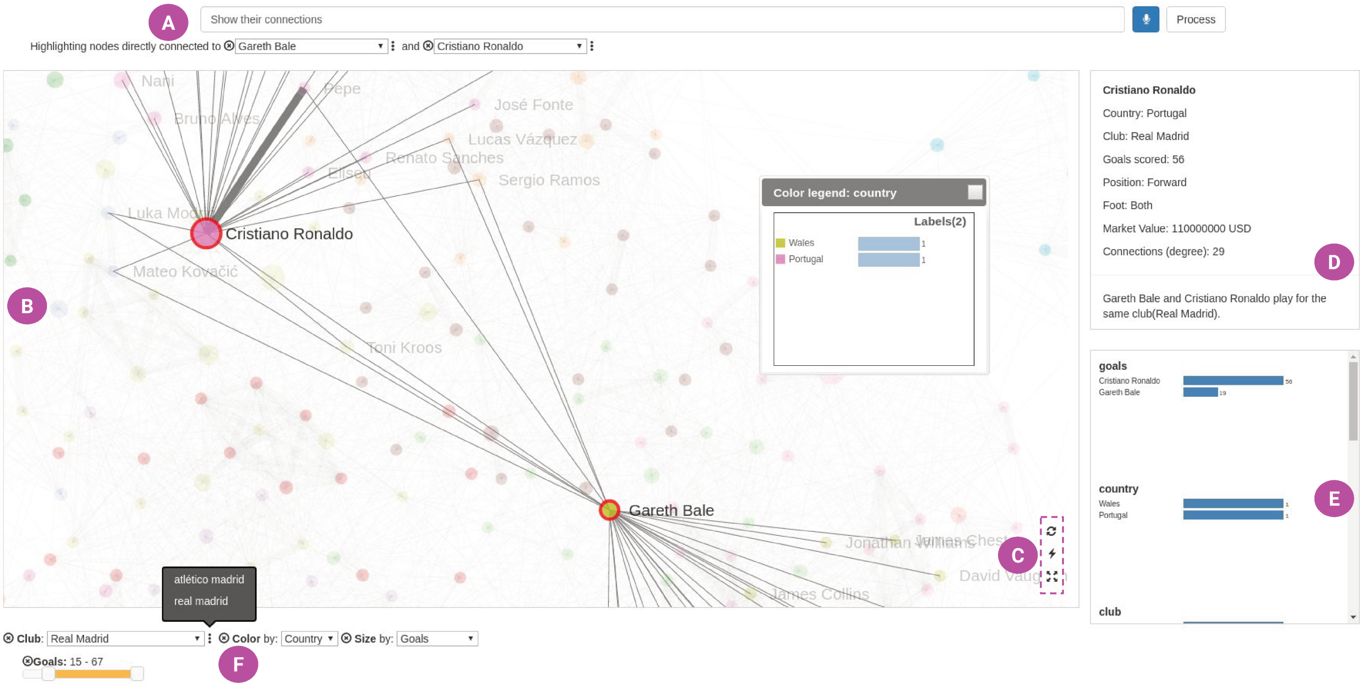
Orko: Facilitating Multimodal Interaction for Visual Exploration and Analysis of Networks

Arjun Srinivasan and John Stasko



Show their connections

Highlighting nodes directly connected to Gareth Bale and Cristiano Ronaldo

Cristiano Ronaldo
Country: Portugal
Club: Real Madrid
Goals scored: 56
Position: Forward
Foot: Both
Market Value: 110000000 USD
Connections (degree): 29

Gareth Bale and Cristiano Ronaldo play for the same club(Real Madrid).

Club: Real Madrid | Color by: Country | Size by: Goals
Goals: 15 - 67

Fig. 1. Orko's user interface being used to explore a network of European soccer players. Cristiano Ronaldo, Gareth Bale, and their connections are highlighted. Connected nodes with lower opacity do not meet all the filtering criteria. Interface components: (A) Natural language input and action feedback row (B) Network canvas (C) Quick access icons (D) Details container (E) Summary container (F) Filters and visual encodings row.

Abstract—Data visualization systems have predominantly been developed for WIMP-based direct manipulation interfaces. Only recently have other forms of interaction begun to appear, such as natural language or touch-based interaction, though usually operating only independently. Prior evaluations of natural language interfaces for visualization have indicated potential value in combining direct manipulation and natural language as complementary interaction techniques. We hypothesize that truly multimodal interfaces for visualization, those providing users with freedom of expression via both natural language and touch-based direct manipulation input, may provide an effective and engaging user experience. Unfortunately, however, little work has been done in exploring such multimodal visualization interfaces. To address this gap, we have created an architecture and a prototype visualization system called Orko that facilitates both natural language and direct manipulation input. Specifically, Orko focuses on the domain of network visualization, one that has largely relied on WIMP-based interfaces and direct manipulation interaction, and has little or no prior research exploring natural language interaction. We report results from an initial evaluation study of Orko, and use our observations to discuss opportunities and challenges for future work in multimodal network visualization interfaces.

Index Terms—Multimodal interaction, network visualization, natural language input, direct manipulation, multitouch input

1 Introduction

Data visualization systems have predominantly been developed for WIMP-based direct manipulation interfaces. More recently, there has been increased interest within the visualization community to explore visualization on devices and settings where conventional input modalities such as keyboard and mouse are not available (commonly referred to as post-WIMP systems [32]).

One line of research has explored how data visualization can be conducted on large (wall or tabletop) and small (tablets) displays facilitating touch input via a finger or pen [44,45,54]. These efforts have shown that developing visualization systems on new devices requires significant changes in the interaction style of a visualization system's interface [8,20,56].

Another line of research has explored natural language as an input modality for visualization systems [21,60,64]. Natural language interfaces (NLIs) take a person's utterance as input, and then create or modify visualizations in response to the utterance. Natural language is a promising interaction modality for visualization systems because

- *Arjun Srinivasan and John Stasko are with Georgia Institute of Technology. E-mail: arjun010@gatech.edu, stasko@cc.gatech.edu*

Manuscript received 31 Mar. 2017; accepted 1 Aug. 2017.
Date of publication 28 Aug. 2017; date of current version 1 Oct. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2017.2745219

people often can express their questions and commands more easily using natural language than by translating their intentions to interface actions [6, 24].

While recent research has explored touch and natural language input, each modality largely has been explored on its own. Prior work within the HCI community has shown, however, that *multimodal* interaction can significantly enhance user experience and system usability. For instance, in a study comparing the use of speech and pen-based input individually to a combination of both input modalities in the context of an interactive mapping system, evaluations showed that multimodal interaction significantly improved error handling and reliability: people made 36% fewer errors with a multimodal interface [46]. A recent evaluation of an NLI for visualization also indicated potential value in combining direct manipulation and natural language as complementary interaction techniques [60].

In our work we explore multimodal interaction with visualization, with a particular focus on network-based data. Network visualizations are useful for describing and exploring data relationships in many domains such as transportation planning [39], biology [41], and the social sciences [43]. Interaction plays an important role in network visualization systems because users need to engage with elements of interest (e.g., nodes, links) and interact with interface widgets (e.g., sliders, dropdown menus) in order to better understand the data.

Until now, little work has been done in exploring natural language and multimodal interfaces for network visualizations. We hypothesize that the freedom of expression provided by natural language can be a powerful addition to direct manipulation-based network visualization tools. Natural language combined with direct manipulation may facilitate a better analytical flow by allowing people to more naturally communicate operations such as finding nodes and paths, even while interacting with different parts of the visualization system.

While multimodal interfaces for network visualization appear to be a promising idea, clearly further research and evaluation is needed to determine whether the conjectures above are true. Will such interfaces facilitate common network exploration and analysis tasks? Will they lead to an improved user experience? To answer such questions, we developed a system, *Orko*, that facilitates direct manipulation and natural language based multimodal interaction with network visualizations. The primary contributions of our work are as follows:

- Building upon existing visualization task taxonomies, we highlight the types of queries and interaction patterns that a multimodal network visualization tool needs to support.
- Through the design and implementation of Orko, we exemplify how multimodal input can be processed to generate context that can be used to complement the individual modalities. We discuss how coupling this context with time lags between inputs helps facilitate unimodal (touch or speech only), and simultaneous or sequential multimodal interaction with a given visualization.
- We report observations from an evaluation of Orko that show people naturally use multimodal input when performing network visualization tasks. Further, we discuss varying preferences for modalities and interaction patterns highlighting the need for future visualization tools to further explore multimodal interaction.

