DESIGN AND IMPLEMENTATION OF AN INTERACTIVE VISUAL QUERYING SYSTEM FOR MARITIME DATA

by

Seyed Adel Ghaeinian

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
June 2024

Table of Contents

List of Tables . . . . **vi**

List of Figures . . . . **vii**

Abstract . . . . **xi**

List of Abbreviations and Symbols Used . . . . **xii**

Acknowledgements . . . . **xiv**

Chapter 1 Introduction **1**

1.1 Background and motivation 1

1.2 Research problem 3

1.3 Research Aim and Objectives 4

1.4 Research Questions (RQs) 5

1.5 Research Contribution 5

1.6 Research Challenges and Limitations 6

1.7 Thesis Structure 7

Chapter 2 Literature review **8**

2.1 AIS Data and Its Applications in Maritime Analysis 8
2.1.1 Anomalous Data Detection in AIS 9
2.1.2 AIS Data Visualization 9

2.2 Visual Query Languages (VQLs) 10
2.2.1 Visual Query Languages for GIS 15

2.3 Direct Manipulation Techniques 17

2.4 Collaboration in Maritime Operations 19

2.5 Large Tabletop Interfaces 19

2.6 Augmented Reality in GIS 22

2.7 Summary and Gap Identification 23

Chapter 3 Design and Evaluation of the Visual Query Builder (VQB) System . . . 25

3.1 Overall System Architecture . . . 25
3.1.1 Components Breakdown . . . 27
3.1.2 Scalability . . . 31

3.2 Gathering and Refining Data . . . 32
3.2.1 Processing of AIS Tracks . . . 32
3.2.2 Protectedseas Areas: GeoJson Data [2] . . . 33

3.3 API Server and Database Integration . . . 34

3.4 React Visualization App Development . . . 35
3.4.1 Direct Manipulation on the Map . . . 35
3.4.2 User Interface and Interaction Design . . . 37
3.4.3 Temporal and Spatial Filtering . . . 39

3.5 Visual Query Builder . . . 40
3.5.1 Query Formulation . . . 46

3.6 Query Parsing and Execution Process . . . 49

3.7 Cognitive Walkthrough . . . 54
3.7.1 Methodology . . . 54
3.7.2 First Iteration . . . 55
3.7.3 Second Iteration . . . 57
3.7.4 Summary of Findings and Improvements . . . 57

3.8 Summary of the Chapter . . . 58

Chapter 4 Augmented Reality Exploration . . . 60

4.1 Introduction . . . 60

4.2 AR Integration with Tabletop . . . 60

4.3 Synchronization of Augmented Visuals . . . 61

4.4 Relation Indicator Arches . . . 62

4.5 Time Range Sliders in AR . . . 63

4.6 Challenges, Exploration Outcomes and Insights . . . 64

4.7 Future Work . . . 67

Chapter 5 Methodology for Comparative Study . . . 68

5.1 Research Design . . . 68

5.2 Interfaces for Comparison (VQB, SPARQL) . . . 69
5.2.1 VQB . . . 69
5.2.2 SPARQL Editor . . . 69

5.3 Selection of Query Tasks . . . 73

5.4 Participants and Sampling . . . 74
5.4.1 Demographics . . . 74

5.5 Study Procedure . . . 75

5.6 Data Collection . . . 77
5.6.1 Demographic questionnaire . . . 77
5.6.2 Screen recording of the querying tasks . . . 78
5.6.3 Time to finish document reading . . . 78
5.6.4 Time to finish tasks . . . 79
5.6.5 Time spent on documents during the tasks . . . 79
5.6.6 Number of referring back to the documentation during the tasks 79
5.6.7 Given hints during the tasks . . . 79
5.6.8 Submitted queries and results . . . 80
5.6.9 NASA-TLX . . . 80
5.6.10 System Usability Scale (SUS) . . . 80
5.6.11 Custom questionnaire . . . 81
5.6.12 Observer notes . . . 81

Chapter 6 Comparative Evaluation and Analysis . . . 84

6.1 Time Efficiency and Accuracy . . . 84

6.2 User Experience and Preferences . . . 89

Chapter 7 Discussion and Implications . . . 101

7.1 Interpretation of the Results . . . 101
7.1.1 Time Efficiency and Accuracy . . . 101
7.1.2 User Experience and Preferences . . . 102

7.2 Implications of the Findings . . . 104

7.3 Limitations and Future Research Directions . . . 106

Chapter 8 Conclusion, Contributions, and Future Work **108**

8.1 Summary of Contributions 108

8.2 Recommendations for Future Research 110

8.3 Final Remarks 111

Bibliography **113**

Appendix A User study materials and instructions **119**

A.1 Introduction to SPARQL Querying In Maritime Context 119

A.2 Introduction to Visual Query Builder (VQB) Interface 129

A.3 Research Ethics Board (REB) Application 140

Appendix B Statistical Analysis of NASA-TLX Questions Individually **172**

List of Tables

5.1 Filtering Actions Required for Querying Tasks 74

5.2 Within-subject counterbalancing 74

6.1 Univariate Type III Repeated-Measures ANOVA Assuming Sphericity 87

6.2 Participant Given Hints Types Table: Indicating the Common Problems 90

6.3 Univariate Type III Repeated-Measures ANOVA Assuming Sphericity 93

6.4 Mauchly's Test for Sphericity 93

6.5 Descriptive Statistics for Reference Counts by Task and Interface 98

6.6 Univariate Type III Repeated-Measures ANOVA Assuming Sphericity 100

6.7 Mauchly's Test for Sphericity 100

6.8 Greenhouse-Geisser and Huynh-Feldt Corrections for Departure from Sphericity 100

List of Figures

2.1 QueryVOWL result of a SPARQL query: 102 persons who starred in movies together with at least one of their children. This figure has been reconstructed from [20]. The red stroke represents the model that is being focused on. 13

2.2 Visual query of RDF explorer interface representing the query of: Finding politicians born in Auckland. Image taken from [57] © 2019 Springer Nature Switzerland AG; used with permission. 14

2.3 The GeoGate system showcase, wherein each view illustrates a different perspective of the Space-Time Cube (STC) oriented towards the observer. Progressing in a counter-clockwise direction, these perspectives represent the left, top, front, back, and right sides of the STC [52] © 2019 IEEE. Reprinted with permission. 21

3.1 Physical system architecture diagram 26

3.2 The interface of the visualization application 29

3.3 Placing the tangible cube on the tabletop opens a ring Menu . 30

3.4 Broadcasting map state via socket connection; required for synchronization of the AR clients 31

3.5 AIS Data structure A) Raw data AIS data from CSV B) preprocessed AIS data 33

3.6 ProtectedSeas geoJson areas 34

3.7 Detailed information panel displayed upon selecting a single vessel on the map. 37

3.8 A user-defined polygonal area in edit mode, showcasing adjustable nodes for refining the shape. 38

3.9 Augmented Reality Demonstration: Lifting a vessel to view its detailed information. 39

3.10 An augmented reality double-pinch slider, allowing users to adjust the ”startTime” and ”endTime” for temporal filtering. . . 40

3.11 An example query generated using the visual query builder interface, which queries cargo vessels inside the polygon or fishing vessels that have a speed over ground exceeding 50 miles per hour. 41

3.12 Visual query builder palette, A) Entity nodes B) Conditional nodes C) Selection Nodes (User selections over the map) D) Functional nodes E) Template nodes F) Query history 42

3.13 Visual Query Builder (VQB) Interface for Maritime Querying, the Latest Version 44

3.14 Vessel type prompt allowing users to query an exact type of vessels 44

3.15 Prompt asking users to input the values of a condition node before initialization. 46

3.16 Two examples of filling the condition node initialization prompt: The first one contains numerical conditioning and the unit and value, and the second one is for spatial querying, asking for the spatial area input 47

3.17 A pop-up dialog prompt that asks the user to input the required output details 48

3.18 An example of the output node usage, querying average SOG of the cargo vessels 49

3.19 Formulating a simple query: querying all cargo vessels during ”2023-01-01 13:30:00” to ”2023-01-02 23:30:00” 50

3.20 Executing the query and visualizing the result of it on the map visualization application 51

3.21 Visualizing the formulation of a more complex query 52

3.22 In (A) users are collaboratively interacting with the tabletop and (B), a user wears a HoloLens to view AR content while interacting with the tabletop display. 56

3.23 Direct Manipulation (DM) and Visual query builder (VQB) Tasks. In the VQB task image, the first-person vision of augmented reality interaction over the shared display showing the AR guide to Visual query builder (VQB) shape selection. . . . 59

4.1 A: Pinch gesture initiates scanning of the QR code, B: A vessel and its route displayed in AR after synchronizing with the tabletop environment. 61

4.2 Relation indicator arch: the arch visualization in AR conveys the relation between an element in VQB and its source on the map 63

4.3 Determining the arch's initial and terminal points: X, Y represent the absolute position of the workspace relative to the tabletop's origin; x, y indicate the selection node's relative position within the workspace; lat,lng denote the geospatial coordinates of the center of the selection (highlighted as the yellow circle in this figure). 64

4.4 Snapshots of a vessel at different timestamps: This feature facilitates users in comparing vessel positions across various moments, deepening their analytical perspective. As depicted, three distinct snapshots capture the vessel's positions at separate times. 65

4.5 Visualizing vessels' routes using the STC approach in the Unity simulation: Cubes of identical colors represent the detections of a specific vessel. By linking these detection points, a route is established. 66

5.1 SPARQL Editor Interface 70

5.2 Age Distribution 75

5.3 Gender Distribution 76

5.4 Data Management Tools and Interfaces Experience 77

5.5 GIS Familiarity 78

5.6 NASA-TLX Workload Scale Form 82

5.7 System Usability Scale (SUS) Questionnaire [7] 83

6.1 Average Time-to-finish Per Interface For Each Task 86

6.2 Line chart representing the time efficiency of the participants for each task, per interface 87

6.3 Box-plot visualization of the performance on each task 88

6.4 Violin-plot performance visualization 89

6.5 Given Hints Rate Sorted By Frequency 91

6.6 Total given hints per interface 92

6.7 Mean NASA-TLX Scores Per Task (Tasks 1 to 4) 93

6.8 Interaction of Interface and Task on NASA-TLX workload . . 94

6.9 Interaction Between Tasks, Interface, and Group on the Workload Score 95

6.10 Grade rankings of SUS scores [8] 96

6.11 Scatter-plot visualizing of the SUS scores 97

6.12 Box-plot visualization of the SUS scores 98

6.13 Box Plot Visualization of Reference Counts 99

Abstract

Automatic Identification System (AIS) data is a crucial foundation of maritime operations such as navigation, traffic management, and safety monitoring. This requires robust management, visualization, and interactive visual analytics systems for the operators. Maritime data analytics is an open-ended problem. Moreover, the increasing volume and complexity of data make it essential to incorporate the latest advancements in user interaction and analytics. These advancements are necessary to handle the challenges effectively, ensuring operators can make timely and informed decisions. This research explores the design and implementation of a novel system architecture for maritime data querying and exploration, enabling enhanced user interactions through direct manipulation techniques. This system architecture provides a collaborative environment that incorporates Mixed Reality (MR), a touchable tabletop interface, as well as iterative design and evaluation of a graph-based Visual Query Builder (VQB). The aim of developing the VQB interface is to allow non-experts to explore and query maritime data without the need to have technical skills while enhancing the decision-making process and spatial awareness of maritime operations. This research further evaluates the VQB interface in terms of efficiency, accuracy, and user preferences by conducting a user study. For this study, we developed a baseline web-based textual interface for SPARQL queries, enhanced with auto-correction features and additional spatial querying capabilities, to effectively measure and fairly compare the performance of the VQB interface. 20 students from the faculty of computer science participated in this study without prior knowledge of RDF data querying and SPARQL language. Overall, the results of this study were promising as it showed a higher efficiency and accuracy rate as well as less perceived workload among participants in the VQB interface compared to the baseline. These findings highlight the role of visual query interfaces in improving the user experience as well as elevating efficiency, especially for non-experts, allowing them to explore and query maritime data without the need to learn technical skills.

List of Abbreviations and Symbols Used

AIS	Automatic Identification System
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
API	Application Programming Interface
AR	Augmented Reality
COG	Course Over Ground
CSV	Comma-Seperated Value
CW	Cognitive Walkthrough
DBMS	Data Base Management Systems
DM	Direct Manipulation
DRDC	Defence Research and Development Canada
GIS	Geographic Information System
HCI	Human-Computer Interaction
HMD	Head-Mounted Displays
IMO	International Maritime Organization
LAT	Latitude
LON	Longitude
M	Mean
MMSI	Maritime Mobile Service Identity
MR	Mixed Reality
MRTK	Mixed Reality Took Kit

NASA-TLX	NASA Task Load Index
NOAA	National Oceanic and Atmospheric Administration
PUN	Photon Unity Networking
RDF	Resource Description Framework
RQ	Research Question
S-TVQE	Spatio-Temporal Visual Query Environment
SOG	Speed Over Ground
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
STC	Space-Time Cube
STD	Standard Deviation
SUS	System Usability Scale
VQB	Visual Query Builder
VQL	Visual Query Language
WLAN	Wireless Local Area Network
WQH	Wikidata Query Helper

Acknowledgements

I would like to thank my supervisor, Dr. Joseph Malloch, for his guidance and support throughout this research. I also extend my gratitude to Professor Derek Reilly, head of the HCI and Graphic Visualization cluster at Dalhousie, for providing a great experience and a supportive lab environment.

Chapter 1

Introduction

Data has become a foundation of decision-making in various domains and fields in the digital era and the maritime domain is not an exception. Maritime data management, querying, analysis, interaction, and visualizations, are crucial aspects of this domain since these are playing a significant role in maritime operations. This thesis explores the complexities of maritime data analysis and further highlights the possible room for improvement in the existing tools and interfaces for maritime data exploration and analysis. It proposes a novel graph-based Visual Query Builder (VQB) interface with the aim of enhancing the user experience, efficiency, collaboration, and decision-making in maritime operations while opening a path for non-experts to proceed with their maritime operations without the need for familiarity with query languages. Additionally, this research aims to iteratively develop and evaluate this prototype that integrates with novel direct manipulation techniques. This chapter of the thesis outlines the motivation behind the study, defines the research problem, articulates the aims and objectives, and previews the structure of the thesis, It prepares the ground for a detailed discussion on improving maritime data interaction.

1.1 Background and motivation

Data plays an important role in the tech era we live in. Every day, the amount of data that we produce gets higher in every aspect, such as industry, academia, healthcare, and other fields. Just like other fields in our tech era, maritime data benefits from technological advancement, with new equipment such as radars, sensors, and cameras emerging to generate crucial maritime data. Evidently, the analysis of maritime data is significantly important in many aspects, such as vessels' traffic control and collision avoidance, traffic observation for security insurance [27], controlling illegal fishing, monitoring prohibited traffic areas, anomaly detection [32], human rescue operations, etc. The importance of maritime surveillance and providing safe and seamless maritime traffic gave rise to the development of Automatic Identification System (AIS) [13]. AIS data provides real-time information for the vessels that can significantly help maritime human operators, guards, experts, and in general maritime decision-makers to operate. However, working with raw AIS data by human operators in this field could be challenging and even in some situations impossible without having proper data refinements, visual analytics, and querying systems. Proper data visualization techniques can help experts to have a better understanding of the data by mapping the data to visual elements [28].

