems in co-located team interactions. Taken together, our research shows how the theory of effective use can be applied to improve the interaction of users with BI&A systems and advances our understanding of how users interact with MUIs.

Our evaluation also sheds light on additional design issues, which offer valuable starting points for a further improvement of multimodal BI&A systems. First, one weakness of multimodal BI&A systems derived in our evaluation indicates that the users need to be

able to perform adaptation actions on the speech interaction itself. If the speech interaction feels unnatural to team members or the system repeatedly fails to understand their speech input, teams are unlikely to use multimodal BI&A systems. To provide the system with the capabilities of adapting its speech interaction and to facilitate transparent interaction, T. J.-J. Li, Azaria, et al. (2017) propose to make multimodal systems “instructable”. This would imply that, if the multimodal BI&A system fails to understand the teams’ intention or input, team members are able to provide feedback back to the system. More specifically, teams could not only mark their input as interpreted incorrectly but also demonstrate the correct intention to the system using touch which the multimodal BI&A system provides due to its multimodal nature. For example, if a user wants to “Filter for the critical customers”, the system would not know what critical customers are. Therefore, the user can demonstrate for future cases using touch that critical customers have an order volume of higher than 1 million and a remaining contract term of 1 year. Therefore, providing MUIs with the ability to improve their recognition of intentions for a certain modality using input from another modality could facilitate the effective use of MUIs in general.

Furthermore, our results suggest that an initial onboarding could further facilitate effective use as it helps teams to learn how to interact properly with the system. Using multimodal BI&A systems during co-located team interactions allows everyone to interact with the system and contribute equally to the discussion and derivation of insights. However, this brings new challenges to the moderator of the discussion and the proper interaction with the system. Therefore, teams should be guided through the system in an onboarding phase to help them adapt their behavior to the system (e.g., how to formulate their questions in natural language) and show them how to get information using which modalities. Furthermore, during the use of the multimodal BI&A system, feedback should be provided based on the current interactions to help teams understand which information is further needed by the system, where the boundaries of the system are, and what modalities are available. Our reactive feedback (DP2) already provides feedback to teams on their current interactions. However, it does not provide explicit suggestions on how to interact with the system and how teams may adapt the multimodal BI&A systems in accordance with their team characteristics. This reactive feedback could be enhanced with further inquiries, suggestions, and insights in order to make the interaction between the team and the multimodal BI&A system not a one-way, but a two-way conversation.

Finally, there are also some limitations of work that should be considered. First, our multimodal BI&A system only implemented two modalities: touch and speech. Although

they are generally considered to be important modalities in HCI, future research could evaluate how other modalities (e.g., gaze and speech) complement each other and can be integrated into multimodal BI&A systems to facilitate effective use. Second, we instantiated our design principles on a large interactive display. However, the size, as well as the appearance of the interactive surface, may influence how people interact with our artifact. Consequently, future research could evaluate the influence of the type of device used for the provision of the artifact. Finally, we used a confirmatory focus group to perform a qualitative evaluation of the impact of the software artifact on the facilitation of effective use. Although we argue that this approach is appropriate given the innovative nature of multimodal BI&A systems, further research using quantitative evaluation methods is needed. Therefore, a quantitative field-based study could provide additional insights into the impact of multimodal BI&A systems on their effective use in co-located team interaction.

### **3.6 Conclusion**

This paper reports the results of the first cycle of a DSR project focusing on the design of multimodal BI&A systems for co-located team interactions. Overall, our DSR project contributes with design knowledge that can be applied to facilitate the effective use of multimodal BI&A systems in co-located team interactions. In particular, we contribute with three design principles in order to provide a multimodal BI&A system to teams consisting of user-defined multimodal interactions as well as feedback and signaling affordances for speech interaction. The design principles were derived based on the theory of effective use, guidelines for the design of MUIs, and empirical insights of an interaction-elicitation study. We instantiated our design principles and developed a running software artifact based on state-of-the-art technology. Finally, our evaluation of the software artifact in the form of a confirmatory focus group with an industry partner demonstrates the potential of our proposed software artifact.

# 4. Study III: ONYX: Assisting Users in Teaching Natural Language Interfaces Through Multi-Modal Interactive Task Learning<sup>4</sup>

## 4.1 Introduction

With the recent advances in natural language processing, users are increasingly provided with the ability to interact through natural language (NL) with their smartphones (e.g., Siri; Google Assistant), the web (e.g., FireFox Voice (Cambre et al., 2021)) or specific applications like data visualization tools (e.g., Power BI Q&A; Tableau’s Ask Data). While these NL interfaces (NLI) initially pursued a one-size-fits-all design, developers soon realized that different users or contexts require supporting more personalized NL inputs (Grudin & Jacques, 2019) since users are otherwise often quitting those NLI instead of retrying after they face breakdowns (C.-H. Li et al., 2020). Therefore, users are able to teach some existing NLI how to handle new NL inputs in limited ways. For example, NLI can be taught by end users to perform custom procedures combining multiple functionalities of the underlying system (*procedural knowledge*), such as the custom commands provided by Apple’s Siri (Apple, 2022). Another example is how end users can provide synonyms to existing concepts or define new concepts (*declarative knowledge*) to improve the NLI’s understanding (e.g., what constitutes a “crucial” customer in Power BI Q&A (Microsoft, 2022a)).

To empower end users in extending the previously mentioned NLI, end user development techniques, such as visual programming (e.g., Siri (Apple, 2022); Google Assistant (Google, 2019)) or simpler form-filling techniques (e.g., Power BI Q&A (Microsoft, 2022a)), are utilized to lower the barriers. While visual programming is currently the key approach, it has been shown over decades that creating programs with visual programming languages is still relatively complex for most users (Booth & Stumpf, 2013; Eagan & Stasko, 2008; João et al., 2019; Myers, Ko, Scaffidi, et al., 2017). First, end users struggle to select the correct visual programming blocks from the extensive options to create their intended program (Booth & Stumpf, 2013). Second, detecting and fixing errors in their visual programs is still

---

<sup>4</sup>This chapter is based on the following studies which are published or in work: Ruoff, Myers, et al. (2021) and Ruoff, Myers, et al. (2023).

a stumbling block (Ko et al., 2006; Myers, Ko, Scaffidi, et al., 2017). A promising approach to address these challenges is interactive task learning (ITL). ITL-based systems do not require end users to search for correct visual programming blocks or familiarize themselves with a programming language. Instead, ITL-based systems combine NL programming and programming-by-demonstration to learn from multi-modal user demonstrations of the task in the actual system (Laird et al., 2017).

A fundamental challenge in ITL-based systems is their ability to not only learn macro recordings of specific user demonstrations but to be able to *generalize* the derived knowledge to support users in performing *similar* tasks. While previous research on ITL has improved the NLI’s ability to generalize *declarative* knowledge (i.e., concepts such as *hot* and *cold* (T. J.-J. Li, Radensky, et al., 2019) and values such as customer names (Krosnick & Oney, 2022; Leshed et al., 2008)) and utilize previously defined declarative knowledge during future demonstrations (T. J.-J. Li, Radensky, et al., 2019), research opportunities still remain in generalizing and reusing *procedural* knowledge. Improving the generalization of procedural knowledge is crucial since ambiguities contained in demonstrations through direct manipulation either lead to a narrow understanding of the demonstrated task or require many demonstrations of the same task (Zong et al., 2021). Additionally, end users want to build on existing user-defined procedures and therefore need insight into which parts of a new NL input the NLI understands and can already handle.

Therefore, we introduce *ONYX*, an intelligent agent that is able to learn both procedural and declarative knowledge through ITL. Essential capabilities of *ONYX* are its ability to generalize procedural knowledge, and its ability to provide insight into existing declarative and procedural knowledge during the teaching of new NL inputs. *ONYX* learns both from users’ direct manipulations (programming-by-demonstration) and NL inputs (NL programming) after encountering a new NL input. Three key novel aspects of *ONYX*’s design are the (i) *suggestions*, (ii) *follow-up questions*, and (iii) *guidance through visual and textual aids* provided by *ONYX*. First, through *suggestions* *ONYX* describes how it can handle new NL inputs based on previously learned concepts and user-defined procedures. Second, *follow-up questions* are utilized to accurately abstract and generalize procedural knowledge by clarifying possible ambiguities in direct manipulation demonstrations by end users. Third, to provide users with *guidance* at crucial stages of the demonstration process, *ONYX* provides visual and textual aids, such as connecting the concepts *ONYX* understood in the new NL input to their associated visual elements in the GUI (i.e., buttons and data fields).



We demonstrate *ONYX*'s capabilities in a custom-built data visualization tool since data visualization tools (1) facilitate complex actions which possibly exhibit ambiguities (Zong et al., 2021), (2) have a wide range of NL inputs users want to utilize while little labeled NL input exists to train these NLI to understand this variety (Srinivasan, Nyapathy, et al., 2021), and (3) the expressiveness of current learning approaches of new NL inputs is limited (e.g., (Microsoft, 2022a; Tableau, 2019)). We integrated a dataset about the COVID-19 pandemic (Dong et al., 2020) since end users are familiar with this data and the possible insights they might want to derive.

In developing *ONYX*, we took a user-centered design approach using iterative participatory design with 10 participants to explore what issues end users face in NLI with ITL capabilities and to derive new designs for *ONYX* to address these challenges. Over the course of four months, we performed six iterations of *ONYX* with 2 - 4 participants per iteration. Each participant took part in two consecutive iterations so we could get feedback from them both when they have minimal knowledge of *ONYX* in their initial iteration, and in the subsequent iteration where they have a deeper understanding.

After building *ONYX* based on that feedback, we performed an online summative evaluation with 42 participants. Further, a think-aloud study with 5 participants provided evidence for the effectiveness and usability of our final design and offered additional details on how users utilize our novel features.

To summarize, the contributions of this paper are:

1. A multi-modal ITL approach to enhance existing NLI iteratively through programming-by-demonstration and NL programming, with the following major advantages:
  - a) **Suggestions provided during the demonstration process** give users insight into how the ITL agent could handle new NL inputs based on previously learned concepts or user-defined procedures and enable users to focus on concepts and procedures that are currently unknown to the ITL agent.
  - b) The ITL agent uses **follow-up questions** to clarify ambiguities in the user's direct manipulation demonstrations to facilitate the abstraction and generalizability of the derived procedural knowledge.
  - c) Users receive a **display of the ITL agent's understanding** of the new NL input grounded in known concepts and visually tied to GUI elements to provide users a deeper understanding of the ITL agent's declarative knowledge.

2. The *ONYX* system: an implementation of the aforementioned approach, along with a formative study ( $n = 10$ ) highlighting issues end users face in existing NLI systems with ITL capabilities, along with an online experiment ( $n = 42$ ) and think-aloud study ( $n = 5$ ) summatively evaluating its effectiveness and usability. The final evaluation shows that users provided with *ONYX*'s suggestions and follow-up questions achieved significantly ( $p < 0.001$ ) higher accuracy in teaching new NL inputs (median: 93.3%) in contrast to those without (median: 73.3%).

## 4.2 Related Work

In this section, we discuss prior work in three areas related to our *ONYX* system: ITL in general, NLI systems with learning capabilities, and NLI systems for data visualization tools.

### 4.2.1 Learning Tasks through Demonstrations

Interactive Task Learning enables end users to automate tasks without requiring them to write code. Instead, users demonstrate the actions required to complete the task, similar to how they would perform the task without ITL. These demonstrations are then utilized to extract the procedural (i.e., relevant actions) and declarative knowledge (i.e., concepts and values) to create a script. Extensive possibilities of tracking and learning from demonstrations (e.g., through APIs (T. J.-J. Li, Azaria, et al., 2017; Maués & Barbosa, 2013) or through internal data of the system (Allen et al., 2007; Krosnick & Oney, 2022; Vaithilingam & Guo, 2019)) have been used in diverse application areas including the creation of GUIs (Myers, McDaniel, et al., 1993; Vaithilingam & Guo, 2019) or information visualizations (Zong et al., 2021), and the automatization of tasks on mobile phones (T. J.-J. Li, Azaria, et al., 2017; Sereshkeh et al., 2020), on the web (Fischer et al., 2021; Leshed et al., 2008), or with robots (Suddrey et al., 2022; Thomason et al., 2015).

A major challenge in all application areas is the generalization of the underlying procedural and declarative knowledge to support users in performing *similar* tasks to those demonstrated. Existing systems focus on generalizing the declarative knowledge by parameterizing the utilized values and concepts either by involving users during the demonstration process (e.g., (Fischer et al., 2021; Krosnick & Oney, 2022)) or through the system automatically based on internal knowledge (e.g., underlying datasets (Leshed et al., 2008)). However, besides generalizing declarative knowledge, it is also crucial to generalize procedural knowledge. Most importantly, direct manipulations, the main source currently for demonstrations in ITL-based systems, only communicate *what* a user does and not *why* or *how* to perform

these actions in varying contexts. Therefore, possible ambiguities in the direct manipulation need to be clarified to accurately generalize the scripts by deriving the underlying procedural knowledge (T. J.-J. Li, Radensky, et al., 2019; Zong et al., 2021). APPINITE (T. J.-J. Li, Labutov, et al., 2018) aims to address this challenge by involving users to clarify ambiguities. Specifically, after APPINITE detects an ambiguity it opens an empty text input in which users can describe their intention through NL in detail. However, solely relying on end users to clarify the ambiguities is risky as end users have problems expressing in NL the correct conditions without assistance from the system (X. Chen et al., 2018). Hence, we investigate how end users can be assisted by *ONYX* in clarifying ambiguities to abstract direct manipulation demonstrations by answering follow-up questions asked by *ONYX* that only require users to choose between possible interpretations of that ambiguity.

#### 4.2.2 Natural Language Interfaces with Learning Capabilities

Learning new NL inputs is an important application for ITL since NL allows end users to easily perform complex tasks (i.e., conditional tasks (T. J.-J. Li, Radensky, et al., 2019)), access infrequently used actions (Y.-S. Kim et al., 2019), or provide more natural phrasings for existing NL inputs (S. I. Wang et al., 2017).

To assist end users in teaching new NL inputs, end users require insight into which parts (e.g., words, phrases, and concepts) of a new NL input the NLI understands and can already handle. Most NLIs with learning capabilities, however, do not provide insights into existing declarative (i.e., concepts) and procedural (i.e., actions) knowledge during the demonstration process (e.g., (Azaria, Srivastava, et al., 2020; Y.-S. Kim et al., 2019; T. J.-J. Li, Azaria, et al., 2017; S. I. Wang et al., 2017)). In contrast, PUMICE (T. J.-J. Li, Radensky, et al., 2019) provides insight into previously learned declarative knowledge like the concepts *hot* or *cold* during the demonstration process through a multi-turn conversation. However, PUMICE’s multi-turn conversation does not provide insights into existing procedural knowledge (i.e., user-defined procedures) that the system might have learned for the parts of the articulated NL input that is new to the NLI. Providing insight into procedural knowledge differs, as while PUMICE has to clarify a maximum of two concepts for its boolean or value conditions, *ONYX* has to provide insights into procedural knowledge that can consist of a previously unknown number of actions and hence needs to be handled differently. Similarly to PUMICE, AutoVCI (Pan et al., 2022) only provides the means to utilize existing procedural knowledge for complete NL inputs to teach synonyms for these. In contrast, *ONYX* additionally allows the combination of several previous procedural knowledge to inform the teaching of new NL inputs.

To address these issues, we investigate how to provide end users insights into how *ONYX* could handle new NL inputs based on previously learned concepts (declarative knowledge) *and* user-defined procedures (procedural knowledge).

### 4.2.3 Natural Language Interfaces for Data Visualization Tools

NLIs have been increasingly utilized to assist users in analyzing and exploring data in data visualization tools (e.g., (Gao et al., 2015; Y.-H. Kim et al., 2021; Luo et al., 2022; Setlur, Battersby, et al., 2016; Srinivasan, Lee, et al., 2020)). Previous studies showed that extending data visualization tools through NLIs particularly helps users perform tasks that would otherwise require multiple adjustments in the GUI (Setlur, Battersby, et al., 2016) or complex filter settings (Y.-H. Kim et al., 2021). However, while the variety of use cases has grown over the last decades, a large gulf between user expectations and the capabilities of NLIs still exists (Tory & Setlur, 2019).

The major challenge of NLIs in data visualization tools is that users expect the NLI to understand a wide range of NL inputs to create and adapt the data visualizations (Srinivasan, Nyapathy, et al., 2021), and expect the NLIs to have a deep understanding of the context and dataset the users are currently working with (Tory & Setlur, 2019). When current NLIs for data visualization tools fail to understand the users' goal of an NL input they either prompt users to retry their action differently (Y.-H. Kim et al., 2021; Srinivasan, Lee, et al., 2020) or involve the user to clarify possible misunderstandings in the NL input (Gao et al., 2015; Narechania, Fourney, et al., 2021; Setlur, Battersby, et al., 2016). While the latter already improves the user experience of NLIs for data visualization tools (Gao et al., 2015), existing systems do not learn from these demonstrations for future interactions except for simple form-filling techniques (e.g., (Microsoft, 2022a)) and therefore continuously require end users to clarify the same misunderstandings.

Hence, we investigate *ONYX* in the context of data visualization tools to showcase how an NLI can learn from demonstrations to increasingly improve its coverage of the NL inputs users want to utilize.

## 4.3 Formative Study & Design Goals

We took a user-centered approach (Myers, Ko, LaToza, et al., 2016) for designing an NLI with ITL capabilities. Using participatory design, we studied how to address issues end users face in existing NLIs with ITL capabilities. We recruited 10 participants (8 males; 2 females;  $M = 27.5$  years  $SD = 11.4$ ) over the course of 4 months. Each participant took

part in two sessions one week apart (about 1 hour per session). In each session, participants first completed target replication tasks and afterward an open-ended data exploration to additionally understand how *ONYX* impacts users' analytic flow. We recorded their think-aloud statements, coded them, and derived (among others) the following insights.

We summarize the most relevant insights from the formative study for the design of *ONYX* next. From these insights, we distilled six design goals (**DGs**) for enabling effective ITL for NLI.

#### 4.3.1 Understanding of the NLI's Existing Knowledge

In the earlier stage of the participatory design process, while users generally understood *how* they could demonstrate the meaning of an NL input to the system due to an initial introduction by the supervising researcher and a textual introduction by the ITL agent after the breakdown, they did not know *what* they need to teach *ONYX*. Especially because *ONYX*'s error messages were generic, like in many other NLIs (Ashktorab et al., 2019). Participants, therefore, tried to derive the NLI's existing understanding of their NL input during the demonstration process in a trial-and-error approach by changing the NL input incrementally and paying attention to whether *ONYX* understands this adapted NL input as part of its NL programming capabilities. However, this caused significant disruptions to their analytic flow.

**DG1. Be specific about what parts of the NL input the NLI understood and did not understand.**

Participants utilized this trial-and-error approach to get insight into both the (i) procedural and (ii) declarative knowledge the NLI possesses. First, when *ONYX* failed to understand a joined NL input (e.g., ☹ Remove Deaths **and** focus on Cases or ☹ Show states **with** more than 1 million cases) some participants would enter the parts of the NL input during the demonstration separately to check whether *ONYX* possesses procedural knowledge for parts of the NL input.

**DG2. Provide suggestions based on the parts of the NL input the NLI understood.**

Second, participants were often unsure whether *ONYX* failed due to missing declarative or procedural knowledge. For example, in the NL input ☹ Give me TX *ONYX* could either lack procedural knowledge for how to handle **Give me [something]**, lack declarative

knowledge that **TX** is an abbreviation for **Texas** or both. Hence, participants were comparing parts of the NL inputs with labels in the GUI elements to reassure themselves which concepts they think *ONYX* understood, and which it did not.

**DG3. Ground the NLI’s declarative knowledge for parts of the NL input through visual and textual aids in the GUI elements.**

### 4.3.2 Ambiguous Direct Manipulation Demonstrations

Participants mostly utilized direct manipulations for their demonstrations, similar to users in other ITL-based systems (T. J.-J. Li, Azaria, et al., 2017). Hence, abstracting and generalizing the direct manipulation demonstrations is especially important to derive an accurate script.

However, participants realized that when ambiguities occur in direct manipulation interactions, *ONYX* occasionally chose the wrong interpretation of their demonstration. In our initial version, participants appreciated the ability to directly edit a visual representation of the script during the demonstration as this simplified changes and deletions in the script if they noticed misunderstandings. Participants additionally acquired an understanding of the current interpretation of their actions by *ONYX* through this visual representation.

**DG4. Enable users to edit the ITL agent’s understanding of the demonstrations performed.**

However, they highlighted two shortcomings of solely relying on this visual representation of the script for addressing misunderstandings. First, if users did not notice the incorrect interpretation of the ambiguous direct manipulation they would not try to clarify. Second, if users did notice the incorrect interpretation, they would only realize that the interpretation is incorrect, but not what caused this unwanted behavior. Hence, participants specified the need for *ONYX* to actively notify users of ambiguities and subsequently describe the possible interpretations for users to choose from.

**DG5. Address ambiguities in direct manipulation demonstrations through follow-up questions.**

### 4.3.3 Design of Assistance

After addressing the challenges mentioned above by assisting users through *suggestions*, *follow-up questions*, and additional *visual and textual aids*, the participants highlighted the

positive effect of these features on lowering the disruption of participants’ analytic flow. However, participants additionally highlighted two design trade-offs with the design of this assistance:

**Timing of Assistance:** While asynchronously providing assistance would minimize disruptions, synchronous assistance helps users better understand the information *ONYX* requires to learn how to accurately handle the new NL input.

**Modality of Assistance:** While visually presenting follow-up questions as GUI elements helps users make fast decisions, users can better understand the ambiguity *ONYX* is trying to address when using text.

In our context, we opted to have *ONYX* use synchronous assistance and textual follow-up questions. After additional iterations of participatory design, participants highlighted that a deep understanding of the information *ONYX* requires is more urgent than being incrementally faster. Furthermore, the disruptions of synchronous assistance were minor due to additional visual and textual aids to guide end users.

**DG6. Guide users’ attention during assistance through visual and textual aids to minimize the disruption caused by the interruption.**

## 4.4 ONYX

In this section, we describe how *ONYX* incorporates the previously articulated design goals. Specifically, we describe an example scenario that illustrates how users can personalize the NLI. Subsequently, we detail how *ONYX*: (i) learns from multi-modal demonstrations, (ii) derives suggestions, (iii) identifies ambiguities in the demonstrated actions, (iv) provides visual and textual aids, and (v) generalizes the derived scripts.

### 4.4.1 Example Scenario

Mikki, a student from North Carolina recently moved to Pennsylvania for her studies. She aims to utilize *ONYX*, which integrates a dataset (Dong et al., 2020) about the COVID-19 pandemic in the United States of America. We will utilize this dataset throughout this paper to provide consistency.

To get an understanding of the course of the pandemic in her home state, Mikki starts with a scatterplot visualization that depicts deaths on the y-axis, fully vaccinated on the x-axis, and a color encoding based on the dates since the beginning of the pandemic filtered for North Carolina (see Figure 4.1 [A](#)). To focus on some points of interest she enters the

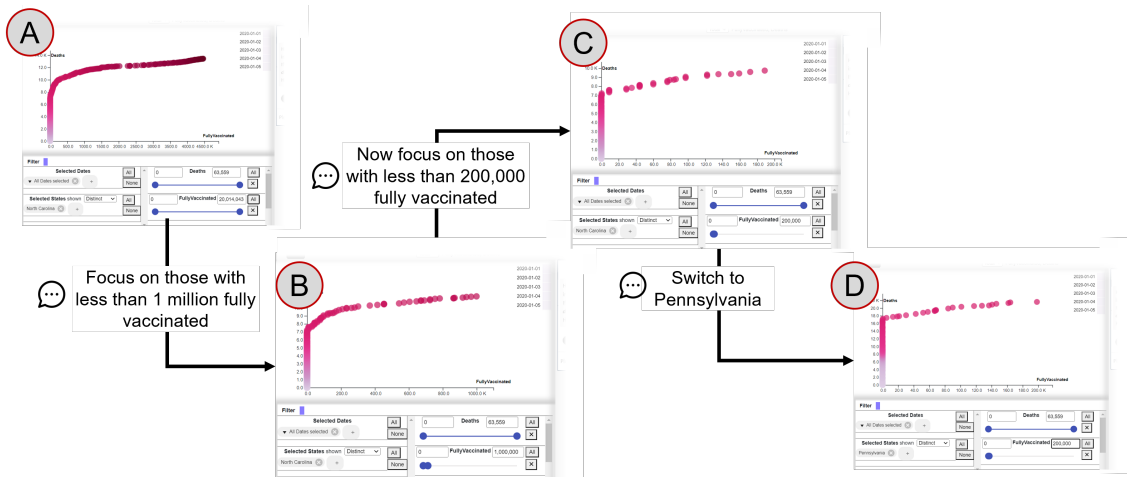


Figure 4.1: Data Visualizations and NL Inputs utilized in the Scenario.

NL input ☺ Focus on those with **less than 1 million fully vaccinated** into the NLI. While *ONYX* would understand other NL inputs for this goal, it does not yet understand how to handle the NL input that Mikki used. However, Mikki would prefer to use her own NL input as it feels more natural to her. After deciding to teach *ONYX*, *ONYX* provides feedback in the NLI and provides in its training interface an explanation of *ONYX*'s understanding of the NL input (**DG1 & DG3**). Additionally, *ONYX* provides suggestions on how to handle parts of the NL input for Mikki to reuse and build on (**DG2**). In this case, *ONYX* reuses procedural knowledge for **less than 1 million fully vaccinated**. The changes are directly carried out in the data visualization tool (see Figure 5.1 **C**) and added to the *current understanding* section of the UI (see Figure 5.1 **E3**). *ONYX* further gives a brief explanation in the NLI on which part of the NL input the suggestion is based on (see Figure 5.1 **D**). If the suggestion would not fit Mikki's understanding, she could refine or delete the suggestion (see Figure 5.1 **E3**). However, since the suggestion fits her understanding, she checks the visual representation of the script and finishes the demonstration as the visual representation of the script already fits her understanding of the NL input. *ONYX* immediately responds that this NL input is now available to her.

She wants to focus on the vaccination roll-out and therefore enters ☺ Now focus on those **with less than 200,000 fully vaccinated**. Due to her previous demonstration, *ONYX* knows how to handle this NL input and adapts the data visualization accordingly (see Figure 4.1 **C**).