2 Related Work

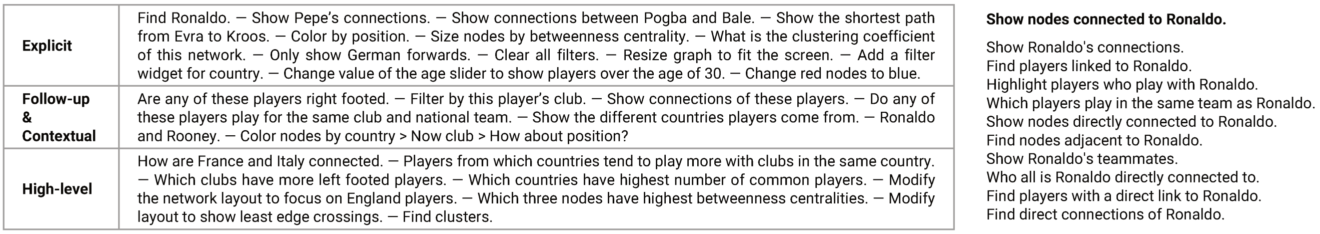
Networks have been studied extensively by the visualization community. Many existing systems (e.g., [5, 19, 66]) allow people to interactively explore networks by visualizing them using different layouts and representations. Various researchers have proposed different task taxonomies for network visualizations. Lee et al. [33] present a list of tasks commonly encountered while analyzing network data. They define network specific objects and demonstrate how complex tasks could be seen as a series of low-level tasks [2] performed on those objects. Pretorius et al. [53] give an overview of the entities and properties of multivariate networks and present a taxonomy for general visualization tasks. They describe how multivariate network tasks can be composed of lower-level tasks of the general taxonomy. Saket et al. [57] present a group-level task taxonomy for network visualizations and characterize a subset of the proposed tasks using a multi-level typology of abstract visualization tasks [11]. As part of our work, we utilized these taxonomies to understand the tasks that our system would need to support and the types of questions people may ask.

A large part of our motivation to explore input modalities (e.g., natural language and touch) that are afforded by post-WIMP interfaces is based on opportunities and challenges highlighted by Lee et al. [32]. The authors specifically identify "going beyond the mouse and keyboard" and "providing a high freedom of expression" as two of the five key opportunities for research within the visualization community. Given the widespread adoption of direct manipulation as an interaction technique, visualization systems on post-WIMP interfaces have largely been explored using touch-based input [8, 18, 45, 54–56]. Along similar lines, there has been work exploring network visualizations, particularly node-link diagrams, on post-WIMP interfaces. Schmidt et al. [59] proposed a set of multi-touch gestures for selections in network visualizations. Frisch et al. [20] explored how people interact with network visualizations on interactive tabletop surfaces using multimodal interaction in the form of touch and pen-based input. More recently, Cordeil [15] et al. investigated the relative advantages of immersive technologies like CAVE-style environments and low-cost head-mounted displays (HMDs) for collaborative analysis of network connectivity.

Another input modality that has recently gained renewed interest for data analysis and visualization is natural language. There are several NLIs that allow users to ask questions of their data in the context of databases (NLIDBs) [1, 4, 30, 62]. More recently, Li and Jagadish [35] showed how even novice users were able to specify complex SQL queries using natural language. NLIs for visualization, have been explored both in the research community and as commercial software (e.g., IBM Watson Analytics). Cox et al. [16] presented some of the earliest work in the space of NLIs for visualization. They combined natural language and direct manipulation in a data visualization environment and showed that multimodal input provides more expressibility than a single modality [49]. The Articulate system [64] presents a natural language interface for visualization. It maps user queries to tasks and uses these tasks in combination with data attributes to generate required visualizations. DataTone [21] is another system that allows users to generate visualizations using natural language queries. It specifically focuses on detecting ambiguity in natural language queries and uses a mixed-initiative approach to resolve this ambiguity and help users iteratively construct visualizations. Kumar et al. [31] present Articulate2 an initial prototype of a conversational interface for visualization which aims to explore the dialogue between a user and a system to generate visualizations. The Eviza system [60] presents a visualization and allows users to ask questions in the context of the given visualization. In doing so, Eviza enables users to have an interactive conversation with the system. By emphasizing the ability for a user to continually revise and update their queries, Eviza seeks to provide a rich dialog with the visualization.

We also use the context of a given network visualization as a starting point for a conversation between our system and its users. Our work builds upon techniques presented by prior work and extends them to support tasks required to interact with network visualizations. We use a combination of grammar-based and lexicon-based parsing techniques to interpret natural language queries. Further, while existing NLIs for visualization facilitate some level of multimodal input (e.g., Eviza lets users ask a query and then select points on a map), these systems focus more on responding to user queries rather than exploring how people may use multiple modalities. Additionally, most existing NLIs focus on WIMP-based settings and largely let users interact via a mouse and keyboard.

The broader HCI community, on the other hand, has explored multimodal interfaces facilitating natural language in post-WIMP settings [65]. Possibly the first, and one of the best known multimodal systems was presented in Bolt's article "Put-that-there" [9] in 1980. Following this, there were many systems that explored multimodal interaction using a combination of of touch or pointing devices and natural language for a variety of applications including graphics manipula-



Explicit	Find Ronaldo. — Show Pepe's connections. — Show connections between Pogba and Bale. — Show the shortest path from Evra to Kroos. — Color by position. — Size nodes by betweenness centrality. — What is the clustering coefficient of this network. — Only show German forwards. — Clear all filters. — Resize graph to fit the screen. — Add a filter widget for country. — Change value of the age slider to show players over the age of 30. — Change red nodes to blue.
Follow-up & Contextual	Are any of these players right footed. — Filter by this player's club. — Show connections of these players. — Do any of these players play for the same club and national team. — Show the different countries players come from. — Ronaldo and Rooney. — Color nodes by country > Now club > How about position?
High-level	How are France and Italy connected. — Players from which countries tend to play more with clubs in the same country. — Which clubs have more left footed players. — Which countries have highest number of common players. — Modify the network layout to focus on England players. — Which three nodes have highest betweenness centralities. — Modify layout to show least edge crossings. — Find clusters.

***(a)** Possible query types*

Show nodes connected to Ronaldo.

Show Ronaldo's connections.
Find players linked to Ronaldo.
Highlight players who play with Ronaldo.
Which players play in the same team as Ronaldo.
Show nodes directly connected to Ronaldo.
Find nodes adjacent to Ronaldo.
Show Ronaldo's teammates.
Who all is Ronaldo directly connected to.
Find players with a direct link to Ronaldo.
Find direct connections of Ronaldo.

***(b)** Different ways of asking the same query*

Fig. 2. An illustration of the variety of potential natural language utterances.

tion [28,52,68], writing and painting [23,67], exploring maps [14,46], among many others. As part of our work, we investigate how the two emerging input modalities of touch and natural language can be combined to facilitate multimodal interaction with network visualizations. While we focus on touch and speech based input, we designed our prototype such that it also works on WIMP-based systems to enable future comparisons between the two settings.

3 Challenges in Interpreting Natural Language Input for Network Visualization

While the designers of network visualization systems generally understand the challenges and issues of implementing direct manipulation interfaces, natural language interfaces provide an altogether different set of challenges. For instance, consider the different types of queries that people may pose to such a system. (We use the term "query" throughout the remainder of the article to refer to any type of utterance such as a command, comment, or question from a person.)

To help understand the possibilities, we collected a set of sample queries by referring to existing network visualization task taxonomies [33, 53, 57] and through pilot studies with students and research colleagues. We then used an affinity diagramming approach, grouping similar queries and iteratively refining the groups according to how precise queries were in terms of conveying user intent. We made the assumption that user intent is conveyed by tasks and the values they map to. We then combined groups under broader categories. This process resulted in three higher-order categories of queries: *explicit*, *contextual and follow-up*, and *high-level* (Figure 2a).

For the remainder of the article, we will use a specific example, a network of European football (soccer) players, to help ground our discussions and make concepts more explicit. The network contains 552 players (nodes) and 6772 connections (links) between those players. A link is created between two players if they play for the same club team (league team) or the same national team. In addition to the name, club, and country information, other attributes associated with players include number of goals scored, market value (in USD), age, club, country, preferred foot, and field position.

Of the three categories of queries introduced above, explicit queries typically provide sufficient information in terms of both tasks and values for a system to parse the query and generate a response. Command-like queries can be considered as a subset of explicit queries. Examples of these types of queries include "Find Ronaldo" or "Show the shortest path from Evra to Kroos".

Given the conversational nature of NLIs, users may frequently pose queries that are follow-ups to their previous queries. Such queries typically lack references to tasks or values associated with a task. For example, consider the query "Color nodes by country" followed by "Now club", followed by "How about position?". While the queries following the first one appear incomplete individually, they refer to the coloring task implied by the first query. In a multimodal setting, users may even present queries that are follow-ups to their actions on the direct manipulation interface. We refer to such questions or queries as "contextual" queries. For example, if the user selects a subset of nodes and utters the query "Show connections of these players", the system needs to detect that the user is referring to the selected players and automatically map the task of finding connections to those players.

Finally, high-level queries are generally open-ended user questions. These questions typically do not specify explicit tasks and can be interpreted and answered in multiple ways depending on the interpretation. Examples include questions like "How are France and Italy connected?" or "Players from which countries tend to play more for clubs in the same country?" To generate a response for such queries, a system typically needs to break the question into smaller tasks, solve those tasks and combine the results into a final response.