Given the critical role of effective visualization systems, numerous visualization analytic systems have been proposed and developed to support maritime experts in their operations and aid their decision-making process [16, 29, 24, 42, 34, 60]. The visualization systems proposed in the maritime domain are typically demand-driven, and designed to address specific challenges. For instance, a model developed for anomaly detection in vessel trajectories may not perform as effectively when applied to traffic assessment and observation. While visualization systems bring insights into maritime data, their true value lies in enabling interaction with these systems allowing operators to filter the data both spatially and temporally, delve deeper, and even take actions more quickly by interacting with visual elements without the necessity of having knowledge of any query languages such as SQL, SPARQL, etc. With advancements in human-computer interaction (HCI) , the development and evaluation of visual querying systems have drawn researcher's attention due to their capabilities and benefits that they offer. Studies have shown promising results for the effectiveness of visual querying systems, especially allowing non-experts to formulate their queries [57, 12]. Visual querying systems can lower the barrier to entry for experts who may not have extensive technical backgrounds such as query languages.

Akin to improvements in the HCI field, hardware tool designs have advanced significantly in recent years, opening new paths toward interaction design. Developments such as large touchable and interactive interfaces, augmented reality, virtual reality, wearable devices, and tangible interactive made novel user interactions possible, and these novel interactions could potentially benefit the user experience, and productivity and even facilitate easier collaboration among users [47]. These advancements could enhance user experience, productivity, and accuracy [52], which is the focus of our evaluation in this thesis. As we mentioned earlier, having proper visualizations and user interactions in the maritime context is significantly important due to its nature since there are use cases such as surveillance operations and observation of the safety of vessel traffic, that are crucial to be performed accurately and quickly. In the maritime domain, the usage of mixed reality has grown recently. Maritime simulations in mixed reality environments offer valuable benefits such as reducing the cost of training, increasing accessibility, and also the ability to conduct training sessions remotely [37]. The use of AR in the maritime can also enhance the decision-making process and increase spatial awareness and can aid maritime activities such as navigation, maintenance, construction as well as training [55].

1.2 Research problem

In the maritime data analysis context, the data querying and exploration have traditionally been navigated through textual query languages such as SQL. Stojanovic et al. highlighted that traditional SQL-like languages fail to efficiently address the spatial properties of real-world entities, often leading to a complex and user-unfriendly interface for geographic information system (GIS) applications [53]. Due to the importance of spatial querying and addressing the need for spatial-driven applications, some database management systems (DBMS) have developed support for spatial querying by providing spatial querying modules such as the PostGIS module in PostgreSQL, and the support of geographic data by main DBMSs such as Microsoft SQL, MySQL, SQLite [5]. These querying modules, while robust, need querying language knowledge and technical skills to operate with, presenting a barrier to non-technical users. For instance, while PostGIS offers a valuable querying framework for spatial data, it does not provide a direct user interface for manipulating spatial data. It allows storing and querying data about location and mapping but operates primarily through SQL commands.

To address this challenge GIS visual analytics systems, like ArcGIS [1] and QGIS [2] have been developed to streamline maritime querying and visualization processes by providing interfaces that simplify the complexities associated with SQL commands.

[1] ArcGIS website URL: https://www.arcgis.com/index.html

[2] QGIS website URL: https://www.qgis.org/en/site/

Yet these systems expect users to possess a grasp of GIS principles, which may pose a challenge, for individuals lacking a background. Additionally, although these tools offer some data interaction capabilities they do not fully harness the benefits of direct manipulation interfaces, which have the potential to improve comprehension and operational effectiveness by enabling users to interact directly with the data.

1.3 Research Aim and Objectives

This research focuses on maritime querying and analytics systems to be able to investigate the effectiveness of the query systems deeply and thoroughly, by narrowing the context to the maritime domain. Given the mentioned research problems, this research aims to develop a prototype of a visual querying method tailored for maritime spatiotemporal data querying tasks in a human-centric fashion with multiple enhancement iterations. By conducting a user study, this research aims to comparatively evaluate the developed prototype with the traditional querying method interfaces. Moreover, this research will innovate system design in the maritime context to allow multiple users to connect to a session and perform their querying and exploration tasks. This research aims to integrate high-end querying methods and technologies in the system while offering novel human interaction models and ideas.

- **Objective 1**: To design and develop a robust system architecture that enables multiple users to connect and collaboratively perform maritime-related data explorations and manipulations.

- **Objective 2**: To design and develop an initial prototype of a visual query builder interface tailored for the maritime domain, and include the basic spatiotemporal querying functionalities.

- **Objective 3**: To iteratively enhance the visual query builder and gradually integrate advanced querying methods suitable for maritime querying.

- **Objective 4**: To compare and contrast the implemented visual query builder prototype with traditional querying interfaces in terms of effectiveness, learning curve, accuracy, and user preference.

This method guarantees an investigation into search techniques in maritime systems with the goal of enhancing the user-friendliness and effectiveness of tools for analyzing maritime data.

1.4 Research Questions (RQs)

This research tries to explore the design and implementation of visual query builders and evaluate and compare them with traditional querying interfaces in the maritime context. The focus of this evaluation is on both quantitative factors such as accuracy and time-efficiency, as well as qualitative factors such as the cognitive load on the users and the user preferences.

- **RQ 1**: How visual query builder interface can affect performance and accuracy in maritime querying tasks compared to the textual query interface? This RQ aims to assess the operational performance of the visual query builder interface in comparison with the textual query interface. To answer this research question, metrics such as task completion times and error rate should be analyzed and evaluated.

- **RQ 2**: What are the cognitive effects of the visual query builder on its users compared to the textual query interface? This RQ aims to comparatively evaluate the mental workload, physical workload, user satisfaction, and overall user experience of using the visual query builder interface compared to the textual interface.

- **RQ 3**: How can we enhance the design of a visual querying interface to improve user interaction, engagement, and overall user experience in maritime data manipulation and exploration tasks? This RQ leads to the exploration of the design elements that are most effective in enhancing cognitive and physical interaction, user satisfaction, and overall experience with the system.

1.5 Research Contribution

This study contributes to the field of HCI by providing a detailed comparative analysis of user interactions with visual querying systems in the context of maritime data exploration. Specifically, it examines the effectiveness of different design elements in enhancing user engagement and interaction efficiency. The methodology chapter can guide prospective researchers in this field through the study design as well as the human-centric methods of the implementation and iterative enhancement of the visual query builder systems. Additionally, the result of the study fills the gap in the literature regarding the lack of user studies investigating the effectiveness of visual query builders in the maritime domain. This research will benefit maritime industries by providing a detailed analysis and evaluation of the Visual Query Builder (VQB). This evaluation includes its novel user interaction and direct manipulation techniques, as well as its limitations and challenges.

Lastly, this research aims to implement the visual query builder in a modular way so it facilitates prospective researchers to scale up the project and expand the interface to conduct their future user studies and research in this field.

1.6 Research Challenges and Limitations

Before starting the actual research it is reasonable to investigate and acknowledge the limitations and challenges that we might encounter during the research and the user study. Here are some challenges and limitations of the research that we have anticipated:

- **Sampling of the User Study** One of the challenges could be hiring maritime experts and operators in this domain to conduct the user study, potentially limiting the diversity of the user study.

- **Time Limitations** As this research is a result of a master's thesis, one of the limitations of the study is the time limitation. This limitation could impact the scope of the study.

- **Generalizability** The result of this research might not applied to all domains since we are focusing on the effectiveness of the visual query builders and query system designs for the maritime domain by implementing some functions such as spatial and temporal filtering that are tailored for the maritime querying and exploration tasks.

- **Data Gathering** Access to certain types of maritime data can be challenging due to security concerns, high costs, or confidentiality agreements. Consequently, it is essential to carefully select and engage with reputable maritime data providers who can provide the necessary data while compiling with privacy standards.

- **Data Processing** Various data sources and types would be used for this research, such as the AIS data, data of areas of interest, and the user interaction data with the system. Maritime datasets usually consume large amounts of disk space and the processing of them could be challenging and requires high processing power if we want to ensure seamless user interactions with the system. Therefore, simplifications and other filtering methods are required to overcome this challenge and offer consistent, quick, and interactive data visualizations.

1.7 Thesis Structure

Starting by analyzing and reviewing the related works and the current common knowledge in this field, the gaps in the research will be presented, and our approach for filling the gap in the literature will be mentioned in the literature review chapter. Furthermore, in the design and evaluation of the VQB, this thesis represents the detailed actions that have been taken to accomplish the implementation process of the maritime querying and exploration system and the rationale behind the actions, as well as the improvement of the system over time. The augmented reality exploration chapter describes the integration of the implemented AR application component with the rest of the system, especially the tabletop and also this chapter describes what we took away from this exploration. In the following chapters, the design, result, and analysis of the user study for evaluating the effectiveness of the implemented VQB component in querying maritime data will be elaborated with the following discussion chapter as well as the summary of the research and the related limitations.

Chapter 2

Literature review

The purpose of this chapter is to review and synthesize the existing literature on visual querying systems, technologies, hardware, and system designs in the maritime domain using a thematic approach. Additionally, this chapter highlights the research works that have influenced our work, the limitations in current literature, and the approach that our research adopts to expand upon these findings.

2.1 AIS Data and Its Applications in Maritime Analysis

Automatic Identification System (AIS) has been invented by international maritime organizations to track vessels and maritime targets in order to provide safety and collision avoidance [41] and improve the way of navigation. Also, it helps with maritime target tracking and identification [22]. Each AIS transmitter on any vessel usually sends static data information about that vessel regarding its size, weight, Maritime Mobile Service Identity, abbreviated as MMSI, and dynamic information about the speed and position of the vessel. The vessel's speed is displayed as its course over the ground, its position is displayed as latitude and longitude, and its heading indicates its orientation.

Since 2002, AIS has been a mandatory feature on all new ships over 4,300 gross tonnage and on all passenger vessels, regardless of size [41]. This mandate caused a large number of vessels to transmit their data, making the management, validation, and visualization of this information a complex challenge. Regarding the challenge of validation, some individuals may manipulate the AIS data they intend to transmit. They might intentionally turn their transmitters on or off or even attempt to interrupt the receptors to conceal illegal activities [39].

2.1.1 Anomalous Data Detection in AIS

Lane et al. classified anomalies in the AIS data into kinematic behavior anomalies, AIS transmission anomalies, and other types of anomalies [32].

1. Kinematic Behavior Anomalies: These are the first category of anomalies in AIS data, associated with the movement and navigation of vessels.
2. AIS Transmission Anomalies: These anomalies may be caused by unexpected switching on or off of the AIS transmitter.
3. Other Types of Anomalies: This category includes other atypical behaviors not captured in the above categories, such as alterations in ship registration details.

One common approach to detecting anomalies in vessel paths and AIS data is using machine learning classification methods. Singh et al. proposed an artificial neural network (ANN) model to detect anomalies in the AIS data and anomalous vessel paths [51]. In the system that the researchers proposed, AIS data is first received, pre-processed, and then subjected to feature extraction given speed, course, position, and vessel timing as inputs. Once this preparation is complete, the data is ready for anomaly testing, and the system will generate alerts if any anomalies are detected. Additionally, their proposed system is able to classify the type of anomaly with %99.9 accuracy, whether it's because of their making U-turns or entering a prohibited zone, etc.

2.1.2 AIS Data Visualization

When confronted with a problem, good visualization can greatly assist in making the right choice and solving the issue [56]. The maritime domain is not an exception. Since maritime data is relatively large data, good visualization can provide decision-makers with good insight into the data, subsequently helping them with their decision-making process. Wang et al. provided a summary of existing visualization techniques for maritime data [59]. According to this research, for maritime visualization, some simplification algorithms (such as the Douglas-Peucker algorithm) have been widely used [35] to simplify vessel trajectories prior to visualization. The second step, Visual Transformation, changes data structures into visual ones, like line plots, bar graphs, pie charts, and dot charts. The third step, Visual Mapping, puts these visual structures into the right visualization scenarios, such as animations, infographics, or dynamic videos.

An innovative approach to AIS data visualization was proposed by Lei, using tools such as Tableau and E-chart for dynamic visualization of AIS data [34]. Their methodology involved compressing the motion tracks of AIS data to extract vessel motion characteristics, thereby offering a more intuitive method for analyzing regional vessel movements. Their work serves as an example of how effective visualization can be when analyzing complex behaviors in areas of interest, such as ports.

He et al. offered a visual analytic tool called ADQvis, which helps experts detect faulty and problematic AIS data for quality validation purposes [25]. ADQvis provides temporal filtering in order to select a span of time for analysis. In their visualization section, they use scatterplot which enables users to identify the data quality problems as quickly as possible. Additionally, the stacked graph allows for a comparative analysis of different base stations at various times. The tool also utilizes Open Street Map for maritime data display on the map, revealing the geographical distribution of trajectory points.

Li et al. utilized the aforementioned Douglas-Peucker simplification algorithm to simplify the trajectories in the AIS dataset. Furthermore, they have used a kernel density estimation (KDE)-based method to identify and visualize the density of the vessels on the map [35]. In order to find a balance between the visualization performance and simplification parameters, the researchers in the study conducted several user experiment studies. The result of their studies shows that their proposed method can effectively visualize AIS data density while reducing the visualization time.

2.2 Visual Query Languages (VQLs)

As we've entered an era where data collection has skyrocketed, the need to efficiently manage and interact with this data has grown significantly. Therefore, the necessity to design query systems that are both effective and user-friendly is apparent, and this is where visual query builders come into play. They've emerged as a tool, providing an easier way to work with and make sense of complex data sets usually without the need of knowing the structural query languages such as SQL, SPARQL, etc.

Among the reviewed papers, a recurring theme about visual query languages is their capacity to assist non-experts in crafting queries and engaging with various datasets. Additionally, certain visual query languages offer the functionality to modify these datasets through interaction with visual components. Not only can a visual query builder help non-experts explore and manipulate the data, but it can also support query language learners in understanding query generation through visuals, as was the aim of the researchers in the study by Miedema and Fletcher [40]. In this study, SQLVis was introduced and evaluated. This tool aims to assist users in crafting SQL queries with greater accuracy and efficiency. Notably, participants who employed SQLVis generated significantly more accurate queries and committed fewer errors compared to other control groups. This observation is particularly relevant for the HCI community as it highlights the potential benefits of visual aids in complex tasks such as SQL query generation. Although this paper represents a comprehensive accuracy and error-rate comparison between SQLVis and traditional SQL querying, it does not address the evaluation of other critical factors like consistency and time efficiency. Consistency is an important factor because it ensures predictable results, which is crucial for reliable data analysis and decision-making.

Vega-Gorgojo et al. [58] proposed and implemented the OptiqueVQS interface which is a querying tool for querying and exploration of the semantic web (RDF data). The design goal of this interface was to achieve a balance between expressiveness and usability and hiding complex RDF and SPARQL syntax, and simplifying the user experience. This interface uses nodes and links for query construction allowing users to form their queries in a graph-based format. By executing the query in this interface, the result is represented in a tabular format as an output. In this research, they have conducted a user study with 15 participants to comparatively evaluate the usability and effectiveness of this interface over the PepeSearch[1] interface which is a form-based visual querying tool for RDF data. The user interface of PepeSearch works in a way that the form starts with a selected class and shows collapsible for related classes, each with specific controls for refining attributes based on the characteristics of data. Similarly, the result of the query is shown in a tabular format via sorting functionality. The result of this study shows that both interfaces,

[1] Official source-code of the PepeSearch on Github: https://github.com/guiveg/pepesearch

Pepesearch and OptiqueVQS, are promising in terms of effectiveness in searching and exploring RDF data. Although OptiqueVQS was preferred among expert searchers due to its power and expressiveness, mainstream participants preferred the form-based interface due to its simplicity and intuitiveness. The result of this study showed that the graph-based visual querying interfaces, while powerful, they pose usability challenges for non-tech users. On the other hand, form-based querying interfaces are more simple and intuitive, while sacrificing expressiveness for constructing more complex queries. Although both tools that have been mentioned (OptiqueVQS and PepeSearch) provide various useful general querying functionalities, especially for non-expert users, they are not effective in querying domains that involve spatiotemporal data querying due to their lack of support for geospatial querying features and spatial representation of the results. In our research, we consider the maritime domain querying as our focus area that involves both temporal and spatial data. Therefore we aim to define and implement interactive spatial representation and enhanced temporal data handling features.