Now Mikki wants to know if Pennsylvania has a similar trend. Hence, Mikki enters ☺ Switch to **Pennsylvania**. Again, *ONYX* does not know how to handle the NL input and therefore initiates the demonstration mode (see Figure 4.3 **A**). After deciding to



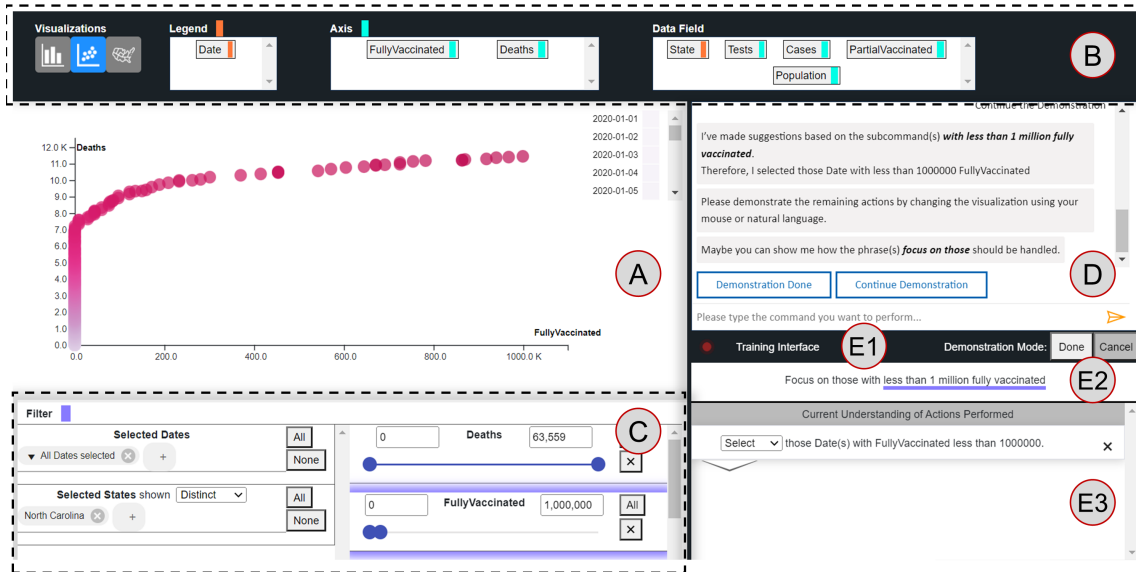


Figure 4.2: User Interface of the Data Visualization Tool with integrated ITL-based NLI during the Demonstration Process. (A) Visualization canvas, (B) Buttons to adapt chart type and encodings, (C) Filter pane to provide constraints, (D) NLI providing text input and feedback, (E1) Signalling indicator and buttons to finish the demonstration mode, (E2) NL input to be demonstrated, (E3) Visual Representation of the Script.

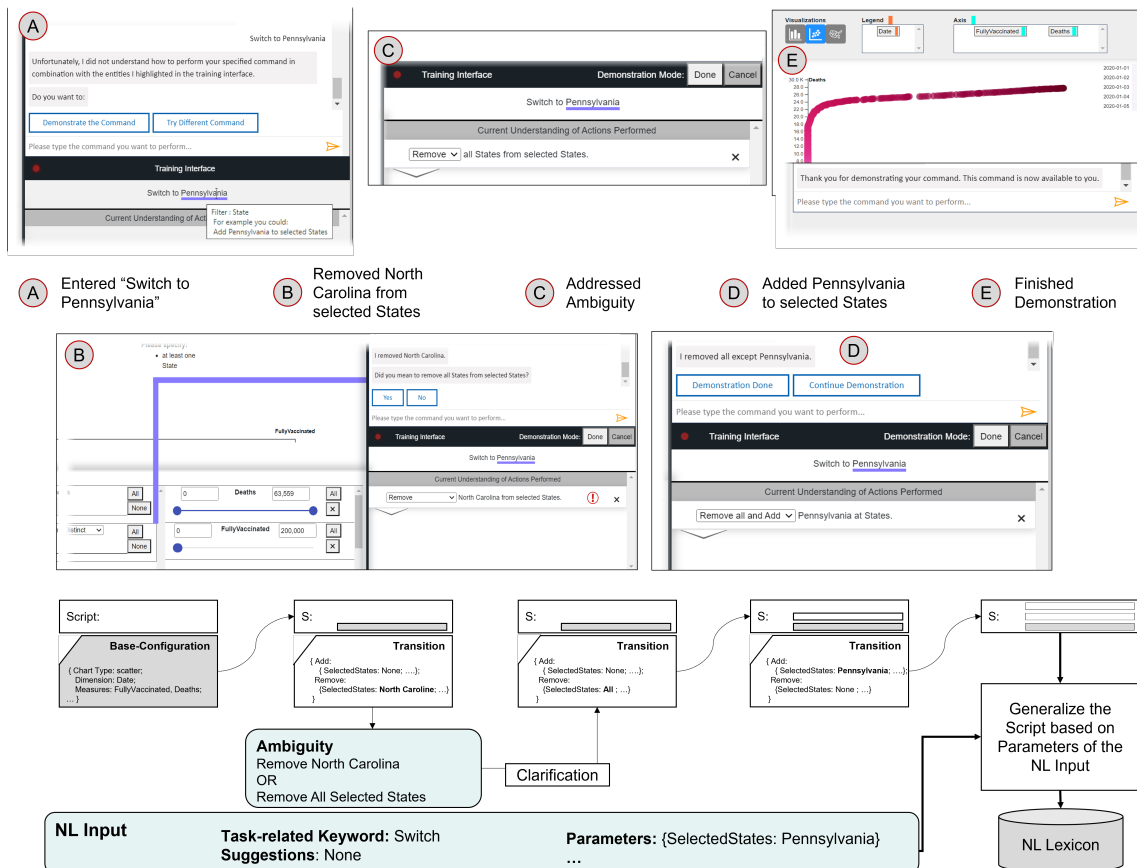


Figure 4.3: Progression of the User Interface, the Script, and the ITL Agent during the Training for the NL Input "Switch to Pennsylvania".

demonstrate the NL input, Mikki directly removes **North Carolina** from the current selection of states displayed in the visualization. This triggers an ambiguity in the back-end of *ONYX* since *ONYX* is unsure if Mikki wanted to (i) specifically delete **North Carolina** or (ii) if *all* states should be removed from the selected states. This is crucial since Mikki might later use this NL input again with different or even multiple states selected. Hence, *ONYX* prompts a follow-up question in the NLI which asks whether Mikki wanted to remove all states from selected states as a yes/no question (**DG5 & DG6**) (see Figure 4.3 **B**). After reading the question, she quickly selects *Yes* as the appropriate answer and *ONYX* adapts its understanding in response (see Figure 4.3 **C**). Mikki performs the rest of the actions and finishes the demonstration mode to which *ONYX* responds that the action is now available to her (see Figure 4.3 **E**).

Mikki might now use similar NL inputs during further exploration with different parameters in the NL input such as different values in the numeric relation of the first NL input or different states instead of Pennsylvania in the second NL input.

## 4.4.2 Key Design Features

### 4.4.2.1 Learning from Multi-Modal User Demonstrations

To allow users to demonstrate actions using both direct manipulation and NL input, *ONYX* utilizes programming-by-demonstration and NL programming to translate both types of interactions into a common script. Users are able to adapt the script by adapting its visual representation (see Figure 5.1 **E3**) by selecting a different keyword in the display of an action to correct minor misunderstandings (**DG4**) (see Figure 4.4 **B**). Major misunderstandings can be addressed by deleting the incorrect parts of the script and redoing them. To continuously check whether the goal of the NL input might be attained, *ONYX* compares the concepts the user utilized in the script with the ones extracted as parameters from the initial NL input. If all parameters have been utilized, then *ONYX* inquires whether users completed their demonstration (see Figure 4.4 **A**).

### 4.4.2.2 Suggestions

*ONYX* provides suggestions based on procedural knowledge associated with (i) *existing user-defined procedures* and (ii) *known concepts utilized in the NL input*. First, suggestions can be triggered if parts of the NL input are similar to known NL inputs and fulfill the

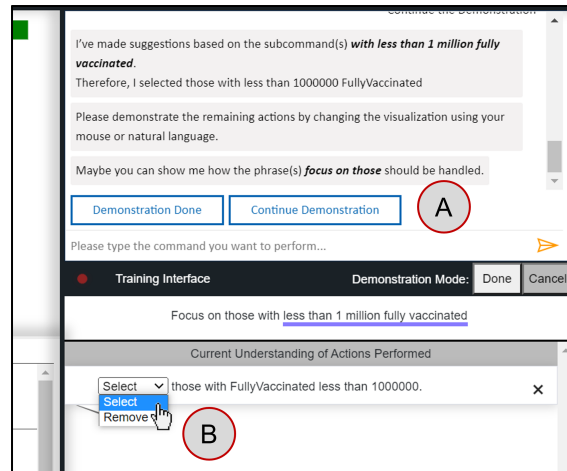


Figure 4.4: Screenshot of the UI when *ONYX* detects a possible Goal Attainment during the Demonstration Process.

requirements of its associated user-defined procedures (see Figure 4.5 (B)). Second, to provide insightful suggestions when only a few user-defined procedures exist, *ONYX* additionally provides suggestions based on known concepts. Only numeric filters and chart types can trigger suggestions (see Figure 4.5 (A)) as they are associated with one specific type of procedural knowledge. Concepts such as dimensions, on the other hand, can be associated with multiple (e.g., filtering, selecting/removing as axis, etc.) and hence could lead to incorrect suggestions.

After receiving the NL input from the NL Parser and its associated information (i.e., dependency tree, numeric relations, and named entities), *ONYX* searches in a depth-first manner for parts of the NL input *ONYX* knows how to handle (**DG2**) (see Figure 4.5). It saves the suggestions together with neighboring parts of the NL input that need to be demonstrated. *ONYX* provides these suggestions in the order they occur in the NL input. If not actively requested by users, *ONYX* only provides the next suggestion after the parameters contained in the neighboring parts of the NL input are utilized in the users' demonstrations.

*ONYX* makes a suggestion by performing the associated actions directly in the data visualization tool and updates the script and the associated visual representation (see Figure 4.4 (B)). Furthermore, in the NLI, *ONYX* provides a short explanation of which part of the NL input its suggestion is based on, what actions it performed, and which parts of the NL input associated with the suggestion it did not understand (see Figure 4.4 (A)).

In the example depicted in Figure 4.5, *ONYX* provides the first suggestion regarding the numeric relation (see Figure 4.5 (A)). It requests the user to demonstrate what the component `Show me all states` means because it recognized that these components are

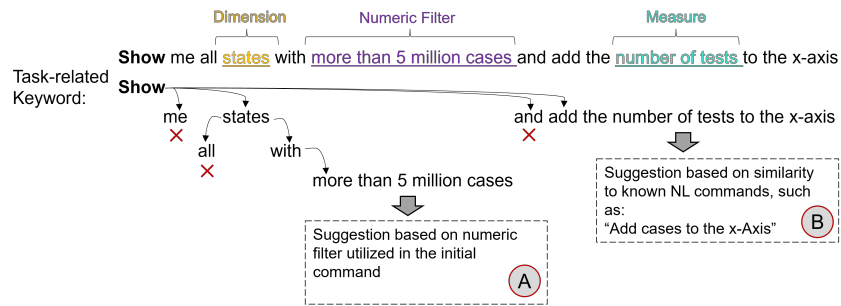


Figure 4.5: A Sample Sentence tagged and processed by *ONYX* to derive Suggestions. The Words of the Sentence are connected with directed Arrows based on their Dependency Structure.

connected. The suggestion in Fig. 4.5 (B) is only provided after *ONYX* recognizes that the user demonstrated something connected with the concept “states” (i.e., selecting a state filter). Users can also actively request additional suggestions by utilizing the NLI (e.g.,  or

#### 4.4.2.3 Follow-Up Questions

Follow-up questions are used in *ONYX* to abstract the meaning of low-level direct manipulations to a higher-level understanding by addressing possible ambiguities. To detect these ambiguities in the users’ demonstrations, *ONYX* utilizes conditions based on the parameters extracted from the NL input, the current script, and the demonstration performed (DG5). The conditions integrated into the current instantiation of *ONYX* are able to detect (i) *direct manipulation demonstration ambiguities*, such as whether users wanted to remove the specific state manipulated or all states if the selected states are empty afterward and (ii) *language ambiguities*, such as if states in  refers to the filter or also to the legend of the data visualization.

*Direct manipulation demonstration ambiguities* are only triggered if *ONYX* decides that it can not clarify the ambiguity on its own by utilizing information from the articulated NL input to preserve users from unnecessary interruptions. For example, when *ONYX* is unsure whether users only want to remove a specific state or all states, *ONYX* assumes the former is the correct interpretation without involving users if the removed state was utilized in the NL input.

If *ONYX* decides that the user is required to abstract the meaning of the direct manipulation demonstration, then *ONYX* directly asks the follow-up question after one of the conditions is triggered (synchronous assistance) (see Figure 4.6 (C)). Users can directly address the

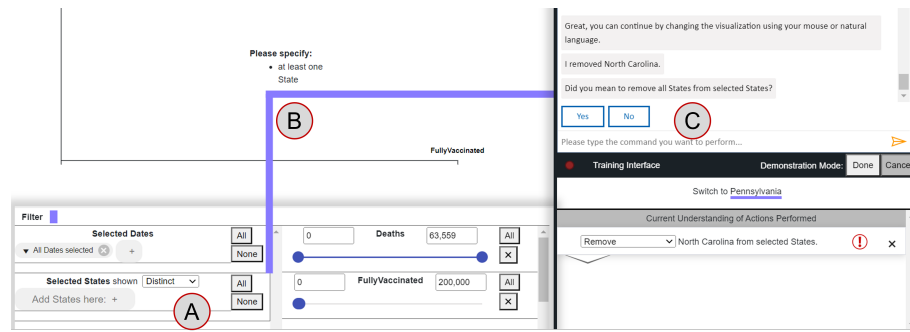


Figure 4.6: Follow-Up Question, after *ONYX* detects a Direct Manipulation Ambiguity in the last User Action.

ambiguity by selecting Yes or No in the NLI or they can continue the demonstration process and address the ambiguity in the visual representation of the script. If multiple conditions for ambiguities are triggered, then *ONYX* prompts the next follow-up question only after the previous one is clarified to avoid information overload.

#### 4.4.2.4 Visual and Textual Aids

**Aids for the NL Input.** To provide users at the beginning of the demonstration process an understanding of why the NLI failed, *ONYX* utilizes the extracted procedural and declarative knowledge to highlight the parts it did understand (**DG1**). Users can request a visual and textual aid by hovering over the underlined named entities (i.e., measures, dimensions, and categorical filters) and numeric relations of the NL input (see Figure 4.7 **B**). Upon hovering, *ONYX* provides a short explanation of its declarative knowledge (e.g., that it recognizes it as a filter) and exemplary actions based on the current context of the data visualization tool (procedural knowledge) as a textual aid. Furthermore, *ONYX* provides visual aids to ground these concepts in the GUI by highlighting the corresponding GUI elements, such as filter panes or labels (see Figure 4.7 **A**) (**DG3**).

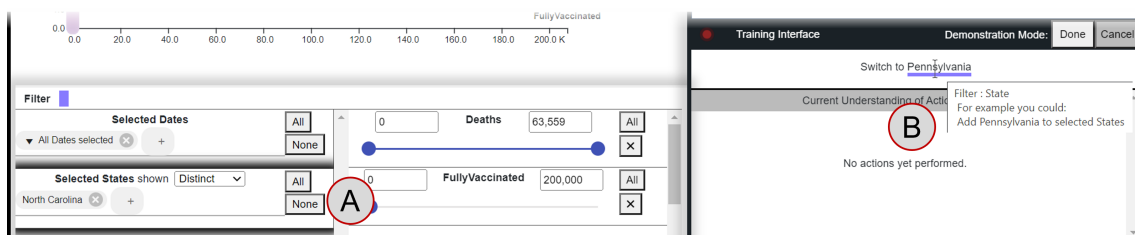


Figure 4.7: UI when Users hover over underlined Parameters during the Demonstration Process, with highlighted States Filter at **A**.

**Aids for Follow-up Questions.** Visual aids for follow-up questions are requested by users either by hovering over the follow-up question in the NLI (see Figure 4.6 **C**) or by hovering

over the GUI element that triggered the ambiguity (see Figure 4.6 **A**). *ONYX* guides the attention of users by connecting the GUI element associated with the ambiguity to the follow-up question with a color-coded line (see Figure 4.6 **B**) (**DG6**).

#### 4.4.2.5 Generalization.

*ONYX* is able to learn knowledge on three levels of generalizability by utilizing the named entities and numeric relations to parameterize both the derived script and the associated NL input. First, verbs (e.g., **Giving** [something]/ **Focusing on** [something]) and their connected procedural knowledge are generalized across various datasets and are only dependent on the functionality provided by the GUI. Second, *ONYX* links named entities (e.g., States) and relations to abstract concepts specific to data visualizations (e.g., Dimensions, Measures) (see Figure 4.5). The ITL agent only utilizes the abstract concept for deriving the possible values for parameters in the NL input and is, hence, able to generalize NL inputs across different datasets. However, currently, a JSON-File must be changed manually that maps data fields in the data set to their abstract concepts (e.g., `{'Measures': ['Deaths',...], 'Dimensions': ['States',...]}`). Third, *ONYX* is able to learn a narrow understanding of concepts that are connected to a specific data field (e.g., **southern** states, **soaring** number of deaths) and therefore would, e.g., support asking about southern states in other datasets, but not southern countries. This is due to the fact that *ONYX* connects this to a well-defined condition (e.g., southern states are Texas, Mississippi, ... / soaring refers to values higher than ...).

#### 4.4.3 System Architecture

The *ONYX* system employs a web-based, client-server model. It utilizes HTML5, CSS3, and JavaScript for its web-client and Python for the access and processing of the dataset. The interface manager coordinates the communication between the user interface of *ONYX* and its NL Parser and ITL agent. The data visualization tool of *ONYX* is built on the D3.js library (Bostock et al., 2011) and supports bar charts, scatterplots, and map charts as visualizations as well as categorical and numeric filters.

Figure 4.8 depicts *ONYX*'s architecture with its three main components: (i) Interface Manager, (ii) NL Parser, and (iii) ITL agent.

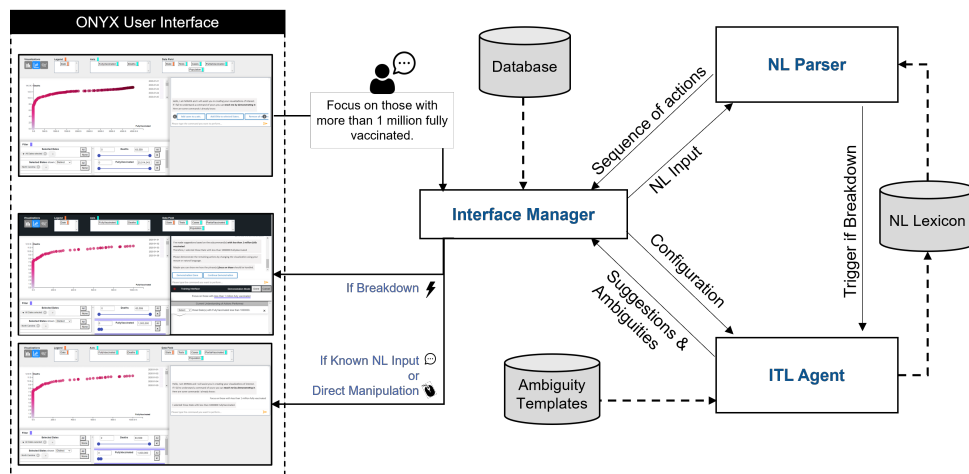


Figure 4.8: System Architecture Overview.

#### 4.4.3.1 Natural Language Parser

The NL input entered in the NLI is forwarded to an NL Parser. First, the NL Parser utilizes the results from Microsoft’s Language Understanding and Intelligent Service (LUIS) as well as the parts of speech tags and the dependency tree from Google Cloud Natural Language API to extract named entities and relations. Subsequently, the NL Parser identifies whether the users’ NL input is targeted at (i) *adapting the visualization* (e.g., Switch to Pennsylvania) or at (ii) *interacting with the ITL agent* (e.g., Finish the demonstration) by utilizing LUIS intent classification that is fine-tuned on examples extracted from the formative study and additional examples generated by the researchers. If the NL input is targeted at adapting the data visualization, the NL Parser applies a lexicon-based approach utilizing a Bigram Dependency Kernel (Özates et al., 2016) to derive the associated script from the NL Lexicon. The NL Lexicon is instantiated as a table consisting of entries for all previously trained NL inputs specifying (i) the parameterized trained NL input as an index, (ii) a parameterized bigram-representation of the NL input, (iii) the associated parameterized procedural knowledge, (iv) the required concepts that need to be included in the NL input, and (v) the ID of the user who taught the NL input or whether it is a foundational NL input provided by developers. Subsequently, the NL Parser augments the generalized script with the previously extracted named entities and relations and forwards it to the Interface Manager to execute the augmented script. If the NL Parser does not find an associated script in the NL Lexicon for the articulated NL input, the ITL agent gets triggered.

#### 4.4.3.2 Interactive Task Learning Agent

The ITL agent of *ONYX* utilizes the internal state of the data visualization tool to access user demonstrations. If the demonstration mode is active, the ITL agent derives a script from the continuous user demonstrations. Through this unified script, *ONYX* is capable of learning how to handle NL inputs that can also be achieved through direct manipulation. After completing the demonstration mode, the ITL agent checks whether named entities or numeric relations utilized in the NL input also have been utilized in the derived script. The detected instances are then parameterized in the derived script and referenced to the named entity or numeric relation in the NL input that has been utilized for parameterization to enable later augmentation during future usage. The NL input is then stored as a new entry in the NL Lexicon.

### 4.5 Evaluative User Studies

We conducted an online experiment and a think-aloud study to evaluate the effectiveness of *ONYX*. We deliberately chose to initially provide *ONYX* with only a small set of existing procedural knowledge that covered manipulating all GUI elements. In this, our goal is to show that even with this limited set of known concepts and user-defined procedures, users are still effectively assisted by *ONYX* in accurately demonstrating new NL inputs.

In both studies, participants were asked to address three breakdowns of *ONYX* by demonstrating the meaning of the NL input through direct manipulation and/or NL inputs. With the online experiment, we aimed to provide evidence for the effectiveness of the design goals instantiated in *ONYX* by utilizing a version of *ONYX* *without* the suggestions (**DG2**) and follow-up questions (**DG5** & **DG6**) as a baseline condition as we could not find similar tools that supports users with little programming expertise to complete the target tasks. The baseline version of *ONYX* still includes the ability to learn from NL inputs (**DG1**: see Section 4.4.2.1) and basic visual and textual aids (**DG3**: see Figure 4.7) to avoid an unfair comparison. The think-aloud study provided us with a deeper insight into the behaviors of participants.

#### 4.5.1 Participants

For our **online experiment**, we recruited 42 participants (20 male, 20 female, 2 non-binary) aged between 26 - 62 ( $M = 40.3$ ,  $SD = 8.8$ ) on Amazon Mechanical Turk. The participants were randomly assigned to one of the two conditions (baseline = 21; treatment = 21).



Across both treatments, they exhibited minor experience in programming with 81% rating their experience as poor or fair and an average experience of 2.3 years in programming (SD = 4.8). 90% of our participants rated their experience with NLIs as either average or better than average. Participants are denoted as PO1.T-PO21.T for the treatment and PO1.B-PO21.B for the baseline condition in subsequent sections.

The 5 participants (3 male, 2 female) recruited for our **think-aloud evaluation** from a signup list of university students and the general public exhibited similar characteristics, with ages ranging between 23 - 68 (M = 40.4 SD = 20.9) and minor experience in programming (100% said their programming experience was poor). 3 out of the 5 participants (60%) rated their experience with NLIs as either average or better than average. All think-aloud participants were provided with the complete version of *ONYX*. These participants are denoted as PT1-PT5 in subsequent sections.

#### 4.5.2 Procedure

At the beginning of both the online experiment and the think-aloud study, participants agreed to our IRB-approved consent form and were then provided with a pre-study questionnaire to elicit their demographics and their experience with programming and NLIs. Then, participants received an interactive guided tour that trained them in interacting with the data visualization tool. This tour did **not** include an introduction to the ITL aspects of *ONYX*. This provided participants with a basic understanding of the data visualization tool and ensured that the studies evaluated the effectiveness of the ITL and NLI aspects of *ONYX* and not the data visualization aspects. The interactive guided tour took participants around 4 minutes to complete.

After the interactive guided tour, participants were provided with three tasks in a randomized order. Participants were required to proceed to the next task after they finished the demonstration process when they felt they accurately taught *ONYX* how to handle the NL input. After finishing they were not able to test their demonstration. The participants in the think-aloud study were additionally encouraged during this phase to think aloud and both their voice and the screen containing *ONYX* were recorded for later analysis.

After completing the tasks, participants were requested to fill out a post-study questionnaire regarding their subjective experience with the different features of *ONYX*. Finally, they were able to provide feedback about *ONYX* in a free-text field.

### 4.5.3 Tasks

The three tasks consisted of an NL input and a short description of its meaning, similar to previous evaluations of NLIs with ITL capabilities (T. J.-J. Li, Azaria, et al., 2017). We derived the tasks based on common NL inputs in the formative study and ensured that key features of *ONYX* are covered by the tasks. To make sure that users needed to train the system with new NL inputs, we provided participants with NL inputs that *ONYX* did not yet know how to handle. We further confirmed that existing NLIs, such as Tableau’s Ask Data and Microsofts Q&A, were not able to perform these NL inputs correctly without user involvement (e.g., in Task C Ask Data only understood numeric filters and was *not* able to adapt the States filter accordingly or understood the goal of *combining*).

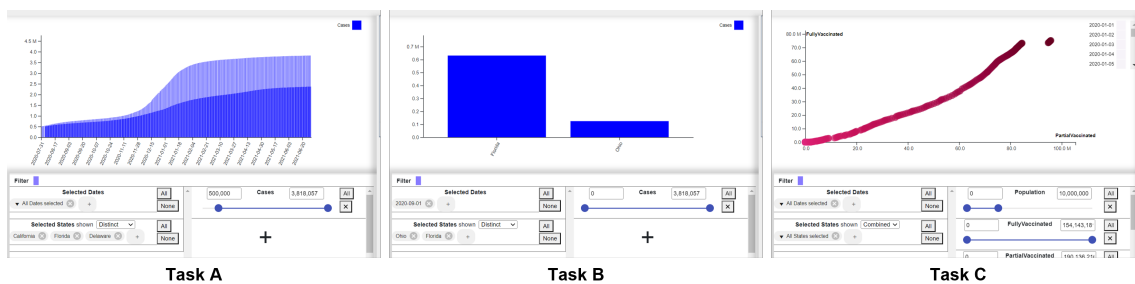


Figure 4.9: Target Visualizations after completing each Task A-C.

*Task A:* Demonstrating the NL input ☺ Display all dates with **more than 500,000 cases**. This task focuses on providing suggestions (bold parts of the NL input) and addressing language ambiguities. Online participants in both treatments required on average 93 seconds (Min = 36, Max = 231, SD = 46.8) for the demonstration process, which was significantly faster than *Task B* and *Task C*.

*Task B:* Demonstrating the NL input ☺ Only show Ohio and Florida on the 1st of September 2020, which should *only* show Ohio and Florida for the 1st of September 2020 even if other states or dates had been previously displayed. This task focuses on addressing direct manipulation ambiguities. Online participants in both treatments required on average 123 seconds (Min = 56, Max = 257, SD = 51) for the demonstration process.

*Task C:* Demonstrating the NL input ☺ Combine all states with **less than 10 million inhabitants**. This task focuses on providing suggestions (bold parts of the NL input). Online participants in both treatments required on average 157 seconds (Min = 37, Max = 609, SD = 134.6) for the demonstration process.

## 4.5.4 Results

### 4.5.4.1 Suggestions and Follow-Up Questions

For participants in the online experiment, we measured the accuracy of how well their demonstration fit the meaning of the NL input by comparing the learned script from *ONYX* to a gold standard that was derived from the task descriptions. Specifically, for all three tasks, we calculated the accuracy as a percentage of how many of the requirements are included in the learned script and whether requirements are included that are not required based on the task description. Participants interacting with the full version of *ONYX* had significantly higher accuracy (median: 93.3%) than those in the baseline condition (median: 73.3%) ( $U = 67.5$ ;  $p < 0.001$ ), based on a Mann-Whitney U test (see Figure 4.10). The difference in the time it took the participants to demonstrate the NL inputs (averaged across all tasks) was not significant at a 0.05 level for the participants that interacted with the full version (median: 120 s) and the baseline version (median: 110.5 s) of *ONYX* ( $U = 209$ ;  $p = 0.78$ ).