The sheer variety of ways of saying something poses another challenge. Given the freedom of expression associated with natural language, a person's particular intent can be stated in multiple ways. For example, Figure 2b shows some of the many ways that a person could state a query to find the connections of a node. Additionally, other challenges of natural language such as ambiguity exist as well. Ambiguity may exist not only at a syntactic level (e.g., misspelled words) but also at a semantic level in the context of words (e.g., synonyms and hypernyms) and the data (e.g., "high goal scorers" can refer to players with over 10 goals, 20 goals, and so on).

The presented examples and classifications in Figure 2 are not exhaustive, nor is our goal to provide a definitive taxonomy of query types. Instead, the intent of this section is to present a general overview of the input space of natural language queries for network visualizations in the targeted multimodal setting, and highlight the associated complexities. We will reference these different query types later when describing Orko's functionality.

4 Orko

4.1 Design Goals

Although we sought to achieve a variety of objectives while building Orko, two primary high-level goals drove the design of the system.

DG1. Facilitate a variety of network visualization tasks. A core goal for Orko was to support exploratory analysis similar to that done in existing desktop-based network visualization systems (e.g., [7,37]), but in a multimodal setting. More specifically, we wanted to focus on supporting a variety of tasks including topology-based, attribute-based, and browsing tasks in context of the taxonomy by Lee et al. [33], a subset of structure-based and attribute-based tasks at a node (entity) level per the taxonomy by Pretorius et al. [53], and finally, a small subset of group-only, and group-node tasks as specified in the taxonomy by Saket et al. [57].

DG2. Facilitate a variety of input integration patterns for multimodal interaction. Multimodal interfaces provide more freedom of expression allowing users to interact with the system in multiple ways. For such systems, input patterns are typically categorized based on the number of modalities used and temporal synchronization between modalities [48, 50, 51]. Given the system's primary usage setting (touch and speech input), our goal was to support a variety of these input patterns, including unimodal input (touch-only, speech-only), sequential multimodal input (e.g., selecting nodes via touch followed by a pause followed by a *find connections* query), and simultaneous input (e.g., issuing a *Color nodes* query while highlighting a node's connections). More specifically, in addition to incorporating both input modalities individually, this goal required us to consider synergies between the two modalities while designing the interface, and support not just explicit and follow-up but also contextual queries (Figure 2a). We chose not to focus on high-level questions (Figure 2a) as we believe they are more specific to NLIs. Further, these queries pose a challenge of overcoming the variability in responses that is beyond the scope of our current work focusing on multimodal interaction.

4.2 System Overview and User Interface

Implemented as a web-based tool, Orko runs on both conventional desktops/laptops and other devices supporting touch and speech interaction. Figure 1 shows the system's user interface, and we provide an overview of its capabilities and operations below.

At the top of the window (Figure 1A), is an input box that shows the typed or spoken natural language query. Users can input natural language queries in two ways: pressing the microphone button (🎙) next to the input box and speaking the query, or by saying "System" and then speaking the actual query (similar to the interaction with Amazon.com's Alexa [3] or Google's Assistant [22]). Below the input text box is an action feedback row that conveys the changes made to the interface as part of the response to a query. Orko also provides optional audio feedback where the system narrates the content of the feedback row. Some examples of feedback messages include "Highlighting nodes directly connected to Gareth Bale", "Changed coloring attribute to country", or "Sorry I'm unable to process that query. Can you try asking the question differently?". In some cases, the feedback row is interactive, allowing users to modify the input query (discussed further in section 4.4.3).

The network canvas (Figure 1B) presents the network visualization with entities represented as circles and connections between entities represented as lines connecting the circles. Node positions are determined by D3's force-directed layout [10]. By default, labels are hidden to avoid clutter. Labels are only shown when nodes are selected or highlighted. A click or tap on the canvas background clears selections. Quick access icons (Figure 1C) are provided at the bottom right of the canvas to reset the view by clearing all selections and filters (↻), unpin all pinned nodes and reset the force-layout (ϟ), and re-center the network (⊠).

As mentioned earlier, we designed Orko primarily for touchscreen devices. Consequently, we implemented the interactions within the visualization such that they do not rely on hover (unavailable on commonly found touchscreens [29]) and can work on both touch and pointing devices (e.g., mouse, stylus). Users can single-tap on nodes to get details, double-tap to highlight a node's connections, drag a node to modify the layout by pinning a node, long press (individually) on two nodes to highlight the shortest path between them, zoom using a pinch-gesture, and pan using a single finger drag on the background of the canvas. When a node's connections are highlighted, details of individual links can be seen using a single-tap on the link. Keeping in mind issues like the fat-finger problem [61], we sized nodes such that it is easy to tap them and add buffer space while detecting interactions with links to handle offset touches. The details container (Figure 1D) shows attributes of selected nodes and link descriptions.

The summary container (Figure 1E) shows bar charts that complement the selections on the network visualization. The charts present attribute-level summaries for active (highlighted) nodes. The bar charts are coordinated with the network visualization and facilitate brushing and linking. The bars within charts are sorted in descending order of width from top to bottom to facilitate ordered comparisons. The charts dynamically reorder based on the sequence of user interactions with attributes—the most recently used attribute is always shown on top of the container. We made this design decision of reordering charts based on two hypotheses. First, users would find it beneficial to see the summary statistics for attributes they most recently used on top to answer possible questions they have in mind for the attribute (e.g., if a user filters nodes by goals scored, the summary chart for goals would show up on top presenting a ranked list of the highlighted players and the number of goals they scored). Second, since the container shows all attributes available in the dataset, it could help facilitate an analytical conversation by triggering questions in users' minds about potential attributes they may not have considered.

The filters and encodings row (Figure 1F) presents the various filtering and encoding options. Filtering widgets include range sliders for numerical values and dropdown menus for categorical values. Visual encoding widgets include dropdowns for assigning node coloring and sizing attributes. Users can choose to keep the widgets on or remove them at any point using the ⊗ icon next to each widget.

Operation	Applicable to	Sample queries
Find nodes	Label attribute	"Find Wayne Rooney", "Show Bale and Ronaldo", "Highlight Iniesta"
Find connections	Label attribute	"Show Griezmann's teammates", "Highlight players connected to David de Gea", "Find players who play with Sergio Ramos", "Show players connected to these players"
Find path	Label attribute	"How are James McCarthy and Toni Kroos connected", "Show connection between Giroud and Neuer", "Highlight a path from Sergio Busquets to Patrice Evra", "Show a path between these nodes"
Filter nodes	Categorical & numerical attributes	"Show left footed Madrid players", "Find players with more than 15 goals", "Highlight German players over the age of 28", "Remove age and market value filters"
Color nodes	Categorial attributes	"Color by country", "Highlight player positions", "Color players by their field positions", "Can you add coloring based on foot"
Size nodes	Numerical attributes	"Size by market value", "Resize nodes to highlight age", "Size nodes by betweenness centrality"
Interface actions	-	"Refresh view", "Clear all filters and selections", "Deselect all nodes", "Re-center graph"

Fig. 3. Currently supported operations with sample queries. The *label* attributes refer to the attributes defining the nodes.

To support the targeted categories of network analysis tasks **(DG1)**, Orko currently provides a set of seven more specific low-level operations listed in Figure 3. In the context of a recent categorization of potential tasks people try to perform in visualization related NLIs [63], Orko currently focuses on supporting visualization-specific interactions, and provides basic support for low-level analytical operations and system control-related tasks.

4.3 Usage Scenario: European Soccer Players

To provide a better sense of how the system works, we describe a hypothetical usage scenario below. Imagine Joe, a visitor at the FIFA World Football Museum, interacting with the European soccer player network using Orko on a touch and speech-enabled display (The scenario is illustrated more explicitly in a storyboard sequence and a video as part of the supplementary material for this paper.)

Orko: Shows the network using a force-directed layout.

Joe: To focus on high scoring players, says *"Show top goal scorers"*.

Orko: Identifies ambiguity in the question due to the word 'top'. Consequently, adds a slider for the *goals* attribute (Figure 4a).

Joe: Adjusts the slider and observes the summary charts to highlight the top five goal scorers in the network (Robbie Keane, Zlatan Ibrahimovic, Cristiano Ronaldo, Wayne Rooney, Lucas Podolski). To see connections of the highlighted players, says *"Show players connected to these players"*.

Orko: Detects that by "these" Joe is referring to the filtered set of five players, thus highlights their connections. Preserving the goals filter, shows the connections as faded nodes (similar to faded nodes in Figure 1B).