Haag et al. developed a prototype of a visual query builder for querying RDF data called QueryVOWL [20]. QueryVOWL is a graph-based visual query builder utilizing nodes and links to formulate SPARQL queries in a visual representation. In this prototype, various visual elements and connection visuals have been defined to cover most of the SPARQL syntax, such as triple definition, identifiers, union, filter, etc. Fig 2.1 shows an example of a visual representation of a SPARQL query in the QueryVOWL prototype. In this prototype, the scale and of the nodes will change based on the result (count of the result entities). This research uses a qualitative user study with 6 participants to evaluate the effectiveness of the QueryVOWL prototype. The results were promising since the participants could complete most of the tasks. Another report of the study was that the node-link representations were intuitive for the participants and they could comprehend those graphs, while some confusion arose over the properties and classes. All of the participants believed that QueryVOWL can be beneficial for various use cases and scenarios; while highlighting its potential for data exploration.

Haag et al. development of QueryVOWL demonstrates the benefits of graph-based visual querying systems. In our research, we extend this concept by adding

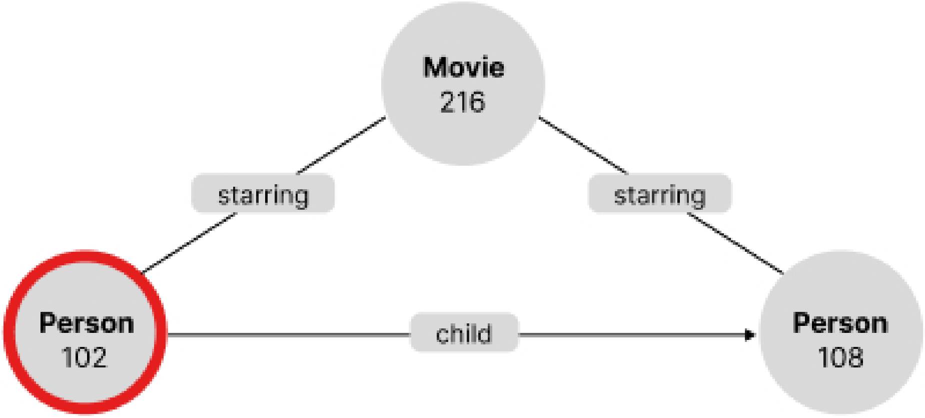


Figure 2.1: QueryVOWL result of a SPARQL query: 102 persons who starred in movies together with at least one of their children. This figure has been reconstructed from [20]. The red stroke represents the model that is being focused on.

spatial and temporal filtering features which are absent in the QueryVOWL prototype but crucial for effectively querying spatiotemporal data, particularly in the maritime domain. Additionally, our research considers increased the population size of our user study, to have more reliable results and robust analysis.

Vargas et al. presented a visual RDF Explorer (RE), a visual query builder aimed to allow non-expert users to navigate and query RDF graphs with minimized cognitive effort [57]. The interface of RE is composed of six main components displayed in three panes, including a search panel, visual query editor, and node detail view. Users can visually generate a SPARQL query by adding variables as nodes and edges as properties. The system allows users to explore and receive feedback on the results generated thus far, guiding the next steps. A user study has been conducted in this research to comparatively evaluate the RE interface over the baseline interface. The baseline interface in this study is called Wikidata Query Helper (WQH) which includes a text box that shows the SPARQL query and any changes made in WQH are reflected in the query and vice versa. WQH offers two functions; the option to set a filter to choose a property (p) and an object (o) and the capability to display data beyond the filter by fixing a property value and introducing a variable linked to its respective object. WQH also offers the autocompletion feature. 28 non-expert students with no familiarity with SPARQL and RDF data participated in this user study. Researchers used the NASA-TLX workload questionnaire and a Likert scale to evaluate the RE interface in terms of workload scores, usability, and user satisfaction. The results of this study show that the participants completed their tasks using the RE interface more accurately (higher success rate) than the baseline. Participants favored this interface more than the baseline interface due to its more intuitive and easier user interface. In terms of a learning curve, RE was initially complex for the participants and the participants learned to use it more effectively over time. Lastly, although RDF Explorer required more physical interactions (e.g., clicks and drags), it was found to demand less mental effort compared to the baseline, which is beneficial for user engagement and satisfaction.

Figure 2.2 depicts an RDF visual query example made by the paper.

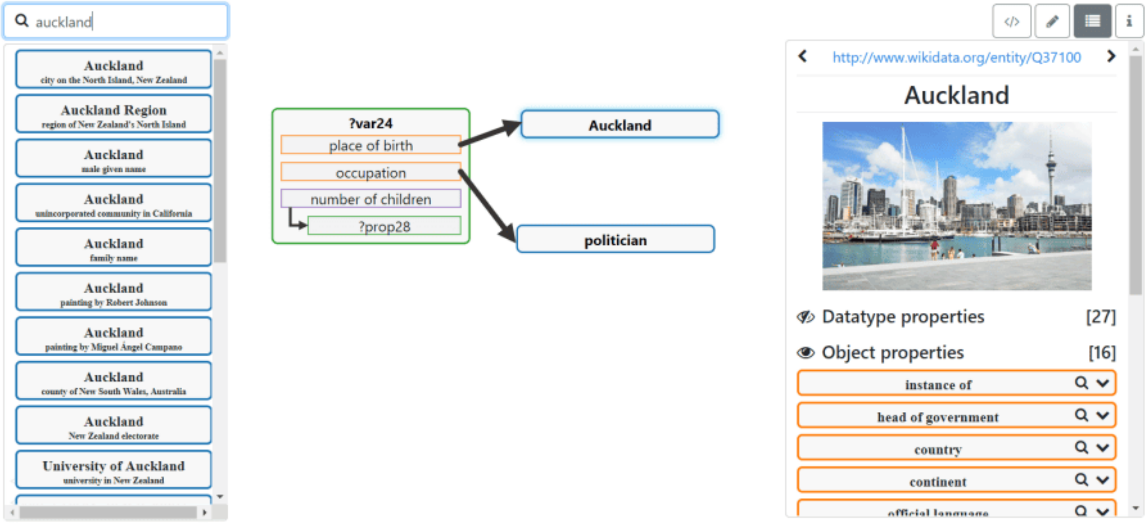


Figure 2.2: Visual query of RDF explorer interface representing the query of: Finding politicians born in Auckland. Image taken from [57] © 2019 Springer Nature Switzerland AG; used with permission.

Ambrus et al. presented the Konduit visual query builder, which incorporates the development of four visual query builder techniques, aiming to facilitate user access to the growing pool of semantic data [4]. Two techniques are schema-based, utilizing ontologies to allow users to construct tree-like SPARQL SELECT and CONSTRUCT queries, focusing on simplifying the process and aiding users to explore local schema structures. The other two techniques are instance-based, employing both schema and instance information from the Nepomuk system to suggest possible options to users for constructing SPARQL queries, effectively enabling exploration of local RDF repositories, and allowing the addition of non-existing properties or conversion of instances from one ontology to another. The paper emphasizes the balance between these techniques' accessibility for novice users and the robustness required by advanced users, suggesting future usability assessments to determine the most suitable SPARQL query builder for a diverse range of users. Furthermore, the focus on usability assessments in [4] has guided our approach to iterative testing and refinement, ensuring that our visual query builder meets the diverse needs of maritime operators.

Svarre et al. conducted a comparative study to evaluate the effectiveness of visual query builders against traditional form-based query builders. The study involved 14 research librarians. PubMed was used to represent the traditional form-based querying method, while their developed interface, named 2Dsearch, represented the visual query builder. The tasks were designed to test the librarians' technical and professional search skills in the healthcare and life sciences domain. The visual method scored higher on user experience metrics. However, some participants still favored the familiarity of form-based interfaces. The study concludes that while visual interfaces are promising, there is still a place for traditional form-based methods [54].

2.2.1 Visual Query Languages for GIS

Maritime systems rely heavily on spatiotemporal data, which is vital for operators and decision-makers to effectively monitor vessel movements, track maritime traffic, and identify potential security threats or hazards. Over the past few years, numerous studies and research papers have introduced visual query languages in the context of the geographic information system (GIS). In the following paragraphs, we will review significant works highlighting different approaches and methodologies in querying geospatial data. While there are many works on visual querying of geospatial and temporal data separately, we found only one work that supports both temporal and spatial queries.

According to the research by Guerroudji-Meddah et al., there are four main ways of querying and interacting with geospatial data. The initial approach is related to principles of algebra and geometry [19]. Another approach is using formal languages such as SQL, SPARQL, etc. A downside to this approach is that it requires knowledge of specific languages, which often come with their own learning curves. The paper also discusses a third approach is using natural languages for querying [31, 62, 21]. This concept was quite complex to execute back in 2009 when the paper was written. However, with the emergence and advancement of large generative language models in recent years, the implementation of this approach has significantly improved and has become considerably easier. The last-mentioned approach uses and interacts with visual elements such as icons, diagrams, and drawings to formulate the query.

Salgado et al. propose a visual query framework for constructing queries for geospatial data, named GeoVisualQL [46]. This framework tackles the challenge of creating complex spatial queries by offering a visual interface that simplifies the user process. GeoVisualQL adheres to the main structures of the Spatial Query Language (SQL) specification by the Open GIS Consortium, enabling users to build SQL-like queries visually. This feature is particularly advantageous for users unfamiliar with textual query languages, as it facilitates interaction with geospatial data through the direct manipulation of visual elements.

The GeoVisual Interface, an integral part of the framework, provides a user-friendly environment for query formulation. It includes areas for data visualization, query composition, and message display, which enhance the overall user experience. By incorporating visual elements such as points, lines, and polygons, users can intuitively construct and execute spatial queries without needing specialized knowledge of underlying database schemas or query languages. The framework also features a Query Manager module that converts visual queries into textual SQL queries, ensuring compatibility with various GIS systems. However, a notable limitation of this interface is that it only supports spatial data, whereas AIS data is spatiotemporal. Thus, while the concepts of querying spatial data with this interface might be applicable, it cannot be utilized for querying AIS data that includes vessel tracks with timestamps.

The study by Calcinelli and Mainguenaud defines a visual query language (VQL) as a query language where the query is expressed by a drawing, called Cigales [10]. Moreover, this study proposes one of the earliest visual query languages in the GIS data context. The interface of the visual query builder features five spatial operators, including inclusion, intersection, bordering, path, and distance. The interface also includes 'with' and 'without' buttons to express or cancel the concept of negation among the operators. In a nutshell, the authors of this paper argue that visual query languages are more user-friendly for non-programmers than alphanumeric query languages. However, the authors note that spatial relationships are currently not well-handled in their visual query language implementation, as they appear only through the straight line operator, and the relative orientation in the query area is difficult to handle correctly.

Cavalcanti et al. present the Spatio-Temporal Visual Query Environment (S-TVQE) interface, a system for visually querying spatiotemporal data. This web-based, user-friendly interface is designed to enable users to query spatiotemporal databases without needing to understand database schema or query language syntax. The S-TVQE interface consists of three main parts: a map and attribute list on the left for selecting spatial and temporal elements, a middle section for specifying and displaying query constraints, and a right side with icons representing relational, spatial, and temporal operators. This setup allows users to visually construct and execute complex spatio-temporal queries, with results displayed in various formats such as maps, graphs, and tables [11]. While this interface supports querying both spatial data and temporal data, it is limited in its ability to handle complex queries and lacks advanced interactive features for detailed analysis. Additionally, this interface does not support compound queries. Our VQB overcomes these limitations by providing a more intuitive, node-based interface that simplifies the construction of complex queries and offers enhanced interactive capabilities for detailed spatiotemporal data analysis.

2.3 Direct Manipulation Techniques

Direct Manipulation (DM) techniques have been widely used in user interfaces for a long time. The paper by Fishkin et al. defines direct manipulation as interaction with digital objects in a way that makes users feel as though they are directly manipulating the objects. This includes actions such as dragging, dropping, resizing, or even certain physical actions, for example, turning pages in an e-book [17]. One characteristic of

DM actions is that they are similar to real-world actions, intuitive, and aligned with users' mental models in such a way that users can predict the result of their actions.

Authors in the study by Endert et al. claimed that in the past, direct manipulation actions have been done using control panels [14]. However, the emerging trend is using direct manipulation approaches to manipulate model outputs directly, where the cognitive process of the users is involved. In this paper, the authors delved into the principles of direct manipulation and discussed the concept of spatializations, which visually represent data based on similarity. The paper highlighted emerging trends in visual analytics systems that allow users to directly manipulate data within spatializations, thereby guiding and refining the data layout. After thoroughly analyzing different levels of interactivity and their inherent challenges, the authors highlighted the importance of balancing human thought processes with computational methods, promoting a more user-friendly and cooperative approach to visual analytics.

According to the study by Saket et al., embedded interactions are subsets of the direct manipulation concept, which allow users to directly adjust visual marks, like bars in a chart, rather than using traditional menus. This method aims to make the action closely match the user's intention, making tasks more intuitive [45]. The study examines how people directly interact with visual graphics instead of using traditional tools like checkboxes. The researchers conducted a qualitative study to observe how participants interacted with visual graphics, performing 15 different operations. Participants commonly employed direct manipulation of graphical encodings for visualization tasks. By observing participants, the researchers discovered common interaction patterns. However, the chosen strategies of direct manipulation varied among participants. For example, participants sometimes manipulated different graphical encodings to achieve the same goal. Some adjusted the height of bars to navigate temporal data, while others used color changes or repositioning. The complexity of strategies also varied, with some participants using simple, direct manipulations, while others combined multiple actions, such as resizing and repositioning or recoloring and dragging.

This approach aligns with my research objectives, which aim to enhance the usability and intuitiveness of maritime data exploration tools by employing innovative interaction techniques that match the user's intentions. In my research, the principles of embedded interactions are integrated into the design of the Visual Query Builder (VQB) and other interface components such as the map visualization application and the AR component. By allowing users to directly manipulate visual elements on a map or within AR environments, my system enhances the user's ability to perform complex queries and data manipulations more intuitively. For example, users can select and filter their areas of interest by directly interacting with the visual representations on a tabletop interface or drag and drop nodes in the VQB from the palette to the workspace to generate nodes and formulate their queries.

2.4 Collaboration in Maritime Operations

Collaboration is crucial in designing a system architecture for maritime operations because effective decision-making often requires input from multiple operators and stakeholders with different roles and varying levels of expertise. Collaboration matters in the maritime context because it enhances decision-making, situation awareness, and operational efficiency, thereby improving maritime safety and effectiveness in various operations [55]. Lee et al. highlighted that collaboration in maritime operations is crucial for managing safety, encompassing both formal and informal interactions among crew members, between ships, and with shore-based entities, which helps in adjusting to variable and uncertain situations [33]. Collaboration in maritime operations is prevalent in various types of operations, including ship navigation (traffic control, surface navigation, underwater navigation), crisis operations, maritime construction, and cargo handling in ports [55]. Because of these factors, we considered collaboration in our system architecture design to enable operators to collaborate effectively while using our system for both exploring and querying maritime data.