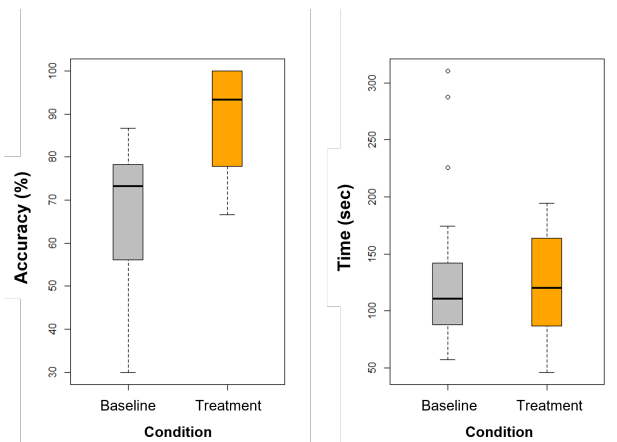


Figure 4.10: Boxplot of the Accuracy and Time for both Conditions for all 3 Tasks. For Accuracy, a higher Rating means better Accuracy. For Time, a lower Rating means better Time.

For insights into why the accuracy of the two conditions of the online experiment differed, we analyzed the reason for the errors. We specifically investigated whether *ONYX*'s follow-up questions and suggestions helped reduce the errors for the treatment condition in contrast to the baseline condition. Therefore, we labeled each incorrect section of the learned scripts based on whether *ONYX* provided no assistance for such errors (*other*) or if *ONYX* would assist in avoiding such errors through *follow-up questions* or *suggestions* (see Figure 4.11). In Task A and Task C, a third of the errors in the baseline condition was

associated with a lack of suggestions (Task A: 32.7%: Task C: 34.1%). A lack of follow-up questions targeting language ambiguities was associated with 18.8% of the errors in the baseline condition in Task A. In contrast, a lack of follow-up questions targeting direct manipulation ambiguities was associated with 84.2% of the errors in the baseline condition in Task B. However, participants in the treatment condition in Task B also exhibited errors associated with follow-up questions as they were able to ignore or decline the prompted follow-up questions.

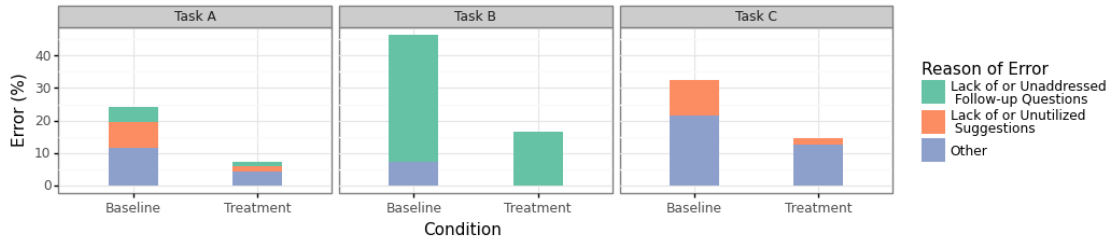


Figure 4.11: The Average Error in the Learned Scripts across Tasks A - C colored by the Reason of the Error.

Additionally, we asked participants in the post-study questionnaire of the treatment condition to rate statements regarding the effectiveness of suggestions and follow-up questions of *ONYX* on a 5-point Likert scale where 1 is “strongly disagree” and 5 is “strongly agree”. *ONYX* achieved an average score of 4.3 on “*The assistant supports me through its suggestions*” and 4.3 on “*The assistant supports me through its follow-up questions to my actions*”, further supporting the effectiveness of both the suggestions and follow-up questions (see Figure 4.12). These ratings were further supported by statements made by participants and through a post-hoc analysis of the log data derived from user interactions.

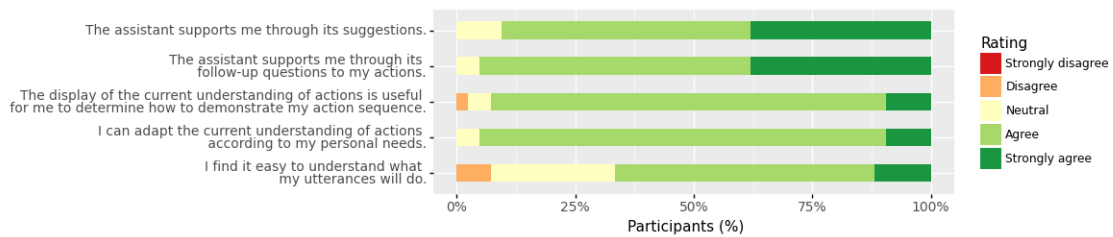


Figure 4.12: Responses to Post-Study Likert-scale Questions about the Experience of Participants with *ONYX*'s Features.

### Suggestions.

10 participants (47.6%) in the treatment condition of the online experiment explicitly stated that the suggestions helped them in their task in the free-text feedback field at the end of the post-study questionnaire. Specifically, through these suggestions participants

( $n = 7$ ) understood what information *ONYX* required from them and helped them start demonstrating to *ONYX* how to handle the NL input. The statement “[*ONYX*] makes suggestions that are logical” by PO8.T further illustrates the point of four participants that the suggestions they received were easy to interpret.

In the think-aloud study, we were able to investigate more closely how participants utilized the suggestions. All five participants noticed at the beginning of the demonstration mode what *ONYX* suggested, then checked if the suggestion fit their understanding of the NL input, and finally proceeded to demonstrate the part of the NL input *ONYX* did not yet know how to handle. For example, at the start of the demonstration PT4 said “*I would have done the same*” when checking the suggestions and PT3 said regarding the process of using the suggestions: “*I just had to check if it fits my own thinking*”. Two participants (40%) in the think-aloud study additionally stated that the suggestions helped them to stay in the analytic flow since the suggestions provided a good transition into the demonstration process without major disruptions.

#### **Follow-up questions.**

The positive aspects of the follow-up questions were explicitly mentioned by six participants (28.6%) in the treatment group of the online experiment in the free-text feedback field. Participants especially highlighted the understandability ( $n = 4$ ) and timing ( $n = 3$ ) of the follow-up questions. This is represented by the statement “*The questions were understandable and timely; that is, [ONYX] asked for clarifications at appropriate times and confirmed that it understood specific requests at appropriate times, as well*” by PO20.T. However, one participant in the online experiment also highlighted a negative aspect of the follow-up questions. PO15.T said: “*I found it difficult to demonstrate one NL input at a time and respond to the chat*”. Additionally, in the think-aloud study, two participants highlighted that they had problems with the follow-up questions at first because they did not notice the textual aid in the NLI as their focus was on another GUI element. For example, PT5 stated that the focus of their attention “*was over in the filter section. And not really looking at all the questions*”. This was supported by our log data as while 55.2% of follow-up questions were addressed by participants, 44.8% were unnoticed or incorrectly disregarded. However, PT5 further stated that it was easy to learn how to spot follow-up questions after noticing them for the first time, suggesting a learning effect.

To investigate this learning effect associated with follow-up questions closer, we analyzed the log data regarding the timing of follow-up questions prompted by *ONYX* and whether participants addressed them. Across all tasks, participants received a maximum of three

follow-up questions per task. Participants sometimes received few or no follow-up questions during certain tasks which complicated the derivation of clear measures to assess the learning effect. Especially in Task C, only 4 participants received follow-up questions since performing the correct demonstrations did not trigger any follow-up questions. However, if participants still received a follow-up question due to additional incorrect demonstrations, they correctly addressed these follow-up questions. In Task A and Task B, all participants received at least one follow-up question. However, only six participants (28.6%) addressed this first follow-up question in Task A and only 10 (47.6%) in Task B. When participants received a second follow-up question, they increasingly addressed this follow-up question (Task A: 58.8%; Task B: 65%). In Task B, 11 participants received a third follow-up question, which was noticed and addressed by 81.8%.

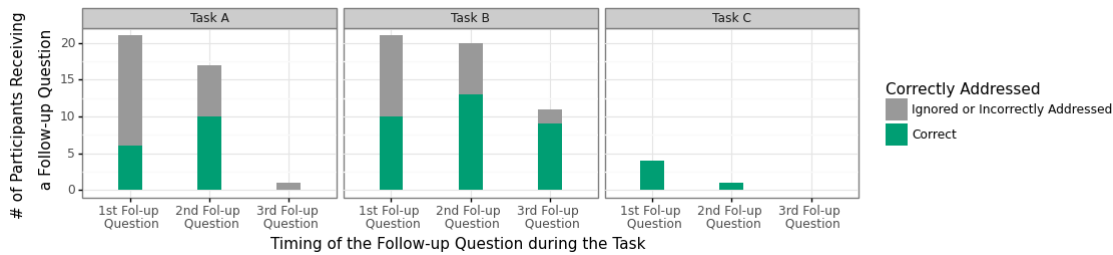


Figure 4.13: The Number of Participants receiving and addressing Follow-up Questions at different Timings during the Task

#### 4.5.4.2 Visual and Textual Aids

Seven participants (16.7%) of both conditions in the online experiment highlighted in the free-text feedback that the visual and textual aids helped them better understand what *ONYX* understood and what *ONYX* did not understand in their NL input. This helped them “*learn what information [ONYX] needs*” as stated by PO19.T and two additional participants.

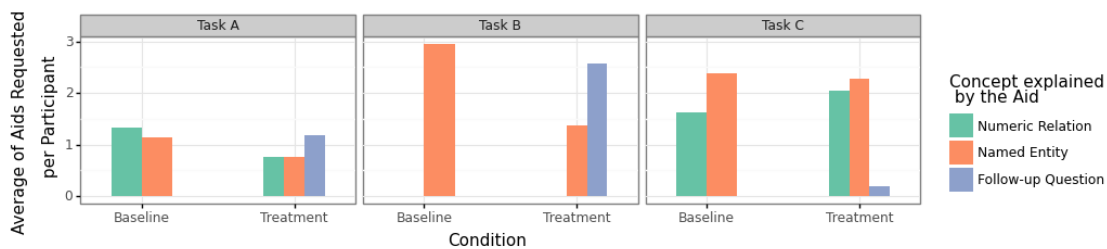


Figure 4.14: The Average Number of Visual and Textual Aids requested by Participants across Tasks A - C in the Baseline and Treatment Conditions. The Values are colored by the Concept explained by the Visual and Textual Aid.

Based on the log data of participants, we analyzed for which concepts participants requested visual and textual aids. The concepts that participants requested visual and textual aids for can be categorized into visual and textual aids helping participants understand the concepts in the NL input (e.g., numeric relations, named entities) and visual aids helping participants understand follow-up questions (see Figure 4.14). While both conditions received visual and textual aids regarding the NL input, only the participants in the treatment condition could receive visual and textual aids regarding follow-up questions (see Section 4.4.2.4). If the NL input included concepts associated with named entities (e.g., *Dates*, *Ohio*, or *1st of September 2020*), participants requested the connected visual and textual aid on average 1.82 times per task. Visual and textual aids for numeric relations (e.g., *more than 500,00 cases*) were requested on average 1.44 times per task. Participants mostly requested visual aids for named entities and numeric relations *before* performing the first demonstration to get insights into the NLI’s understanding of their NL input. Participants requested visual aids for follow-up questions in the treatment condition on average 1.32 times per task and 0.29 times per follow-up question.

#### 4.5.4.3 Display and Adaptation of *ONYX*’s Understanding

Regarding the effectiveness of the visual representation of *ONYX*’s understanding during the demonstration, 93% of participants agreed or strongly agreed with “*The display of the current understanding of actions is useful for me to determine how to demonstrate my action sequence*”, 95% with “*I can adapt the current understanding of actions according to my personal needs*” and 67% with “*I find it easy to understand what my utterances will do*” (see Figure 4.12).

## 4.6 Discussion

The results of our formative study highlighted the need for assisting users in teaching NLI’s how to handle new NL inputs through multi-modal ITL. The outcome of the user studies shows that *ONYX*’s suggestions, follow-up questions, and visual and textual aids are effective features to provide users with this kind of assistance. Participants in our user studies were able to significantly reduce errors in the learned scripts by requesting visual aids to receive a better understanding of *ONYX*’s knowledge, following *ONYX*’s suggestions, and clarifying the follow-up questions prompted by *ONYX*.

**Assisting Users to Understand *ONYX*’s Interpretations.**

In our formative study, we learned that users have problems understanding how the older version of *ONYX* interpreted their NL input at the *start* of the demonstration process and how it interprets their demonstrations *during* the demonstration process. To address these issues, our final design of *ONYX* assists users through suggestions, follow-up questions, and visual and textual aids that are targeted at improving the users' understanding. Our quantitative and qualitative results of the final user studies provide evidence that *ONYX*'s clear suggestions at the start of the demonstration process assist users in understanding existing knowledge of *ONYX*. Through users' improved understanding, they are able to better derive what demonstrations they need to perform. We further learned that users are often unaware of possible ambiguities in their direct manipulation demonstrations. This is highlighted in Task B of the online experiment in which errors due to ambiguity accounted for 84.2% of the errors made by participants in the baseline condition. Providing assistance through follow-up questions helped *ONYX* reduce the errors due to ambiguities on average by 65.4%.

#### **Learning to Utilize the Assistance.**

In our user studies, we learned that users do not always follow the advice of the system in a mixed-initiative approach, even if the advice is correct. A central problem was that users first needed to learn *when* and *how* *ONYX* provided them with assistance. After they learned these two aspects they were able to improve their utilization of *ONYX*'s assistance in our user studies. Perhaps explicitly introducing users to the assistance the first time they receive it could further improve the utilization of *ONYX*'s assistance by users as their learning is accelerated.

#### **Making the Reason of Assistance Explicit.**

Participants in our formative study highlighted that providing assistance without clarifying the reason for the assistance can even negatively influence the overall performance. This was because the assistance can disrupt the analytic flow. These findings highlight that more assistance is not always better. Our results on the final system suggest that users increasingly benefit from the assistance by *ONYX* if they are guided during the assistance through visual and textual aids. Furthermore, providing assistance directly after the event that triggered the need for assistance (synchronous assistance) helps users better map the assistance to its reason. Developers of ITL-based systems, therefore, need to ensure that the assistance is designed in such a manner that the benefit of the assistance outweighs



its negative effects (e.g., interruptions). The certainty of the ITL-based system about the correctness of the assistance and its timing have to be taken into account to assess this trade-off.

#### **Adaptability of *ONYX*'s Features.**

Although we demonstrated *ONYX*'s capabilities in just one data visualization tool custom-built utilizing D3.js, we believe that the strategies *ONYX* uses would generalize to other data visualization tools. We point to similarities with other data visualization tools, such as Power BI, Tableau, or open-source frameworks like vega-lite. For any of these visualization tools, *ONYX*'s features could be adapted to assist users in teaching the NLI how to specify (i) the encodings of the data visualization (e.g., the axes), (ii) the chart type, and (iii) numeric or categorical filters. *ONYX* is able to learn to perform these functionalities independent of the dataset when provided with the ability to track and manipulate the state of the data visualization tool either through an API or by directly integrating *ONYX* as in our case. However, *ONYX* is restricted to functionalities that users can perform through direct manipulation. Therefore, *ONYX* is, for example, not able to learn to internally calculate additional metrics, such as the average across the visualized data points, unless there is a built-in control for that in the GUI. We further believe that with additional future work our insights can be adapted to domains in which systems possess clear pre-defined functionalities and the state of the system can be translated into structured data. These insights can then be applied to augment existing ITL-based systems, such as VoiceCuts integrated into Photoshop (Y.-S. Kim et al., 2019).

## **4.7 Limitations and Future Work**

#### **NL Inputs with Multiple Meanings.**

*ONYX* cannot distinguish between multiple meanings of the same NL input based on the situational context. For example, the NL input ⊗ Remove states was used by users in the formative study both (i) to remove states from the x-Axis and (ii) to remove the filter associated with states. Therefore, the situational context, such as current configuration, previous interactions, or time and location of users, would be informative and relevant for *ONYX* to distinguish between similar NL inputs with varying meanings in different situational contexts. An important focus of our future work is to enable users to clarify the aspects of the situational context that lead to a varying meaning of similar NL inputs and to enable *ONYX* to learn from these clarifications.

### Recognition of Important Words, Synonyms, and Antonyms.

The current NL Parser enables *ONYX* to distinguish between NL inputs based on task-related keywords, the parameters, and the sentence structure. However, *ONYX* has no semantic understanding of important words contained in the NL input. For example, when *ONYX* learns how to interpret ☺ Only show Florida, it does *not* learn what the specific word **Only** means. Furthermore, the NL Parser is not able to identify synonyms and antonyms. To address these shortcomings, we plan to integrate post-processing of the learned NL inputs utilizing TF-IDF and the dependency structure to identify important words and their connection with parameters and parts of the scripts. This could then be utilized to inform additional suggestions provided by *ONYX* to assist users during the demonstration process. Furthermore, we plan to utilize open-source knowledge provided by ontologies (e.g., WordNet, VerbNet) to inform this post-processing and to enable *ONYX* to process synonyms and antonyms.

### Cold-Start Problem.

Lastly, *ONYX* initially only knew how to handle a limited set of NL inputs without additional training and was therefore only able to support users with suggestions based on a limited set of user-defined procedures. To address this issue, we plan to integrate our ITL approach with *existing* NL toolkits for creating data visualizations (e.g., NL4DV (Narechania, Srinivasan, et al., 2021), which is grammar-based, or ncNet (Luo et al., 2022), which is example-based) by extracting existing knowledge from these NL toolkits and providing this knowledge through *ONYX* to users throughout the demonstration process. Through this approach, the ITL capabilities of *ONYX* would not be an alternative, but an extension to the existing advances in NL processing, such as systems similar to GPT-3. This would be beneficial as while the learned insights from example-based NLI are more generalizable, existing alternatives still require numerous examples to train new tasks and are a black box to users (e.g., (Luo et al., 2022)).

## 4.8 Conclusion

Users are increasingly empowered to personalize natural language interfaces (NLIs) by teaching how to handle new natural language (NL) inputs. In this paper, we introduce *ONYX* which integrates a multi-modal interactive task learning (ITL) approach that assists users during the demonstration process to improve the accuracy of the learned

script. Specifically, *ONYX* assists users through suggestions based on parts of the NL input *ONYX* understood, follow-up questions to address ambiguities in direct manipulation demonstrations, and guidance through visual and textual aids. The results of our user studies show that the proposed *ONYX* features help users significantly improve the accuracy of the learned script for the NL input without requiring more time. Furthermore, participants appreciated how these features are integrated into *ONYX* and how we addressed the features' trade-offs. More broadly, our work demonstrates how users can be assisted during the demonstration process by an ITL agent to create a synergetic experience in personalizing an NLI in a multi-modal system.

# 5. Study IV: ContexIT - Interactively Contextualizing Natural Language Inputs in Data Visualization Tools <sup>5</sup>

## 5.1 Introduction

Natural language interfaces (NLIs) are increasingly shifting away from a one-size-fits-all design by allowing end users to teach NLIs how to correctly perform their natural language (NL) inputs (e.g., Siri (Apple, 2022); Google Assistant (Google, 2019)). A promising concept for lowering the barrier for end users in teaching NLIs is interactive task learning (ITL) (T. J.-J. Li, Azaria, et al., 2017; Ruoff, Myers, et al., 2022; S. I. Wang et al., 2017). ITL-based systems are inspired by how we teach humans new tasks and do not require end users to familiarize themselves with a programming language (Laird et al., 2017). Instead, ITL-based systems leverage programming-by-demonstration and NL programming to learn the correct interpretation for new NL inputs from multi-modal demonstrations by end users during their task completion in the actual system.


A key challenge in ITL-based systems is how to learn to correctly interpret NL inputs under consideration of the semantic and pragmatic level. Especially, as the meaning of an NL input depends on the person uttering it and their current context (Hawkes, 1977). Initial ITL-based systems only supported the learning of very narrow interpretations for new NL inputs, such as macro-recordings of user demonstrations. Recent ITL-based systems started to support the learning of broader interpretations by trying to capture the semantics of an NL input. This is done by generalizing the taught procedures (e.g., demonstrated clicks) (T. J.-J. Li, Azaria, et al., 2017; Pan et al., 2022; Ruoff, Myers, et al., 2022) and explained concepts (e.g., when is something hot or cold) (T. J.-J. Li, Radensky, et al., 2019). On the one hand, this enables end users to perform similar tasks to the ones demonstrated by utilizing different NL inputs that have the same meaning. On the other hand, this opens up the issue of performing incorrect interpretations in new situations.

The key problem is that the *pragmatic level* of the NL input in a specific situation is not yet considered (Pan et al., 2022). In general, pragmatics is used in NL processing to disambiguate a particular NL input under consideration of context (Setlur, Battersby, et al., 2016). We consider context as “any information that can be used to characterize

---

<sup>5</sup>This chapter is based on the following studies which are published or in work: Ruoff, Myers, et al. (n.d.).

the situation of an entity [...]” (Dey, 2001), such as the current location, time, or the state of the system the user is currently interacting with. NLI need to take this context into account for correctly interpreting an NL input, as NL inputs only mean *something* to *someone* (Grudin & Jacques, 2019; Hawkes, 1977; L. H. Leong et al., 2005; Misra et al., 2016; Tian et al., 2017). We performed an initial study with 22 participants where we elicited NL inputs, and found that 12.9% of the total 1,646 elicited NL inputs exhibited ambiguities with regard to which goals users want to achieve with a specific NL input, such as if they want to highlight or filter for something. When utilizing the current state of a system as additional information to the NL input, 71.4% of the ambiguous NL inputs can be clarified. Therefore, ITL-based systems need not only to generalize NL inputs for similar tasks but also to utilize the context of the taught NL inputs to clarify the correct interpretation if an ambiguity arises due to multiple reasonable interpretations.

To address this challenge, we introduce *ContextIT*, an intelligent agent integrated into a data visualization tool that is able to learn new NL inputs based on their sentence structure (syntactic level), their linguistic meaning (semantic level), and their meaning in a certain context (pragmatic level) through ITL. In addition to existing systems that are able to learn how to handle new NL inputs from users’ direct manipulations (programming-by-demonstration) and NL inputs (NL programming) on a syntactic and semantic level, *ContextIT* allows end users to contextualize their NL input in a second step by providing further information about how to consider the context. Specifically, after demonstrating how to handle a new NL input, end users can specify statements about the state of data fields, filters and the data visualization in general that need to be true for their demonstrated interpretation of the NL input to be the correct interpretation - hereafter referred to as contextual conditions. For example, if users utter the NL input  Show amount invested, then users can specify that the system should highlight the data field “Amount Invested” in the data visualization only if “Amount Invested” is already visualized. They can further specify that otherwise *ContextIT* should add the data associated with “Amount Invested” to the data visualization if the NL input is entered.

Three novel features of *ContextIT* support users in specifying these contextual conditions: (i) *suggestion of contextual conditions to users by ContextIT for the new interpretation*, (ii) *refinement of contextual conditions to differentiate existing interpretations*, and (iii) *guidance of the users by ContextIT through visual and textual aids*. First, *ContextIT* suggests possible contextual conditions stated in a declarative manner (e.g., “Amount Invested was in Values”) that could be important for selecting the demonstrated interpretations

considering the GUI elements utilized in the demonstration of the interpretation and based on previously existing contextual conditions of conflicting interpretations. Second, *ContexIT* provides users with the ability to add new contextual conditions to already existing interpretations that conflict with the interpretation that is currently contextualized. Specifically, *ContexIT* suggests possible contextual conditions to add to the existing conflicting interpretations to accurately understand when *not* to use these existing conflicting interpretations. Third, *ContexIT* provides *guidance* at crucial stages of specifying contextual conditions. Specifically, *ContexIT* provides visual and textual aids, such as displaying existing conflicting interpretations (see Figure 5.1 <sup>A</sup>) and connecting contextual conditions to their associated GUI representation to overcome pitfalls we identified in our participatory design studies, such as users prematurely finishing the teaching process.

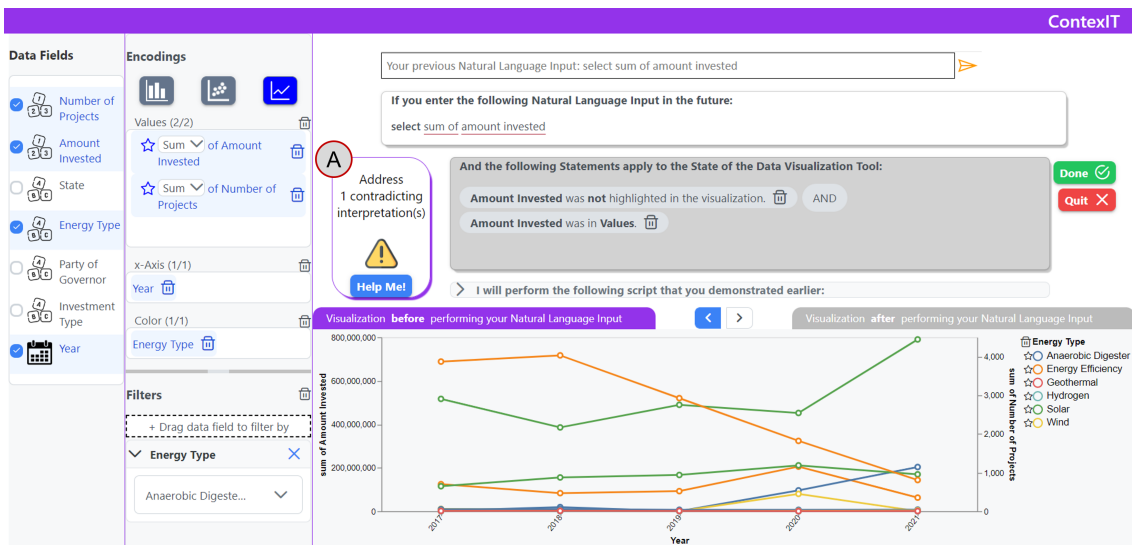


Figure 5.1: User Interface of *ContexIT* during the Contextualization Process.

We demonstrate *ContexIT*'s capabilities in a custom-built data visualization tool since data visualization tools (1) benefit from additionally enabling users to interact through NL (Setlur, Battersby, et al., 2016; Srinivasan, Lee, et al., 2020), (2) context is known to be crucial for understanding NL inputs in data visualization tools (Setlur, Battersby, et al., 2016; Tory & Setlur, 2019), and (3) the context of data visualization tools is finite and can be translated into a structured model (e.g., data fields, filters,...) (Tory & Setlur, 2019). We used a dataset about renewable energy projects in the United States as an example, however, the insights can be generalized across different datasets since both our data visualization tool and ITL-agent work independently of the dataset.

In developing *ContexIT*, we followed a human-centered approach by first identifying the reasons and circumstances for possible conflicting interpretations in NL inputs of users

with the data visualization tool by performing an initial NL elicitation study with 22 participants and collecting 1,646 NL inputs from these users that felt intuitive to them to invoke various functionalities of *ContextIT*'s data visualization tool. Subsequently, we iteratively developed the ITL capabilities together with end users by conducting a workshop ( $n = 5$ ) and seven participatory design sessions that include *ContextIT* at increasing levels of completeness. Afterward, we summatively evaluated our final design in eight think-aloud studies. The think-aloud sessions provide evidence that users are able to specify contextual conditions in *ContextIT* accurately (Accuracy:  $M = 92.5\%$ ) and therefore enable users to provide *ContextIT* with the needed information to choose in future contexts the correct interpretation of an NL input if multiple interpretations are reasonable.