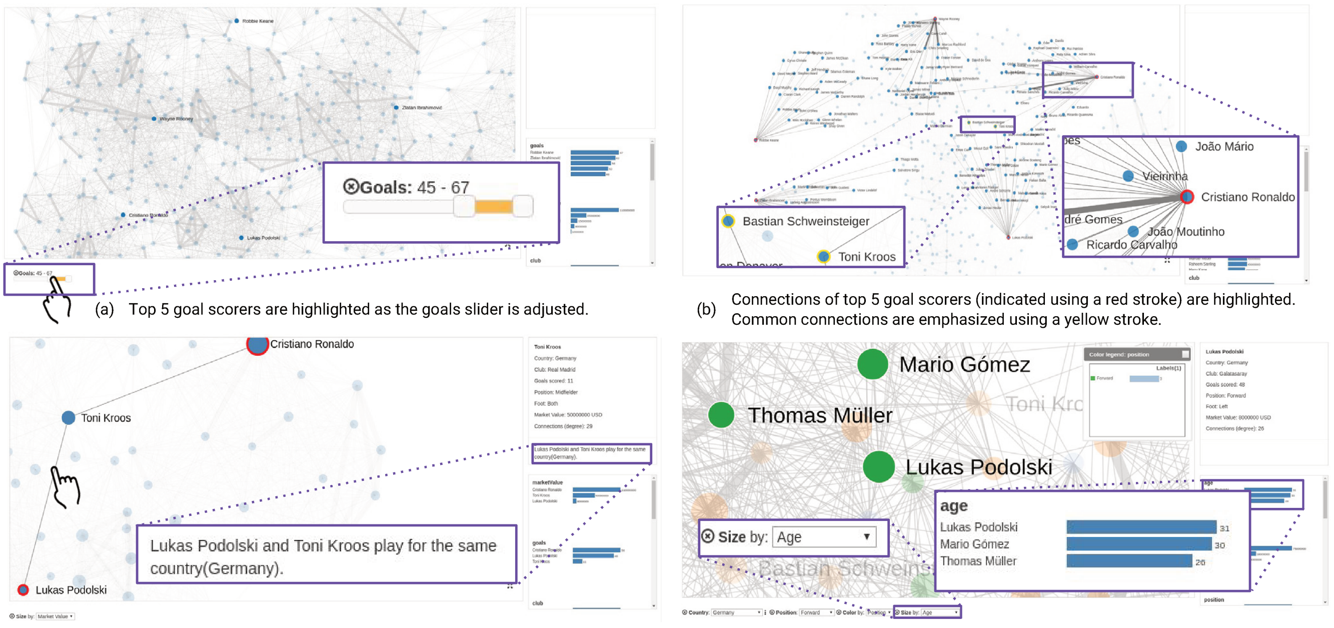
Joe: To highlight all connections of top five goal scorers, removes the goals filter.

Orko: Highlights top five goal scorers and their connections. Also highlights common connections between two or more of the top five goal scorers using a yellow stroke around the node (Figure 4b).

Joe: Modifies the layout by pinning the top five players in different locations. Using the modified layout and highlighted common connections, observes that Cristiano Ronaldo and Lucas Podolski have a common connection (Toni Kroos). Wayne Rooney and Lucas Podolski also have a common connection (Bastian Schweinsteiger). Robbie Keane and Zlatan Ibrahimovic have no common connections with the other top goal scorers. Wondering about the market value of players, says *"Size players by their salaries"*

Orko: Maps the word "salaries" to the attribute *market value* and resizes nodes by the market values of players.

Joe: Observes that Podolski, even though among the top five goal scorers, is paid much less in comparison to his teammates. Observes that Toni Kroos, who is the common connection between Podolski and Ronaldo is paid notably more than Podolski. Intrigued by this, says *"Highlight this connection between Ronaldo and Podolski"*.



(a) Top 5 goal scorers are highlighted as the goals slider is adjusted.

(b) Connections of top 5 goal scorers (indicated using a red stroke) are highlighted. Common connections are emphasized using a yellow stroke.

(c) Path between Ronaldo to Podolski is highlighted. Connection between Podolski and Kroos is displayed.

(d) Player nodes are re-sized by age. Summary panel is re-organized to facilitate expedited exploration.

Fig. 4. Scenes from the Orko usage scenario. Sub-figure captions summarize system states.

Orko: Highlights the path from Ronaldo to Podolski via Kroos (Figure 4c).
Joe: By tapping on the highlighted links, understands that both Kroos and Ronaldo play for the same club (Real Madrid) and Kroos and Podolski play for the same country (Germany). Curious to learn more about the differences in salaries between Podolski and his club and national teammates, double-taps on Podolski.
Orko: Highlights Podolski and his connections.
Joe: Notices that Podolski is a striker with a high number of goals and still is not even in the top ten players among his teammates (shown by the summary charts) in terms of market value. Wondering if field position has any correlation with salaries, says *"Highlight field positions of these players"*.
Orko: Colors nodes by the positions the players play in.
Joe: Scanning the highlighted players, notices three German strikers (Muller, Gomez, Podolski). Muller is paid most, followed by Gomez and Podolski. Using the summary charts, notices that in terms of goals, Podolski has most goals (48) followed by Muller (31) and Gomez (27). Intrigued by this fact, wonders about other attributes that may account for the differences in salaries between German strikers. To focus on the specific players, says *"Just highlight the German strikers"*.
Orko: Highlights the three German strikers and updates the summary container.
Joe: To compare the three players across different attributes, toggles through the list of attributes available for node-sizing.
Orko: Re-sizes nodes and re-orders the summary charts so that recently used attributes are shown on top (Figure 4d).
Joe: Stops at age attribute upon noticing that nodes representing Podolski and Gomez are much larger than the node representing Muller. Confirming the values using the age bar chart at the top of the summary container, hypothesizes that the age may be the factor leading to salary differences (since younger players are typically paid more).

4.4 System Architecture and Design

To facilitate multimodal interaction and support the different combinations of input patterns **(DG2)**, Orko employs a client-server architecture shown in Figure 5. Below we describe the individual components highlighting their specific functions and how they communicate with each other to facilitate multimodal interaction.

4.4.1 Processing direct manipulation or natural language input

All direct manipulation (e.g., mouse, pen, touch) events triggered on Orko's interface (e.g., tapping a node, changing a dropdown value, adjusting a slider) are collected and handled by the *interface manager*. We use the HTML5 webkit speech recognition API for detecting voice-input and performing speech-to-text translation. Once a query string is available (either via speech-to-text or user-typed), the interface manager sends it to the server for interpretation. On the server, the *query processor* parses natural language queries and generates a list of actions that need to be taken in response to a query.

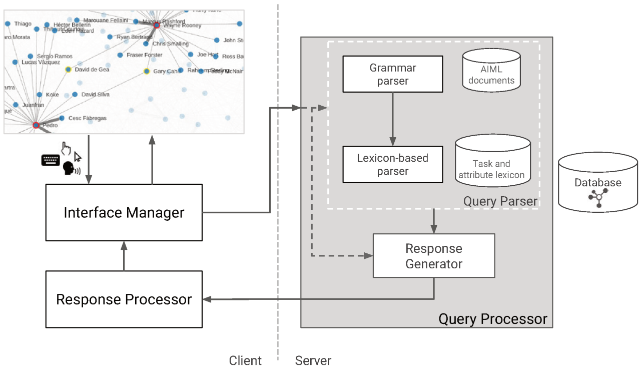


Fig. 5. Orko system architecture.

To parse natural language queries, in comparison to existing NLIs for visualization that tend to use either a grammar-based approach (e.g., [16, 64]) or a lexicon-based approach (e.g., [21]), we implement a two-step approach combining both grammar and lexicon based parsing techniques. This approach lets the system parse queries that match a grammar pattern instantaneously while at the same time, also having a more general back-up parsing strategy based on a combination of keyword-matching and heuristics for query patterns not covered by the grammar.

Patterns for the grammar parser are specified using Artificial Intelligence Markup Language (AIML) [13]. The patterns were generated based on question types and examples presented in existing task taxonomies [33, 53, 57] and pilot studies with students and research colleagues. We use a python based AIML interpreter (PyAIML) trained with the AIML files to parse incoming query strings. The interpreter builds a directed pattern tree and employs a backtracking depth-first search algorithm for pattern matching. For a given query, the grammar parser seeks to identify operations (Figure 3) specified and substrings containing references to attributes or values the operations apply to (analogous to a set of non-terminal symbols in a context-free grammar [17]). If there is no matching pattern found, the entire query is forwarded to the second parser, else, only the target referencing substring is sent to the lexicon-based parser. For instance, given a query like "Show connections of Ronaldo", the grammar parser identifies that the operation is *find connections* and the target is 'Ronaldo' (which is sent to the second parser). Alternatively, a query like "Show only if Barcelona and left foot" may not match an existing pattern and will be forwarded as-is to the lexicon-based parser. The lexicon used consists of attributes derived from the dataset (e.g. *goals*, *country*, *names*) and manually specified keywords (e.g., teammates, adjacent, striker) that help identify attributes, values, and operations in a given query. Some of these keywords are generic and apply to multiple datasets (e.g., adjacent) while others are dataset specific (e.g., striker, teammate). While we leverage existing lexical databases like WordNet [42] to support using synonyms (discussed further below), there always will be dataset-specific cases that are not supported by such general databases (e.g., using "striker" instead of "forward" for position). For such cases, in our current implementation, both, domain-specific grammar patterns and dataset-specific keywords should be manually added the first time a dataset is loaded.