2.5 Large Tabletop Interfaces

An interactive tabletop interface is nothing but a wide display screen that you can put on a table. It allows one or more users to give it commands by touching the screen with a special pen or their fingers. Tabletops allow multiple users to collaboratively work on it and freely move around it, therefore, this type of interface has been considered as an ideal interface for a collaborative workspace for a small number of users [49].

Since these interfaces usually have a large touchable screen, they allow maritime decision-makers to have a good spatial understanding of maritime data. Tabletop interfaces enable direct manipulation of digital objects, allowing its users to interactively analyze and control maritime operations. Additionally, some tabletop interfaces can detect tangible objects placed on them. This ability assists developers and software architects in adding more interactive control visual elements. These elements, working in harmony with tangible items, enhance the tabletop application, allowing users to interact more effectively. The GeoGate research [52] provides a compelling example of how a tangible item can be used on a tabletop. This research will be elaborated on in the subsequent paragraph. Given the advantageous features of tabletops that have been highlighted, these interfaces are widely used in researches related to sea operations and analyzing maritime data [47, 61, 30, 52, 26].

The GeoGate research by Ssin et al. is a system designed to help explore the relation between multiple geo-temporal datasets. They utilized the tabletop in their system to visualize maritime trajectories [52]. GeoGate system utilizes an augmented reality approach to visualize the maritime trajectories in a space-time cube (STC) format. They have also considered a physical ring device whose position can be detected by the tabletop via TUIO commands. This tangible element helps users select their region of interest on the map to have a better focus by spatial filtering of the visual element on the map and augmented visuals over the map. To evaluate GeoGate, a user study was conducted to compare it with traditional 2D desktop systems used for vessel movement visualization. The study aimed to test the performance and accuracy of GeoGate in understanding the geo-temporal movements of multiple objects. The result showed that GeoGate was as accurate, or in some cases more accurate, than traditional 2D desktop systems in most tasks. However, the performance of GeoGate was similar to the traditional 2D interface except in tasks involving the identification of specific data points based on latitude and longitude, where GeoGate was less efficient. Figure 2.3 shows the way that the information about the STC object has been augmented over the tabletop.

Our research builds upon the GeoGate research by providing a visual query builder tool for the maritime exploring system, that is interconnected with the tabletop in an interactive way to add additional capabilities to support spatiotemporal querying in the maritime domain.

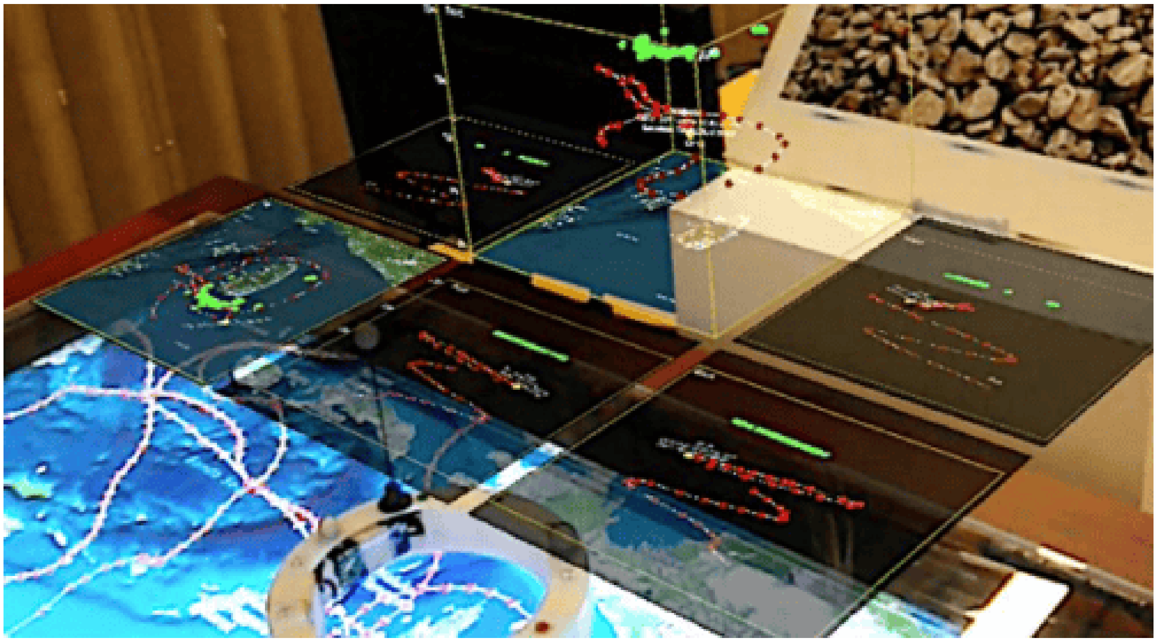


Figure 2.3: The GeoGate system showcase, wherein each view illustrates a different perspective of the Space-Time Cube (STC) oriented towards the observer. Progressing in a counter-clockwise direction, these perspectives represent the left, top, front, back, and right sides of the STC [52] © 2019 IEEE. Reprinted with permission.

Scott et al. developed a prototype application that focuses on planning and decision-making in maritime operations, which runs on a pen-based tabletop [47]. On their prototype, they have considered a map visualization that allows users to view and edit the vessels' tracks. Their prototype is made so that many users can work together at the same time collaboratively, with each person viewing things on the screen that match their specific task needs. Additionally, they demonstrated and briefed their prototype to the Canadian Forces Maritime Warfare Centre, the I/ITSEC conference (2009), and the DRDC Atlantic scientific community; and have received positive and promising feedback. However, there were some challenges reported regarding the pen-based approach of the tabletop and the wireless internet that might not be available in some complex military environments.

The research by Ens et al. introduces Uplift, a system prototype that was co-designed with smart-grid experts to aid data exploration among stakeholders with various expertise [15]. The Uplift prototype incorporates tabletop interfaces to visualize the map layout of a location of study on it using the Mapbox[2] framework. In this prototype, they have also utilized tangible objects that represent the chunk of area grids inside the whole area that are trackable using the AR device. The Augmented reality devices in this study are interactively entangled with the tangible objects in a way that augments data visualization of the grids on top of each tangible object, and as you lift the part, it shows more detailed information about that part such as the yearly power consumption and power generation. The user study that they conducted also showed promising results which participants found the Uplift prototype engaging while mentioning some potential challenges.

2.6 Augmented Reality in GIS

Augmented Reality (AR) offers a digital overlay that can educate, alert, or support users in various contexts. Many AR systems, such as Microsoft Hololens which we used later in the project, can interpret hand gestures and eye movements as interactive commands. Additionally, the portability of modern AR devices, thanks to their battery-powered design, allows them to be used anywhere. One of AR's standout features is its immersive and captivating user experience. Unlike traditional displays, which are bound by pixel constraints, AR operates within a broader spatial context, virtually eliminating size limitations. This means that challenges, like the size restrictions of a tabletop interface, can be overcome by augmenting visuals beyond its boundaries using AR. Unsurprisingly, These features made AR devices an appealing tool to utilize and explore in the maritime environment.

Helzel et al. [26] developed a mixed reality (MR) prototype with the goal of enhancing decision-making and spatial awareness in the context of hazard detection, offering a dynamic visualization of the disaster scenarios to the operators. Two scenarios have been implemented; large fire hazard and earthquake. In both scenarios, they have offered a dynamic visualization of the hazard information such as the location of the hazard and the damaged structures by utilizing a mobile device AR for projecting the information over a tabletop. This research has evaluated the prototype

[2] Mapbox official website is accessible from https://www.mapbox.com/

with the help of the experts and the result was promising as the experts' reviews revealed positive feedback on the potential of the system in enhancing spatial awareness and decision-making. However, this research was not free of limitations as the use of mobile devices for augmenting visuals was discouraged and they suggested using head-mounted displays (HMD) for the augmentation purpose to enhance usability.

Another usage of mixed reality in a geospatial context can be found in Rydvanskiy et al. research [44]. In this study, they developed a mixed-reality system that visualizes flood scenarios and different flood levels. This system utilizes Hololens 2 for augmented reality purposes and Photon for enabling synchronization and collaboration between multiple AR clients. This system is highly interactive in a way that a user can move around, alter the strength of the flood models, and change the flood models while keeping synchronized with the environment. The user study conducted in this research suggests a significant potential for enhancing the understanding of flood risks and adaptations while having some challenges such as content management when the user environment is poorly mapped and hardware constraints such as the poor performance of the Hololens devices on bright environments. This study influenced our research in synchronizing multiple AR clients in a single collaborative session using the Photon library.

In the study by Liu et al., researchers introduced an AR navigation aid for vessels that combines AIS data with visual detections, optimizing the utilization of both visual and positional information. This fusion approach aims to enhance traffic situation awareness for autonomous sea vehicles. Their experimental results highlighted the system's effectiveness under various navigational conditions such as lack of light, haze, etc. [36].

2.7 Summary and Gap Identification

In this chapter, related research works and tools for maritime data exploration and querying have been discussed in various categories. The reviewed studies in visual querying systems have advanced in this field by proposing and developing intuitive visual querying interfaces. The common goal of visual querying systems is to enhance accessibility for non-expert users by providing an intuitive interface to work with without the necessity of having knowledge in querying languages and other technical aspects [40, 58, 57, 20]. Another consideration of developing a visual querying system is to find a balance between intuitiveness and ease of use for non-expert users and the usability and robustness of the system for more advanced users. Despite the progress in visual querying systems, several gaps remain. While various visual query systems for general data querying were proposed and developed their ability to handle basic and complex spatial relationships and temporal filtering in queries is still limited. These features are crucial for exploring and querying spatiotemporal data such as maritime data.

Due to the importance of improved user experience, efficiency, and collaboration in maritime operations, several researches have been conducted to propose, implement, and evaluate maritime querying systems that involve novel interaction methods such as tangible objects, large touchable interface table-tops, and augmented reality environments [52, 15, 48, 30]. To enhance the existing body of knowledge, this research seeks to develop a system design tailored for collaborative maritime querying and exploration tasks by leveraging established methods that have been used in the literature while including novel interaction and direct manipulation methods.

Chapter 3

Design and Evaluation of the Visual Query Builder (VQB) System

3.1 Overall System Architecture

Maritime datasets can benefit from the comprehensive platform known as the Visual Query Builder (VQB) System. With the use of a sizable tabletop interface, augmented reality elements, and a visual query-building process, the architecture goal is to provide a natural and participatory experience. The system is designed to visualize and manage AIS data, enhancing the efficiency and effectiveness of marine exploration and operations. Our architecture comprises several core components:

- **API Server**: Responsible for data retrieval, processing, and delivering the requested data to the front-end visualization app.
- **MongoDB Dataset**: Acts as the primary storage solution, holding the AIS data, user interactions, protectedSeas regional data, and interacts closely with the API server.
- **React Visualization App**: The front-end application allows users to visually interact, and explore maritime data.
- **Visual Query Builder**: This interface allows users to formulate maritime queries by adding nodes to the workspace and linking them.
- **Tabletop Interaction**: Enables users to use tangible items for direct manipulation on the map, enhancing user experience.
- **Augmented Reality Components**: Provides an immersive experience by integrating AR visuals and tools with the main system.

The components we're working with all come together, creating a smooth flow of data. It starts with pulling in the raw AIS data and goes all the way to how things get shown and interacted with on the tabletop. We've set things up so the data moves efficiently, making sure it's pulled, processed, and displayed just how we want it.

Figure 3.1 illustrates the physical system architecture. As depicted, the tabletop display is powered by a dedicated tabletop processor. This processor communicates with the main server via a wireless local area network (WLAN) connection. Through this, it sends requests to the API server, and showcases the results on the tabletop upon receiving responses from the server. Parallelly, AR devices also interface with the API, ensuring they remain synchronized with the tabletop's state. To initiate this synchronization, AR devices must scan a dedicated QR code placed on the tabletop each time they attempt a connection. The role of this QR code is to spatially synchronize the augmented visuals with the map application visualized on the tabletop.

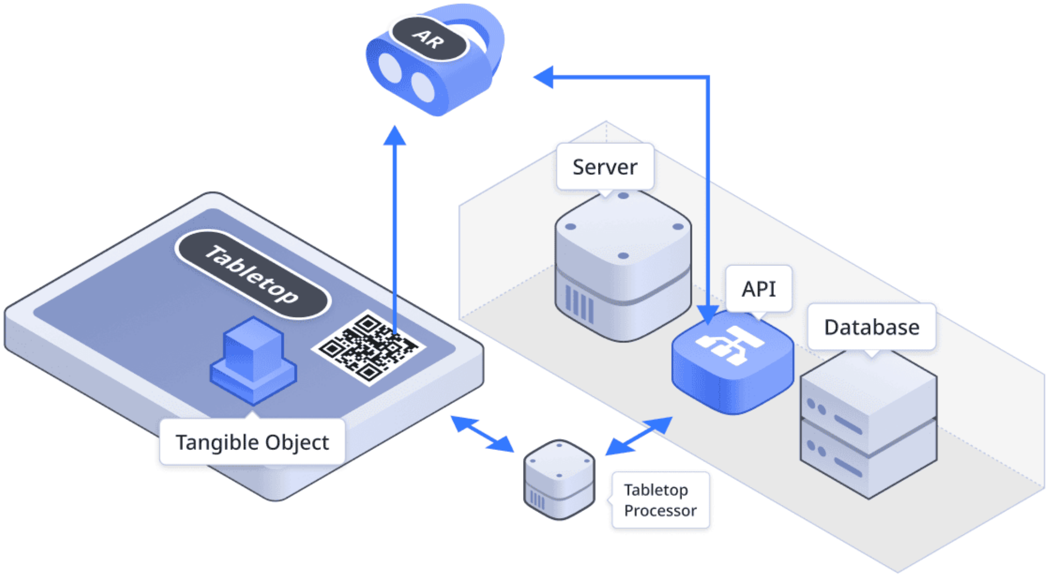


Figure 3.1: Physical system architecture diagram

In essence, the architecture of the VQB system has been meticulously designed to ensure efficiency, scalability, and user-centricity, enabling maritime researchers and operators to derive insights from AIS data. In the following subsection, each of the main components that we have mentioned briefly earlier will be elaborated upon in detail.

3.1.1 Components Breakdown

Here in this part, we will delve deeper into each primary element of our system, shedding light on its purpose, interactions, and significance within the overall architecture.

API Server

The API Server is the central hub for data movement in our system. For the backend, we employed ExpressJS[1] with NodeJS[2]. The decision to utilize ExpressJS was based on its efficient handling of asynchronous operations, which is critical for real-time applications like ours. Additionally, its wide range of plugins and active community support made it a favorable choice. This component is mainly responsible for:

- Processing requests from both the tabletop display and AR devices.
- Fetching the required AIS data and the protectedSeas regional data from the database.
- Simplifying the areas or vessels' tracks before sending them as a result.
- Ensuring real-time and efficient communication between different components, such as the tabletop and AR devices.

MongoDB Dataset

We chose MongoDB[3] as our database mainly because it is a great Data Base Management System (DBMS); especially, for working with spatial data, which is crucial for our maritime project. A standout feature of MongoDB is its '2dSphere' indexing. This indexing mechanism not only supports queries that calculate geometries on an earth-like sphere but also significantly enhances the speed of spatial data querying. Also, MongoDB's flexibility and scalability make it an ideal choice for handling diverse datasets like ours. Our MongoDB dataset serves as the primary storage for:

[1] ExpressJS official website: https://expressjs.com/

[2] NodeJs official website: https://nodejs.org/

[3] MongoDB official website: https://www.mongodb.com/

- AIS Data: Providing insights about vessel movements and behaviors.
- ProtectedSeas Regional Data: Polygon-shaped data points highlighting specific maritime areas.
- User Sessions: User sessions data stores the interaction data of users. It helps in managing multiple user sessions concurrently, sharing and restoring a user session, as well as further analysis of user experience.