## 5.2 Related Work

### 5.2.1 Natural Language Interfaces with Learning Capabilities

Even though the improvements in NL processing techniques, such as the introduction of GPT-3, reduce the breakdowns of NLI by increasing the coverage of possible NL inputs, existing NLIs still break down in unknown situations or perform incorrect interpretations (Lin et al., 2022; Luger & Sellen, 2016; Trummer, 2022; Zamora, 2017). Therefore, NLIs need to provide end users the ability to repair these breakdowns (Ashktorab et al., 2019). As end users can be motivated to improve the NLIs themselves if they break down (T. J.-J. Li, Azaria, et al., 2017), one promising approach is to enable NLIs to interactively learn from users how to accurately handle these missing or incorrect NL inputs during the interaction between users and the NLI (e.g., (Allen et al., 2007; Azaria, Krishnamurthy, et al., 2016; T. J.-J. Li, Azaria, et al., 2017; T. J.-J. Li, Radensky, et al., 2019; Pan et al., 2022; S. I. Wang et al., 2017)). Specifically, end users are enabled to teach the NLIs new procedures (e.g., demonstrated sequence of clicks (S. I. Wang et al., 2017)) and new concepts (e.g., *hot*, *cold* (T. J.-J. Li, Radensky, et al., 2019)).

To generalize the procedures and concepts from these one-shot demonstrations, NLIs with learning capabilities often utilize the NL inputs to generalize direct manipulation interactions and employ some form of mixed-initiative learning. SUGILITE (T. J.-J. Li, Azaria, et al., 2017), AutoVCI (Pan et al., 2022), and Voxelurn (S. I. Wang et al., 2017) generalize the learned procedures by automatically searching for entities in the NL inputs that can be utilized for parameterizing actions in the script of the learned interpretation. ParamMacros (Krosnick & Oney, 2022) builds on this approach and involves the user after detecting parameters to specify the set of possible alternatives that can be utilized as

instances of that parameter. APPINITE (T. J.-J. Li, Labutov, et al., 2018) and ONYX (Ruoff, Myers, et al., 2023) further aim to address ambiguities in the direct manipulation demonstrations either by asking users to describe their intention in NL (T. J.-J. Li, Labutov, et al., 2018) or by asking them to answer specific follow-up questions regarding the ambiguity at hand (Ruoff, Myers, et al., 2023). While these approaches broaden the possibility of tasks that the NL inputs can be used for, the NLIs do not have an understanding of whether possible conditions apply based on the current context since they only utilize the syntactic and semantic level of the NL inputs to calculate the most probable interpretation (Pan et al., 2022).

PUMICE aims to address the missing consideration of context by forcing users to specify IF conditions in the NL inputs (e.g., “If it is hot, then order a cup of Iced Cappuccino”) to allow users to execute IF-THEN procedures and to include pragmatics into the interpretation. These conditions are also flexible based on the task domain and allow the NLI to have multiple interpretations for the concepts used in the conditions (e.g., hot weather vs. hot ovens). However, users are often unspecific in their NL inputs (Srinivasan, Nyapathy, et al., 2021) and have difficulties in accurately specifying conditions on their own (Pane et al., 2001). Furthermore, PUMICE enables users to specify only one contextual condition per NL input, which is in contrast to the multiple rules that need to be specified when more than two interpretations are reasonable. To address these shortcomings, *ContexIT* aims to enable the NLIs to learn the pragmatics of NL inputs. Specifically, *ContexIT* utilizes the context to choose the correct interpretation without requiring users to specify the contextual conditions in each NL input by remembering and considering the contextual conditions the NLI previously learned for the NL input.

### 5.2.2 Natural Language Interfaces for Data Visualization Tools

Giving users of conventional data visualization tools the ability to additionally utilize NL inputs makes it easier for users to find the necessary information (Gao et al., 2015; Setlur, Battersby, et al., 2016; Srinivasan, Lee, et al., 2020), to specify complex filters (Y.-H. Kim et al., 2021), and to author the data visualization (e.g., changing colors) (Y. Wang et al., 2022). While NLIs for data visualization tools increasingly cover a wide variety of use cases and are also integrated into commercial products (e.g., Tableau (Tableau, 2019), PowerBI (Microsoft, 2022a)), they still have no precise solution for which parts of the previous context to retain (e.g., filters) and which to replace based on an NL input (Tory & Setlur, 2019). Often these decisions should be dependent on the visualization state of the data



visualization tool which needs to be taken into account by the NLI to determine how to correctly perform the NL input (Setlur, Battersby, et al., 2016).

However, the majority of existing NLIs for data visualization tools only take the intent and the linguistic properties (e.g., syntax, semantic, sentiment) of the NL input into account and need to involve the users to clarify ambiguities repeatedly (e.g., (Gao et al., 2015; Hoque et al., 2018; Setlur, Tory, & Djalali, 2019)) without the NLIs learning from these past clarifications.

Both *ONYX* (Ruoff, Myers, et al., 2022) and *Eviza* (Setlur, Battersby, et al., 2016) aim to address this challenge. After a breakdown, *ONYX* (Ruoff, Myers, et al., 2022) can learn how to correctly handle an NL input after users demonstrate to *ONYX* how to perform their intended task. However, *ONYX* only learns one interpretation for each NL input and still can not take the context into account. In contrast, *ContextIT* enables NLIs to differentiate between multiple interpretations of the same NL input based on the context. *Eviza* (Setlur, Battersby, et al., 2016) on the other hand tries to prevent breakdowns by augmenting the NL input of users with text included in the context (e.g., data fields, previous NL inputs). However, *Eviza* currently takes every text included in the context into account and is not aware of which parts of the context are especially important and which need to be ignored. While this is a minor issue in *Eviza* with its simplified context, as users can only specify one filter for the data visualization, any useful data visualization tools for data exploration will have more complex contexts with multiple selected data fields and data field types (e.g., categorical, temporal, numeric), filters, encodings, and transformations (e.g., average, max, min,...). Hence, to accurately utilize the context to differentiate multiple conflicting interpretations of an NL input, *ContextIT* learns which parts of the context are relevant to its current decision and how to leverage this information.


### 5.3 Formative Studies & Design Goals

For designing our *ContextIT* system we conducted three consecutive formative studies. First, we conducted an initial NL elicitation study to understand which NL inputs can be ambiguous and whether the context (e.g., the current visualization state of the data visualization tool) could be utilized to clarify these ambiguous NL inputs. We recruited 22 participants from Amazon’s MTurk (Gender: 13 female, 9 male; Age:  $M = 37.5$  years,  $SD = 11.2$ ) and asked them to provide two NL inputs that felt intuitive to them to invoke a functionality for each of 40 demonstrations of functionalities in our data visualization tool. We excluded 6.5% of the total 1,760 NL inputs that were completely irrelevant or

apparently due to laziness, resulting in 1,646 NL inputs included in our subsequent analysis. Second, we conducted a design workshop with five participants from our university (Gender: 4 female; 1 male; Age:  $M = 23.6$  years;  $SD = 2.88$ ) to derive initial requirements for a system supporting users in contextualizing NL inputs and developed an initial version of *ContextIT* based on those requirements. Third, we iteratively improved our initial version of *ContextIT* through seven participatory design studies (Gender: 3 female; 4 male; Age:  $M = 40.43$  years;  $SD = 10.56$ ) with 1-3 weeks between sessions.

From the insights we gained in these three studies we distilled **five** design goals (**DGs**) for enabling users to contextualize NL inputs through ITL.

### 5.3.1 Clarifying Ambiguities in NL Inputs through Context

Users usually assume that the NLI considers the same information for interpreting their NL inputs as they do (Reinhart, 1981), including also the context of the NL inputs. This incorrect assumption was further evidenced by our initial NL elicitation study as 12.9% of the NL inputs are ambiguous in regards to users' goals. To clarify these ambiguities, additional information about the NL input itself is required to choose the correct interpretation. For example, to correctly interpret the NL input  (a blue rounded rectangle containing a speech bubble icon and the text "Show energy types"), the NLI has to consider whether energy types are already visualized and if there is a filter set for specific energy types. One approach to clarify these ambiguities is utilizing the current visualization state (e.g., selected data fields) in addition to the NL input as this clarifies 71.4% of the ambiguous NL inputs in our dataset. Our analysis further provides evidence that even with insights from 22 participants we did not reach coverage of all possible ambiguities as the last participant still introduced new ambiguities to the functionalities we were eliciting NL inputs for. This highlights that it is difficult for developers to foresee all possible ambiguities for NLIs even through extensive user testing. To enable users to clarify the 71.4% of ambiguities that can be addressed by considering the context, users should be enabled to specify contextual conditions themselves during the actual interaction as a response when the NLI did not choose the appropriate interpretation that the user had in mind. The NLI should then leverage these contextual conditions for future interactions.

**DG1 - Contextualizing:** Enable users to interactively specify contextual conditions for their NL inputs based on the current visualization state.

When teaching NLIs it is essential for users to understand which parts of the NL input and the context the system thinks are actually important for choosing an interpretation for

an NL input (Lim & Dey, 2010). Participants in the workshop highlighted that *ContexIT* should display in NL **what** part of the visualization state is important for considering the right interpretation to give users a better understanding of the system’s model. The key argument for declarative contextual conditions was according to participants that end users can easily check whether the GUI elements that are important to *ContexIT* for the interpretation were also important for their interpretation. Especially as users with limited technical expertise often do not have a strong understanding of the underlying functionality of the system and, therefore, need an abstracted version of the system’s model. The feedback in the subsequent participatory design studies in which users interacted with a live version of *ContexIT* further strengthened this choice while also prompting us to refine the terms used in the contextual conditions as they initially included too many technical terms (e.g., temporal data fields) that end users were not familiar with.

**DG2 - Declarative Contextual Conditions:** State the contextual conditions in a declarative manner to improve their understandability by users.

### 5.3.2 Supporting Users in Contextualizing NL inputs

End users generally have issues knowing what information the system requires and how they need to provide it when teaching a system (McDaniel & Myers, 1999). Therefore, as *ContexIT* can track what parts of the visualization state were altered by users when correcting the interpretation of the NLI, users should be supported by suggestions of contextual conditions derived from these affected parts of the visualization state. Furthermore, *ContexIT* has knowledge about contextual conditions for existing interpretations of similar NL inputs and can utilize them to provide suggestions to differentiate the current interpretation from these existing conflicting interpretations. A remaining challenge is to derive the correct abstraction level for the suggestions (e.g., “The values field was not empty” vs. “State was in the values field”). Hence, the users should be provided with multiple levels of abstractions for the derived contextual conditions to enable users to choose the appropriate ones from the suggestions.

**DG3 - Suggestions:** Support users in specifying contextual conditions through multiple suggestions that users can choose from.

While the suggestions helped participants in our participatory design studies to specify when to use the new interpretation to address the current ambiguity, they often did not specify enough contextual conditions in order for *ContexIT* to understand when **not** to use their interpretation to clarify future ambiguities that they were not aware of yet. For example, multiple participants taught *ContexIT* how to remove certain filters and did not specify any conditions but later in their session realized that they would like to use the same NL input to remove highlights of data points and that they would require to contextualize both interpretations to achieve this goal. Hence, it is important for users to be able not only to specify contextual conditions for new interpretations but also to refine the contextual conditions of existing interpretations when new ambiguities arise.

**DG4 - Refinement:** Enable users to continuously refine contextual conditions of existing interpretations of NL inputs.

### 5.3.3 Guiding Users in Contextualizing NL inputs

To clarify an ambiguity in *ContexIT*, users first have to demonstrate the correct interpretation and subsequently specify contextual conditions for the new interpretation and differentiate existing conflicting interpretations. Throughout this process, users are supported by suggestions which lead to even more information that users have to process. While teaching our initial version of *ContexIT*, participants of our participatory design studies were occasionally confused about which mode the *ContexIT* system is currently in (interact vs. demonstrate vs. contextualize), whether they already addressed all ambiguities, and which part of the visualization state a suggested contextual condition was associated with. Therefore, we integrated several textual and visual aids to guide users in creating and contextualizing the new interpretation. This helped users to process the information more easily and to avoid confusion in the subsequent participatory design studies. We continued to improve the phrasings and design of the visual and textual aids as our initial design and phrasings were too technical.

**DG5 - Visual & Textual Aids:** Guide users in contextualizing the interpretation of the NL input through visual and textual aids to avoid confusion and minimize disruption.

## 5.4 *ContextIT*

In the subsequent section, we detail how *ContextIT* incorporates the previously derived five DGs. Specifically, we first describe an example scenario of using *ContextIT*. Second, we highlight the features of *ContextIT* that (i) enable users to specify contextual conditions, (ii) provide suggestions of contextual conditions to users, (iii) enable users to refine existing interpretations, and (iv) enable *ContextIT* to generalize contextual conditions to new situations.

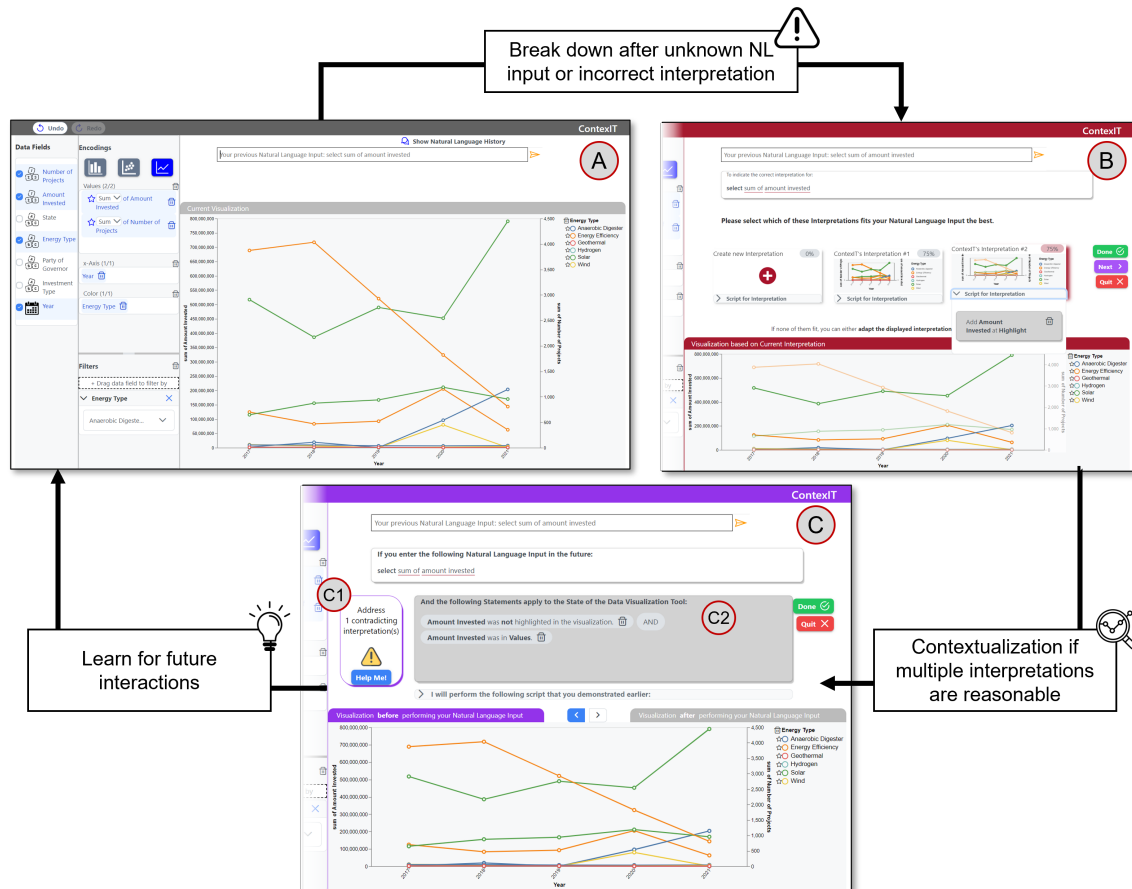







Figure 5.2: *ContextIT*'s User Interface depicted in its three Modes: (i) Interaction **A**, (ii) Demonstration **B**, and (iii) Contextualization **C**.

### 5.4.1 Example Scenario

This section illustrates how *ContextIT* works through an example scenario. Suppose Rosie, a business user at a USA NGO, wants to explore the number of renewable energy projects and the amount invested in them across the years and by energy type. She, therefore, created a line chart as visualized in Figure 5.2 **A**. To highlight the amount invested in the visualization while retaining the number of projects as well and she asks the NLI to **Select sum of amount invested**. *ContextIT* interprets the NL input and incorrectly

concludes that Rosie wants to add the amount invested to the values. However, as this action performed by *ContexIT* does not lead to any change in the data visualization, *ContexIT* asks Rosie whether it has incorrectly interpreted the NL input. As a response, Rosie signals that she wants to teach *ContexIT* how to correctly interpret the NL input.

Subsequently, *ContexIT* switches to its demonstration mode (Figure 5.2 ) and provides suggestions to Rosie on various possible interpretations of her NL input. Rosie selects an interpretation that looks similar to the data visualization she wants to create, checks the script of the interpretation on whether it conforms to her interpretation, and clicks . *ContexIT* then asks her whether she could help contextualize the NL input as *ContexIT* has another conflicting interpretation of this NL input.

When Rosie confirms she will contextualize her NL input, *ContexIT* shifts to the contextualization mode. *ContexIT* supports Rosie through suggestions that the system derived based on the demonstrated interpretation and existing contextual conditions regarding the conflicting interpretation. She now goes through the suggestions and selects the contextual conditions that best reflect when to utilize her interpretation of the NL input. *ContexIT* adds those contextual conditions to the GUI to visualize its decision-making (Figure 5.2 ) and updates how many conflicting interpretations remain (Figure 5.2 ). Rosie is now satisfied with the contextual conditions that she selected for her interpretation, but one conflicting interpretation still remains. *ContexIT*, therefore, enables her to refine the contextual conditions of the existing conflicting interpretation by specifying which of the previously selected contextual conditions can best be utilized to differentiate those two conflicting interpretations. Afterward, no conflicting interpretations remain and she clicks  to finish the training.

*ContexIT* now learns the newly taught interpretation by Rosie in combination with her specified contextual conditions and additionally refines *ContexIT*'s understanding of the existing conflicting interpretation by adding the contextual condition that best differentiates the two conflicting interpretations.

### 5.4.2 Key Features

In the following, we will focus on discussing how *ContexIT* enables and supports users in contextualizing NL inputs and not on how its users can demonstrate the interpretations because *ContexIT*'s ability for learning and generalizing procedural and conceptual knowledge builds on the underlying mechanisms of our previous *ONYX* system (Ruoff, Myers, et al., 2022).

## 5.4.2.1 Specifying Contextual Conditions.

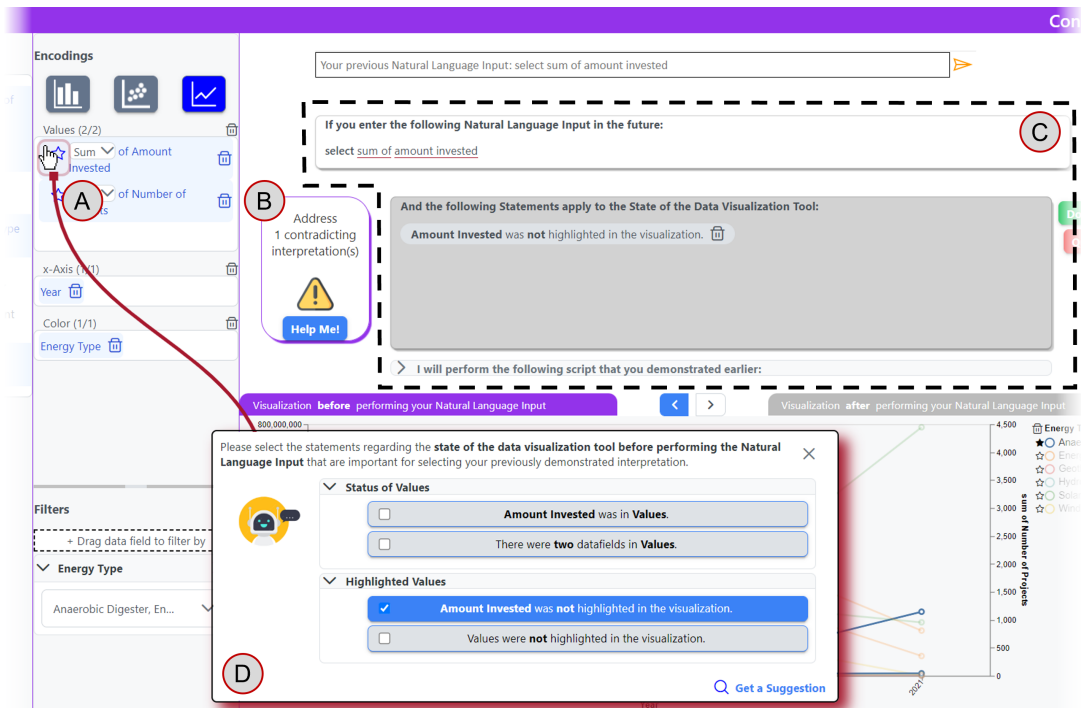


Figure 5.3: *ContextIT*'s User Interface during the Contextualization with the Visual and Textual Aids (B) & (B) and depicting the Contextual Conditions (D) when Users are hovering over a GUI Element (A).

In the formative studies, we identified that users often point to elements in the GUI when talking about the reasons why they interpreted the NL input in a certain way. We translated this behavior to *ContextIT* in order for end users to interactively specify contextual conditions without the need to know a programming language (DG1). First, users hover with their mouse pointer over the associated GUI elements (Figure 5.3 (A)) similar to how they would point to it when talking about their reasoning. Second, based on the GUI element the user is referring to, *ContextIT* provides declarative contextual conditions as options for the user to select from (Figure 5.3 (D)) (DG2). The more specific the target GUI element that the user is hovering over, the more specific the contextual conditions are that *ContextIT* provides users as an option. For example, if users hover over the overall values field they would get more general contextual conditions as options that refer either to the amount invested, the number of projects, or both but nothing regarding the highlighting of amount invested. To derive the contextual conditions, *ContextIT* checks which states and configurations are associated with the targeted GUI element (e.g., selections, aggregates,

display in the data visualization) and creates the contextual conditions utilizing templates that take the target, the relationship and the property as an input. The relationships can either be *equals*, *contains*, *has range*, or *of type* as well as their negated counterparts. *ContexIT* then clusters contextual conditions together that convey a similar message but on different abstraction levels. To achieve this, *ContexIT* utilized the conceptual hierarchy of the object (e.g., solar being a subcategory of energy types) and their visual placement (e.g., if a data field is selected as the color in encodings). Hence, users can easily compare which abstraction level best fits their reasoning and can then select the appropriate contextual condition accordingly. When users select a contextual condition in the pop-up window, then *ContexIT* adds this contextual condition to the display of its reasoning (see Figure 5.3 ©)

#### 5.4.2.2 Guiding Users through Visual and Textual Cues.

To guide users in accurately contextualizing their NL input, *ContexIT* implements several visual and textual cues (DG5). First, the display of *ContexIT*'s reasoning is displayed as a production rule since production rules are a structure that end users are familiar with even if they lack technical expertise (Pane et al., 2001) (Figure 5.3 ©). Through this structure, they know what the trigger for their interpretation is (their NL input), what the conditions are (the selected contextual conditions), and what their interpretation encapsulates (their demonstrated actions). Second, to guide users' attention and to ground the contextual conditions that *ContexIT* provides in the GUI elements that users are familiar with, *ContexIT* associates the GUI element the user is hovering over with the pop-up containing the contextual conditions through a dynamic line. Third, *ContexIT* continuously informs users about how many conflicting interpretations remain (Figure 5.3 ©). This is especially important, as users on their own had issues understanding when they selected enough contextual conditions to accurately contextualize their NL input and whether they needed to differentiate their interpretation against existing conflicting interpretations. Fourth, users are guided through textual prompts throughout the demonstration and contextualization of their interpretation of the NL input. *ContexIT* explains the changes in its user interface to the user after a switch of modes (e.g., from demonstration to contextualization mode; Figure 5.4 (A)) as users can have difficulties noticing when there is a change in the mode of a system (Myers, McDaniel, et al., 1993) and adapts its color scheme to highlight the change in modes. Furthermore, when users request help from *ContexIT* in the contextualization mode (Figure 5.3 ©) then *ContexIT* explains through textual cues what options users



have to contextualize their interpretation. While all options are available in Figure 5.4 (B), *ContextIT* adapts these options, for example, if it is not possible yet to differentiate the current interpretation against existing conflicting interpretations.

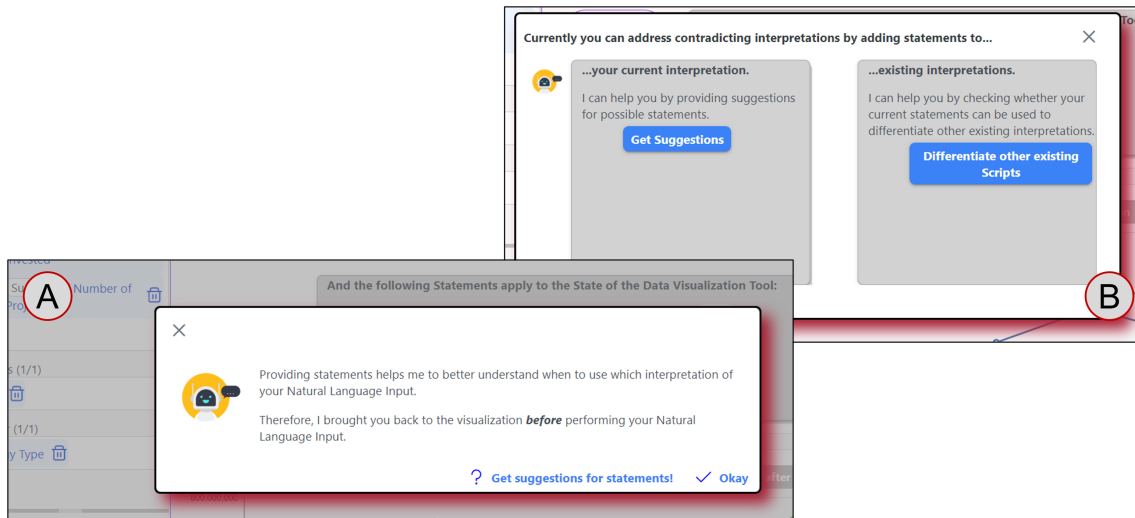


Figure 5.4: *ContextIT*'s Textual Aids when switching from Demonstration to Contextualization Mode (A) and when Users ask for Assistance during the Contextualization (B).

#### 5.4.2.3 Suggesting Contextual Conditions.

To support users in contextualizing their interpretation through suggestions (DG3), *ContextIT* leverages both (i) the GUI elements affected by the demonstration of the interpretation and (ii) the contextual conditions of existing conflicting interpretations. First, from the demonstrated interpretation, *ContextIT* derives the affected GUI elements and clusters them based on the area of the user interface that the GUI elements are situated in. Based on pre-defined templates (see Section 5.4.2.1), *ContextIT* suggests contextual conditions that could help contextualize the interpretation.

Second, *ContextIT* suggests contextual conditions that could be utilized to differentiate the current interpretation from existing conflicting interpretations. The underlying assumption is that when a part of the visualization state is important to contextualize a conflicting interpretation and this contextual condition is violated by the initial state of the current interpretation, then the opposite of that contextual condition could be utilized to contextualize the current interpretation. Therefore, *ContextIT* first checks which interpretations of the NL input are conflicting with the current one. Then *ContextIT* extracts those contextual conditions of existing conflicting interpretations that are violated in the initial state of the current interpretation. Finally, from the extracted set of contextual conditions, *ContextIT*

derives the targeted GUI element and proposes based on pre-defined templates contextual conditions on multiple abstraction levels that are the opposite of the contextual condition of the existing conflicting interpretation.