Given a portion of the query or the entire query string, the lexicon-based parser first performs stemming and removes stop words (with the exception of conjunction/disjunction phrases). It then extracts n-grams (with n ranging from 1 to the number of words in the input string). For each n-gram, it identifies POS tags (e.g., noun, verb) and entity types (e.g., person, location, organization) using NLTK [36] and Stanford CoreNLP [40]. n-grams not containing entity types relevant to the dataset or values that may apply to filters (e.g., numbers) are discarded. This filtering helps improve performance by ignoring n-grams that do not contain relevant information. Next, the relevant n-grams are compared to logically similar lexical entries (those that have related POS tags or entity types). This similarity-based comparison again helps improve performance by avoiding matches against potentially irrelevant values (e.g., comparing people to locations). Building upon existing work [21, 60], we use the cosine similarity and the Wu-Palmer similarity score [70] when comparing n-grams to lexical entries. These scores help in detecting both syntactic (e.g., misspelled words) and semantic (e.g., synonyms, hypernyms) ambiguities. If there are no operations identified by the grammar parser, similar to Gao et al. [21], we use keyword-matching and a combination of POS-tags and dependency parsing techniques [40] to identify operations specified in a query. In summary, for the query "Show only if Barcelona and left foot", the lexicon-based parser identifies a *filter* operation, a *club* ("Barcelona FC"), and a value for *foot* ("left").

4.4.2 Managing multimodal input

In addition to its focus on network visualizations, Orko's primary difference compared to related systems (e.g., [16, 60]) is its support for various multimodal interaction patterns listed in **DG2**. As an example of the input patterns the current framework supports, consider the case of finding connections of a set of top goal scoring players for England. A user could accomplish this via only touch by applying multiple filters (for country and goals) and double tapping nodes to highlight connections. Instead, one could also use speech alone to perform the same task (using a single query like "Show connections of English players with more than 20 goals" or multiple smaller queries). Alternatively, a user could use a combination of the two modalities and: (1) apply filters (via touch) and follow it with a spoken query (e.g.,"show adjacent nodes"), or (2) apply filters via speech and then double-tap nodes, or (3) do both filtering and uttering a query simultaneously (starting with either of the two modalities). In cases (1) and (3), the context generated by one input is used to complement the second and highlight connections of the filtered nodes. For (2), the system processes the two inputs individually as described in the subsection above, preserving filters from the spoken query.

To support the patterns described above, the system needs to first classify input patterns and then share relevant information collected across input modes to appropriately respond to the user input. To accomplish this, Orko first classifies an input pattern as unimodal, sequential, or simultaneous. To classify an input pattern, we use a combination of interface context and time lag between user inputs. The interface context is tracked using an object that stores information about active/highlighted nodes, filters applied, encodings used, previous interaction modality used, and operations and target values in the last specified query. Both the interface manager and the query processor continually update this context object based on user inputs and actions.

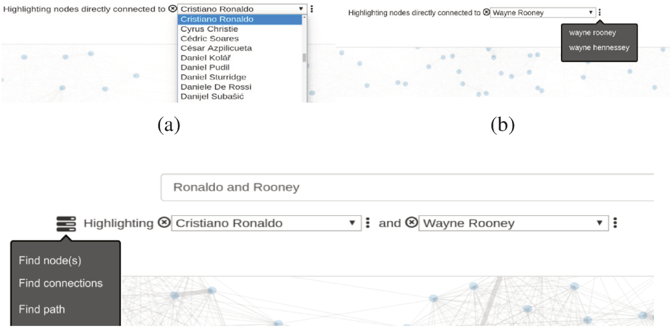
When a user input (touch or speech) event occurs, we check in paralell if there is a change in the modality used between inputs. If so, we further check if there is any missing information in the input (e.g., missing target value in a query) and a corresponding interface context that can be applied to the current input. For example, if a user selects two nodes (via touch) and then issues a query "Find connections", Orko can leverage the context of the selected nodes and apply it to the user query. We use a heuristic approach and mappings between operations and attribute types (Figure 3) to decide if a context applies to an input. However, an applicable interface context could be generated in both sequential and simultaneous input.

To differentiate between the two, when there is an applicable interface context, we also check the time lag between the previous and recent input. Based on prior work on multimodal input patterns [51] and our pilot studies, we differentiate between sequential and simultaneous input based on a time lag of two seconds between modalities. We make this differentiation to decide when context from the previous input should not be applied to the current one. For example, consider a case where there are no selected nodes and a user issues a query "Show nodes connected to" and follows it by a long pause. Now, the user adds a filter (via touch) to highlight the Spanish players. If the context from the query is applied by default, connections of the filtered nodes will automatically be shown. However, after such a pause, it is likely that the user was trying to perform a new filter action and wanted to ignore the previous query. In such cases, since we know that the pattern in this case was sequential, we can choose to not apply the system context and ignore the previous query instead.

4.4.3 Supporting follow-up queries and query refinement

To handle follow-up queries (discussed in Section 3), we implement a conversational centering [26, 27] (or immediate focusing [25]) based approach. The centers are maintained by the query processor and include operations, attributes, and values. We *retain*, *shift*, or *continue* centers across utterances [27] depending on the information shared or added between queries. Consider the query "Show only Real Madrid players" followed by "Show strikers". In this case, the *club* filter "Real Madrid" is retained across queries, and a *position* filter ("striker") is added after the second query (a continue operation). Now, when another query "Show defenders for Barcelona" is presented, the center is shifted to "defender" and "Barcelona FC" respectively.

Since we wanted to test Orko in a speech+touch setting where typing to modify specific parts of a query or repeatedly uttering similar queries can become tedious, while designing Orko, we also considered interface elements that may assist users in modifying their queries. We considered ways in which we could assist users to ask follow-up queries that refer to the same operation but different target(s) (e.g. "Show Real Madrid players" followed by "Show Barcelona players"). Such queries can be very common during network exploration, particularly while users try to scan through different node groups. To assist in constructing such follow-up queries, Orko adds *query manipulation widgets* (dropdowns and sliders) to the action feedback row highlighting the domain of input values for an attribute alongside the operation being performed (e.g., Figure 6a). Hence, for the club example discussed above, the user can specify a *filter* by club query once and then keep updating the club names for consecutive queries using the dropdown. Users can still ask follow-up queries or modify existing queries by typing if they prefer to do so. To allow users to instantly revert back to the original query, it is preserved in the text box (Figure 1A).



(a) (b)

(c)

Fig. 6. Query manipulation and ambiguity widgets. (a) Dropdown menu to change player name in a query. (b) Tooltip showing values matched to an ambiguous word “Wayne”. (c) Tooltip suggesting tasks guessed by the system for an underspecified query.

4.4.4 Highlighting ambiguity in queries

Similar to prior work [21, 60], we also provide ambiguity widgets to highlight and help users resolve ambiguity via Orko's interface. Currently supported ambiguity widgets include range sliders (for numerical attributes) and interactive tooltips (for label and categorical attributes). The interactive tooltips work as follows: vertical ellipsis icons (⋮) are added next to query manipulation dropdowns to notify users that the system detected ambiguous matches. When pressed, these icons display a tooltip showing the list of matched values (Figures 1F and 6b). Selecting an item in the tooltip updates the value of the adjacent dropdown. By default, the query manipulation dropdown is populated with the value most similar to the ambiguous string.

We also explored how the system can suggest operations in cases where a presented query lacks references to operations (or there is ambiguity in operations) and there is no preceding query or an applicable interface context to leverage. Such queries can be common with issues in speech detection that lead to partial detection of queries (e.g., “Rooney and Ronaldo” instead of “Find connections of Rooney and Ronaldo”). When the response generator detects attributes or values in a query but is unable to map them to a specific operation, it makes a “guess” at the operation that a user could perform based on a reverse mapping from attribute types to the list of available operations (Figure 3). An example of this is shown in Figure 6c where three operations, *find nodes*, *find connections*, and *find path* are suggested in response to an underspecified query using an interactive tooltip.

5 Evaluation

We conducted a user study to evaluate Orko's design and multimodal approach for interacting with network visualisations. We had three main goals: (1) assess basic usability of the system, (2) collect qualitative feedback on Orko's features and design, and (3) collect observational data on how people interact with network visualizations when they have the option of using multimodal input.

We initially considered performing a comparative study to measure usability and performance but struggled to find the right comparison since we could not find any publicly available network visualization tool that supported multimodal interaction. We also considered comparisons to unimodal interfaces (touch-only, NL-only) but decided against doing so because that evaluation would focus more on an examination of unimodal versus multimodal interaction, which was not our goal in this study. We finally decided on a study where all participants would interact with Orko and perform the same set of tasks using the European soccer player network described earlier.