Visualization App

React[4] is a well-known library for creating web applications that provide effective state management and data handling capabilities, making it a great choice for rapid prototyping. For this reason, we have considered this tool to be the core component of our visualization system.

The visualization application incorporates a map visualization tool from Leaflet[5]. This tool offers numerous map visualization features and provides intuitive direct manipulation interactions, enabling users to effortlessly explore and navigate through the map. Additionally, a visual querying panel was implemented using GoJs[6], a well-known tool for its rapid diagram generation capabilities, making it especially advantageous for prototyping. In the next subsection, the visual query builder interface will be discussed in detail. Figure 3.2 depicts the appearance of the visualization interface. This application offers:

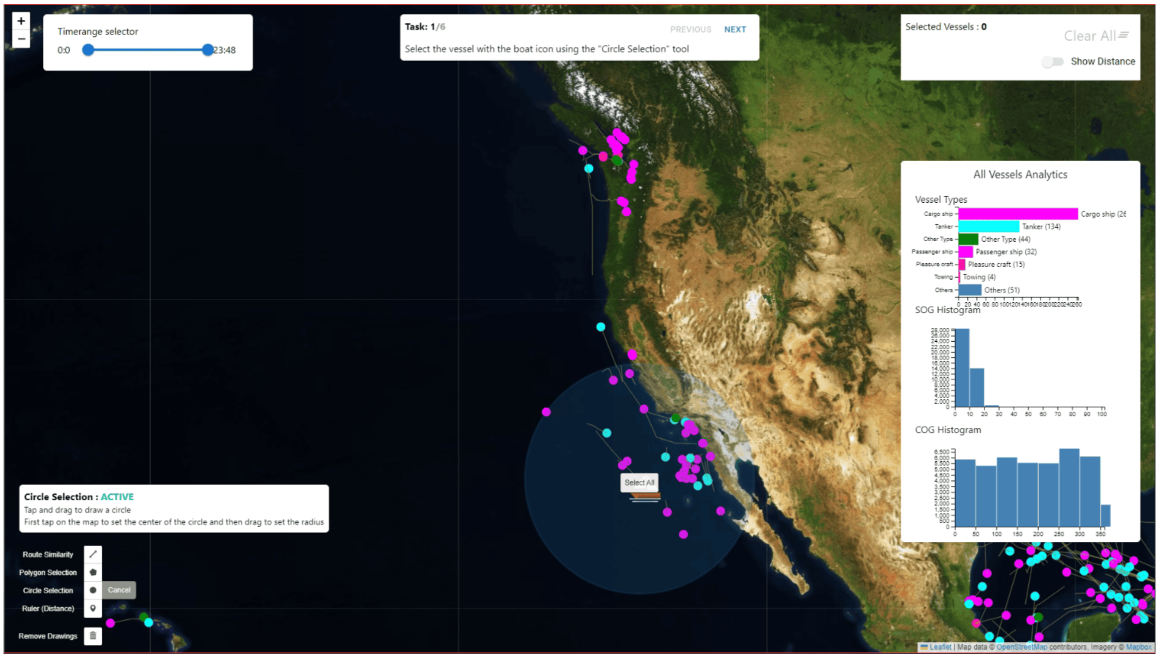
- A visual interface for users to interact with the data
- A map interface for visualizing maritime objects and other related spatiotemporal data such as AIS
- Menus that offer various interactive features, allowing users to draw on the map to select a region of interest, as well as control elements for filtering time ranges and other related functions
- Visual Query Builder (VQB) tool for query formulation using visual elements

[4] React official website: https://react.dev/

[5] Leaflet official website: https://leafletjs.com/

[6] More information about GoJs can be found at https://gojs.net/

- Direct manipulation on the map for intuitive interactions



Task: 1/6

Select the vessel with the boat icon using the "Circle Selection" tool

Circle Selection : ACTIVE

Tap and drag to draw a circle

First tap on the map to set the center of the circle and then drag to set the radius

Selected Vessels : 0

Figure 3.2: The interface of the visualization application

Visual Query Builder

We have iteratively enhanced and implemented this interface. The goal of this interface is to allow novice users to work with maritime data querying. In this interface, a palette of pre-defined nodes has been provided, which by dragging a node out of that in the workspace, we can initiate that node. The nodes have ports on their sides, which can be used for node linkage to define the relations between them. More detail about the inner components of this interface and the implementation journey, its enhancements over time, and some example queries will be elaborated on later in this chapter.

Tabletop Interaction

We're using a MultiTaction tabletop for our setup. This tabletop can detect multiple touches and QR codes at once, and it communicates using TUIO[7] commands.

[7] TUIO framework website: https://www.tuio.org/

We made a script to turn these commands into appropriate visuals on our display program. In our project, users can place a tangible object on the table. This action triggers a ring menu to appear around the object. From this menu, users can pick how they want to interact with the system. They might select the kind of action they want or switch between drawing the area of their interest on the map and a navigation mode, which lets them move around and explore the map (Fig 3.3).

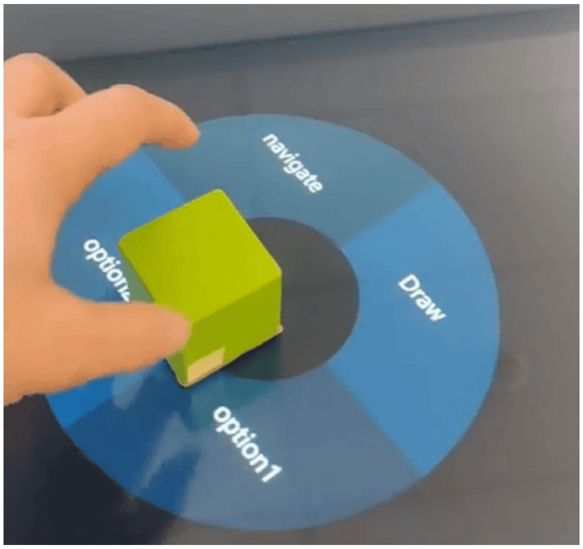


Figure 3.3: Placing the tangible cube on the tabletop opens a ring Menu

Augmented Reality (AR)

The augmented reality tool must be synchronized with the map's current state. For instance, if we display vessels on the tabletop in augmented reality at specific locations and a user alters the map's zoom or boundaries, the augmented visuals must adjust accordingly. To ensure this synchronization, we integrated a socket broadcasting server within our tabletop server. This server receives the map's current state and then broadcasts it to all AR clients (View Figure 3.4), ensuring they align with the tabletop's display. For optimal performance, we only transmit map states from the visualization application when a user modifies the map state.

The AR component offers:

- Synchronized augmented visuals adjusted according to the tabletop's map state.
- Interaction elements that allow the user to adjust the time range

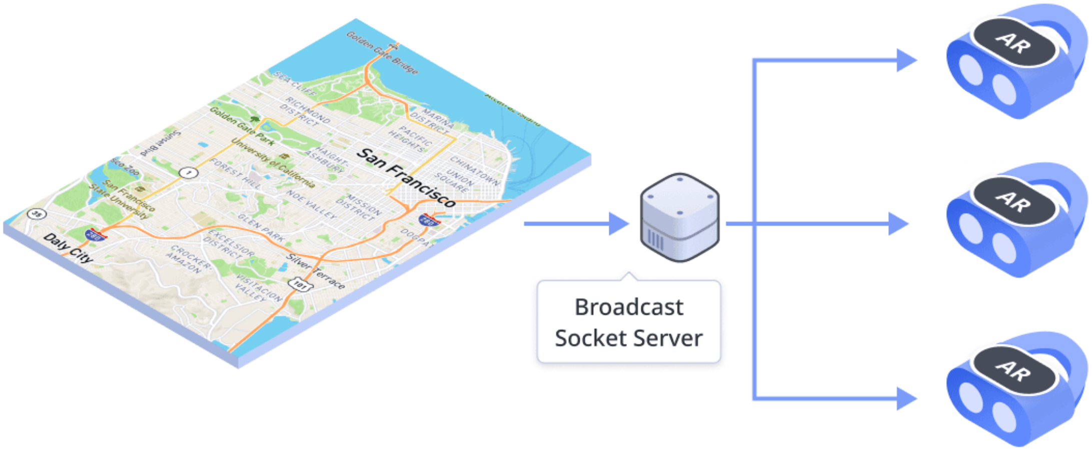


Figure 3.4: Broadcasting map state via socket connection; required for synchronization of the AR clients

- QR code scanning for spatially syncing AR-device to the tabletop.

In summary, each component of our system has been designed to offer a smooth, efficient, and immersive user experience. They work in harmony to ensure efficient data retrieval, storage, processing, and representation.

3.1.2 Scalability

The map visualization application has been implemented in a way that accepts AIS data for visualization. Therefore, it is scalable and can be connected to a live data server and visualize the real-time vessel data. Moreover, we translate the drawings that a user generates on the map to geoJson standard format, interpretable by most of the datasets, such as MongoDB, PostgreSQL with PostGIS, MySQL, etc. The visual query builder interface, however, is not compatible with other types of datasets, and its parser and compiler have to be implemented individually. Moreover, by adding any new feature to the query builder interface, we have to reflect the changes to the visual query parser define the new feature, and implement the conversion of nodes and links data to a query that the database can handle. This process could be time-consuming and complex as the visual query interface features grow, and we can count it as the interface limitation.

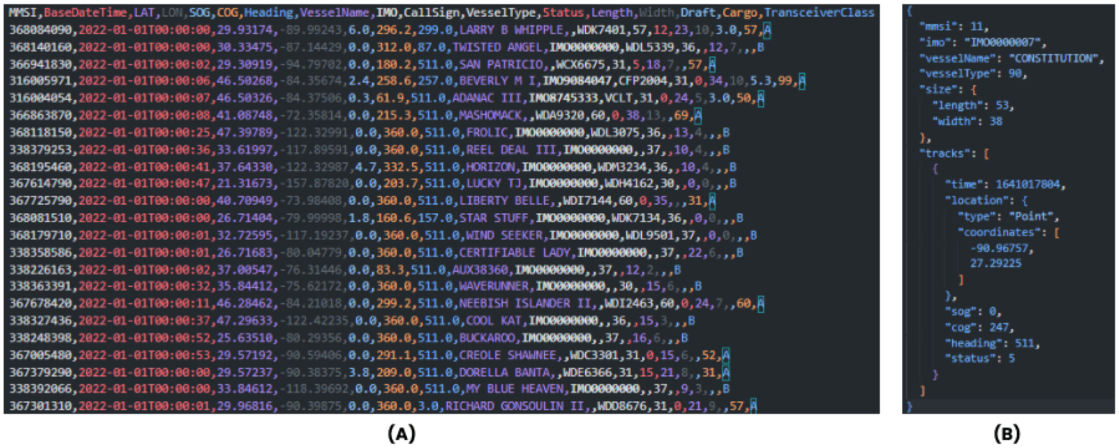
3.2 Gathering and Refining Data

There are two main data sources that we have used in this project: AIS tracks data and protectedSeas regional data. Before showing the AIS data and specific geo-areas, we need to prepare the data. This means extracting the right information and setting it up for clear visualization. We clean, filter, and combine different datasets to get a complete and correct view of maritime traffic and areas. Following this section, the characteristics of each data source, and our approach to processing that data will be explained.

3.2.1 Processing of AIS Tracks

The AIS tracks data that we use in this study have been acquired from the website marinecadastre.gov, which is maintained by the National Oceanic and Atmospheric Administration (NOAA) of the United States. The website [3] described that the AIS data is collected by the U.S. Coast Guard and contains vessel traffic information such as location, time, vessel type, speed, length, beam, and draft. The data has been repurposed and made available for analysis in desktop GIS software by the Bureau of Ocean Energy Management, NOAA, and the U.S. Coast Guard Navigation Center. The data available on the website is provided in a compressed ZIP format, containing recorded tracks of each day individually. The uncompressed file includes a CSV file that uses comma-separated values and for each recorded track, contains information such as MMSI, BaseDateTime, LAT, LON, SOG, COG, Heading, VesselName, IMO, CallSign, VesselType, Status, Length, Width, Draft, Cargo, and TransceiverClass.

As some data related to vessels such as MMSI and VesselName remain constant over time, we have combined the data from these columns and stored the time-related data in an array named tracks. The original format of the AIS data is illustrated in Figure 3.5:A, while the processed data is presented in Figure 3.5:B. It is important to note that the example in Figure 3.5:B only displays a single track with one row of AIS data for illustration purposes. The updated data format, shown in Figure 3.5:B, includes a modification to the spatial data format. The coordinates (LAT, LON) have been transformed into a location object using a well-known geoJSON format that consists of a point type and an array of coordinates [longitude, latitude]. This enables the use of geospatial-based operations such as querying geoJSON points within a geoJSON area, such as multi-polygon-shaped areas, utilizing functions provided by the MongoDB database platform.



```
MMSI,BaseDateTime,LAT,LON,SOG,COG,Heading,VesselName,IMO,CallSign,VesselType,Status,Length,Width,Draft,Cargo,TransceiverClass
368084090,2022-01-01T00:00:00,29.93174,-89.99243,6.0,296.2,299.0,LARRY B WHIPPLE,,WDK7401,57,12,23,10,3.0,57,A
368140160,2022-01-01T00:00:00,30.33475,-87.14429,0.0,312.0,87.0,TWISTED ANGEL,IMO0000000,WDL5339,36,,12,7,,,B
366941830,2022-01-01T00:00:02,29.30919,-94.79702,0.0,180.2,511.0,SAN PATRICIO,,WCX6675,31,5,18,7,,57,A
316005971,2022-01-01T00:00:06,46.50268,-84.35674,2.4,258.6,257.0,BEVERLY M I,IMO9084047,CFP2004,31,0,34,10,5.3,99,A
316004054,2022-01-01T00:00:07,46.50326,-84.37506,0.3,61.9,511.0,ADANAC III,IMO8745333,VCLT,31,0,24,5,3.0,50,A
366863870,2022-01-01T00:00:08,41.08748,-72.35814,0.0,215.3,511.0,MASHOMACK,,WDA9320,60,0,38,13,,69,A
368118150,2022-01-01T00:00:25,47.39789,-122.32991,0.0,360.0,511.0,FROLIC,IMO0000000,WDL3075,36,,13,4,,,B
338379253,2022-01-01T00:00:36,33.61997,-117.89591,0.0,360.0,511.0,REEL DEAL III,IMO0000000,,37,,10,4,,,B
368195460,2022-01-01T00:00:41,37.64330,-122.32987,4.7,332.5,511.0,HORIZON,IMO0000000,WDM3234,36,,10,4,,,B
367614790,2022-01-01T00:00:47,21.31673,-157.87820,0.0,203.7,511.0,LUCKY TJ,IMO0000000,WDH4162,30,,0,0,,,B
367725790,2022-01-01T00:00:00,40.70949,-73.98408,0.0,360.0,511.0,LIBERTY BELLE,,WDI7144,60,0,35,,,31,A
368081510,2022-01-01T00:00:00,26.71404,-79.99998,1.8,160.6,157.0,STAR STUFF,IMO0000000,WDK7134,36,,0,0,,,B
368179710,2022-01-01T00:00:01,32.72595,-117.19237,0.0,360.0,511.0,WIND SEEKER,IMO0000000,WDL9501,37,,0,0,,,B
338358586,2022-01-01T00:00:01,26.71683,-80.04779,0.0,360.0,511.0,CERTIFIABLE LADY,IMO0000000,,37,,22,6,,,B
338226163,2022-01-01T00:00:02,37.00547,-76.31446,0.0,83.3,511.0,AUX38360,IMO0000000,,37,,12,2,,,B
338363391,2022-01-01T00:00:32,35.84412,-75.62172,0.0,360.0,511.0,WAVERUNNER,IMO0000000,,30,,15,6,,,B
367678420,2022-01-01T00:00:11,46.28462,-84.21018,0.0,299.2,511.0,NEEBISH ISLANDER II,,WDI2463,60,0,24,7,,60,A
338327436,2022-01-01T00:00:37,47.29633,-122.42235,0.0,360.0,511.0,COOL KAT,IMO0000000,,36,,15,3,,,B
338248398,2022-01-01T00:00:52,25.63510,-80.29356,0.0,360.0,511.0,BUCKAROO,IMO0000000,,37,,16,6,,,B
367005480,2022-01-01T00:00:53,29.57192,-90.59406,0.0,291.1,511.0,CREOLE SHAWNEE,,WDC3301,31,0,15,6,,52,A
367379290,2022-01-01T00:00:00,29.57237,-90.38375,3.8,209.0,511.0,DORELLA BANTA,,WDE6366,31,15,21,8,,31,A
338392066,2022-01-01T00:00:00,33.84612,-118.39692,0.0,360.0,511.0,MY BLUE HEAVEN,IMO0000000,,37,,9,3,,,B
367301310,2022-01-01T00:00:01,29.96816,-90.39875,0.0,360.0,3.0,RICHARD GONSOULIN II,,WDD8676,31,0,21,9,,57,A
```

```
{
  "mmsi": 11,
  "imo": "IMO0000007",
  "vesselName": "CONSTITUTION",
  "vesselType": 90,
  "size": {
    "length": 53,
    "width": 38
  },
  "tracks": [
    {
      "time": 1641017804,
      "location": {
        "type": "Point",
        "coordinates": [
          -90.96757,
          27.29225
        ]
      },
      "sog": 0,
      "cog": 247,
      "heading": 511,
      "status": 5
    }
  ]
}
```