#### 5.4.2.4 Refining Existing Conflicting Interpretations.

While it is relatively easier for users to specify positive contextual conditions that need to apply to the initial state of the current interpretation, it is more difficult for users to specify what should **not** apply to other existing conflicting interpretations. Therefore, to support users in refining existing conflicting interpretations, we utilize the specified contextual conditions for the current interpretation, check which are violated in existing conflicting interpretations and therefore could be used to differentiate the two, and ask the users to specify which of the final selection is really the contextual condition that would best differentiate the two interpretations (see Figure 5.5). Through this approach, users do not have to specify negative contextual conditions about the existing conflicting interpretation, which is difficult for them, but can specify which of the contextual conditions are most important for the current interpretation and *ContextIT* then converts this to a negative contextual condition in the back-end.

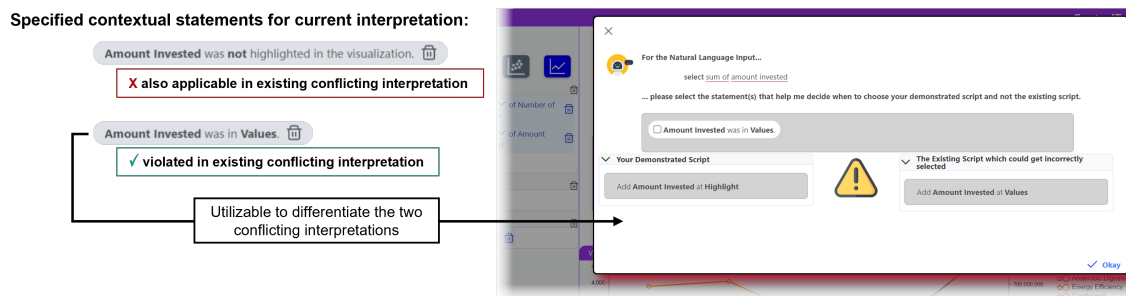


Figure 5.5: *ContextIT*'s decision-making for deriving the Contextual Conditions for differentiating two conflicting Interpretations.

In the example depicted in Figure 5.3 with the goal of highlighting the Amount Invested in the data visualization, the user selected the two contextual conditions “*Amount Invested was not highlighted in the visualization.*” and “*Amount Invested was in Values.*” to contextualize their interpretation of ☺ Select sum of amount invested for this initial state. When selecting to differentiate the existing conflicting interpretation of just adding Amount Invested to values, then *ContextIT* checks which of the contextual conditions are violated in the initial state of this existing interpretation. In this case, users now have one single option (“*Amount Invested was in Values.*”) to differentiate the two conflicting interpretations, however, there can be multiple options to select from depending on how many of the

specified contextual conditions are violated in the existing interpretation. Now users have to decide whether this option is reasonable to differentiate the two interpretations or whether they might need to add additional contextual conditions to the current interpretation to subsequently check again whether the added contextual conditions can be better used for differentiation.

#### 5.4.2.5 Generalization of Contextual Conditions.

*ContexIT* is able to learn contextual conditions on two generalization levels: (i) fixed and (ii) parameterized. Fixed contextual conditions are important to convey abstract states of the data visualization, such as that the values field was empty. In this case, the target (values field), the relationship (equals), and the property (empty) are fixed and need to apply for future interpretations in that specific way. Parameterized contextual conditions utilize the entities extracted from the NL input to parameterize the target and the property if possible. This is important for contextual conditions, such as “*Amount Invested was not in Values*” for the NL input ☺ Select amount invested as this contextual condition should also be applicable if users utilize the “number of projects” in the NL input. Furthermore, this parameterization allows in addition to the implicit utilization of the context which is of focus in our *ContexIT* system, also the explicit utilization of the context in the NL input (e.g., PUMICE (T. J.-J. Li, Radensky, et al., 2019)). Hence, users can teach NL inputs such as ☺ If Year is in the x-Axis, switch to a line chart.

### 5.4.3 System Architecture

The *ContexIT* system employs a web-based, client-server model. It utilizes HTML5, CSS3, and JavaScript for its web-client. For the NL Parser and ITL agent, *ContexIT* utilizes Python. The data visualizations are rendered utilizing the vega-lite library (Satyanarayan et al., 2017) which also enables the interaction through direct manipulation with the data integrated.

#### 5.4.3.1 NL Parser

When provided with an NL input, the NL parser of *ContexIT* aims to find parts of the NL input that contain meaning within the context of the dataset. Therefore, the NL parser extracts n-grams, ranging from a single word to the length of the complete NL input. For each n-gram, we traverse a knowledge tree to extract the most probable meaning of the n-gram. The knowledge tree consists of a base node and three connected layers. The first

layer represents the syntactic similarity (e.g., similar structure of words used). The second represents semantic similarity (e.g., words used have a similar meaning). Finally, the third layer represents the pragmatic layer which checks whether the context of the NL input fits the pragmatics of the leaf node that encapsulates the meaning. After calculating for each n-gram the similarity score for the most probable meaning, the NL Parser rates the possible combination of the n-grams based on their similarity scores, the number of unused words, and the granularity of n-grams. If the score of the most probable combination is equal to or exceeds a threshold (during our formative studies we have found 0.7 on a scale from 0 to 1 is a good threshold), then the interpretation of the NL input is directly performed. Otherwise, the ITL agent is triggered.

#### 5.4.3.2 ITL Agent

The functionality of tracking, generalizing, and learning from user demonstrations is adapted from our previous *ONYX* system. We extended the ability of *ONYX* to learn from and generalize contextualizations of NL inputs. After completing the contextualization, *ContextIT* searches for named entities and numeric relations that are utilized both in the contextual conditions as well as in the NL input that users want to contextualize. The instances detected through this search are then parameterized and the parameterized instances are connected to the position of the NL input that needs to be utilized for augmenting the contextual conditions during future usages of similar NL inputs.

## 5.5 Evaluative User Study

We conducted a think-aloud study to evaluate how effective *ContextIT* is in enabling users to contextualize their NL inputs. The think-aloud study further provides us with insights into the design goals instantiated in *ContextIT* and how users utilize the different features associated with these **DGs**.

### 5.5.1 Participants



For our think-aloud study, we recruited eight participants (Gender: 5 female, 3 male; Age:  $M = 46.1$  years,  $SD = 10.48$ ) from the crowdsourcing platform Prolific. The participants had minor previous experience in programming ( $M = 2.4$  years,  $SD = 3.58$ ) and 75% of the participants rated their own programming skills either as poor or fair ( $M = 1.24$  on a 5-point Likert scale). Every participant rated their skills with NLI either average or higher ( $M = 3.34$  on a 5-point Likert scale) and they all stated that their first language

is English. All think-aloud participants were provided with the final version of *ContextIT*. These participants are denoted as T1-T8 in subsequent sections.

### 5.5.2 Procedure

The study was remotely conducted over Zoom. The participants were sharing their screen that included the instantiation of *ContextIT* during the introduction and the main part of the study. Throughout the study, they were asked to think aloud and the researcher additionally asked participants questions when they were not thinking aloud or when they performed interactions of interest.

After consenting to the study, participants were given an introduction to the data visualization tool. During the introduction, users were asked to perform various mouse-based interactions to give them insight into both the functionalities of the data visualization tool and the dataset. They also had the opportunity to play around with the data visualization tool. The introduction took around 20 minutes.

Subsequently, users performed a tutorial task to introduce them to the framing of our tasks and to the capabilities of *ContextIT* to learn and contextualize interpretations of NL inputs. Users were provided with two conflicting interpretations of  (e.g. ) and they were guided by the researcher to teach *ContextIT* (i) when to add the data field to the x-axis and (ii) when to add the data field to color instead. *ContextIT* was already taught the former interpretation of the NL input and users had to demonstrate and contextualize the latter interpretation of the NL input. In doing so they had to additionally refine the former interpretation of the NL input to correctly contextualize the NL input. The tutorial took around 20 minutes.

After completing the tutorial task, participants were provided with two NL inputs in random order. After they felt that they accurately contextualized all interpretations of an NL input they were able to proceed to the next.

After contextualizing the two NL inputs, users were asked to fill out a post-study questionnaire about their experience with *ContextIT* and its features. Furthermore, they provided their demographics in the post-study questionnaire.

### 5.5.3 Tasks

We selected two NL inputs that were among the most frequently elicited NL inputs with ambiguities in our data set from the NL elicitation study. For each NL input, participants were provided with multiple interpretations. One interpretation was already known to

Task	NL Input	Contextual Conditions		Description
		Specify	Refine	
A.1	Display all [Data Field]	-	x	Remove the focus from specific entities of the data field in the visualization
A.2	Display all [Data Field]	x	-	Remove the filter for specific entities of the data field
B.1	Select [Aggregate] of [Data Field]	-	x	Add the data field to the values field
B.2	Select [Aggregate] of [Data Field]	x	(x)	Highlight the data field in the data visualization
B.3	Select [Aggregate] of [Data Field]	x	-	Change the aggregate of the data field

Table 5.1: Tasks covered by the User Study, whether Users are required to Specify or Refine Contextual Conditions to correctly contextualize the Interpretation, and a short Description of the Interpretation.

*ContextIT* without any contextual conditions specified. The other interpretations have to be demonstrated and contextualized by the participants and the already-known interpretation has to be refined. To understand the reasoning when *ContextIT* should utilize which interpretation, users were provided with two examples per interpretation of an NL input as videos on how the data visualization should look like before uttering the NL input and how it should respond to the NL input. They were also asked that for all the examples their contextualization has to work correctly so that they can utilize the examples as test cases. We utilized for each example different data fields (e.g., energy types, investment types) and aggregates (e.g., mean, sum, max) to provide a broad spectrum of contexts and NL inputs. This additionally prevented participants from adding unnecessary contextual conditions as *ContextIT* must choose after the training the correct interpretation for all examples given in the task description.

## 5.5.4 Results

### 5.5.4.1 Overall Effectiveness of ContextIT.

To calculate how accurately participants contextualized the different interpretations of the provided NL inputs, we counted how many of the provided exemplary contexts *ContextIT* would perform the correct interpretation. We additionally labeled examples as incorrect when *ContextIT* would perform the correct interpretation but would have a possible ambiguity. We added this check to control for the order of training, as *ContextIT* uses the first interpretation it learns as a default when it has multiple interpretations with an identical likelihood that both exceed the similarity threshold of the NLI.

Overall, participants achieved an accuracy of 92.5% (SD: 0.12) across all participants and tasks. 5 participants (62.5%) contextualized all interpretations of the two NL inputs correctly. 7 (87.5%) succeeded in 4 out of the 5 interpretations.

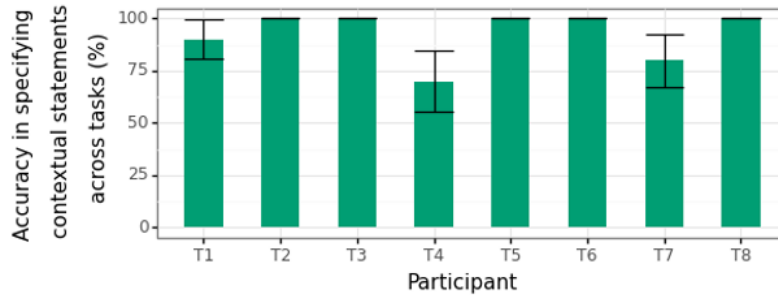


Figure 5.6: Accuracy in specifying Contextual Conditions across Tasks for each Participant.

There were three reasons for errors that occurred equally often. First, T4 incorrectly demonstrated the interpretation of task B.3 (see Table 5.5.3). Since T4 did not realize the incorrect demonstration, the participant was confused during the contextualization by the suggestions of *ContexIT* that were based on the previous incorrect demonstration. Second, T7 did not refine the contextual conditions of B.1 during the contextualization of B.2. This led to multiple reasonable interpretations when performing the NL input in the exemplary contexts of B.2 due to a lack of differentiation. Third, both T4 and T1 overspecified their contextualization of B.2 based on the first exemplary context. This resulted in *ContexIT* not utilizing the demonstrated interpretation in the second exemplary context since a contextual condition that was specific for the first was violated in the second exemplary context.

Additionally, we asked participants in the post-study questionnaire of the treatment condition to rate statements regarding the effectiveness of *ContexIT* and its features on a 5-point Likert scale where 1 is “strongly disagree” and 5 is “strongly agree”. *ContexIT* achieved an average score of 4.5 on “*The system is successful in learning to understand new natural language inputs according to my personal needs.*” and 4.375 on “*It is important to me to be able to teach the system to understand new natural language inputs.*”, further supporting the effectiveness of *ContexIT* and the relevance of teaching and contextualizing NL inputs (see Figure 5.7). Only T3 thought it is “*much easier just to be taught how to use it rather than teach it*”.

Furthermore, three participants (37.5%) specifically stated that they “*like the natural conditions, it’s really good. It helps so anyone can understand them*” (T7), providing evidence for the effectiveness of **DG2 (Declarative Contextual Conditions)**.

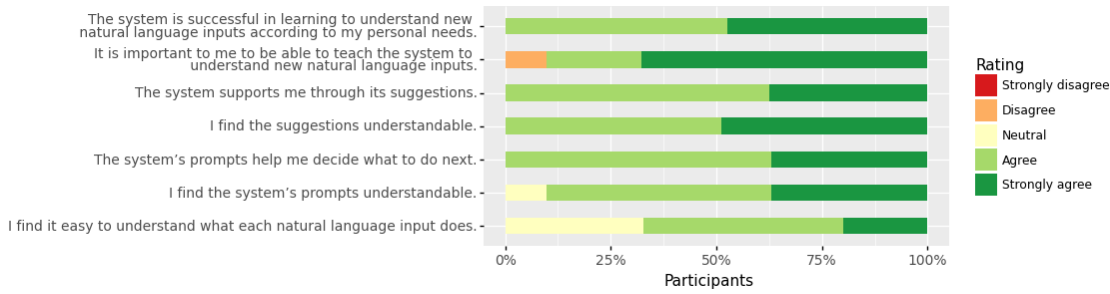


Figure 5.7: Responses to Post-Study Likert-scale Questions about the Experience of Participants with *ContextIT*'s Features.

To get a better understanding of whether participants utilized similar contextual conditions for contextualizing their NL input, we grouped participants for each task together that utilized the same contextual conditions to contextualize that interpretation (see Figure 5.8). The groups only include participants that correctly solved that task. For both interpretations of the NL input `☺ Display all [Data Fields]` participants can be separated into two groups. The biggest group of participants solving the task similarly consists both times of over 60% of the overall participants. However, for the NL input `☺ Select [Aggregate] of [Data Field]` the contextualizations of the interpretations differed more strongly. For the second interpretation (B.2) the biggest group only consisted of two participants.

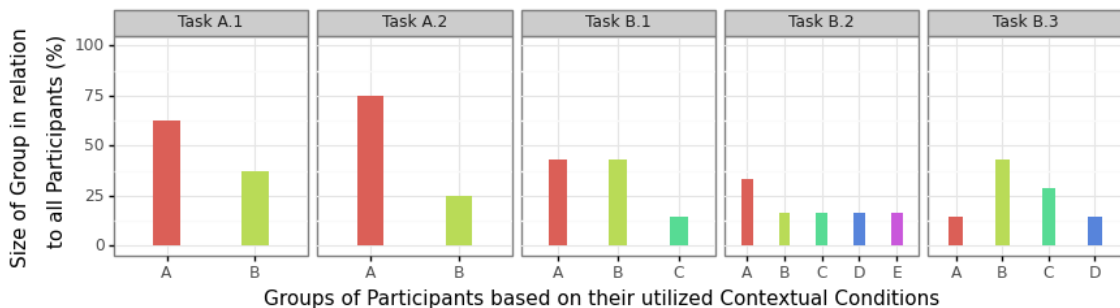


Figure 5.8: Size of the Groups of Participants utilizing the same Contextual Conditions in relation to all Participants across the five Tasks.

#### 5.5.4.2 Effectiveness of Support by *ContextIT*.

All participants found that *ContextIT* supported them in contextualizing the NL inputs through its suggestions ( $M = 4.375$ ) and found the suggestions understandable ( $M = 4.5$ ) (see Figure 5.7) (**DG3**). Two participants (25%) even stated during the think-aloud part that without the suggestion they would not be able to contextualize the NL inputs as it would be difficult for them to grasp the information that *ContextIT* would need.

Participants showed a common process on how they leveraged the suggestions. For each



suggested contextual condition they would first check whether they thought that this contextual condition would be important for contextualizing their interpretation of the NL input. If so, they would check whether they already selected another contextual condition that has the same message on a different abstraction level (e.g., “Amount Invested was in Values” vs. “The Values field was not empty”). If they haven’t previously selected a similar contextual condition or if the current one fits better they would select the new option. If the previously selected contextual condition fits better, they would continue.

This process was additionally highlighted by three participants (37.5%) during the think-aloud study, as exemplified by T1’s statement: *“I had to just follow my natural logic and checking, double checking. Really, that’s the way I’ve worked it out with the suggestions”*.

Among all possible suggestions of *ContexIT* that can include both relevant and irrelevant suggestions, participants successfully chose only 41.7% of the provided suggestions. This provides further evidence that participants make a deliberate choice between similar suggestions on a different abstraction level. Especially in task A.2 where participants received many suggestions with similar messages on different abstraction levels, participants successfully selected 25% of the provided suggestions (see Figure 5.9).

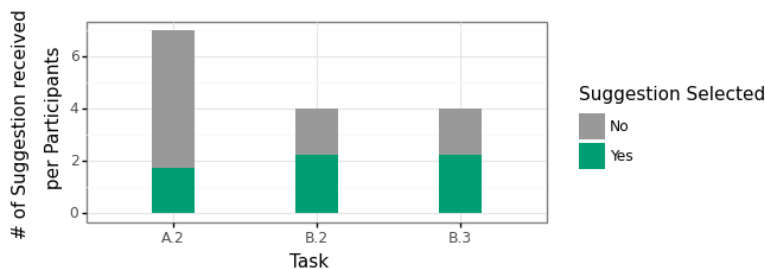


Figure 5.9: Number of Suggestions received per Participant and whether they have been selected for Contextualizing the NL Input for Specification Tasks (A.2, B.2, and B.3.)

However, one participant (12.5%) highlighted that the number of suggestions and the multiple contextual conditions with similar messages but different abstraction levels can also be confusing as it leads to more information that needs to be processed.

Regarding the refinement of existing conflicting interpretations (**DG4**), there are similar results. Participants do not select all options (66.7%) that *ContexIT* provides them when refining the contextual conditions of existing conflicting interpretations (see Figure 5.10). When refining the contextual conditions for B.1 only 58.3% of the provided contextual conditions are utilized.

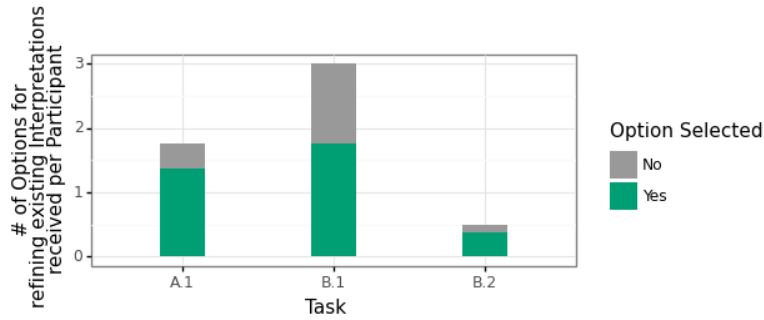


Figure 5.10: Number of Options for refining existing conflicting Interpretations received per Participant and whether they have been selected for differentiating the conflicting Interpretations for the Tasks A.1, B.1, and B.2.

#### 5.5.4.3 Effectiveness of Guidance by *ContextIT*.

The textual aids targeted at guiding users through the contextualization process are both understandable and effective as evidenced by our post-study questionnaire. All participants agreed or strongly agreed that *ContextIT*'s textual aids helped them decide what to do next ( $M = 4.375$ ) and 87.5% agreed or strongly agreed that the textual aids were understandable ( $M = 4.25$ ) (see Figure 5.7). Four participants (50%) specifically highlighted that the textual aids “*display very well and it guides you through*” (T1) and that it “*picks up on when you’ve not been very specific*” (T3).

Furthermore, two participants (25%) explained that the visual aids “*showed me exactly where the [contextual conditions] are coming from*” (T6). They liked that it interactively grounded the textual contextual conditions to the GUI elements.

## 5.6 Discussion

Our formative study provided evidence that users should be enabled to contextualize NL inputs through ITL by specifying contextual conditions that need to be true for the NLI to perform the corresponding interpretation. The results of our think-aloud study show that users are able to accurately contextualize NL inputs with *ContextIT* when they are supported during this process through suggestions as well as visual and textual aids and when they are provided the ability to continuously refine these contextualizations.

#### Mutual Dependence of *ContextIT* and its Users.

While our implemented approach in *ContextIT* to derive contextual conditions as suggestions for users can be improved by gathering training data to rank the assumed importance of these derived contextual conditions, the results of our think-aloud study show that *ContextIT*

can not completely do this without user involvement. Especially as the contextualization can differ across participants as highlighted by the small size of groups of users with similar contextualizations of the interpretations derived in our evaluation for NL input B. Furthermore, the results highlight that most users prefer being involved to provide this input instead of being submitted to possible misinterpretations by *ContextIT*.


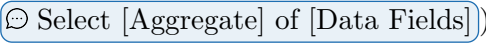
On the other hand, users could not accurately contextualize the NL inputs without support from *ContextIT* as evidenced by our formative studies and the think-aloud study. Hence, ITL-based systems that enable the contextualization of NL inputs need to leverage the advantages of the system to derive an initial foundation for the decision and the ability of the users to be able to accurately select the correct abstraction level of contextual conditions.

#### **More Suggestions are not always better.**

The formative studies and the think-aloud study provide evidence that the suggestions by *ContextIT* help users contextualize the NL inputs. Especially, the feature of providing multiple contextual conditions with similar messages but different abstraction levels has been utilized by participants in their workflow. However, participants also stated that the number of suggestions can be overwhelming.

*ContextIT* therefore has to assess the trade-off between being precise about what *ContextIT* thinks could be relevant to limit information overload and at the same time identifying all the parts of the context that could be relevant since users have issues specifying these parts without suggestions. Previous contextualizations could be utilized in future work to further inform this decision in addition to the currently used information derived from the demonstrated interpretation and the conflicting interpretations.

#### **Learning across Different Users.**

While the contextualization of participants can be easily separated into a few clusters for some NL inputs (e.g., ), contextualizations for other NL inputs (e.g., ) can vary greatly across participants, indicating that contextual conditions could be generalized across users for certain NL inputs. This would further improve the usability of NLI that are able to contextualize their NL inputs through user involvement, as not every participant has to contextualize every ambiguity. Instead, the NLI could learn a default contextualization for the majority of users that might have to be adapted by some individual users. While the small number of

utilized NL inputs in the summative evaluation limits the generalizability of this statement, it would be of interest to investigate in future research how contextualizations can be learned across different users.

## 5.7 Conclusion

While end users are increasingly enabled to teach natural language interfaces (NLIs) how to handle new natural language (NL) inputs through interactive task learning (ITL) both in research prototypes (e.g., (T. J.-J. Li, Azaria, et al., 2017)) and commercial products (e.g., Siri), these NLIs are still unable to learn from users how to correctly leverage the context for the interpretation of their NL input. In this paper, we introduce *ContexIT*, an intelligent agent integrated into a data visualization tool that is able to learn new NL inputs on a syntactic, semantic, and pragmatic level through ITL. Specifically, *ContexIT* supports users in contextualizing their NL inputs through suggestions based on their demonstrated interpretation and on existing conflicting interpretations of the NL input, supports them by enabling users to refine the contextualization of existing interpretations, and guides users during the contextualization through visual and textual aids. Our evaluative study provides evidence that users appreciate the ability to contextualize their NL inputs and the features provided by *ContexIT*. In general, our results show the mutual dependence of the users and *ContexIT* during the contextualization of the NL inputs as each party on their own would not be able to achieve the correct results.

## 6. Discussion

Today's data visualization tools are primarily targeted at supporting business analysts that are interacting with these systems on a frequent basis. While end users with limited technical expertise are also interested in exploring and analyzing data on their own, they are still struggling to effectively use data visualization tools for achieving their goals. To address the challenges end users are facing, natural language interfaces (NLIs) are a promising approach to make the interaction between end users and data visualization tools more intuitive by enabling users to just ask the data visualization tool their questions of interest. However, the design of the interaction between end users and multi-modal systems in general and data visualization tools integrating an NLI in particular is not well understood. Furthermore, further research is required to understand how end users can be enabled to continuously improve the NLIs integrated into these data visualization tools.

In this thesis, I explore the design of effective NLIs for data visualization tools. Specifically, I investigate the two design challenges of effective NLIs for data visualization tools, namely how to facilitate the effective use of data visualization tools integrating NLIs by end users and how to enable users to continuously improve the NLIs to reduce future breakdowns. To address these design challenges, I designed, developed, and evaluated four systems. The results of these studies have several theoretical contributions and practical implications, which I will discuss in the following. Subsequently, I will discuss the major limitations of these studies and propose promising future work that addresses these limitations and go beyond the insights derived from the previous studies.

### 6.1 Theoretical Contributions

This thesis makes several theoretical contributions that are summarized in Table 6.1. First, I will present the theoretical contributions for each study individually and how they address their associated research question. Subsequently, I will summarize the overall theoretical contributions of this thesis.

In studies I and II, I explored how end users can be enabled to effectively use data visualization tools by integrating an NLI. This is important, as end users are interested in performing data exploration and analysis on their own. However, end users are currently limited in effectively gaining insights from data on their own as they do not understand how to properly interact with or navigate data visualization tools. The main reason is that

Table 6.1: Summary of the Theoretical Contributions of this Dissertation.

Study	Theoretical Contributions
<b>Study I</b>	<ul style="list-style-type: none"> <li>- Prescriptive knowledge in the form of a nascent design theory to inform the design of NLIs for data visualization tools increasing their transparent interaction and ultimately the effective use by end users</li> <li>- Identification of design issues when introducing new ways of interaction and prescriptive knowledge on how to address them through conversational onboarding</li> </ul>
<b>Study II</b>	<ul style="list-style-type: none"> <li>- Prescriptive knowledge in the form of three design principles to inform the design of NLIs for data visualization tools increasing the effective use in the context of co-located team interactions</li> <li>- Identification of design issues on how the personal needs of teams require the NLI to be adaptive by the end users themselves</li> </ul>
<b>Study III</b>	<ul style="list-style-type: none"> <li>- Prescriptive knowledge in the form of design goals to inform the design of NLIs for data visualization tools that support end users to effectively teach the NLI how to perform new NL inputs</li> <li>- Identification of design issues on how end users leverage the assistance by the NLI during the teaching process</li> </ul>
<b>Study IV</b>	<ul style="list-style-type: none"> <li>- Prescriptive knowledge in the form of design goals to inform the design of NLIs for data visualization tools that support end users to effectively contextualize NL inputs</li> <li>- Identification of design issues on how the mutual dependence of the end user and the NLI have to be addressed to translate it into a synergetic interaction</li> </ul>

existing data visualization tools still focus on supporting the data analysis and exploration of business analysts or other experts and not users with limited technical expertise. To fill the existing research gaps, I conducted studies I and II to investigate a theory-driven design of data visualization tools for end users in different usage scenarios, namely in crisis response and in co-located team interactions.