Selecting and phrasing the study tasks themselves was another challenge. Due to the availability of speech as an input modality, posing tasks as direct questions was not an option since participants could simply repeat (say) the questions. Thus, we adopted the Jeopardy-style evaluation approach proposed by Gao et al. [21]. We gave participants a set of facts and asked them to modify the visualization to show each fact. For example, one such fact was “Robbie Keane only plays with Irish players.” To “answer” this, participants would need to show that all of Robbie Keane's connections belong to Ireland. (So, participants would need to find connections + color nodes by country, or find connections + scan summary charts, or find connections + scan individual nodes). We also added some entity naming tasks that required participants to explore the network to identify specific entities. The tasks again were framed so that participants could not simply parrot them to get an answer. For example, one of the questions was “Name an FC Barcelona midfielder. Identify at least two non-Barcelona midfielders the player is connected to.” To respond to the first part of this question, participants would have to name (using filter or find) and highlight a Barcelona midfielder. For the second part, they would have to highlight the player's connections with two other non-Barcelona players and show that those two are also midfielders.

To assess the prototype's usability, we used the standard 10 question SUS questionnaire [12]. We used SUS since it has been shown to provide a global measure of system satisfaction and sub-scales of usability and learnability [34]. Further, the correlation between SUS scores and task performance is similar to most other post-test questionnaires [58].

To collect qualitative feedback on Orko's design and features, we encouraged participants to think aloud and interact with the experimenter while performing the given tasks. We also conducted informal interviews at the end of the session asking participants about their experience and feedback on the system features.

5.1 Participants and Experimental Setup

We recruited six participants, ages 22 to 42, five male and one female. All participants had prior experience with visualization tools such as Tableau. All participants had some prior experience with network visualization tools (e.g., Gephi [7]). Two participants stated they had some prior experience working with touch-based visualization systems and only one participant (P2) had never used a voice-based system (e.g., Siri). In terms of domain knowledge, two participants were well acquainted with the sport of soccer and the data, and remaining four stated they were aware of it but did not follow the game or know much about the data. All participants interacted with Orko running on Google's Chrome browser on a 55” Microsoft Perceptive Pixels (PPI) device. The PPI was set to a resolution of 1920 x 1080 pixels.

5.2 Procedure

Participants were given a brief introduction (approximately 5-7 minutes) to the system's UI and the European soccer player dataset. For the UI, we highlighted the different components (Figure 1), possible touch-interactions (tap, double-tap, drag etc.), and told participants that they could use typed (using a virtual keyboard) or speech-based input for natural language interaction. We did not show any trials or videos of how participants could perform any specific task since we wanted to observe their natural behavior and reactions. Participants were then asked to try the touch and speech input using any interactions and commands they wanted to test, until they felt comfortable (approximately 1-3 minutes).

Next, we gave participants a list of 10 tasks printed on a sheet of paper and 30 minutes to interact with the system. The order of the tasks was randomized for each participant. The tasks contained a mix of Jeopardy-style [21] facts and entity identification questions. The questions were constructed using previously defined network visualization task taxonomies [33, 57] and included topology-level tasks, attribute-level tasks, browsing-tasks, and some group-level tasks. Details of the tasks are provided as part of the supplementary materials.

We told the participants that we would not be measuring how quickly they perform the tasks, so they should feel free and interact as naturally as possible. We recorded both video and audio of participants interacting with the system during these 30 minutes. Participants who finished the tasks before 30 minutes could continue exploring the data using the system if desired. Participants were then given a post-session questionnaire that consisted of SUS questions and questions asking them about their experience with Orko. We also conducted informal interviews asking the participants about what they liked/disliked most about the system and recorded their their responses as audio files. Sessions lasted between 40-60 minutes and participants were given a $20 Amazon gift card.

5.3 Results and Observations

5.3.1 System Usability Scale responses

All participants attempted each of the 10 tasks and on average, provided correct responses for 8.67 tasks. Figure 7 (right) summarizes overall SUS scores. Participants gave Orko an average SUS score of 75.42. SUS scores have a range of 0 to 100 and a score of around 60 and above is generally considered as an indicator of good usability [38]. The SUS scores indicate that even though the prototype is in its initial stages, participants in general found the interface and the interactions facilitated by Orko usable.

5.3.2 Interaction Patterns and preferences

Figure 7 (left) summarizes interactions for the six participants for each study task (descriptions provided as supplementary material). The cell values indicate the number of times an input modality was used to accomplish operations in Figure 3. For example, for a *find connections* operation, P1 used a combination of speech and touch (*find* query + double-tap) once and two speech queries (*find* + *find connections*) the second time (first and second row of the table respectively).

Of 181 total constructions, 92 (50.8%) instances of just spoken queries arose, unimodal touch accounted for 55 (30.9%), and multimodal interaction where both speech and touch were used sequentially made up the remaining 33 (18.3%) constructions. No instances existed where modalities were used simultaneously (a myth of multimodal interaction [47]). However, all participants used more than one input modality at least once while performing the study tasks. Interaction patterns varied for the same task across participants (e.g., P1 performed task T1 using a multimodal pattern of speech+touch whereas P2 performed the same task using a single speech query). Similarly, individual participants' patterns varied as they performed similar tasks multiple times too. For instance, P6 performed task T5 using a series of spoken, touch, and multimodal interactions but when performing a similar task T6, used only speech.

In general, speech was typically used for search, filtering, and topology-based tasks involving multiple nodes (e.g., finding path and common connections). Touch, on the other hand was typically used for tasks like highlighting connections of individual nodes and changing values of existing graphical encodings. However, preferences for modalities also varied across task types (Figure 3). For instance, for a *find connections* task, four participants (P1, P3, P4, P5) generally used a combination of speech (for find) and touch (for finding connections) whereas the remaining two participants used only spoken queries to see connections (P2, P6). For *filtering*, all participants used speech at least once (typically at the beginning). For the spoken queries, two participants (P1, P4) used longer queries with multiple filters (e.g., "*Show Barcelona midfielders*") whereas three participants (P2, P5, P6) used multiple single filter queries. One participant (P3) typically followed a spoken query with touch interactions for modifying filters. Preferences even varied for less-visualization specific tasks such as *coloring* and *sizing*. Four participants mostly used spoken queries to change node color/size whereas two (P3, P6) often used a combination of speech and touch for the same.

The use of multiple modalities (individually and together) to accomplish tasks and the variable nature of interaction patterns across participants highlights the need and potential value of multimodal interfaces that accommodate such varying user preferences.

Natural language interaction and interpretation. Participants commended Orko's natural language capabilities and felt it interpreted queries fairly well (Figure 7-right). Multiple participants were initially skeptical about natural language input based on their previous experiences but were pleasantly surprised by the system's capabilities and the usefulness of speech input. For instance, P6, who reported that she frequently used applications like Siri and Notes stated "*I was surprised by the speech feature. I did not expect it to work as well as it did*". She also mentioned that speech not only worked well but actually improved her experience with visualization tools. She said "*having worked with many visualization programs before, having to go through and manually clicking is really annoying epecially when you have a ton of dropdowns. So I really like the speech feature, I know it's still in a rudimentary stage but it does a really good job*".

In terms of query interpretation, there were only seven instances where Orko either did not respond or responded incorrectly to a query. Some of these were queries included operations that were not yet supported (e.g., layout change) while others queries had multiple values that were not separated by conjunctions. For example, for the query "Show connections of Rooney McCarthy and Stones" P5 expected the system to find connections of three players. The system, however, only showed connections of two players (Rooney and Stones), but still listing McCarthy within the ambiguity tooltip widget (Figure 6b). In such cases, participants typically thought of an alternative way to perform operations via touch or broke the query further into more explicit ones.

Although speech recognition was viewed favorably by participants in general, it was not perfect. On average, 16% of queries were missed or incorrectly translated from speech-to-text. The percentage was higher for some participants (e.g., 30% for P3) due to variations in accent and speaking tone. Speech detection issues did cause some frustration among participants. For example, P3 stated "*It was a little frustrating when the system did not understand my voice or did not react at all to voice*". Ambiguity widgets did help for incorrectly detected player names, but only twice. Participants typically used the virtual keyboard to fix their utterances since it happened only occasionally. The more common case was the system failing to detect queries. In such cases, participants either repeated queries by tapping the 🎤 icon (Figure 1A) or used touch input to proceed with the task. For example, when the system did not detect a participant's (P4) *find connections* query, the participant simply double-tapped the node to see its connections.

These observations further motivate the need to study multimodal interaction for visualization systems. With a growing number of NLIs, such examples show how users can leverage alternative modalities to counterbalance issues such as speech detection in NLIs.