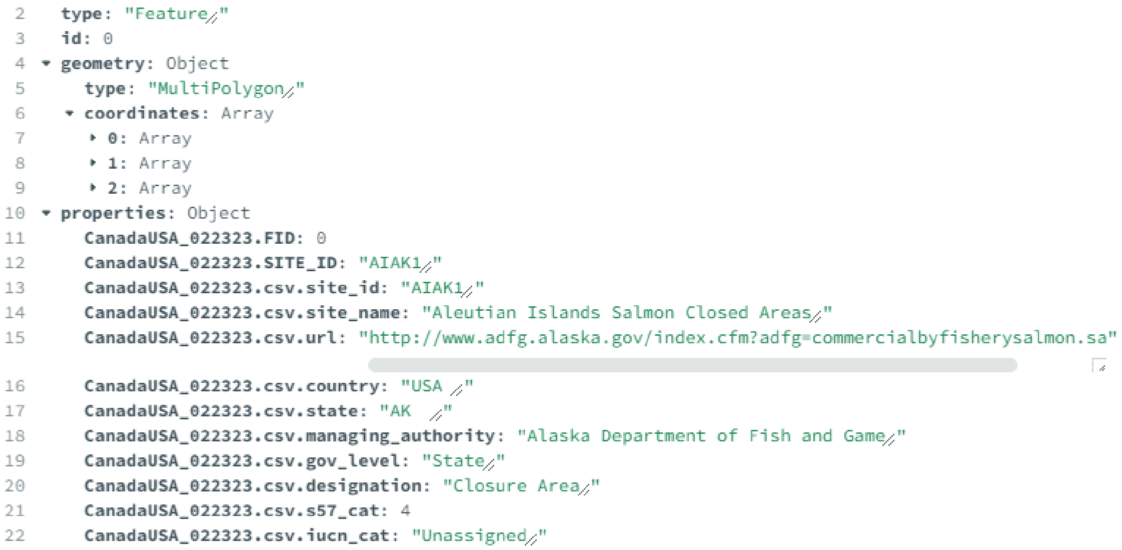
Figure 3.5: AIS Data structure A) Raw data AIS data from CSV B) preprocessed AIS data

3.2.2 Protectedseas Areas: GeoJson Data [2]

We've incorporated ProtectedSeas areas as another data source in the interface. These data come in geoJson format [9] and include information such as the type of area (e.g. polygon or multi-polygon), as well as various identification properties like id and name, and area metadata properties like country and state. Additionally, the data includes information about species of concern, fishing regulations, and other related details about the area. For simplicity, we are not displaying all properties of the imported area, but Figure 3.6 depicts its final appearance after importing to the database.

Given that the total number of ProtectedSeas areas provided is quite high and these areas have high-fidelity shapes with numerous nodes, data transformation from our dataset to the clients could be slow. To address this issue, we have decided to simplify the areas before sending them to clients via the API. We are using the TurfJS library[8] for this purpose, which claims to utilize the Ramer-Douglas-Peucker [18] algorithm for simplification.

[8] TurfJs official website: https://turfjs.org/



```
  type: "Feature"
  id: 0
▾ geometry: Object
    type: "MultiPolygon"
  ▾ coordinates: Array
    ▸ 0: Array
    ▸ 1: Array
    ▸ 2: Array
▾ properties: Object
    CanadaUSA_022323.FID: 0
    CanadaUSA_022323.SITE_ID: "AIAK1"
    CanadaUSA_022323.csv.site_id: "AIAK1"
    CanadaUSA_022323.csv.site_name: "Aleutian Islands Salmon Closed Areas"
    CanadaUSA_022323.csv.url: "http://www.adfg.alaska.gov/index.cfm?adfg=commercialbyfisherysalmon.sa"
    CanadaUSA_022323.csv.country: "USA "
    CanadaUSA_022323.csv.state: "AK  "
    CanadaUSA_022323.csv.managing_authority: "Alaska Department of Fish and Game"
    CanadaUSA_022323.csv.gov_level: "State"
    CanadaUSA_022323.csv.designation: "Closure Area"
    CanadaUSA_022323.csv.s57_cat: 4
    CanadaUSA_022323.csv.iucn_cat: "Unassigned"
```

Figure 3.6: ProtectedSeas geoJson areas

3.3 API Server and Database Integration

To ensure data integrity, scalability, and enhanced performance, we implemented a centralized API. This API processes incoming requests, validates them, and optimizes the results before sending responses. Given the extensive data, with numerous vessels each having its own track composed of multiple nodes, direct visualization on the tabletop could be challenging. The tabletop's processing capability might be overwhelmed, potentially leading to a less-than-smooth user experience. Hence, we found it necessary to simplify the routes before serving the data in response to each request. As mentioned earlier, for this purpose, we are using the Ramer-Douglas-Peucker [18] algorithm to simplify the routes. We input the route in geoJson polyline format to the algorithm, along with a simplification tolerance parameter.

The API provides the following endpoints:

- **GET /ais/tracks**: This endpoint fetches AIS data, returning them as simplified detection tracks in an array. Parameters ‘startTime‘ and ‘endTime‘ are used for temporal filtering. Another parameter, ‘trackLimit‘, is used to define the number of tracks desired for each AIS vessel. If a vessel has track detections exceeding the ‘trackLimit‘ within the stipulated time range, only the specified ‘trackLimit‘ tracks are returned, evenly distributed over the period.

- **GET /ais/track**: This endpoint delivers data for a single AIS vessel based on the ‘startTime‘ and ‘endTime‘ parameters for temporal filtering. It’s beneficial for viewing an individual track in its full detail without the alterations introduced by simplification.

- **POST /ais/queryBySketch**: This endpoint takes a user-drawn polyline over the map to identify vessels following a similar path. Initially, it leverages MongoDB’s ’$geoNear’ function to pinpoint vessels whose paths are in close proximity to the polyline’s starting node. Subsequently, the Hausdorff distance calculates the distance between these vessels’ paths and the input polyline, ranking them accordingly. Essentially, when you provide this endpoint with a polyline, it responds with vessels that have a comparably-shaped path in that vicinity in the specific time range.

- **GET /session/**: This endpoint is for sending the session to the user that contains the state of the map and visual query builders, in addition to the user’s selections and drawing data on the map. If ’sessionId’ is not provided by the user, it will return the latest created session as an output.

- **POST /session/**: This endpoint is for creating a new session by passing the map state.

- **PUT /session/**: This endpoint is for editing a session by passing the ’sessionId’ and all the parameters that need to change for updating it.

3.4 React Visualization App Development

3.4.1 Direct Manipulation on the Map

Direct manipulation actions that have been implemented/considered in the visualization application are as follows:

- **Map navigation**: By pinching and panning on the tabletop’s touchable screen, users can adjust the map to their preferred view. This direct manipulation method is commonly used in many map applications.

- **Data point selection**: Tapping on a vessel on the map opens a panel, allowing users to see information about the vessel's latest track motion data (SOG, COG, spatial position) as well as details about the vessel like its weight, dimensions, name, MMSI, and more. See Fig 3.7 for a visual of this panel.

- **Cross to Select**: Users can freehand-draw a polyline on the map and select vessels by intersecting the polyline with the vessels' previous routes. For this purpose, the "cross to select" tool should be activated.

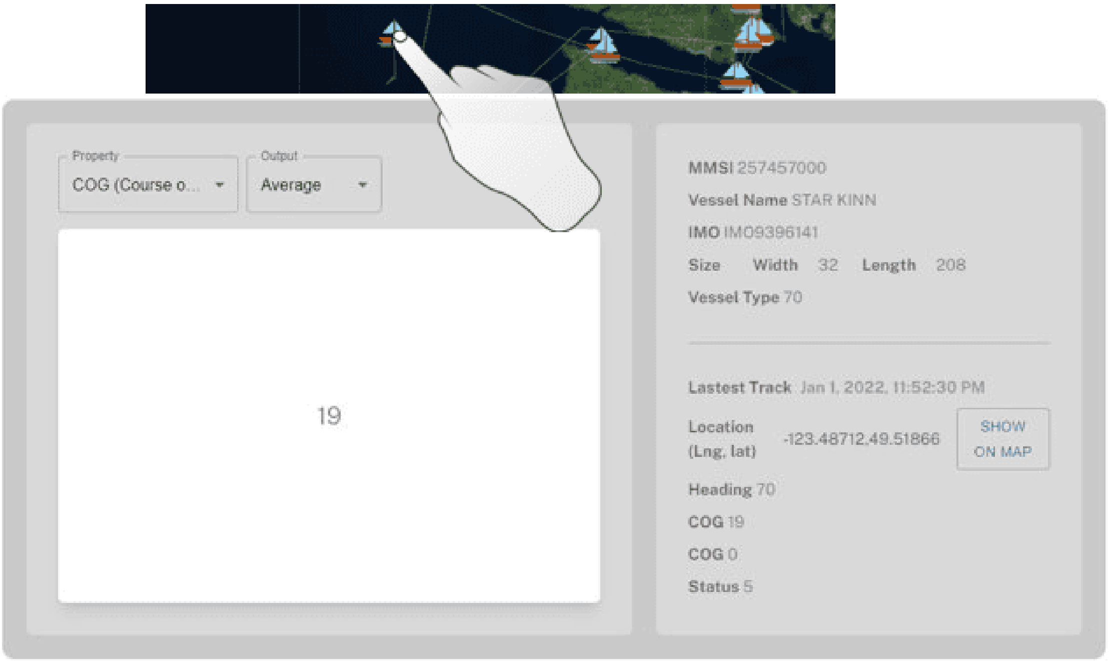
- **Query By Sketch**: By activating the "Query By Sketch" tool, users can draw a polyline on the map and query vessels with a similar route shape compared to the drawn polyline.

- **Spatial Filtering**: Upon activating the "Spatial Filtering" tool, users can define areas of interest on the map. These areas can be defined as circles, polygons, or freehand shapes, which are subsequently simplified into polygonal forms. Users also have the option to refine these drawings; for polygon selections, this is achieved by dragging and dropping its nodes, while for circular selections, the center and radius can be adjusted. Fig 3.8 displays a user-drawn polygon in edit mode, highlighting its adjustable nodes.

Direct manipulation actions that have been implemented/considered in the mixed reality experience are as follows:

- **Lifting a Vessel**: In augmented reality, pinching and lifting a vessel generates its clone, and an AR panel appears displaying general information about the vessel. Elevating the vessel further reveals more detailed information. This DM action is currently under development. Figure 3.9 illustrates the existing functionality, highlighting the vessel and its lifted cloned shadow.

- **Temporal Filtering**: In the AR tool, there has been considered a double-pinch slider that a user can adjust the "startTime" and "endTime" by pinching and releasing these buttons on the slider (view Fig 3.10).



Property: COG (Course o...

Output: Average

19

MMSI 257457000

Vessel Name STAR KINN

IMO IMO9396141

Size Width 32 Length 208

Vessel Type 70

Lastest Track Jan 1, 2022, 11:52:30 PM

Location (Lng, lat) -123.48712,49.51866

Heading 70

COG 19

COG 0

Status 5

Figure 3.7: Detailed information panel displayed upon selecting a single vessel on the map.

3.4.2 User Interface and Interaction Design

This subsection focuses on the implemented interactions and the tools that we have provided for users in order to interact with the system either helping them to formulate their query or helping them in an exploratory way for map navigation. The main components of the user interface of the visualization application are as follows:

- **Map**: Essentially a map showcasing vessels, their paths, and designated regions (from the protectedSeas data). Users can also delineate areas of interest directly on the map. It supports zooming via pinching and panning for optimal viewing.

- **Drawing tools**: These include tools for polygon selection, circular selection, freehand selection, tap-to-select, and polyline selection (used for querying vessels based on route similarity).

- **Temporal filtering slider**: This component is essentially a slider with two buttons, representing the start time and the end time for temporal filtering. Users can adjust it to explore data within their desired time range.