**In study I**, I addressed the challenges of end users to gain insights from data visualization tools during crisis response. Particularly, I derived a theory-driven design for data visualization tools for crisis response integrating an NLI to improve end users' transparent interaction with the data visualization tool and, as a result, its effective use. To accomplish this, I draw on the theory of effective use (TEU) (Burton-Jones & Grange, 2013), which builds on the premise that rather than just being used, systems must be used effectively to obtain maximum benefits. Burton-Jones and Grange (2013) conceptualize effective use as an aggregate construct with three hierarchical dimensions: (1) transparent interaction, (2) representational fidelity, and (3) informed action. In this study, I focused on transparent interaction, as this is the fundamental dimension that is necessary for the system to be used effectively. To improve the transparent interaction Burton-Jones and Grange (2013) identified two major factors that act as drivers of transparent interaction, namely both adapting and learning the surface structure through which end users are interacting with the system. Based on these two drivers of transparent interaction I derived two meta-requirements for the design of a data visualization tool integrating an NLI. Subsequently, I proposed three design principles (DPs) from the meta-requirements and instantiated them in an artifact. The DPs are translated into testable design propositions to evaluate the design in a large-scale online experiment with 292 participants. The results of the evaluation provide evidence that the design increases end users' transparent interaction with the data visualization tool by integrating an NLI to provide a more intuitive way of interacting, ultimately improving the effectiveness and efficiency of the end users. I synthesized the findings into a nascent design theory for data visualization tools for crisis response that integrate an NLI. Furthermore, the results highlight the trade-off for introducing new ways of interacting to end users. Particularly, I highlight how end users can have issues in correctly choosing which way of interacting (e.g., NL or pointing device) to use for which task when provided with multiple to choose from. In the nascent design theory, I introduce one way of addressing this issue, by providing end users an initial conversational onboarding to introduce them to the benefits of each way of interacting. This approach has been shown to significantly improve the transparent interaction of end users when provided with multiple ways of interacting with a data visualization tool.

**In study II**, I investigated how end users can be supported in co-located teams to effectively use data visualization tools for their decision-making. Particularly, I derived a theory-driven design for data visualization tools for co-located teams that improves the effective use of the data visualization tool by providing end users the ability to interact either through touch or speech. Consistently with study I, I draw on the TEU (Burton-Jones & Grange, 2013) to inform the design. Particularly, I derived two meta-requirements from the TEU that posit that users need to be provided with unimpeded access to the system’s underlying data (transparent interaction) and need to be enabled to obtain faithful data visualizations (representational fidelity) during the discussion of the decision in the co-located team. I translated the two meta-requirements into three DPs and instantiated them into a data visualization tool that enables users to interact with the data visualizations through touch and speech on a large interactive display. The results of the evaluation of the DPs in a focus group provide evidence that the combination of touch and speech for multi-modal data visualization tools provides teams with additional possibilities to interact properly based on the team characteristics and context. Furthermore, the evaluation sheds light on additional design issues, which offer valuable starting points for future improvements, such as the need for the team to be able to improve the NLI integrated into the data visualization tool and to adapt the NLI to their personal needs as the individual characteristics among teams can vary greatly.

In studies III and IV, I explored how end users can be enabled to effectively teach NLIs integrated into data visualization tools how to handle new NL inputs. While users are interested in personalizing these NLIs (T. J.-J. Li, Azaria, et al., 2017) and are increasingly provided with the ability to improve NLIs, such as Siri (Apple, 2022) or Google Assistant (Google, 2019), end users still struggle to effectively use the existing approaches to teach these NLIs, such as visual programming languages (João et al., 2019; Myers, Ko, Scaffidi, et al., 2017). To address this issue, I conducted studies III and IV to investigate the design of interactive task learning (ITL)-based NLIs integrated into data visualization tools.

**In study III**, I investigated how to assist end users in effectively teaching new NL inputs through ITL. Particularly, I designed a multi-modal ITL approach to enhance existing NLIs iteratively through programming-by-demonstration and NL programming instantiated in ONYX. End users are supported by suggestions provided during the demonstration process to give users insight into the existing knowledge of ONYX and through follow-up questions by ONYX to clarify ambiguities in the direct manipulation demonstrations of end users. While the participatory design of ONYX highlighted the relevance of this assistance, the



summative evaluation further provided evidence that the features implemented in ONYX improve the effectiveness of the teaching by end users. Hence, I contribute through this study with design goals (DGs) that enable end users to effectively teach an ITL-based NLI integrated into a data visualization tool. Furthermore, the study sheds light on the challenges of end users in interacting with ITL-based NLIs. For example, more assistance is not always better, as participants of the evaluation highlighted the importance of making the reason for the assistance explicit. Additionally, users need to learn how to properly utilize the assistance as they are not yet used to teaching NLIs how to perform new NL inputs in their everyday life. This study, therefore, provides interesting insights for future ITL-based systems to build on my approach.

**In study IV**, I investigated how to enable users to effectively contextualize their NL inputs through ITL. Particularly, I design and developed *ContexIT*, an intelligent agent integrated into a data visualization tool that is able to learn new NL inputs on a syntactic, semantic, and pragmatic level through ITL. This enables *ContexIT* to differentiate between multiple conflicting interpretations of an NL input based on the current context of usage. End users are able to teach *ContexIT* the contextual conditions for their interpretations by defining pre-conditions that need to be true for their demonstrated interpretation of the NL input to be the correct interpretation. In the evaluation of the design of *ContexIT*, I provide evidence that the support by *ContexIT* during the contextualization of the NL input enables users to effectively teach the system. Especially, the suggestions provided by *ContexIT* for possible pre-conditions to differentiate existing interpretations and the ability to continuously refine the understanding of *ContexIT* have been suggested to be effective. Furthermore, in this study, I investigated the mutual dependence of the end users and the system on each other and how important it is to leverage the strengths of both parties and to clearly communicate the current understanding to translate this mutual dependence into a synergistic interaction.

**In summary**, all individual studies provide prescriptive knowledge to shape future research on designing data visualization tools integrating NLIs to support their effective use by end users. With studies I and II, this thesis provides a solid theoretical foundation for data visualization tools integrating an NLI and how to facilitate the interaction by end users by including the users' perspective utilizing a theoretical lens, namely the TEU (Burton-Jones & Grange, 2013). In both studies, I was able to show that the design for extending the data visualization tool with an NLI and the derived prescriptive knowledge can lead to an increase in effective use by end users. In studies III and IV, I focussed on

enabling the effective teaching of new NL inputs by end users to the NLI in the context of data visualization tools. Again, in both studies, I was able to provide evidence that the prescriptive knowledge can be utilized to design NLIs integrated into data visualization tools that enable end users to effectively teach both new NL inputs as well as contextual conditions. Across all studies, I demonstrated how important it is to not only focus on the technological aspect of the interaction with NLIs but to further include the users' perspective, such as previous experience with technology. This is especially important because the studies showed that the end users and the NLIs for data visualization tools have a mutual dependence on each other, which needs to be considered in the design of their interaction.

## 6.2 Practical Implications

This thesis offers important practical implications for the design of effective NLIs for data visualization tools. **First**, with studies I and II, I provide high-level guidance that developers and providers of data visualization tools can utilize as a blueprint for developing data visualization tools with NLIs for end users. Specifically, the in-depth description can be leveraged by developers as actionable guidance to instantiate effective NLIs for data visualization tools. Furthermore, common approaches across the systems introduced in this thesis can be utilized to develop and provide NLIs for data visualization tools for more diverse use cases. **Second**, the results of this thesis show that providing effective NLIs for data visualization tools is not only achieved by integrating a technologically functional NLI into a data visualization tool but that the users' perspective has to be considered and the design adapted accordingly. I, therefore, provide several practical insights that developers of NLIs for data visualization tools can draw upon. While study I shows that end users have issues selecting the appropriate way of interacting for the task at hand, the explained design of study I demonstrates how this issue can be addressed by developers through conversational onboarding. Furthermore, I describe actionable guidance on how this feature can be implemented. **Third**, I provide foundational design knowledge to enable users to effectively teach the NLIs integrated into data visualization tools. This knowledge can be directly utilized in commercial data visualization products, such as Tableau's AskData, to enable the NLI to learn from user involvement after the NLI fails to correctly interpret the NL input of users. **Fourth**, I demonstrate the importance of considering the context of the system in the interpretation of NL inputs by NLIs. Furthermore, I showcase how developers can enhance existing NLIs through ITL to enable these NLIs to learn from end users themselves how the context has to be considered in the interpretation of NL inputs.

## 6.3 Limitations and Future Research

While all studies were conducted in a rigorous manner, they still have limitations. Hence, I will explain the limitations of the studies of this thesis and highlight the future work that these limitations open up.

### 6.3.1 Large-Scale Quantitative Evaluation & Field Studies

Studies II and IV focused on qualitatively evaluating the systems with participants that are similar to the end users that would utilize these data visualization tools in the real world. While these qualitative evaluations provided a deeper understanding of how end users utilize the designs introduced in these studies, I was not able to evaluate the two NLIs for data visualization tools on a larger scale based on quantitative behavioral data. In studies I and III, I utilized both qualitative and quantitative approaches to evaluate the designs which provided us with additional insights that were valuable to better understand the effectiveness of the designs and enabled me to derive more generalizable insights from the evaluation. Hence, a promising future avenue would be to evaluate the artifacts of studies II and IV in a large-scale quantitative evaluation to generate more generalizable insights.

All studies were conducted in an artificial environment with pre-selected tasks. Furthermore, apart from study II, which was conducted with employees from a large European energy provider, all other studies were conducted with crowd workers that had similar characteristics to the targeted end users. While controlling for the task and context of use allowed me to evaluate the innovative features of my artifacts more closely, conducting a subsequent evaluation in the field could provide additional insights into how end users would use the artifacts in everyday life, which issues they are facing and especially the longitudinal implications of the design. Additionally, this would enable me to strengthen the external validity of the insights identified during my evaluations. For example, this could provide insights into how end users reuse NL inputs they have taught in the artifacts of studies III and IV. Furthermore, this could allow researchers to more closely investigate when users choose to adapt the artifact, switch between the modalities provided by the artifact for interaction, and when they would adapt themselves by just learning to say the NL inputs the way the NLI already understands. Hence, conducting a field study with the artifacts introduced in this thesis could lead to a deeper understanding of how users utilize NLIs for data visualization tools and how their interaction evolves over time. In order to achieve this, the features of the artifacts of studies I-IV would need to be introduced into

real-world data visualization tools, such as PowerBI or Tableau, that the end users are already interacting with on a regular basis to go beyond the artificial environment and to get externally valid insights into their interactions with the NLI for data visualization tools.

### **6.3.2 Improving the NL Processing**

In my studies, I utilized for the NL processing both machine learning-based approaches (studies I and II) and grammar-based approaches (studies III and IV). While I collected an initial sample of exemplary NL inputs for the machine learning-based NLI, their accuracy could further be improved by conducting a larger NL input elicitation study for the tasks that the NLI should support. The grammar-based approaches could further be improved by utilizing additional information in the pre-processing of the NL inputs. Neither synonyms nor antonyms are currently utilized to better interpret unknown NL inputs in my artifacts in studies III and IV. Hence, the knowledge provided by ontologies (e.g., WordNet, VerbNet) could be utilized to further improve the accuracy of the grammar-based NLI. This increase in the accuracy of the NLI is only a technological improvement of the artifacts. However, it could further influence how end users utilize the NLI for data visualization tools and especially how and when they switch between the modalities provided by the artifacts.

Furthermore, all NLI implemented are only able to process stand-alone NL inputs and not NL inputs that build onto each other (e.g., “Highlight Florida in the visualization”  $\implies$  “Now remove everything except for that”). To enable this, the NLI would require memory of the previous interactions and the utilized actions, and more importantly, the NLI would need to understand pronouns and reference words (e.g., this, that, the previous) and to which element they are referencing. While this is an issue that could be more easily implemented in the artifacts of studies I & II by adding additional processing rules, this would be a greater challenge for the ITL-based NLI. Particularly because it would be harder for users to demonstrate during the ITL process what reference word is connected to which previously utilized action or element and how the NLI should generalize this for future interactions.

### **6.3.3 Improving the Interaction with the NLI**

End users with limited experience with the technology often have issues knowing exactly what to ask the NLI or provide vague NL inputs. Therefore, NLI increasingly support end users through suggestions, such as autocompletion, and recommendations of what NLI they could use next (Srinivasan, Nyapathy, et al., 2021). While both the NLI

developed in studies III & IV provide suggestions after the NLI breaks down due to a missing understanding of what actions to perform based on the NL input, these suggestions could still be improved and additional assistance provided before the NLI even breaks down.

Before entering the NL input, users could be additionally supported through suggestions of possible NL inputs of interest. While previous research showed that these suggestions based on a well-curated set of NL inputs can help users during the interaction with an NLI (Srinivasan, Nyapathy, et al., 2021), it is not well understood how users could be provided with suggestions by the NLI based on a continuously growing set of NL inputs in which it is unknown how high the quality of the trained NL inputs is and which NL inputs could be of interest in which situations. Hence, a new metric would need to be derived that indicates how relevant an NL input is in the current situation of the user based on factors such as the previous interactions, the current state of the visualization, how often the NL input is utilized, and by whom it was taught to the NLI.

For suggestions after a breakdown, the NLIs in studies III & IV both utilize their internal knowledge and the previously trained NL inputs to derive suggestions. However, it could be additionally of interest how ITL-based NLIs could utilize general knowledge about the verbs used and the background of the dataset to provide better suggestions when the NLI breaks down. Furthermore, conversations are bi-directional and hence the NLI could ask specific follow-up questions to elicit the goal of users or their current background to utilize this information in deriving more relevant suggestions after a breakdown.

#### **6.3.4 Integrating Richer Context into the NLI's Interpretation**

Apart from the NLI for data visualization tools in study IV, none of my NLIs utilize the current context in the NLI's interpretation of an NL input. However, as highlighted in study IV, context is important to accurately understand the actual goal of a user's NL input (Reinhart, 1981; Setlur, Battersby, et al., 2016). Context includes among others the characteristics of users (e.g., personality, experience), the situation of the system (e.g., current state, size of the screen, previous interactions), the physical context (e.g., noise level), and the time context (e.g., time of day or month). While I demonstrated in study IV how the current state of the data visualization tool (part of the context of the system) can be utilized in the interpretation of an NL input by the NLI and how users can teach NLIs to correctly utilize this type of context, future work should investigate which additional contexts of the interaction are also important for the correct interpretation of an NL

input and further, how users can be enabled to contextualize the NL inputs utilizing these contexts. Promising contexts for future works are especially the individual characteristics of users and understanding how user behavior in the form of previous interactions across modalities can influence the interpretation of future NL inputs as both greatly influence users' interactions with NLI (e.g., follow-up questions by end users building on previous results).

### **6.3.5 Extending the Tasks Supported by the ITL-based NLI for Data Visualization Tools**

Currently, the NLIs introduced in this thesis are able to change the selected data fields in the data visualization, filter for numeric and categorical values (studies I - IV), change the visualization (studies III & IV), and adapt the aggregation operation (e.g., max, min, average,...) of the data fields (study IV). However, data visualization tools can integrate even more functionalities that could be supported by the NLI, such as changing the color of certain points (e.g., Y. Wang et al., 2022) and ordering points based on value or name (e.g., Srinivasan, Nyapathy, et al., 2021). This is especially important when considering the real-world application of the ITL-based NLIs for data visualization tools. Users expect the NLI to be able to perform all tasks that they would be able to perform in traditional GUI-based data visualization tools (Tory & Setlur, 2019). Hence, future work should investigate whether the approaches for ITL-based NLIs introduced in studies III and IV and tested on a subset of all possible functionalities of data visualization tools are still applicable to the functionalities that have not been included in the data visualization tools.

### **6.3.6 Switching more easily between Datasets**

While the datasets implemented in the data visualization tools differed between studies and encompassed datasets on the topic of COVID-19, renewable energy projects, and sales forecasting, users were not able to change the dataset on their own during the studies. In GUI-based data visualization tools, this functionality is comparably straightforward as the GUI does not need to be adapted if different datasets are used and only the measures and dimensions available for the user to select from have to be altered. However, with NLIs this process is more complicated since to work properly, the NLI requires information regarding the measures and dimensions as well as synonyms and abbreviations for the terms included in the dataset. While this task is usually performed in the real world by data analysts and developers, this would not be feasible for ITL-based NLIs as the main goal of these NLIs is to make end users independent of data analysts and developers. Therefore, when provided

with a new dataset, the NLI should first extract relevant information from the dataset, such as whether a data field is numerical, temporal, or categorical based on the structure of the data. Furthermore, after extracting all relevant terms from the dataset, the NLI could utilize general knowledge, such as by crawling the web, to extract abbreviations and synonyms for these terms. Subsequently, the end users could be asked to clarify terms where the system is unsure if it extracted the correct information. Finally, the NLI would need to learn whether procedural and declarative knowledge it learned on a different dataset still applies in the same way to the current dataset or whether their meaning has shifted. How all of this could be integrated into an ITL-based NLI for data visualization tools is not well understood and calls for future research.

### **6.3.7 ITL-based NLIs beyond Data Visualization Tools**

In this thesis, I designed and developed effective NLIs for data visualization tools. However, the advantages of NLIs are not limited to the domain of data exploration and analysis. Increasingly GUI-based tools, such as Photoshop (Y.-S. Kim et al., 2019) or proprietary tools (S. I. Wang et al., 2017) are enriched through NLIs. As these domains share common characteristics with my domain, such as clear pre-defined functionalities and a state of the system that can be translated into structured data, I believe that the insights generated in this thesis can be translated to these domains. However, while features like suggestions and follow-up questions (study III) could be more easily translated, translating the process of contextualizing the NL inputs (study IV) could be more difficult as the way people interact with tools differs greatly across domains. Hence, it would be of interest to understand how these features could be applied in various domains and which insights could be generalized or need to be adapted.

### **6.3.8 Connecting Natural Language and Pointing Devices more tightly**

Systems that provide multiple modalities for users to interact with can be categorized based on whether the two modalities can be used complementary (e.g., clicking on something and then saying something) or if the modalities are equivalent to each other (e.g., performing a task either by entering an NL input or by clicking somewhere) (Coutaz et al., 1995). The artifacts introduced in the studies of this thesis can be assigned to the equivalence category, as users are able to complete a task either by using an NL input or by using the GUI elements of the data visualization tools. While previous research also investigated the usability of providing NL-based interaction and touch as complementary modalities (e.g., Srinivasan, Lee, et al., 2020), to the best of my knowledge no study has investigated

how end users could be enabled to teach the NLI to interpret complementary interactions across multiple modalities through ITL in a data visualization tool. A challenge in this endeavor is to clearly define the trigger when the data visualization tool should perform which task and which information to utilize from which modality. In the data visualization tools, the trigger could consist of only NL, only direct manipulation of the GUI elements, and a combination of both, such as uttering “Remove these” and then clicking data points in the visualization. Since the artifacts in studies III and IV only addressed the possibility of triggering a task either through NL or through direct manipulation of the GUI elements, future research could investigate how end users could be supported in specifying when and which task the data visualization tool should trigger based on a complementary combination of both modalities.

### **6.3.9 Beyond the Combination of Natural Language and Pointing Devices**

In this thesis, I investigated how to design effective NLIs for data visualization tools. Hence, my thesis focuses on providing users the ability to interact with the data visualization tool through NL, touch, and pointing devices. However, due to various technological advances, users are nowadays also able to interact with systems through mid-air gestures (Lee, Srinivasan, et al., 2021), eye-tracking (Toreini et al., 2022), and other modalities (Jaimes & Sebe, 2007). Hence, future work should investigate which of this plethora of possible combinations would be best suitable to enable end users to effectively use a data visualization tool and how end users could be supported in effectively teaching the data visualization tool to correctly interpret their actions in these new modalities.

### **6.3.10 Combining ITL and Machine Learning-based Approaches for NLIs**

Machine learning-based NLIs, such as GPT-3, have the benefit that they have some kind of solution to most users’ NL inputs. However, they still have to possibility of incorrectly interpreting the NL inputs and would require a lot of training data to correct this incorrect interpretation in the language model. Additionally, end users seldom receive insights into why the machine learning-based NLI failed and how the users could adapt the NLI or themselves to avoid future mistakes. ITL-based NLIs that learn a grammar-based representation of the learned NL inputs, on the other hand, do not have an interpretation for all NL inputs and can not scale their existing knowledge across different situations as easily as machine learning-based NLIs. However, they can learn new NL inputs based on one example and can express the reason why a mistake occurred. Hence, future work should investigate how to combine the two approaches in one NLI. This would enable the



NLI to have a general understanding of most NL inputs while at the same time having accelerated learning when users correct the interpretation of the NLI.

### **6.3.11 Learning Across Different Users**

Both artifacts that enable end users to effectively teach NLIs to perform new NL inputs (studies III & IV) focus on personalizing the language model to one individual user. Hence, if the NLI breaks down due to an incorrect interpretation of an NL input and a user corrects this interpretation either by demonstrating a new interpretation (studies III & IV) or by contextualizing the NL input (study IV) then only this user will benefit from the correction. If another user would have the same breakdown, then this additional user would have to correct the NLI again, even though the first user already provided a possible solution. To enable the NLI to learn across users, future work should investigate when a correction of the NLI should be included in the language model utilized by all users, when it should only be provided for a subgroup of users and when a correction should only be applicable for the users that taught the NLI the correction. For this endeavor, future work should utilize the similarity between users to decide whether to suggest a correction that was taught by a user to a different user when similar situations occur as well as how specific the new interpretation or contextualization of an NL input is. However, a major challenge in this endeavor would be to preserve the privacy of users. End users are often hesitant in sharing their demonstrations since personal information could be included in their demonstrations (T. J.-J. Li, Chen, et al., 2020). Hence, ITL-based NLIs that could learn new NL inputs across different users would need to address these privacy concerns by giving users ownership over their personal information leveraged in learning from their demonstrations and enabling them to either abstract this personal information or completely remove it. This would require the ITL-based NLI to understand which information to share across users and which to limit to a specific user to prevent data leakage.

## 7. Conclusion

This thesis explores how to design effective natural language interfaces (NLIs) for data visualization tools. Therefore, I investigated how data visualizations can be extended by an NLI to improve the effective use by end users and how end users can be enabled to effectively teach these NLIs to perform new natural language (NL) inputs. This is an important endeavor as while end users are increasingly interested in exploring and analyzing data on their own, existing data visualization tools do not support end users who have limited knowledge of the technology in effectively using these data visualization tools. Hence, understanding how extending the traditional GUI-based data visualization tools through more intuitive ways of interaction, such as NL, can help provide these end users better access to the data and ultimately improve the effectiveness of their decision-making. Particularly, this thesis outlines and addresses two design challenges of existing NLIs for data visualization tools. First, existing NLIs for data visualization tools still target users, such as business analysts who are familiar with data visualization tools, and therefore the special design requirements of end users are not included in the design of existing NLIs for data visualization tools. Second, as errors still occur in current NLIs for data visualization tools, end users need to be enabled to address these errors and improve the NLIs for future interactions. However, only little work exists that focuses on enabling end users to teach NLIs for data visualization tools how to handle new NL inputs. Hence, this thesis contributes with four studies that address these design challenges and provide insight into the related research questions. Particularly, studies I and II focus on providing prescriptive knowledge on how to design an NLI for data visualization tools that can be effectively used by end users and evaluate the theory-driven designs both in a large-scale online experiment and qualitative studies. Studies III and IV contribute with prescriptive knowledge on how end users can be enabled to effectively teach the NLI how to handle new NL inputs through interactive task learning (ITL) and to teach the contextual pre-conditions of the users' interpretation of the NL input. Again, the designs are evaluated both in an online experiment and qualitative studies to get a deeper understanding of how end users utilize the design instantiated in the artifacts. While this thesis also opens up many opportunities for future research, the work provides an important step towards designing effective NLIs for data visualization tools that can be both effectively used by end users and enable end users to effectively teach the NLIs to handle new NL inputs.

## 8. Appendix

### A Study I: Additional Material for User Evaluation (Online Experiment)

Table A.1: Demographic Information of Participants

		N	%
Gender	Female	121	44.6%
	Male	150	55.4%
Age	18 – 24 years	19	7%
	25 – 34 years	93	34.3%
	35 – 44 years	87	32.1%
	45 – 54 years	41	15.1%
	55+ years	31	11.5%
Education	High School	50	18.5%
	Technical, trade, or business after high school	45	16.6%
	Bachelor’s degree	136	50.2%
	Master’s degree	31	11.4%
	Doctoral degree or professional degree (JD,MD)	9	3.3%
Experience with Dashboards	Never	31	11.4%
	1-2 times a year	46	17%
	1-2 times a month	73	26.9%
	1-2 times a week	69	25.5%
	daily	52	19.2%
Experience with Conversational User Interfaces	Never	25	9.2%
	1-2 times a year	20	7.4%
	1-2 times a month	42	15.5%
	1-2 times a week	82	30.3%
	daily	102	37.6%
Computer Self-Efficacy	M = 6.01 (SD = 1.03)		

Table A.2: Experimental Tasks

1.	Which of the following states had the fewest counties with more than 40,000 confirmed cases on September 29th? (Texas, California, Florida)
2.	Which of the following states had the largest increase in total cases between May 15th and August 31st? (Idaho, Kentucky)
3.	Which of the following region had more cases as of October 25th? (West, Midwest)
4.	How many people had died in Wisconsin, Nebraska, Idaho, and Connecticut combined until May 3rd, 2020?

Table A.3: Calculation of Transparent Interaction based on Users' Navigation Path: Examples

Navig. Steps	Quickest Path Task 2 (Mouse)	Actual Navigation Path Taken (P5004790 – TDB Condition)	Level of Transparent Interaction
1	Filter for Idaho	Drill-down for Idaho	TI = 5/6 = 0.83
2	Filter for 2020-05-15	Filter for 2020-05-15	
3	Filter for Kentucky	Filter for 2020-08-31	
4	Filter for 2020-08-31	Zoom Out	
5	Filter for Idaho	Drill-down for Kentucky	
6		Filter for 2020-05-15	
Navig. Steps	Quickest Path Task 2 (Natural Language)	Actual Navigation Path Taken (P.9109728 CDB-NLO Condition)	Level of Transparent Interaction
1	Filter for Idaho and 2020-05-15	Filter for Idaho	TI = 4/6 = 0.67
2	Filter for Kentucky	Filter for Kentucky	
3	Filter for 2020-08-31	Filter for 2020-05-15	
4	Filter for Idaho	Filter for Idaho	
5		Filter for 2020-08-31	
6		Filter for Kentucky	

## **B References to Code Repositories, Study Procedures, and Data sets**

The code of the prototypes, study procedures, and data sets of the four major studies of this thesis are available (upon request) on the research data repository of KIT.