Contextual and follow-up queries Based on prior work that has shown a high preference for queries where touch (or pen) is followed by speech input [51,69], we hypothesized that such contextual queries (Figure 2a) would be a common pattern. However, this was not the case. Only two participants (P2, P6) uttered such contextual queries that referred to nodes highlighted via touch interaction. Both P2 and P6 used a contextual query when highlighting connections within a group of nodes. They applied a *country* filter through the dropdown and then said "Show the connections of these nodes" (P2) and "Highlight connections" (P6). However, we suspect the nature of the study tasks and prior experience of participants with visualization tools may have had an effect on the reduced usage of this pattern. Observing users (including novice users) perform more open-ended tasks and exploring the use of contextual queries in the context of other visualizations is certainly a direction for future work. Additionally, Orko currently only supports a limited number of touch gestures. Three participants (P1, P4, P5) also expressed a desire in having more expressive multitouch gestures to select and move groups of nodes. Exploring if the availability of additional multitouch gestures [59] increases the use of contextual queries is another open question.

For follow-up utterances, queries involving continue operations [27] were most common (e.g., adding new filters). Follow-up queries with references to new values (e.g., "Filter to show Real Madrid" > "now Barcelona" > "now strikers") were only used five times (thrice by P2 and twice by P5), all during a filtering operation. Instead, participants preferred to repeat entire queries (e.g., "Filter to show Real Madrid" > "Filter to show Barcelona" > "Filter by strikers") and often also repeated existing filters (e.g., "Filter to show strikers" > "Show only strikers for England"). Given this behavior, the

	P1			P2			P3			P4			P5			P6			
	S	T	ST	S	T	TS	S	T	ST	S	T	ST	S	T	ST	S	T	ST	TS
T1			1	2					1			1			1	1			
T2	2			1			1					1			1	1			
T3	2	2	1	3	1		1		1	3	1		3	1		2			
T4	2		1	3			4					3		6		3			
T5	2			2				1	1			1	2	4		4	1	1	
T6	1		1	1				2	1	1			1	3		4			
T7	1	1		2	3		1	1	1		1	1	3	1		2	2		
T8	1		1	1			1	1	1			1	2	1		1			
T9	2			2					2	2			2	1		1		2	
T10	2	2	2	8	1	2		6	2	2	5		2	5		2	3		1

	P1	P2	P3	P4	P5	P6	Average
Overall SUS scores (out of 100)	80	70	82.5	80	52.5	87.5	**75.42**
Would want to use the system frequently (out of 5)	4	5	5	5	3	5	**4.5**
Found various functions well integrated (out of 5)	5	5	4	3	5	4	**4.33**
Natural language query interpretation (out of 5)	4	4	3	4	4	5	**4**

Fig. 7. (Left) Summary of interactions per study task for each participant. *S*: Speech, *T*: Touch, *ST*: Sequential speech+touch, *TS*: Sequential touch+speech. (Right) Participant respones for specific SUS questions and Orko's query interpretation

query manipulation widgets were not frequently used. Based on these observations, we believe future work could focus more on exploring elements like ambiguity widgets [21] and ways to help users correct their queries and potentially less on how systems could help users ask follow-up questions.

5.3.3 Reaction to system feedback and proactive behavior

A recent analysis of NLI utterances to visualization systems [63] highlights instruction and feedback as well as proactive system behavior as two areas for data visualization NLIs to explore. Along these lines, in Orko, we present both audio and textual feedback when responding to natural language queries. However, even after multiple modes of feedback, one participant (P2) repeated his query twice before he realized the query had already been executed. P2 also expressed that he would like the system to show the possible space of input queries and said "*If the system used the keyboard, an auto-complete function would be very helpful*". Such observations and feedback indicate that we need to explore more ways to surface feedback and potentially expose the input query space on post-WIMP interfaces.

Orko exhibits proactive behavior with its suggestion of tasks for underspecified queries and by rearranging the summary charts based on user interactions and queries. The task suggestion feature was only triggered thrice (twice for P5 and once for P3). In both cases, the participants did not detect it and went on to change their query indicating that the feature needs improvement and more importantly, needs to be surfaced in a more detectable way. All participants used the summary charts at least once. Three participants did not realize the charts were changing order but the ones who did (P1, P2, P4) stated they liked the system behavior. P2 stated "*I enjoyed how quickly the system filtered and changed the settings like color and size of the nodes and provided summary statistics like goals, age, market value*". The reordered summary charts also helped trigger new questions in participants' minds. For instance, after applying a club filter, one participant (P1) scanned the summary charts to realize that there was an attribute for position and said "*Oh yeah! There's position too*" and asked the system to filter based on one of the position values. Based on the feedback and our observations, we feel that adding such proactive behavior to complement interactions within the main visualization was a useful design choice. As future work, similar system behavior should be explored to help facilitate an analytical conversation between users and multimodal (and NLIs) visualization systems.

5.3.4 Participants' feedback on multimodal interaction

Participants overall felt that the various features of the system were well integrated (Figure 7-right). They generally found the multimodal interaction to be intuitive and stated they would want to use such a system frequently (Figure 7-right). One participant (P3) wrote "*It was fun to use and a very intuitive way to explore a network.*". Other participants even stated that they felt direct manipulation and speech-based multimodal input should become a part of network visualization tools in general. For example, one participant (P4) who works with network visualizations almost on a daily basis wrote, "*The ability to perform simple actions like "find node" and "find path between two nodes" was really fun to use, and I see this being highly used in general network visualization tools, especially for novice users*". He further stated that he felt that the speech input worked particularly well for navigation and topology-based tasks. He suggested that the natural language modality for such tasks would be a great addition to keyboard and mouse based network visualization systems and it can speed up performance. He did state, however, that he still wanted to use direct manipulation for tasks like selecting specific values for graphical encodings or tuning parameters for analytical operations, emphasizing that he wanted both modalities.

6 Future Work

Facilitating network presentation tasks. One particularly interesting category of tasks that emerged from our experiment were network presentation tasks. Three of the participants mentioned that they found the ability of being able to spatially drag and pin nodes useful. As stated earlier, some of these participants even wanted to drag and pin entire groups. During the session, one participant (P5) even asked "*Can I ask it to modify the layout to something other than force-directed?*". Such observations indicate potential value in exploring ways in which we can help people accomplish network presentation related tasks such as layout modification, bundling or untangling edges, and minimizing edge crossings. Particularly with natural language, one can even envision layouts being set automatically by the system in response to user queries. Another possible extension of this idea is the system suggesting alternative representations (e.g., a matrix instead of a node-link diagram) that can be most effective in answering a given question.

Exploring additional classes of networks. As part of our current work, we have primarily used Orko to explore multivariate, undirected networks. While interactions supported by Orko are rather generic and can be used in other network types such as directed and multipartite networks, we need to explore other types of networks further to identify and support network-specific tasks that people may want to perform. For instance, for a temporal network, in addition to considering what questions people want to answer and how they ask those questions, considering how multimodal interaction can be leveraged for tasks such as navigating through temporal variations in the network structure is another direction for future research.

7 Conclusion

We introduced Orko, a network visualization tool facilitating multimodal interaction via natural language and direct manipulation (e.g., touch) input. To explain the difficulty of providing such an interface, we highlighted challenges associated with interpreting natural language in such a multimodal setting. We presented Orko's architecture describing how it processes multiple input modalities and facilitates multimodal interaction. Through an example scenario of use and descriptions of Orko's capabilities, we sought to illustrate its innovative approach and potential for a new style of network exploration and data analysis. We reported results from an evaluation study of Orko and used our observations to discuss opportunities and challenges for future work in multimodal network visualization interfaces.

Acknowledgments

We thank the anonymous reviewers for the detailed and helpful feedback on the article. This work was supported in part by the National Science Foundation.

References

[1] S. Agrawal, S. Chaudhuri, and G. Das. DBXplorer: A system for keyword-based search over relational databases. In *Proceedings of the 18th International Conference on Data Engineering*, pages 5–16. IEEE, 2002.

[2] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*, pages 111–117. IEEE, Oct. 2005.

[3] Amazon Alexa. `https://en.wikipedia.org/wiki/Amazon_Alexa`.

[4] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. Natural language interfaces to databases–an introduction. *Natural language engineering*, 1(01):29–81, 1995.

[5] D. Archambault, T. Munzner, and D. Auber. TopoLayout: Multilevel Graph Layout by Topological Features. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):305–317, 2007.

[6] J. Aurisano, A. Kumar, A. Gonzales, K. Reda, J. Leigh, B. Di Eugenio, and A. Johnson. "Show me data": Observational study of a conversational interface in visual data exploration (poster paper). In *IEEE VIS '15*, 2015.

[7] M. Bastian, S. Heymann, M. Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. In *Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM*, pages 361–362, May 2009.

[8] D. Baur, B. Lee, and S. Carpendale. TouchWave: kinetic multi-touch manipulation for hierarchical stacked graphs. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces (ITS)*, pages 255–264. ACM, Nov. 2012.

[9] R. A. Bolt. 'put-that-there': Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '80, pages 262–270, 1980.

[10] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[11] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, 2013.

[12] J. Brooke et al. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[13] N. Bush, R. Wallace, T. Ringate, A. Taylor, and J. Baer. Artificial Intelligence Markup Language (AIML) Version 1.0. 1. *ALICE AI Foundation Working Draft*, 2001.