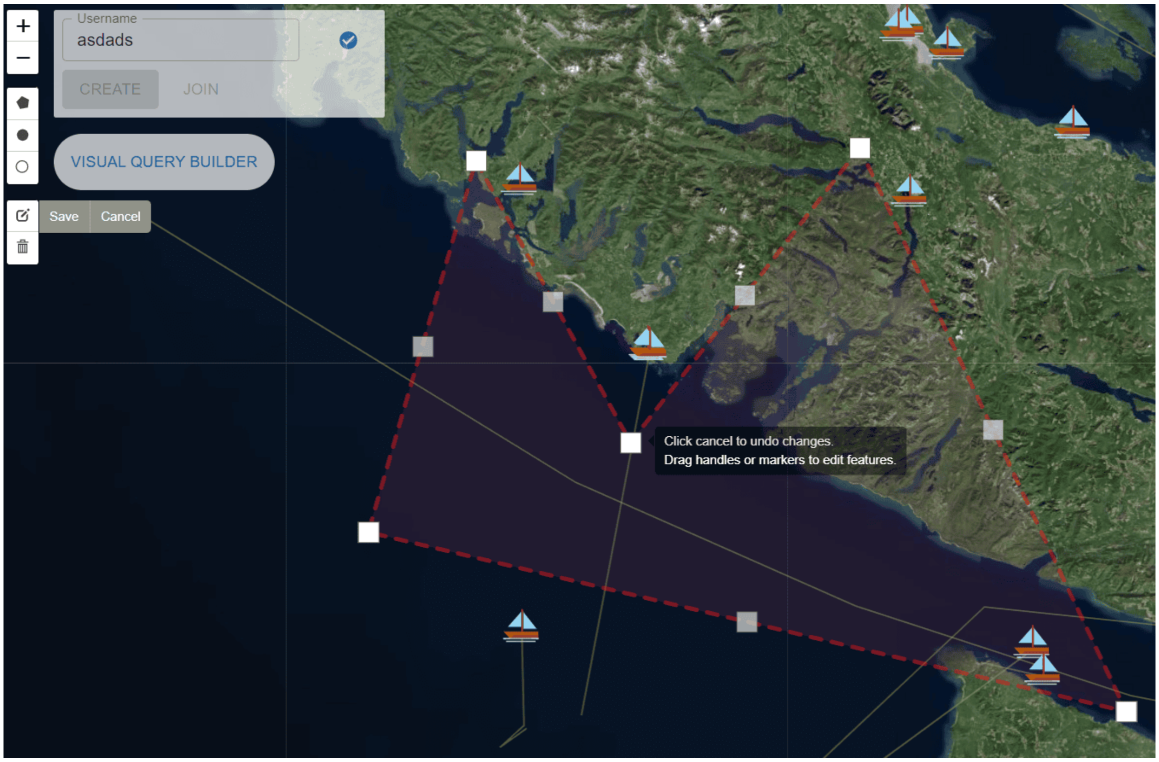

+
−
Username
asdads
CREATE
JOIN
VISUAL QUERY BUILDER
Save
Cancel
Click cancel to undo changes.
Drag handles or markers to edit features.

Figure 3.8: A user-defined polygonal area in edit mode, showcasing adjustable nodes for refining the shape.

- **Visual analytics component**: The visualization tool enables users to select multiple vessels of interest on the map and view the corresponding combined visual analytics for these vessels. These analytics include three charts: a bar chart of vessel types, and histograms showing the Speed Over Ground (SOG) and Course Over Ground (COG) of the selected vessels. At the top of this component, the selected vessels are displayed by their MMSI, allowing users to deselect any vessel if desired.

- **Ring menu**: A customizable menu placement tool. Users position a tangible object with the appropriate QR code on the map where they prefer. Once placed, the ring menu appears above the object, enabling users to select their desired mode – navigation or drawing modes – or to choose specific drawing options. If the user selects the drawing mode, another ring will appear on top of the current rings layer, including options for the drawing modes that can be circular and polygonal selection types.

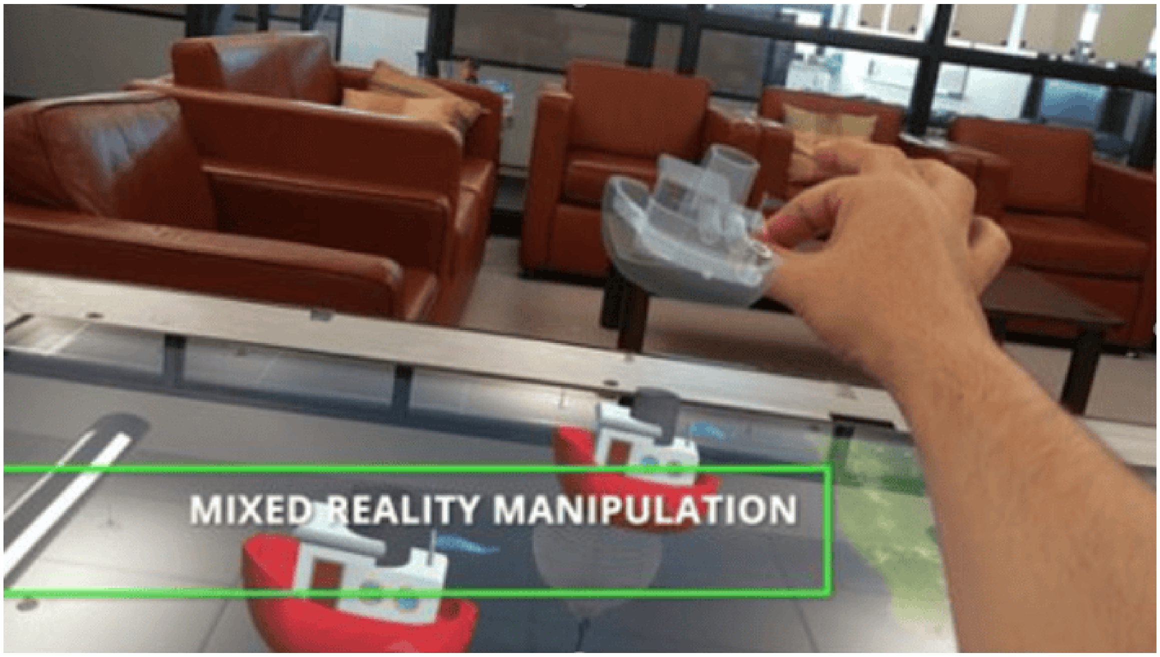


Figure 3.9: Augmented Reality Demonstration: Lifting a vessel to view its detailed information.

- **Ruler tool**: This tool allows the user to measure the distance of two points by placing two markers on the map.

- **Instruction popups**: Certain tools necessitate supplementary information or indicators to show their status. For example, when using the circular selection tool to draw an area of interest, this pop-up underscores that the drawing function is in use. Users are directed to select a point on the map, which will serve as the circle's center, and then drag to determine the circle's radius. This component is specifically designed to both display a tool's current state and guide users through subsequent actions.

3.4.3 Temporal and Spatial Filtering

We have implemented tools for filtering data within the map visualization application, both spatially and temporally. For temporal filtering, two date-time pickers are provided. By adjusting the dates and times of these inputs, users can specify the start and end points of the data. This allows them to analyze the data within their desired timespan. For spatial filtering, users can draw circles or polygonal areas on

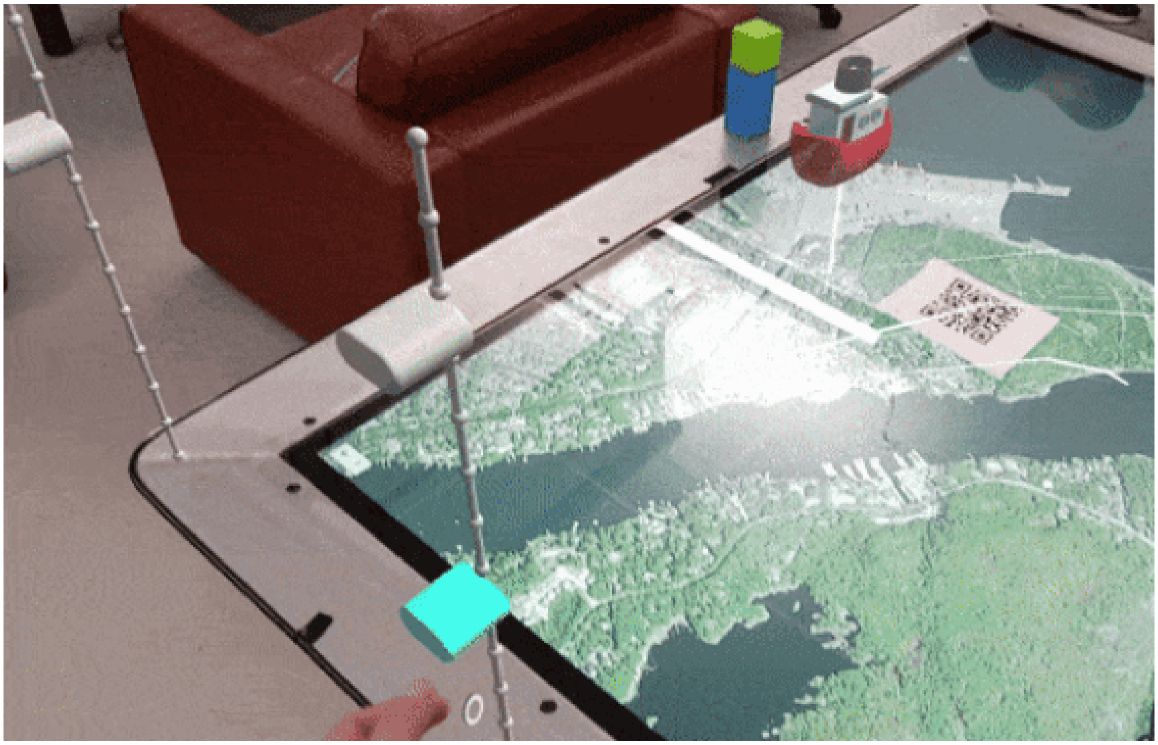


Figure 3.10: An augmented reality double-pinch slider, allowing users to adjust the ”startTime” and ”endTime” for temporal filtering.

the map to filter data directly within those areas, or they can use these areas later in their queries. This can be done either through the visual query builder interface by selecting an area from a list of drawn areas on the map or by using the SPARQL editor and including the area ID in their SPARQL query snippet.

3.5 Visual Query Builder

The visual query builder is divided into two primary sections: the palette and the workspace. Users can select nodes from the palette and drag them to the workspace to aid in formulating their query. As depicted in Fig 3.11, the palette is situated on the left side, while the workspace occupies the right side. To simplify node selection in the palette, we've grouped the nodes into expandable lists based on their attributes, which will be detailed in the subsequent paragraph.

In the palette section, entity nodes represent types of vessels or maritime objects, such as cargo vessels, fishing vessels, or even ice floes. When these nodes are added to the workspace, users can create links based on each entity node's unique

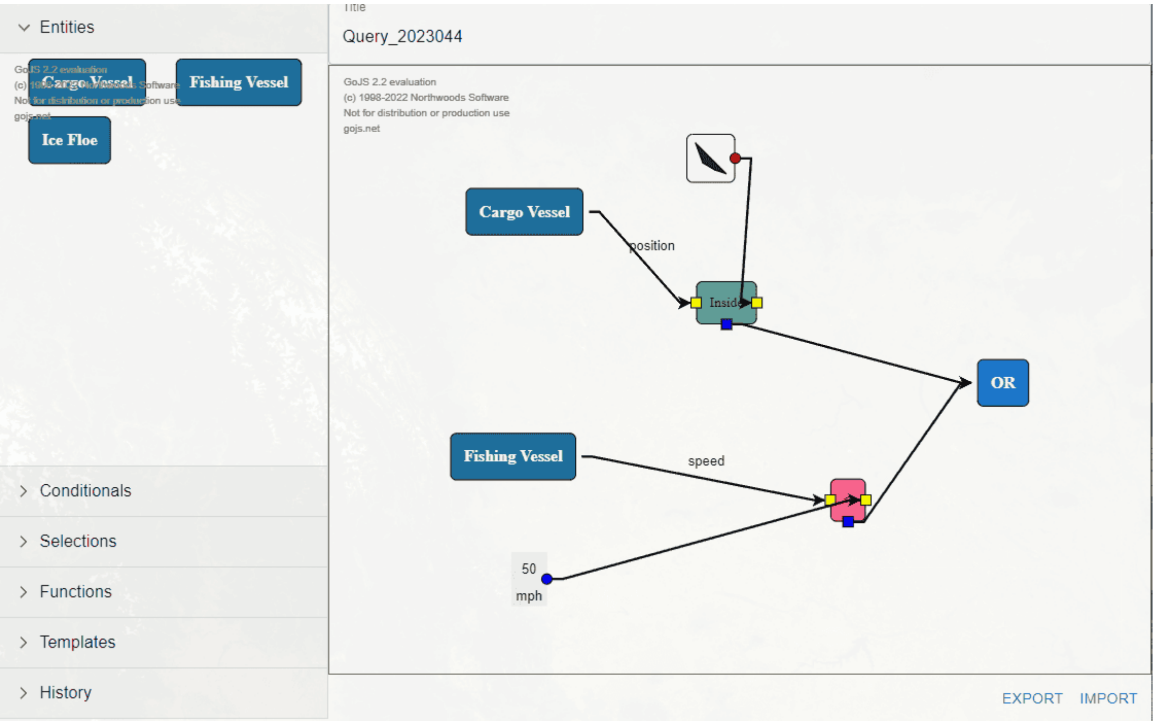


Figure 3.11: An example query generated using the visual query builder interface, which queries cargo vessels inside the polygon or fishing vessels that have a speed over ground exceeding 50 miles per hour.

features. For example, while a cargo vessel might possess a speed attribute, an ice floe wouldn't. Thus, we exclude such options where they don't apply. Conditional nodes, like greater-than, less-than, or equal-to, allow users to establish mathematical or spatial relationships between nodes. Additionally, "AND" and "OR" nodes facilitate the creation of complex queries by combining multiple sub-queries.

Another set of nodes are selection nodes, which represent areas of interest or user-created drawings like polylines (used for route similarity comparisons). This portion of the palette updates dynamically based on user drawings. Following are function nodes that introduce spatial functions to the visual query builder, such as a distance function that calculates the distance between two positions.

The visual query builder also offers template groups, which allow users to drag and drop predefined queries into the Visual Query Builder (VQB) for educational purposes, effectively serving as a hands-on tutorial. Lastly, the history group showcases previously saved user queries, enabling users to revisit or incorporate old queries into their new query constructions. Refer to Fig 3.12 for illustrations of the aforementioned nodes.

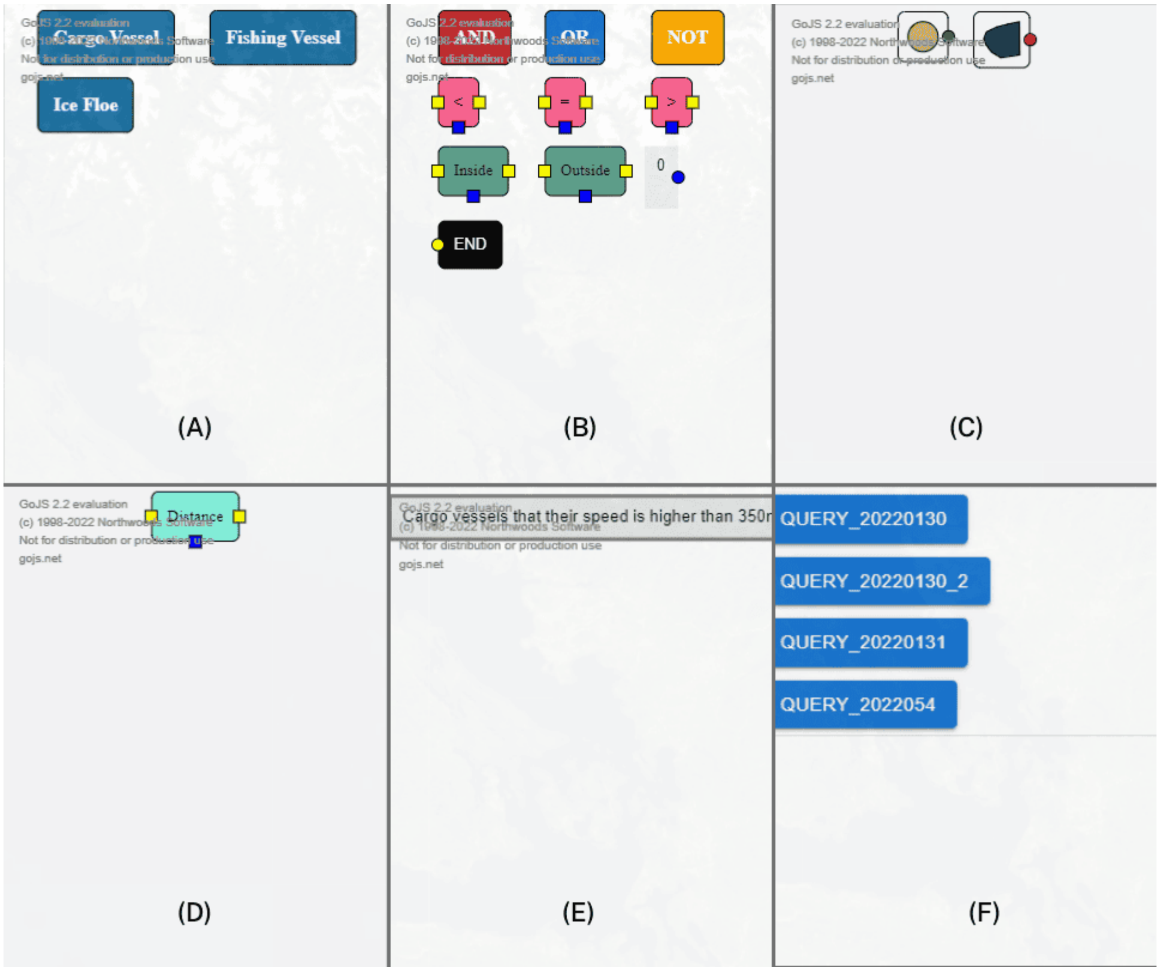


Figure 3.12: Visual query builder palette, A) Entity nodes B) Conditional nodes C) Selection Nodes (User selections over the map) D) Functional nodes E) Template nodes F) Query history

By tapping on each node in the workspace, a ring menu appears over the node, displaying potential links that can be drawn from that node. For instance, if a user taps on the cargo vessel node, they can initiate a link indicating its weight, SOG, COG, or position. Additionally, certain nodes feature a settings button in their ring menu, allowing users to adjust node settings, like specifying a unit or modifying a value. For example, tapping the settings button on a value node enables users to choose a numeric value and its units, such as Kg or tons. To form a query, users drag and connect the generated link to another node using the ring menu. Moreover, there is a delete button in the ring menu for all the nodes that allows users to delete the node if they want to.