### **B.1 Study I:**

**Code:**

<https://git.scc.kit.edu/issd/research/marcel-ruoff-dissertation/Conversational-COVID-19-Dashboard>.  
git

**Study Protocol/Data set:**

<https://radar.kit.edu/radar/en/dataset/okPFEQIIHvnnwJxJ>

### **B.2 Study II:**

**Code:**

<https://git.scc.kit.edu/issd/research/marcel-ruoff-dissertation/ConversationalDashboardBot>.  
git

**Study Protocol/Data set:**

<https://radar.kit.edu/radar/en/dataset/ohfTVtZVJODsZcPr>

### **B.3 Study III:**

**Code:**

<https://git.scc.kit.edu/issd/research/marcel-ruoff-dissertation/Conversational-ITL>.git  
[https://git.scc.kit.edu/issd/research/marcel-ruoff-dissertation/study3\\_bot](https://git.scc.kit.edu/issd/research/marcel-ruoff-dissertation/study3_bot).git

**Study Protocol/Data set:**

<https://radar.kit.edu/radar/en/dataset/PiLQtrsVDFtzwuyG>

### **B.4 Study IV:**

**Code:**

[https://git.scc.kit.edu/issd/research/marcel-ruoff-dissertation/Study4\\_System](https://git.scc.kit.edu/issd/research/marcel-ruoff-dissertation/Study4_System).git  
[https://git.scc.kit.edu/issd/research/marcel-ruoff-dissertation/study4\\_bot](https://git.scc.kit.edu/issd/research/marcel-ruoff-dissertation/study4_bot).git

**Study Protocol/Data set:**

<https://radar.kit.edu/radar/en/dataset/kozWmsajrBDobrYb>

# Bibliography

- Abbasi, A., Sarker, S., & Chiang, R. H. (2016). Big data research in information systems: Toward an inclusive research agenda. *Journal of the Association for Information Systems*, *17*(2), 1–32. <https://doi.org/10.17705/1jais.00423>
- Abelló, A., Darmont, J., Etcheverry, L., Golfarelli, M., Mazón, J. N., Naumann, F., Pedersen, T. B., Rizzi, S., Trujillo, J., Vassiliadis, P., & Vossen, G. (2013). Fusion cubes: Towards self-service business intelligence. *International Journal of Data Warehousing and Mining*, *9*(2), 66–88. <https://doi.org/10.4018/jdwm.2013040104>
- Ahmad, R., Siemon, D., Gnewuch, U., & Robra-Bissantz, S. (2022). Designing Personality-Adaptive Conversational Agents for Mental Health Care. *Information Systems Frontiers*, *1*, 1–21. <https://doi.org/10.1007/s10796-022-10254-9>
- Allen, J., Chambers, N., Ferguson, G., Galescu, L., Jung, H., Swift, M., & Taysom, W. (2007). PLOW: A collaborative task learning agent. *Proceedings of the National Conference on Artificial Intelligence*, *2*, 1514–1519. <https://doi.org/10.5555/1619797.1619888>
- Apple. (2022). Shortcuts and Suggestions - Siri - Human Interface Guidelines - Apple Developer. <https://developer.apple.com/design/human-interface-guidelines/siri/overview/shortcuts-and-suggestions/>
- Arazy, O., Kumar, N., & Shapira, B. (2010). A Theory-Driven Design Framework for Social Recommender Systems. *Journal of the Association for Information Systems*, *11*(9), 455–490. <https://doi.org/10.17705/1jais.00237>
- Ashktorab, Z., Jain, M., Liao, Q. V., & Weisz, J. D. (2019). Resilient chatbots: Repair strategy preferences for conversational breakdowns. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–12. <https://doi.org/10.1145/3290605.3300484>
- Azaria, A., Krishnamurthy, J., & Mitchell, T. M. (2016). Instructable intelligent personal agent. *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2681–2689.
- Azaria, A., Srivastava, S., Krishnamurthy, J., Labutov, I., & Mitchell, T. M. (2020). An agent for learning new natural language commands. *Autonomous Agents and Multi-Agent Systems*, *34*(1), 6. <https://doi.org/10.1007/s10458-019-09425-x>

- Badam, S. K., Amini, F., Elmqvist, N., & Irani, P. (2016). Supporting visual exploration for multiple users in large display environments. *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 1–10. <https://doi.org/10.1109/VAST.2016.7883506>
- Badam, S. K., & Elmqvist, N. (2019). Visfer: Camera-based visual data transfer for cross-device visualization. *Information Visualization*, *18*(1), 68–93. <https://doi.org/10.1177/1473871617725907>
- Barki, H., Titah, R., & Boffo, C. (2007). Information System Use-Related Activity: An Expanded Behavioral Conceptualization of Individual-Level Information System Use. *Information Systems Research*, *18*(2), 173–192. <https://doi.org/10.1287/isre.1070.0122>
- Benke, I., Knierim, M. T., & Maedche, A. (2020). Chatbot-based Emotion Management for Distributed Teams. *Proceedings of the ACM on Human-Computer Interaction*, *4*(CSCW2), 1–30. <https://doi.org/10.1145/3415189>
- Berthold, H., Rösch, P., Zöllner, S., Wortmann, F., Carenini, A., Campbell, S., Bisson, P., & Strohmaier, F. (2010). An architecture for ad-hoc and collaborative business intelligence. *Proceedings of the 1st International Workshop on Data Semantics - DataSem '10*, 1. <https://doi.org/10.1145/1754239.1754254>
- Bolt, R. A. (1980). "Put-that-there": Voice and gesture at the graphics interface. *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1980*, 262–270. <https://doi.org/10.1145/800250.807503>
- Booth, T., & Stumpf, S. (2013). End-user experiences of visual and textual programming environments for Arduino. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7897 LNCS*, 25–39. [https://doi.org/10.1007/978-3-642-38706-7\\_{\\\_}4](https://doi.org/10.1007/978-3-642-38706-7_{\_}4)
- Bostock, M., Ogievetsky, V., & Heer, J. (2011). D<sup>3</sup> Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, *17*(12), 2301–2309. <https://doi.org/10.1109/TVCG.2011.185>
- Buhrmester, M., Kwang, T., & Gosling, S. D. (2011). Amazon's Mechanical Turk. *Perspectives on Psychological Science*, *6*(1), 3–5. <https://doi.org/10.1177/1745691610393980>
- Burstein, F., W. Holsapple, C., Carlsson, S. A., & El Sawy, O. A. (2008). Decision Support in Turbulent and High-Velocity Environments. In *Handbook on decision support systems 2* (pp. 3–17). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-48716-6\\_{\\\_}1](https://doi.org/10.1007/978-3-540-48716-6_{\_}1)

- Burton-Jones, A., & Grange, C. (2013). From Use to Effective Use: A Representation Theory Perspective. *Information Systems Research*, 24(3), 632–658. <https://doi.org/10.1287/isre.1120.0444>
- Burton-Jones, A., & Volkoff, O. (2017). How Can We Develop Contextualized Theories of Effective Use? A Demonstration in the Context of Community-Care Electronic Health Records. *Information Systems Research*, 28(3), 468–489. <https://doi.org/10.1287/isre.2017.0702>
- Cambre, J., Williams, A. C., Razi, A., Bicking, I., Wallin, A., Tsai, J., Kulkarni, C., & Kaye, J. (2021). Firefox Voice: An Open and Extensible Voice Assistant Built Upon the Web. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–18. <https://doi.org/10.1145/3411764.3445409>
- Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think* (S. K. Card, J. D. Mackinlay, & B. Shneiderman, Eds.). Morgan Kaufmann Publishers Inc.
- Castelli, N., Ogonowski, C., Jakobi, T., Stein, M., Stevens, G., & Wulf, V. (2017). What happened in my home?: An end-user development approach for smart home data visualization. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, 2017-May*, 853–866. <https://doi.org/10.1145/3025453.3025485>
- Cay, D., Nagel, T., & Yantac, A. E. (2020). Understanding User Experience of COVID-19 Maps through Remote Elicitation Interviews. *2020 IEEE Workshop on Evaluation and Beyond - Methodological Approaches to Visualization (BELIV)*, 65–73. <https://doi.org/10.1109/BELIV51497.2020.00015>
- Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, 36(4), 1165–1188. <https://doi.org/10.2307/41703503>
- Chen, L., Li, X., Yang, Y., Kurniawati, H., Sheng, Q. Z., Hu, H.-Y., & Huang, N. (2016). Personal health indexing based on medical examinations: A data mining approach. *Decision Support Systems*, 81, 54–65. <https://doi.org/10.1016/j.dss.2015.10.008>
- Chen, X., Self, J. Z., House, L., Wenskovitch, J., Sun, M., Wycoff, N., Evia, J. R., Leman, S., & North, C. (2018). Be the Data: Embodied Visual Analytics. *IEEE Transactions on Learning Technologies*, 11(1), 81–95. <https://doi.org/10.1109/TLT.2017.2757481>
- Cheng, C. K., Ip, D. K., Cowling, B. J., Ho, L. M., Leung, G. M., & Lau, E. H. (2011). Digital dashboard design using multiple data streams for disease surveillance with influenza surveillance as an example. *Journal of Medical Internet Research*, 13(4), e1658. <https://doi.org/10.2196/jmir.1658>



- Chung, H., North, C., Self, J. Z., Chu, S., & Quek, F. (2014). VisPorter: Facilitating information sharing for collaborative sensemaking on multiple displays. *Personal and Ubiquitous Computing*, *18*(5), 1169–1186. <https://doi.org/10.1007/s00779-013-0727-2>
- Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., & Young, R. M. (1995). Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The Care Properties. In *Human—computer interaction* (pp. 115–120). [https://doi.org/10.1007/978-1-5041-2896-4\\\_{-}19](https://doi.org/10.1007/978-1-5041-2896-4\_{-}19)
- Davenport, T., & Malone, K. (2021). Deployment as a Critical Business Data Science Discipline. *Harvard Data Science Review*, *3*(1). <https://doi.org/10.1162/99608f92.90814c32>
- Dayal, U., Vennelakanti, R., Sharma, R., Castellanos, M., Hao, M., & Patel, C. (2008). Collaborative business intelligence: Enabling Collaborative decision making in enterprises. In *Lecture notes in computer science* (pp. 8–25). [https://doi.org/10.1007/978-3-540-88871-0\\\_{-}5](https://doi.org/10.1007/978-3-540-88871-0\_{-}5)
- Deng, L., Wang, Y., Wang, K., Acero, A., Hon, H., Droppo, J., Boulis, C., Mahajan, M., & Huang, X. (2004). Speech and Language Processing for Multimodal Human-Computer Interaction. *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, *36*(2/3), 161–187. <https://doi.org/10.1023/B:VLSI.0000015095.19623.73>
- Dennis, A. R. (1996). Information Exchange and Use in Group Decision Making: You Can Lead a Group to Information, but You Can't Make It Think. *MIS Quarterly*, *20*(4), 433. <https://doi.org/10.2307/249563>
- Dennis, A. R., Wixom, B. H., & Vandenberg, R. J. (2001). Understanding Fit and Appropriation Effects in Group Support Systems via Meta-Analysis. *MIS Quarterly*, *25*(2), 167. <https://doi.org/10.2307/3250928>
- Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, *5*(1), 4–7. <https://doi.org/10.1007/s007790170019>
- Diederich, S., Brendel, A. B., & Kolbe, L. M. (2020). Designing Anthropomorphic Enterprise Conversational Agents. *Business and Information Systems Engineering*, *62*(3), 193–209. <https://doi.org/10.1007/S12599-020-00639-Y/TABLES/6>
- Diederich, S., Brendel, A. B., Morana, S., & Kolbe, L. (2022). On the Design of and Interaction with Conversational Agents: An Organizing and Assessing Review of Human-Computer Interaction Research. *Journal of the Association for Information Systems*, *23*(1), 96–138. <https://doi.org/10.17705/1jais.00724>

- Dimoka, A., Pavlou, P. A., & Davis, F. D. (2011). NeuroIS: The potential of cognitive neuroscience for information systems research. *Information Systems Research*, 22(4), 687–702. <https://doi.org/10.1287/isre.1100.0284>
- Dong, E., Du, H., & Gardner, L. (2020). An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases*, 20(5), 533–534. [https://doi.org/10.1016/S1473-3099\(20\)30120-1](https://doi.org/10.1016/S1473-3099(20)30120-1)
- Eagan, J. R., & Stasko, J. T. (2008). The Buzz: Supporting user tailorability in awareness applications. *Proceedings of the 2008 CHI Conference on Human Factors in Computing Systems*, 1729–1738. <https://doi.org/10.1145/1357054.1357324>
- Fast, E., Chen, B., Mendelsohn, J., Bassen, J., & Bernstein, M. S. (2018). Iris: A conversational agent for complex tasks. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–12. <https://doi.org/10.1145/3173574.3174047>
- Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data* (1st ed.). O'Reilly Media, Inc.
- Fischer, M. H., Campagna, G., Choi, E., & Lam, M. S. (2021). DIY assistant: A multi-modal end-user programmable virtual assistant. *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, 312–327. <https://doi.org/10.1145/3453483.3454046>
- Flowers, J. (2020). The COVID-19 dashboard: bringing together data and statistics in one place - Public health matters. <https://publichealthmatters.blog.gov.uk/2020/09/04/the-covid-19-dashboard-bringing-together-data-and-statistics-in-one-place/>
- Frohlich, D. M. (1993). The history and future of direct manipulation. *Behaviour and Information Technology*, 12(6), 315–329. <https://doi.org/10.1080/01449299308924396>
- Gao, T., Dontcheva, M., Adar, E., Liu, Z., & Karahalios, K. (2015). Datatone: Managing ambiguity in natural language interfaces for data visualization. *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, 489–500. <https://doi.org/10.1145/2807442.2807478>
- Gardner, L., Ratcliff, J., Dong, E., & Katz, A. (2021). A need for open public data standards and sharing in light of COVID-19. *The Lancet Infectious Diseases*, 21(4), e80. [https://doi.org/10.1016/S1473-3099\(20\)30635-6](https://doi.org/10.1016/S1473-3099(20)30635-6)
- Gibson, J. J. (1977). The theory of affordances. In *Perceiving, acting, and knowing* (pp. 67–82). Lawrence Erlbaum. <https://doi.org/10.4324/9781315816852>
- Gnewuch, U., Morana, S., & Maedche, A. (2018). Towards Designing Cooperative and Social Conversational Agents for Customer Service. *Proceedings of the International Conference on Information Systems*.

- Google. (2019). Create commands to control online services and devices. <https://support.google.com/googlenest/answer/7194656>
- Gregor, S., Chandra Kruse, L., & Seidel, S. (2020). Research Perspectives: The Anatomy of a Design Principle. *Journal of the Association for Information Systems*, 21(6), 1622–1652. <https://doi.org/https://doi.org/10.17705/1jais.00649>
- Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, 37(2), 337–355. <https://doi.org/10.25300/MISQ/2013/37.2.01>
- Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association for Information Systems*, 8(5), 312–335. <https://doi.org/10.17705/1jais.00129>
- Grover, V., Chiang, R. H., Liang, T. P., & Zhang, D. (2018). Creating Strategic Business Value from Big Data Analytics: A Research Framework. *Journal of Management Information Systems*, 35(2), 388–423. <https://doi.org/10.1080/07421222.2018.1451951>
- Grudin, J., & Jacques, R. (2019). Chatbots, Humbots, and the Quest for Artificial General Intelligence. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 11, 1–11. <https://doi.org/10.1145/3290605.3300439>
- Gupta, S., & Bostrom, R. P. (2009). Technology-Mediated learning: A comprehensive theoretical model. *Journal of the Association for Information Systems*, 10(9), 686–714. <https://doi.org/10.17705/1jais.00207>
- Gupta, S., Bostrom, R. P., & Huber, M. (2010). End-user training methods. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, 41(4), 9–39. <https://doi.org/10.1145/1899639.1899641>
- Hartson, R. (2003). Cognitive, physical, sensory, and functional affordances in interaction design. *Behaviour & Information Technology*, 22(5), 315–338. <https://doi.org/10.1080/01449290310001592587>
- Hawkes, T. (1977). *Structuralism and semiotics*. Routledge Taylor & Francis Group. <https://doi.org/10.4324/9780203100516>
- Hayes, A. F. (2017). *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach*. Guilford publications.
- Hearst, M., & Tory, M. (2019). Would You Like A Chart With That? Incorporating Visualizations into Conversational Interfaces. *IEEE Visualization Conference (VIS)*, 1–5. <https://doi.org/10.1109/VISUAL.2019.8933766>
- Hendrix, G. G. (1982). Natural-Language Interface. *American Journal of Computational Linguistics*, 8(2), 56–61. <https://aclanthology.org/J82-2002>

- Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19(2), 87–92.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. <https://doi.org/10.2307/25148625>
- Hoque, E., Setlur, V., Tory, M., & Dykeman, I. (2018). Applying Pragmatics Principles for Interaction with Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1), 309–318. <https://doi.org/10.1109/TVCG.2017.2744684>
- Hsieh, J. J. A., Rai, A., & Keil, M. (2008). Understanding digital inequality: Comparing continued use behavioral models of the socio-economically advantaged and disadvantaged. *MIS Quarterly*, 32(1), 97–126. <https://doi.org/10.2307/25148830>
- Hu, W., Almansoori, A., Kannan, P. K., Azarm, S., & Wang, Z. (2012). Corporate dashboards for integrated business and engineering decisions in oil refineries: An agent-based approach. *Decision Support Systems*, 52(3), 729–741. <https://doi.org/10.1016/j.dss.2011.11.019>
- Huang, T.-H. (, Chang, J. C., & Bigham, J. P. (2018). Evorus: A Crowd-powered conversational assistant built to automate itself over time. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–13. <https://doi.org/10.1145/3173574.3173869>
- Isenberg, P., Fisher, D., Paul, S. A., Morris, M. R., Inkpen, K., & Czerwinski, M. (2012). Co-Located Collaborative Visual Analytics around a Tabletop Display. *IEEE Transactions on Visualization and Computer Graphics*, 18(5), 689–702. <https://doi.org/10.1109/TVCG.2011.287>
- Ivanković, D., Barbazza, E., Bos, V., Fernandes, Ó. B., Gilmore, K. J., Jansen, T., Kara, P., Larrain, N., Lu, S., Meza-Torres, B., Mulyanto, J., Poldrugovac, M., Rotar, A., Wang, S., Willmington, C., Yang, Y., Yelgezekova, Z., Allin, S., Klazinga, N., & Kringos, D. (2021). Features constituting actionable COVID-19 dashboards: Descriptive assessment and expert appraisal of 158 public web-based COVID-19 dashboards. *Journal of Medical Internet Research*, 23(2), e25682. <https://doi.org/10.2196/25682>
- Jaimes, A., & Sebe, N. (2007). Multimodal human–computer interaction: A survey. *Computer Vision and Image Understanding*, 108(1-2), 116–134. <https://doi.org/10.1016/j.cviu.2006.10.019>
- Jain, M., Kota, R., Kumar, P., & Patel, S. N. (2018). Convey: Exploring the use of a context view for chatbots. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, (4), 1–6. <https://doi.org/10.1145/3173574.3174042>

- Jetter, H.-C., Gerken, J., Zöllner, M., Reiterer, H., & Milic-Frayling, N. (2011). Materializing the query with facet-streams. *Proceedings of the 2011 CHI Conference on Human Factors in Computing Systems*, 3013. <https://doi.org/10.1145/1978942.1979390>
- João, P., Nuno, D., Fábio, S. F., & Ana, P. (2019). A cross-analysis of block-based and visual programming apps with computer science student-teachers. *Education Sciences*, 9(3). <https://doi.org/10.3390/educsci9030181>
- Kaufmann, J., & Chamoni, P. (2014). Structuring collaborative business intelligence: A literature review. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 3738–3747. <https://doi.org/10.1109/HICSS.2014.465>
- Kim, Y.-S., Dontcheva, M., Adar, E., & Hullman, J. (2019). Vocal Shortcuts for Creative Experts. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 14, 1–14. <https://doi.org/10.1145/3290605.3300562>
- Kim, Y.-H., Lee, B., Srinivasan, A., & Choe, E. K. (2021). Data@Hand: Fostering Visual Exploration of Personal Data on Smartphones Leveraging Speech and Touch Interaction. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–17. <https://doi.org/10.1145/3411764.3445421>
- Knote, R., Janson, A., Söllner, M., & Leimeister, J. M. (2021). Value co-creation in smart services: A functional affordances perspective on smart personal assistants. *Journal of the Association for Information Systems*, 22(2), 418–458. <https://doi.org/10.17705/1jais.00667>
- Ko, A. J., Myers, B. A., Coblenz, M. J., & Aung, H. H. (2006). An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks. *IEEE Transactions on Software Engineering*, 32(12), 971–987. <https://doi.org/10.1109/TSE.2006.116>
- Koch, T. (2021). Welcome to the revolution: COVID-19 and the democratization of spatial-temporal data. *Patterns*, 2(7), 100272. <https://doi.org/10.1016/J.PATTER.2021.100272>
- Krosnick, R., & Oney, S. (2022). ParamMacros : Creating UI Automation Leveraging End-User Natural Language Parameterization. *2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*.
- Kuechler, B., & Vaishnavi, V. (2008). On theory development in design science research: anatomy of a research project. *European Journal of Information Systems*, 17(5), 489–504. <https://doi.org/10.1057/ejis.2008.40>
- Laird, J. E., Gluck, K., Anderson, J., Forbus, K. D., Jenkins, O. C., Lebiere, C., Salvucci, D., Scheutz, M., Thomaz, A., Trafton, G., Wray, R. E., Mohan, S., & Kirk, J. R.

- (2017). Interactive Task Learning. *IEEE Intelligent Systems*, 32(4), 6–21. <https://doi.org/10.1109/MIS.2017.3121552>
- Langner, R., Horak, T., & Dachsel, R. (2018). VisTiles: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 24(1), 626–636. <https://doi.org/10.1109/TVCG.2017.2744019>
- Langner, R., Kister, U., & Dachsel, R. (2019). Multiple Coordinated Views at Large Displays for Multiple Users. *IEEE Transactions on Visualization and Computer Graphics*, 25(1), 608–618. <https://doi.org/10.1109/TVCG.2018.2865235>
- Larson, D., & Chang, V. (2016). A review and future direction of agile, business intelligence, analytics and data science. *International Journal of Information Management*, 36(5), 700–710. <https://doi.org/10.1016/j.ijinfomgt.2016.04.013>
- Lauterbach, J., Mueller, B., Kahrau, F., & Maedche, A. (2020). Achieving Effective Use When Digitalizing Work: The Role of Representational Complexity. *MIS Quarterly*, 44(3), 1023–1048. <https://doi.org/10.25300/MISQ/2020/14583>
- Lee, B., Choe, E. K., Isenberg, P., Marriott, K., & Stasko, J. (2020). Reaching Broader Audiences with Data Visualization. *IEEE Computer Graphics and Applications*, 40(2), 82–90. <https://doi.org/10.1109/MCG.2020.2968244>
- Lee, B., Isenberg, P., Riche, N. H., & Carpendale, S. (2012). Beyond Mouse and Keyboard: Expanding Design Considerations for Information Visualization Interactions. *IEEE Transactions on Visualization and Computer Graphics*, 18(12), 2689–2698. <https://doi.org/10.1109/TVCG.2012.204>
- Lee, B., Smith, G., Riche, N. H., Karlson, A., & Carpendale, S. (2015). SketchInsight: Natural data exploration on interactive whiteboards leveraging pen and touch interaction. *2015 IEEE Pacific Visualization Symposium (PacificVis), 2015-July*, 199–206. <https://doi.org/10.1109/PACIFICVIS.2015.7156378>
- Lee, B., Srinivasan, A., Isenberg, P., & Stasko, J. (2021). Post-wimp interaction for information visualization. *Foundations and Trends in Human-Computer Interaction*, 14(1), 1–95. <https://doi.org/10.1561/11000000081>
- Leong, C., Pan, S. L., Ractham, P., & Kaewkitipong, L. (2015). ICT-enabled community empowerment in crisis response: Social media in Thailand flooding 2011. *Journal of the Association for Information Systems*, 16(3), 174–212. <https://doi.org/10.17705/1jais.00390>

- Leong, L. H., Kobayashi, S., Koshizuka, N., & Sakamura, K. (2005). CASIS: A context-aware speech interface system. *Proceedings of the 10th International Conference on Intelligent User Interfaces*, 231–238. <https://doi.org/10.1145/1040830.1040880>
- Leshed, G., Haber, E. M., Matthews, T., & Lau, T. (2008). CoScripter. *Proceedings of the 2008 CHI Conference on Human Factors in Computing Systems*, 1719. <https://doi.org/10.1145/1357054.1357323>
- Li, C.-H., Yeh, S.-F., Chang, T.-J., Tsai, M.-H., Chen, K., & Chang, Y.-J. (2020). A Conversation Analysis of Non-Progress and Coping Strategies with a Banking Task-Oriented Chatbot. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–12. <https://doi.org/10.1145/3313831.3376209>
- Li, T. J.-J., Azaria, A., & Myers, B. A. (2017). SUGILITE. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, 2017-May*, 6038–6049. <https://doi.org/10.1145/3025453.3025483>
- Li, T. J.-J., Chen, J., Canfield, B., & Myers, B. A. (2020). Privacy-Preserving Script Sharing in GUI-based Programming-by-Demonstration Systems. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1), 1–23. <https://doi.org/10.1145/3392869>
- Li, T. J.-J., Labutov, I., Li, X. N., Zhang, X., Shi, W., Ding, W., Mitchell, T. M., & Myers, B. A. (2018). APPINITE: A Multi-Modal Interface for Specifying Data Descriptions in Programming by Demonstration Using Natural Language Instructions. *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2018-October*, 105–114. <https://doi.org/10.1109/VLHCC.2018.8506506>
- Li, T. J.-J., Radensky, M., Jia, J., Singarajah, K., Mitchell, T. M., & Myers, B. A. (2019). PUMICE: A Multi-Modal Agent that Learns Concepts and Conditionals from Natural Language and Demonstrations. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 577–589. <https://doi.org/10.1145/3332165.3347899>
- Lim, B. Y., & Dey, A. K. (2010). Toolkit to support intelligibility in context-aware applications. *Proceedings of the 12th ACM international conference on Ubiquitous computing*, 13–22. <https://doi.org/10.1145/1864349.1864353>
- Lin, S., Hilton, J., & Evans, O. (2022). TruthfulQA: Measuring How Models Mimic Human Falsehoods. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 1, 3214–3252. <https://doi.org/10.18653/v1/2022.acl-long.229>

- 
- Liu, S. B., & Palen, L. (2010). The new cartographers: Crisis map mashups and the emergence of neogeographic practice. *Cartography and Geographic Information Science*, *37*(1), 69–90. <https://doi.org/10.1559/152304010790588098>
- Liu, Y., Shen, Y., Luo, C., & Chan, H. C. (2021). Reach Out and Touch: Eliciting the Sense of Touch Through Gesture-Based Interaction. *Journal of the Association for Information Systems*, *22*(6), 1686–1714. <https://doi.org/10.17705/1jais.00704>
- Lu, Y., Chen, S., Miao, Z., Delen, D., & Gin, A. (2021). Clustering temporal disease networks to assist clinical decision support systems in visual analytics of comorbidity progression. *Decision Support Systems*, *148*, 113583. <https://doi.org/10.1016/j.dss.2021.113583>
- Luger, E., & Sellen, A. (2016). "Like having a really bad pa": The gulf between user expectation and experience of conversational agents. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 5286–5297. <https://doi.org/10.1145/2858036.2858288>
- Lukyanenko, R., Parsons, J., & Hovorka, D. S. (2020). Research Perspectives: Design Theory Indeterminacy: What Is it, How Can it Be Reduced, and Why Did the Polar Bear Drown? *Journal of the Association for Information Systems*, *21*(5), 1343–1369. <https://doi.org/10.17705/1jais.00639>
- Luo, Y., Tang, N., Li, G., Tang, J., Chai, C., & Qin, X. (2022). Natural Language to Visualization by Neural Machine Translation. *IEEE Transactions on Visualization and Computer Graphics*, *28*(1), 217–226. <https://doi.org/10.1109/TVCG.2021.3114848>
- Majchrzak, A., More, P. H. B., & Faraj, S. (2012). Transcending Knowledge Differences in Cross-Functional Teams. *Organization Science*, *23*(4), 951–970. <https://doi.org/10.1287/orsc.1110.0677>
- Matheus, R., Janssen, M., & Maheshwari, D. (2020). Data science empowering the public: Data-driven dashboards for transparent and accountable decision-making in smart cities. *Government Information Quarterly*, *37*(3), 101284. <https://doi.org/10.1016/j.giq.2018.01.006>
- Maués, R. D. A., & Barbosa, S. D. J. (2013). Keep doing what I just did: Automating smartphones by demonstration. *MobileHCI 2013 - Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 295–303. <https://doi.org/10.1145/2493190.2493216>