[14] P. R. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow. Quickset: Multimodal interaction for simulation set-up and control. In *Proceedings of the fifth conference on Applied natural language processing*, pages 20–24. Association for Computational Linguistics, 1997.

[15] M. Cordeil, T. Dwyer, K. Klein, B. Laha, K. Marriot, and B. H. Thomas. Immersive Collaborative Analysis of Network Connectivity: CAVE-style or Head-Mounted Display? *IEEE Transactions on Visualization and Computer Graphics*, 23(1):441–450, 2017.

[16] K. Cox, R. E. Grinter, S. L. Hibino, L. J. Jagadeesan, and D. Mantilla. A multi-modal natural language interface to an information visualization environment. *International Journal of Speech Technology*, 4(3-4):297–314, 2001.

[17] A. Cremers and S. Ginsburg. Context-free grammar forms. *Journal of Computer and System Sciences*, 11(1):86–117, 1975.

[18] S. M. Drucker, D. Fisher, R. Sadana, J. Herron, et al. TouchViz: a case study comparing two interfaces for data analytics on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2301–2310. ACM, May 2013.

[19] T. Dwyer. Network visualization as a higher-order visual analysis tool. *IEEE Computer Graphics and Applications*, 36(6):78–85, 2016.

[20] M. Frisch, J. Heydekorn, and R. Dachselt. Investigating multi-touch and pen gestures for diagram editing on interactive surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 149–156. ACM, Nov. 2009.

[21] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. DataTone: Managing Ambiguity in Natural Language Interfaces for Data Visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 489–500. ACM, Oct. 2015.

[22] Google Assistant. `https://en.wikipedia.org/wiki/Google_Assistant`.

[23] A. P. Gourdol, L. Nigay, D. Salber, J. Coutaz, et al. Two case studies of software architecture for multimodal interactive systems: Voicepaint and a voice-enabled graphical notebook. *Engineering for Human-Computer Interaction*, 92:271–84, 1992.

[24] L. Grammel, M. Tory, and M.-A. Storey. How information visualization novices construct visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):943–952, 2010.

[25] B. J. Grosz. Focusing and description in natural language dialogues. Technical report, DTIC Document, 1979.

[26] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204, 1986.

[27] B. J. Grosz, S. Weinstein, and A. K. Joshi. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225, 1995.

[28] A. G. Hauptmann. Speech and gestures for graphic image manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '89)*, pages 241–245. ACM, Mar. 1989.

[29] K. Hinckley and D. Wigdor. Input technologies and techniques. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, pages 151–168. ACM, 2002.

[30] V. Hristidis and Y. Papakonstantinou. Discover: Keyword search in relational databases. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 670–681. VLDB Endowment, 2002.

[31] A. Kumar, J. Aurisano, B. Di Eugenio, A. Johnson, A. Gonzalez, and J. Leigh. Towards a dialogue system that supports rich visualizations of data. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 304, 2016.

[32] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2689–2698, 2012.

[33] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task taxonomy for graph visualization. In *Proceedings of the '06 BELIV Workshop*, pages 1–5. ACM, May 2006.

[34] J. R. Lewis and J. Sauro. The factor structure of the system usability scale. In *International Conference on Human Centered Design*, pages 94–103. Springer, 2009.

[35] F. Li and H. Jagadish. Constructing an interactive natural language interface for relational databases. *Proceedings of the VLDB Endowment*, 8(1):73–84, 2014.

[36] E. Loper and S. Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics, 2002.

[37] C. T. Lopes, M. Franz, F. Kazi, S. L. Donaldson, Q. Morris, and G. D. Bader. Cytoscape web: an interactive web-based network browser. *Bioinformatics*, 26(18):2347–2348, 2010.

[38] V. Lopez, M. Fernández, E. Motta, and N. Stieler. PowerAqua: Supporting users in querying and exploring the semantic web. *Semantic Web*, 3(3):249–265, 2012.

[39] T. L. Magnanti and R. T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

[40] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[41] O. Mason and M. Verwoerd. Graph theory and networks in biology. *IET Systems Biology*, 1(2):89–119, 2007.

[42] G. A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[43] J. Moody. The structure of a social science collaboration network: Disciplinary cohesion from 1963 to 1999. *American Sociological Review*, 69(2):213–238, 2004.

[44] T. Nagel, L. Pschetz, M. Stefaner, M. Halkia, and B. Müller. mæve–an interactive tabletop installation for exploring background information in exhibitions. In *International Conference on Human-Computer Interaction*, pages 483–491. Springer, 2009.

[45] C. North, T. Dwyer, B. Lee, D. Fisher, P. Isenberg, G. Robertson, and K. Inkpen. Understanding multi-touch manipulation for surface computing. In *IFIP Conference on Human-Computer Interaction*, pages 236–249. Springer, 2009.

[46] S. Oviatt. Multimodal interactive maps: Designing for human performance. *Human-Computer Interaction*, 12(1):93–129, 1997.

[47] S. Oviatt. Ten myths of multimodal interaction. *Communications of the*

ACM, 42(11):74–81, 1999.

[48] S. Oviatt. Multimodal interfaces. *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications*, 14:286–304, 2003.

[49] S. Oviatt and P. Cohen. Perceptual user interfaces: multimodal interfaces that process what comes naturally. *Communications of the ACM*, 43(3):45–53, 2000.

[50] S. Oviatt, R. Coulston, S. Tomko, B. Xiao, R. Lunsford, M. Wesson, and L. Carmichael. Toward a theory of organized multimodal integration patterns during human-computer interaction. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 44–51. ACM, 2003.

[51] S. Oviatt, A. DeAngeli, and K. Kuhn. Integration and synchronization of input modes during multimodal human-computer interaction. In *Referring Phenomena in a Multimedia Context and their Computational Treatment*, pages 1–13. Association for Computational Linguistics, 1997.

[52] R. Pausch and J. H. Leatherby. An empirical study: Adding voice input to a graphical editor. In *Journal of the American Voice Input/Output Society*. Citeseer, 1991.

[53] A. J. Pretorius, H. C. Purchase, and J. T. Stasko. Tasks for multivariate network analysis. In *Multivariate Network Visualization*, pages 77–95. Springer, 2014.

[54] J. M. Rzeszotarski and A. Kittur. Kinetica: naturalistic multi-touch data visualization. In *Proceedings of the 32nd annual ACM Conference on Human Factors in Computing Systems*, pages 897–906. ACM, 2014.

[55] R. Sadana and J. Stasko. Designing and implementing an interactive scatterplot visualization for a tablet computer. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 265–272. ACM, 2014.

[56] R. Sadana and J. Stasko. Designing multiple coordinated visualizations for tablets. *Computer Graphics Forum*, 35(3):261–270, 2016.

[57] B. Saket, P. Simonetto, and S. Kobourov. Group-level graph visualization taxonomy. *arXiv preprint arXiv:1403.7421*, 2014.

[58] J. Sauro and J. R. Lewis. Correlations among prototypical usability metrics: evidence for the construct of usability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1609–1618. ACM, 2009.

[59] S. Schmidt, M. A. Nacenta, R. Dachselt, and S. Carpendale. A set of multi-touch graph interaction techniques. In *ACM International Conference on Interactive Tabletops and Surfaces*, pages 113–116. ACM, 2010.

[60] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 365–377. ACM, 2016.

[61] K. A. Siek, Y. Rogers, and K. H. Connelly. Fat finger worries: how older and younger users physically interact with pdas. In *IFIP Conference on Human-Computer Interaction*, pages 267–280. Springer, 2005.

[62] A. Simitsis, G. Koutrika, and Y. Ioannidis. Précis: from unstructured keywords as queries to structured databases as answers. *The International Journal on Very Large Data Bases*, 17(1):117–149, 2008.

[63] A. Srinivasan and J. Stasko. Natural language interfaces for data analysis with visualization: Considering what has and could be asked. In *Proceedings of EuroVis '17*, pages 55–59, June 2017.

[64] Y. Sun, J. Leigh, A. Johnson, and S. Lee. Articulate: A semi-automated model for translating natural language queries into meaningful visualizations. In *International Symposium on Smart Graphics*, pages 184–195. Springer, 2010.

[65] A. van Dam. Post-WIMP user interfaces. *Communications of the ACM*, 40(2):63–67, 1997.

[66] F. van Ham and A. Perer. Search, show context, expand on demand: Supporting large graph exploration with degree-of-interest. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):953–960, 2009.

[67] M. T. Vo and C. Wood. Building an application framework for speech and pen input integration in multimodal learning interfaces. In *Proceedings of 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-96)*, volume 6, pages 3545–3548. IEEE, 1996.

[68] G. Woolfe. Making color adjustment accessible to non-experts through the use of language. In *Color and Imaging Conference*, pages 3–7, 2007.

[69] L. Wu, S. L. Oviatt, and P. R. Cohen. Multimodal integration-a statistical view. *IEEE Transactions on Multimedia*, 1(4):334–341, 1999.

[70] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, 1994.