- 
- McDaniel, R. G., & Myers, B. A. (1999). Getting more out of programming-by-demonstration. *Proceedings of the 1999 CHI Conference on Human Factors in Computing Systems*, 442–449. <https://doi.org/10.1145/302979.303127>
- McTear, M. (2017). The Rise of the Conversational Interface: A New Kid on the Block? In *Lecture notes in computer science* (pp. 38–49). Springer Verlag. [https://doi.org/10.1007/978-3-319-69365-1\\_{\\\_}3](https://doi.org/10.1007/978-3-319-69365-1_{\_}3)
- McTear, M., Callejas, Z., & Griol, D. (2016). *The Conversational Interface: Talking to Smart Devices*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-32967-3>
- Microsoft. (2017). Cognitive Services - Enable Natural Language Interaction with LUIS | Microsoft Docs. <https://docs.microsoft.com/en-us/archive/msdn-magazine/2017/january/cognitive-services-enable-natural-language-interaction-with-luis>
- Microsoft. (2021). Microsoft Bot Framework. <https://dev.botframework.com/>
- Microsoft. (2022a). Teach Q&A to understand questions and terms in Power BI Q&A - Power BI | Microsoft Docs. <https://docs.microsoft.com/en-us/power-bi/natural-language/q-and-a-tooling-teach-q-and-a>
- Microsoft. (2022b). The Team Data Science Process lifecycle - Azure Architecture Center | Microsoft Docs. <https://docs.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle-deployment%20https://docs.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle>
- Misra, D. K., Sung, J., Lee, K., & Saxena, A. (2016). Tell me Dave: Context-sensitive grounding of natural language to manipulation instructions. *International Journal of Robotics Research*, 35(1-3), 281–300. <https://doi.org/10.1177/0278364915602060>
- Momenipour, A., Rojas-Murillo, S., Murphy, B., Pennathur, P., & Pennathur, A. (2021). Usability of state public health department websites for communication during a pandemic: A heuristic evaluation. *International Journal of Industrial Ergonomics*, 86, 103216. <https://doi.org/10.1016/j.ergon.2021.103216>
- Morana, S., Kroenung, J., Maedche, A., & Schacht, S. (2019). Designing process guidance systems. *Journal of the Association for Information Systems*, 20(5), 499–535. <https://doi.org/10.17705/1jais.00542>
- Morris, M. R. (2012). Web on the wall. *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces - ITS '12*, 95. <https://doi.org/10.1145/2396636.2396651>

- Myers, B. A., Ko, A. J., LaToza, T. D., & Yoon, Y. (2016). Programmers Are Users Too: Human-Centered Methods for Improving Programming Tools. *Computer*, 49(7), 44–52. <https://doi.org/10.1109/MC.2016.200>
- Myers, B. A., Ko, A. J., Scaffidi, C., Oney, S., Yoon, Y. S., Chang, K., Kery, M. B., & Li, T. J. J. (2017). Making end user development more natural. In *New perspectives in end-user development* (pp. 1–22). <https://doi.org/10.1007/978-3-319-60291-2{-}1>
- Myers, B. A., McDaniel, R. G., & Kosbie, D. S. (1993). Marquise. *Proceedings of the 1993 CHI Conference on Human Factors in Computing Systems*, 293–300. <https://doi.org/10.1145/169059.169225>
- Nadj, M., Maedche, A., & Schieder, C. (2020). The effect of interactive analytical dashboard features on situation awareness and task performance. *Decision Support Systems*, 135, 113322. <https://doi.org/10.1016/j.dss.2020.113322>
- Narechania, A., Fourney, A., Lee, B., & Ramos, G. (2021). DIY: Assessing the Correctness of Natural Language to SQL Systems. *26th International Conference on Intelligent User Interfaces*, 597–607. <https://doi.org/10.1145/3397481.3450667>
- Narechania, A., Srinivasan, A., & Stasko, J. (2021). NL4DV: A Toolkit for Generating Analytic Specifications for Data Visualization from Natural Language Queries. *IEEE Transactions on Visualization and Computer Graphics*, 27(2), 369–379. <https://doi.org/10.1109/TVCG.2020.3030378>
- Nguyen, A., Tuunanen, T., Gardner, L., & Sheridan, D. (2021). Design principles for learning analytics information systems in higher education. *European Journal of Information Systems*, 30(5), 541–568. <https://doi.org/10.1080/0960085X.2020.1816144>
- Nguyen, H., Ketchell, S., Engelke, U., Thomas, B. H., & de Souza, P. (2017). Augmented Reality Based Bee Drift Analysis: A User Study. *2017 International Symposium on Big Data Visual Analytics (BDVA)*, 1–8. <https://doi.org/10.1109/BDVA.2017.8114581>
- Norman, D. (2016). *The Design of Everyday Things*. Vahlen. <https://doi.org/10.15358/9783800648108>
- Nunamaker, J. F., & Deokar, A. V. (2008). GDSS Parameters and Benefits. In *Handbook on decision support systems 1* (pp. 391–414). Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-540-48713-5{-}20>
- Nunamaker, J. F., Derrick, D. C., Elkins, A. C., Burgoon, J. K., & Patton, M. W. (2011). Embodied Conversational Agent-Based Kiosk for Automated Interviewing. *Journal of Management Information Systems*, 28(1), 17–48. <https://doi.org/10.2753/MIS0742-1222280102>

- Oviatt, S. (1999). Ten myths of multimodal interaction. *Communications of the ACM*, 42(11), 74–81. <https://doi.org/10.1145/319382.319398>
- Oviatt, S. (2003). Advances in robust multimodal interface design. *IEEE Computer Graphics and Applications*, 23(5), 62–68. <https://doi.org/10.1109/MCG.2003.1231179>
- Özateş, Ş. B., Özgür, A., & Draradev, G. R. (2016). Sentence similarity based on dependency tree kernels for multi-document summarization. *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 2833–2838. <http://duc.nist.gov/duc2003/tasks.html>
- Pan, L., Yu, C., Li, J., Huang, T., Bi, X., & Shi, Y. (2022). Automatically Generating and Improving Voice Command Interface from Operation Sequences on Smartphones. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–21. <https://doi.org/10.1145/3491102.3517459>
- Pane, J. F., Ratanamahatana, C. A., & Myers, B. A. (2001). Studying the language and structure in non-programmers' solutions to programming problems. *International Journal of Human Computer Studies*, 54(2), 237–264. <https://doi.org/10.1006/ijhc.2000.0410>
- Park, H., Bellamy, M. A., & Basole, R. C. (2016). Visual analytics for supply network management: System design and evaluation. *Decision Support Systems*, 91, 89–102. <https://doi.org/10.1016/J.DSS.2016.08.003>
- Patino, M. (2021). The Rise of the Pandemic Dashboard. <https://www.bloomberg.com/news/features/2021-09-25/why-every-government-needs-a-covid-dashboard>
- Pietz, J., McCoy, S., & Wilck, J. H. (2020). Chasing John Snow: data analytics in the COVID-19 era. *European Journal of Information Systems*, 29(4), 388–404. <https://doi.org/10.1080/0960085X.2020.1793698>
- Pitt, L., Berthon, P., & Robson, K. (2011). Deciding When to Use Tablets for Business Applications. *MIS Quarterly Executive*, 10(3). <https://aisel.aisnet.org/misqe/vol10/iss3/5>
- Porcheron, M., Fischer, J. E., Reeves, S., & Sharples, S. (2018). Voice Interfaces in Everyday Life. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–12. <https://doi.org/10.1145/3173574.3174214>
- Prevent Epidemics. (2020). Tracking COVID-19 in the United States: From Information Catastrophe to Empowered Communities. <https://preventepidemics.org/wp-content/uploads/2020/07/Tracking-COVID-19-in-the-United-States-Report-1.pdf>

- Recker, J. (2021). Improving the state-tracking ability of corona dashboards. *European Journal of Information Systems*, 30(5), 476–495. <https://doi.org/10.1080/0960085X.2021.1907235>
- Reeves, L. M., Lai, J., Larson, J. A., Oviatt, S., Balaji, T. S., Buisine, S., Collings, P., Cohen, P., Kraal, B., Martin, J. C., McTear, M., Raman, T. V., Stanney, K. M., Su, H., & Wang, Q. Y. (2004). Guidelines for multimodal user interface design. *Communications of the ACM*, 47(1), 57–59. <https://doi.org/10.1145/962081.962106>
- Reinhart, T. (1981). Pragmatics and Linguistics: an analysis of Sentence Topics. *Philosophica*, 27(0), 53–94. <https://doi.org/10.21825/philosophica.82606>
- Roberts, J. C., Ritsos, P. D., Badam, S. K., Brodbeck, D., Kennedy, J., & Elmquist, N. (2014). Visualization beyond the desktop-the next big thing. *IEEE Computer Graphics and Applications*, 34(6), 26–34. <https://doi.org/10.1109/MCG.2014.82>
- Ruoff, M., & Gnewuch, U. (2021a). Designing Multimodal BI&A Systems for Co-Located Team Interactions. *29th European Conference on Information Systems*. [https://aisel.aisnet.org/ecis2021\\_rp/113](https://aisel.aisnet.org/ecis2021_rp/113)
- Ruoff, M., & Gnewuch, U. (2021b). Designing Conversational Dashboards for Effective Use in Crisis Response. *29th European Conference on Information Systems*. [https://aisel.aisnet.org/ecis2021\\_rip/45](https://aisel.aisnet.org/ecis2021_rip/45)
- Ruoff, M., Gnewuch, U., & Maedche, A. (2020). Designing Multimodal BI&A Systems for Face-to-Face Team Interactions. *SIGHCI 2020 Proceedings*. <https://aisel.aisnet.org/sighci2020/18>
- Ruoff, M., Gnewuch, U., Maedche, A., & Scheibehenne, B. (2022). Designing Conversational Dashboards for Effective Use in Crisis Response. *J AIS Preprints (Forthcoming)*. <https://doi.org/10.17705/1jais.00801>
- Ruoff, M., & Maedche, A. (2020). Towards Understanding Multimodal Interaction for Visual Data Analysis. *Posters of IEEE Visualization, Oct 2020, Salt Lake City, United States*. <https://publikationen.bibliothek.kit.edu/1000128180>
- Ruoff, M., Myers, B. A., & Maedche, A. (n.d.). ContextIT - Interactively Contextualizing Natural Language Inputs in Data Visualization Tools.
- Ruoff, M., Myers, B. A., & Maedche, A. (2021). ONYX : Towards Extending Natural Language Interfaces for Data Visualization Tools through Interactive Task Learning. *NLVIZ Workshop: Exploring Research Opportunities for Natural Language, Text, and Data Visualization*.
- Ruoff, M., Myers, B. A., & Maedche, A. (2022). ONYX - User Interfaces for Assisting in Interactive Task Learning for Natural Language Interfaces of Data Visualization

- Tools. *Proceedings of the 2022 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 1(1), 1–10. <https://doi.org/10.1145/3491101.3519793>
- Ruoff, M., Myers, B. A., & Maedche, A. (2023). ONYX : Assisting Users in Teaching Natural Language Interfaces Through Multi-Modal Interactive Task Learning. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. <https://doi.org/10.1145/3544548.3580964>
- Saktheeswaran, A., Srinivasan, A., & Stasko, J. (2020). Touch? Speech? or Touch and Speech? Investigating Multimodal Interaction for Visual Network Exploration and Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 26(6), 2168–2179. <https://doi.org/10.1109/TVCG.2020.2970512>
- Satyanarayan, A., Moritz, D., Wongsuphasawat, K., & Heer, J. (2017). Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1), 341–350. <https://doi.org/10.1109/TVCG.2016.2599030>
- Schanke, S., Burtch, G., & Ray, G. (2021). Estimating the impact of “humanizing” customer service chatbots. *Information Systems Research*, 32(3), 736–751. [https://doi.org/10.1287/ISRE.2021.1015/SUPPL{\\\\_}FILE/ISRE.2021.1015.SM1.PDF](https://doi.org/10.1287/ISRE.2021.1015/SUPPL{\\_}FILE/ISRE.2021.1015.SM1.PDF)
- Schmidt, J. B., Montoya-Weiss, M. M., & Massey, A. P. (2001). New Product Development Decision-Making Effectiveness: Comparing Individuals, Face-To-Face Teams, and Virtual Teams. *Decision Sciences*, 32(4), 575–600. <https://doi.org/10.1111/j.1540-5915.2001.tb00973.x>
- Seddon, P. B. (1997). A Respecification and Extension of the DeLone and McLean Model of IS Success. *Information Systems Research*, 8(3), 240–253. <https://doi.org/10.1287/isre.8.3.240>
- Seeger, A. M., Pfeiffer, J., & Heinzl, A. (2021). Texting with humanlike conversational agents: Designing for anthropomorphism. *Journal of the Association for Information Systems*, 22(4), 931–967. <https://doi.org/10.17705/1jais.00685>
- Sereshkeh, A. R., Leung, G., Perumal, K., Phillips, C., Zhang, M., Fazly, A., & Mohamed, I. (2020). VASTA. *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 20, 22–32. <https://doi.org/10.1145/3377325.3377515>
- Setlur, V., Battersby, S. E., Tory, M., Gossweiler, R., & Chang, A. X. (2016). Eviza: A natural language interface for visual analysis. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 365–377. <https://doi.org/10.1145/2984511.2984588>

- Setlur, V., & Kumar, A. (2020). Sentifiers: Interpreting Vague Intent Modifiers in Visual Analysis using Word Co-occurrence and Sentiment Analysis. *2020 IEEE Visualization Conference (VIS)*, 216–220. <https://doi.org/10.1109/VIS47514.2020.00050>
- Setlur, V., & Tory, M. (2022). How do you Converse with an Analytical Chatbot? Revisiting Gricean Maxims for Designing Analytical Conversational Behavior. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–17. <https://doi.org/10.1145/3491102.3501972>
- Setlur, V., Tory, M., & Djalali, A. (2019). Inferencing underspecified natural language utterances in visual analysis. *International Conference on Intelligent User Interfaces, Proceedings IUI, Part F1476*, 40–51. <https://doi.org/10.1145/3301275.3302270>
- Siering, M., Muntermann, J., & Grčar, M. (2021). Design Principles for Robust Fraud Detection: The Case of Stock Market Manipulations. *Journal of the Association for Information Systems*, 22(1), 156–178. <https://doi.org/10.17705/1jais.00657>
- Smuts, M., Scholtz, B., & Calitz, A. (2015). Design Guidelines for Business Intelligence Tools for Novice Users. *Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists - SAICSIT '15, 28-30-Sept*, 1–15. <https://doi.org/10.1145/2815782.2815788>
- Soper, D. S., Demirkan, H., & Schlicher, J. (2021). Analytics and IT in the Response to COVID-19: A Research Framework and Lessons for the Future. *Journal of Decision Systems*, 1–13. <https://doi.org/10.1080/12460125.2021.1899104>
- Srinivasan, A., Lee, B., Henry Riche, N., Drucker, S. M., & Hinckley, K. (2020). InChorus: Designing Consistent Multimodal Interactions for Data Visualization on Tablet Devices. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–13. <https://doi.org/10.1145/3313831.3376782>
- Srinivasan, A., Nyapathy, N., Lee, B., Drucker, S. M., & Stasko, J. (2021). Collecting and Characterizing Natural Language Utterances for Specifying Data Visualizations. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–10. <https://doi.org/10.1145/3411764.3445400>
- Srinivasan, A., & Stasko, J. (2018). Orko: Facilitating Multimodal Interaction for Visual Exploration and Analysis of Networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1), 511–521. <https://doi.org/10.1109/TVCG.2017.2745219>
- Srinivasan, A., & Stasko, J. (2020). How to ask what to say?: Strategies for evaluating natural language interfaces for data visualization. *IEEE Computer Graphics and Applications*, 40(4), 96–103. <https://doi.org/10.1109/MCG.2020.2986902>

- Steelman, Z. R., Hammer, B. I., & Limayem, M. (2014). Data collection in the digital age: Innovative alternatives to student samples. *MIS Quarterly*, *38*(2), 355–378. <https://doi.org/10.25300/MISQ/2014/38.2.02>
- Suddrey, G., Talbot, B., & Maire, F. (2022). Learning and Executing Re-Usable Behaviour Trees From Natural Language Instruction. *IEEE Robotics and Automation Letters*, *7*(4), 10643–10650. <https://doi.org/10.1109/LRA.2022.3194681>
- Sun, Y., Leigh, J., Johnson, A., & Lee, S. (2010). Articulate: A Semi-automated Model for Translating Natural Language Queries into Meaningful Visualizations. In *Lecture notes in computer science* (pp. 184–195). [https://doi.org/10.1007/978-3-642-13544-6{\\\_}18](https://doi.org/10.1007/978-3-642-13544-6{\_}18)
- Sundar, S. S., Jia, H., Waddell, T. F., & Huang, Y. (2015). Toward a Theory of Interactive Media Effects (TIME). In *The handbook of the psychology of communication technology* (pp. 47–86). Wiley. <https://doi.org/10.1002/9781118426456.ch3>
- Surbakti, F. P., Wang, W., Indulska, M., & Sadiq, S. (2020). Factors influencing effective use of big data: A research framework. *Information and Management*, *57*(1), 103146. <https://doi.org/10.1016/j.im.2019.02.001>
- Tableau. (2019). Optimize Data for Ask Data - Tableau. <https://help.tableau.com/current/pro/desktop/en-us/ask%20data%20optimize.htm>
- Thomason, J., Zhang, S., Mooney, R., & Stone, P. (2015). Learning to interpret natural language commands through human-robot dialog. *IJCAI International Joint Conference on Artificial Intelligence, 2015-Janua*(Ijcai), 1923–1929.
- Tian, Z., Yan, R., Mou, L., Song, Y., Feng, Y., & Zhao, D. (2017). How to Make Context More Useful? An Empirical Study on Context-Aware Neural Conversational Models. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, *2*, 231–236. <https://doi.org/10.18653/v1/P17-2036>
- Toreini, P., Langner, M., Maedche, A., Vogel, T., & Morana, S. (2022). Designing Attentive Information Dashboards. *Journal of the Association for Information Systems*, Forthcoming. <https://doi.org/10.17705/1jais.00732>
- Torres, R., & Sidorova, A. (2019). Reconceptualizing information quality as effective use in the context of business intelligence and analytics. *International Journal of Information Management*, *49*, 316–329. <https://doi.org/10.1016/j.ijinfomgt.2019.05.028>
- Tory, M., Bartram, L., Fiore-Gartland, B., & Crisan, A. (2021). Finding Their Data Voice: Practices and Challenges of Dashboard Users. *IEEE Computer Graphics and Applications*, 1–1. <https://doi.org/10.1109/MCG.2021.3136545>

- Tory, M., & Setlur, V. (2019). Do What I Mean, Not What I Say! Design Considerations for Supporting Intent and Context in Analytical Conversation. *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 93–103. <https://doi.org/10.1109/VAST47406.2019.8986918>
- Tremblay, M. C., Hevner, A. R., Berndt, D. J., Tremblay, M., Chiarini, ; & Hevner, A. R. ; (2010). Focus Groups for Artifact Refinement and Evaluation in Design Research. *Communications of the Association for Information Systems*, 26, 599–618. <https://doi.org/10.17705/1CAIS.02627>
- Trieu, V.-H., Burton-Jones, A., Green, P., & Cockcroft, S. (2022). Applying and Extending the Theory of Effective Use in a Business Intelligence Context. *MIS Quarterly*, 46(1), 645–678. <https://doi.org/10.25300/misq/2022/14880>
- Trummer, I. (2022). CodexDB: Synthesizing Code for Query Processing from Natural Language Instructions using GPT-3 Codex. *Proceedings of the VLDB Endowment*, 15(11), 2921–2928. <https://doi.org/10.14778/3551793.3551841>
- Turban, E., & Watkins, P. R. (1986). Integrating expert systems and decision support systems. *MIS Quarterly*, 10(2), 121–136. <https://doi.org/10.2307/249031>
- Turk, M. (2014). Multimodal interaction: A review. *Pattern Recognition Letters*, 36, 189–195. <https://doi.org/10.1016/j.patrec.2013.07.003>
- Vaithilingam, P., & Guo, P. J. (2019). Bespoke: Interactively synthesizing custom GUIs from command-line applications by demonstration. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 563–576. <https://doi.org/10.1145/3332165.3347944>
- Vallurupalli, V., & Bose, I. (2018). Business intelligence for performance measurement: A case based analysis. *Decision Support Systems*, 111(2017), 72–85. <https://doi.org/10.1016/j.dss.2018.05.002>
- Vitalari, N. P. (1985). Knowledge as a basis for expertise in systems analysis: An empirical study. *MIS Quarterly*, 9(3), 221–240. <https://doi.org/10.2307/248950>
- Wang, S. I., Ginn, S., Liang, P., & Manning, C. D. (2017). Naturalizing a Programming Language via Interactive Learning. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1, 929–938. <https://doi.org/10.18653/v1/P17-1086>
- Wang, Y., Hou, Z., Shen, L., Wu, T., Wang, J., Huang, H., Zhang, H., & Zhang, D. (2022). Towards Natural Language-Based Visualization Authoring. *IEEE Transactions on Visualization and Computer Graphics*, 1–11. <https://doi.org/10.1109/TVCG.2022.3209357>



- Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36–45. <https://doi.org/10.1145/365153.365168>
- Widjaja, T., & Gregory, R. W. (2020). Monitoring the Complexity of IT Architectures: Design Principles and an IT Artifact. *Journal of the Association for Information Systems*, 21(3), 664–694. <https://doi.org/10.17705/1jais.00616>
- Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017). A New Chatbot for Customer Service on Social Media. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 3506–3510. <https://doi.org/10.1145/3025453.3025496>
- Yi, J. S., Kang, Y. A., & Stasko, J. (2007). Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 1224–1231. <https://doi.org/10.1109/TVCG.2007.70515>
- Yigitbasioglu, O. M., & Velcu, O. (2012). A review of dashboards in performance management: Implications for design and research. *International Journal of Accounting Information Systems*, 13(1), 41–59. <https://doi.org/10.1016/j.accinf.2011.08.002>
- Young, G. W., & Kitchin, R. (2020). Creating design guidelines for building city dashboards from a user’s perspectives. *International Journal of Human-Computer Studies*, 140, 102429. <https://doi.org/10.1016/j.ijhcs.2020.102429>
- Young, G. W., Kitchin, R., & Naji, J. (2021). Building City Dashboards for Different Types of Users. *Journal of Urban Technology*, 28(1-2), 289–309. <https://doi.org/10.1080/10630732.2020.1759994>
- Yu, B., & Silva, C. T. (2020). FlowSense: A Natural Language Interface for Visual Data Exploration within a Dataflow System. *IEEE Transactions on Visualization and Computer Graphics*, 26(1), 1–11. <https://doi.org/10.1109/TVCG.2019.2934668>
- Zamora, J. (2017). I’m Sorry, Dave, I’m Afraid I Can’t Do That. *Proceedings of the 5th International Conference on Human Agent Interaction*, 253–260. <https://doi.org/10.1145/3125739.3125766>
- Zhang, X. (2017). Knowledge Management System Use and Job Performance: A Multilevel Contingency Model. *MIS Quarterly*, 41(3), 811–840. <https://doi.org/10.25300/MISQ/2017/41.3.07>
- Zhang, Z., Xu, Y., Wang, Y., Yao, B., Ritchie, D., Wu, T., Yu, M., Wang, D., & Li, T. J. J. (2022). StoryBuddy: A Human-AI Collaborative Chatbot for Parent-Child Interactive Storytelling with Flexible Parental Involvement. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–21. <https://doi.org/10.1145/3491102.3517479>

- Zong, J., Barnwal, D., Neogy, R., & Satyanarayan, A. (2021). Lyra 2: Designing Interactive Visualizations by Demonstration. *IEEE Transactions on Visualization and Computer Graphics*, *27*(2), 304–314. <https://doi.org/10.1109/TVCG.2020.3030367>
- Zook, M., Graham, M., Shelton, T., & Gorman, S. (2010). Volunteered Geographic Information and Crowdsourcing Disaster Relief: A Case Study of the Haitian Earthquake. *World Medical & Health Policy*, *2*(2), 6–32. <https://doi.org/10.2202/1948-4682.1069>

# List of Publications

## Journal Publications

**Ruoff, M.**, Gnewuch, U., Maedche, A., & Scheibehenne, B. (Forthcoming). Designing Conversational Dashboards for Effective Use in Crisis Response. In *Journal of the Association for Information Systems*.

Gnewuch, U., **Ruoff, M.**, Peukert, C., & Maedche, A. (2022). Multiexperience. In *Business & Information Systems Engineering*, 64(6), 813-823.

## Conference Proceedings

**Ruoff, M.**, Myers, B. A., & Maedche, A. (2023). ONYX : Assisting Users in Teaching Natural Language Interfaces Through Multi-Modal Interactive Task Learning. In *CHI Conference on Human Factors in Computing Systems*.

Haug, S., **Ruoff, M.**, & Gnewuch, U. (2022). The Impact of Conversational Assistance on the Effective Use of Forecasting Support Systems: A Framed Field Experiment. In *ICIS 2022 Proceedings*.

**Ruoff, M.**, & Gnewuch, U. (2021). Designing Multimodal BI&A Systems for Co-Located Team Interactions. In *29th European Conference on Information Systems*.

**Ruoff, M.**, & Gnewuch, U. (2021). Designing Conversational Dashboards for Effective Use in Crisis Response. In *29th European Conference on Information Systems*.

Fleiner, C., Riedel, T., Beigl, M., & **Ruoff, M.** (2021). Ensuring a Robust Multimodal Conversational User Interface During Maintenance Work. In *Mensch und Computer 2021*.

## Extended Abstracts and Workshop Papers

**Ruoff, M.**, Myers, B. A., & Maedche, A. (2022). Towards Interactively Contextualizing Natural Language Input in Data Visualization Tools. In *NLVIZ Workshop: Exploring Research Opportunities for Natural Language, Text, and Data Visualization*.

**Ruoff, M.**, Myers, B. A., & Maedche, A. (2021). ONYX: Towards Extending Natural Language Interfaces for Data Visualization Tools through Interactive Task Learning. In *NLVIZ Workshop: Exploring Research Opportunities for Natural Language, Text, and Data Visualization*.

**Ruoff, M.**, & Maedche, A. (2020). Towards Understanding Multimodal Interaction for Visual Data Analysis. In *31st IEEE Conference on Information Visualization (2020)*.

# Eidesstattliche Versicherung

gemäß § 13 Abs. 2 Ziff. 3 der Promotionsordnung des Karlsruher  
Instituts für Technologie für die KIT-Fakultät für Wirtschaftswissenschaften

1. Bei der eingereichten Dissertation zu dem Thema *Effective Natural Language Interfaces for Data Visualization Tools* handelt es sich um meine eigenständig erbrachte Leistung.
2. Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtlich oder sinngemäß aus anderen Werken übernommene Inhalte als solche kenntlich gemacht.
3. Die Arbeit oder Teile davon habe ich bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.
4. Die Richtigkeit der vorstehenden Erklärungen bestätige ich.
5. Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt. Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

Karlsruhe, den

**Marcel Ruoff**