Evolution to the Latest Version and Rationale

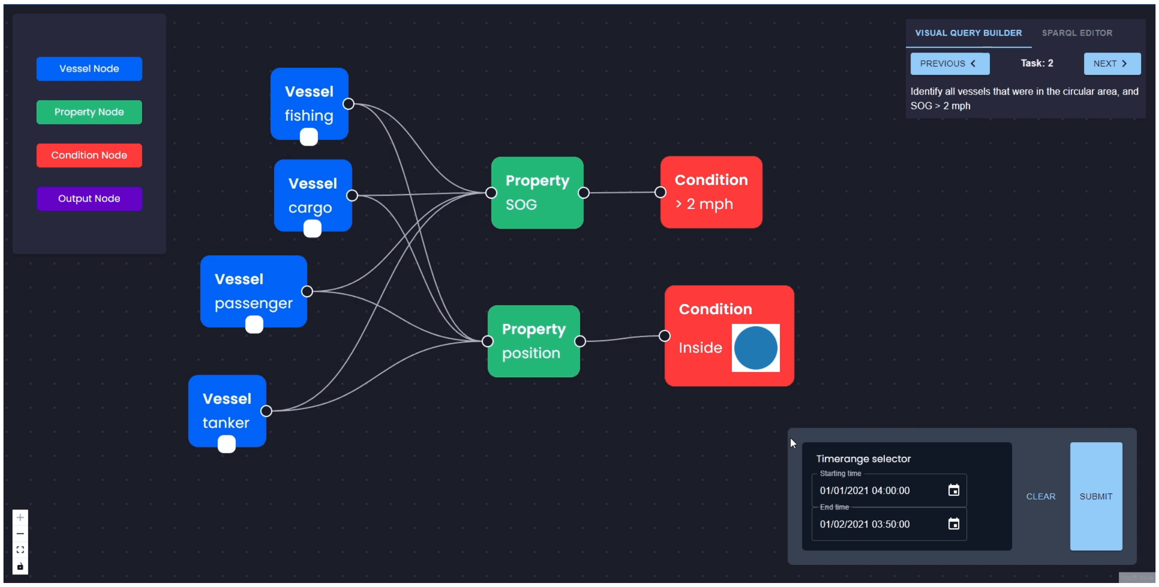
In this research project, we have conducted two cognitive walkthroughs and analyzed the user behaviour while using the VQB interface. Section 3.7 will elaborate upon the cognitive walkthrough in more detail. Briefly speaking, participants felt some difficulties using the earlier version of the VQB interface, especially for constructing compound queries. One major problem with the query interface was working with the ”AND” and ”OR” and other logical operators. In the latest version of the VQB interface, we tried to get rid of these logical operators to make querying more straightforward. Another major problem with this interface was the working with connections (links) and node ports. We also tried to fix that as well. Following this section, the latest version of the VQB interface will be presented.

In the latest node palette of the VQB interface, we have switched from GoJs library to ReactFlow, which is built on a Javascript library[9]. This library allows us to implement diagrams containing nodes and links and it returns us two arrays of data, nodes and links, as the result of the diagram. The updated visual query builder interface that we have implemented offers (vessel node, property node, conditional node, and output node) that lets users drag and drop the nodes that they want to the workspace to create an instance of that node. When a node is released by the user to the workspace, an input prompt will pop up to collect the required information about that node from the user. By connecting two with a link, they can proceed to implement the connection logic that they want in their queries. Another component that has been provided is the timerange selector. This component filters the initial and the ending timestamp of the AIS data that the user is querying for. Fig 3.13 shows a completed query done by utilization of the updated VQB interface.

The nodes that are considered in this interface are as follows:

- **Vessel Nodes:** These nodes clarify the type of vessel(s) that the user is looking for to use in their query. After initializing a vessel node in the workspace, it will ask the type of the vessel and the user will select a vessel type that they are looking for from a drop-down list (Fig 3.14). After finishing the prompt process,

[9] React-Flow library website: https://reactflow.dev



Task: 2

Identify all vessels that were in the circular area, and SOG > 2 mph

Timerange selector

Starting time: 01/01/2021 04:00:00

End time: 01/02/2021 03:50:00

Figure 3.13: Visual Query Builder (VQB) Interface for Maritime Querying, the Latest Version

a vessel node will be added to the querying interface workspace containing the entered vessel type.

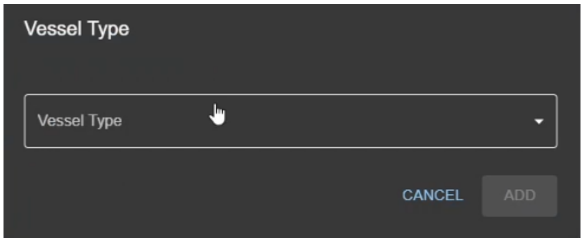


Figure 3.14: Vessel type prompt allowing users to query an exact type of vessels

- **Property Nodes:** A property node is designed to select an AIS property of a vessel in the query. For example, if a user wants to query vessels based on their speed over ground (SOG) property, they would need to add a property node to their workspace and select the SOG property in the property node prompt. The process is similar to filling a vessel node prompt. After that, a property node with the property label will be added to the workspace. Each vessel node can be linked to multiple property nodes, and each property node can be shared among multiple vessel nodes. The value of a property node can be one of these: ”speed over ground (SOG)”, ”course over ground (COG)”, ”weight”, ”height”, "position" (latest geoLocation), and "route". All the other attributes of the AIS data have been discarded from the available values of the prompt for simplicity, however, they can be added later if required. It's worth mentioning that the "route" value allows us to query the vessels based on their route shapes. For example, if a user wants to query vessels that have a route similar to a poly-line shape, they would choose this option for their querying process.

- **Conditional Nodes:** By initializing these nodes in the workspace, users can define the conditions of the properties that they have selected before. Each condition can be linked to multiple or a single property node. Moreover, condition nodes can be shared among multiple property nodes, and each property node can be linked to multiple condition nodes. Before initializing a condition node, users are asked to input the required values of a condition node. After selecting the condition, based on the condition type, other inputs will be adjusted. For example, if a user selects the "Greater than" condition the other two values that are asked from the user would be "value" and the "unit", however, if they want to query vessels that are inside a certain area after they set the condition to "inside", they have to select the area that they want to query vessels in. Fig 3.15 shows the aforementioned visual prompt unfilled, and Fig 3.16 shows the same prompt with two types of filling with different values. For spatial filtering, the input prompt of the condition node is interactively connected to the map application and fetches all the areas that have been drawn on the map. Therefore, if during the initialization of a condition node, the user draws their area of interest on the map, it would appear in the prompt and allow them to select it. Fig 3.16 shows two examples of the filled prompt. The top prompt in that figure shows a numeric condition that represents being greater than 2 mph. However, the second one represents an inside spatial condition with a selected area of interest; in this example, a blue circular area is selected.

- **Output Nodes:** Output nodes are nodes that can be connected to a vessel node, or shared among multiple vessel nodes, after applying the connected conditions to those vessels' properties. The role of the output node is to clarify how we want the final results to be shown. It can be "highlight on the map

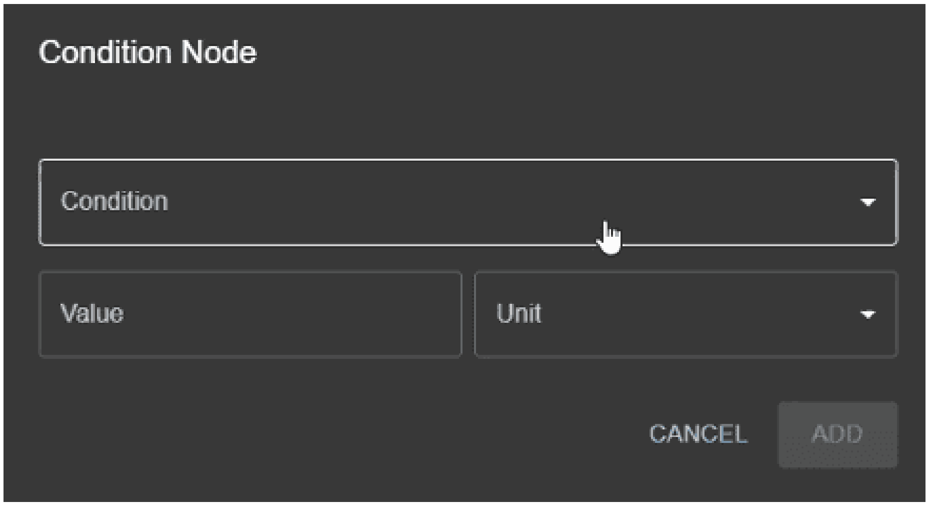
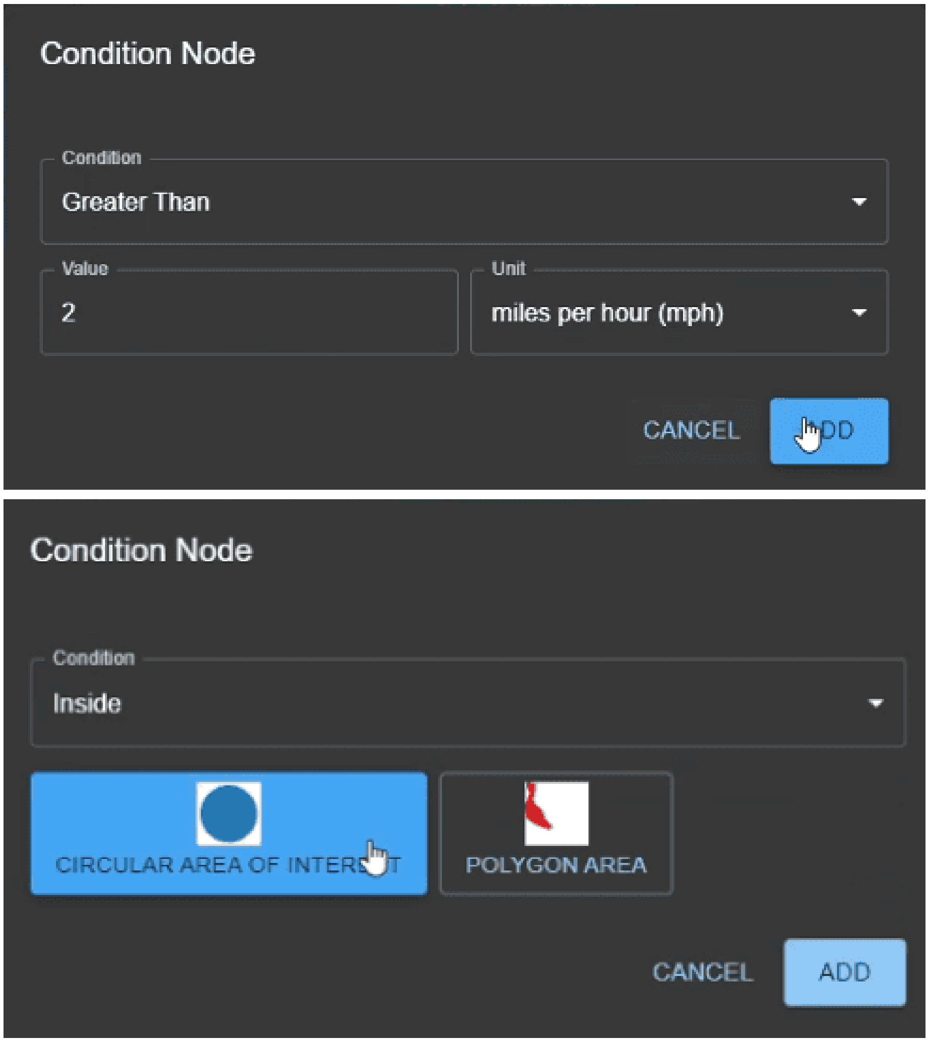


Figure 3.15: Prompt asking users to input the values of a condition node before initialization.

app" or it can be a property of the vessel type such as "SOG", "COG", etc. If the type of output is selected anything except "highlight on the map app", the prompt will ask the user to input the aggregation value, which can be maximum, minimum, or average (Fig 3.17). To give an example, let's say a user wants to select all cargo vessels and view the average speed over the ground of those vessels. To generate such a query, after initializing a cargo vessel node to the querying workspace, the user has to add an output node select "SOG" as the output value, and select "Average" for the aggregation parameter. Eventually, they need to connect the output node to the vessel node (Fig 3.18). To ease the querying process, vessels that are not connected to any output nodes will be highlighted on the map application by default.

3.5.1 Query Formulation

After initializing the nodes, users can define their queries by linking the nodes. A vessel node can be connected to multiple property nodes. This means that in our query we want to add some conditions later for that vessel property. Condition nodes, on the other hand, apply conditions to the vessel properties (property nodes). These nodes can be connected to a property node or multiple property nodes. Also, each property node can be connected to multiple condition nodes, meaning applying



Condition Node

Condition: Greater Than

Value: 2

Unit: miles per hour (mph)

Condition Node

Condition: Inside

Figure 3.16: Two examples of filling the condition node initialization prompt: The first one contains numerical conditioning and the unit and value, and the second one is for spatial querying, asking for the spatial area input

multiple conditions on a vessel property. The shape of the port of the output nodes is different than the other ports, this helps users to distinguish the port node from other ports.

Queries can be formulated as simply as a vessel node, and defining time range which filters data temporally. For example, by adding a cargo vessel node to the workspace and setting the date-time range picker to ”2023-01-01 13:30:00” as the starting date-time and ”2023-01-02 23:30:00” as the ending date-time, we can query all cargo vessels AIS data gathered on that specific period, and visualize them on the map application. Fig 3.19 shows the mentioned example. By executing this

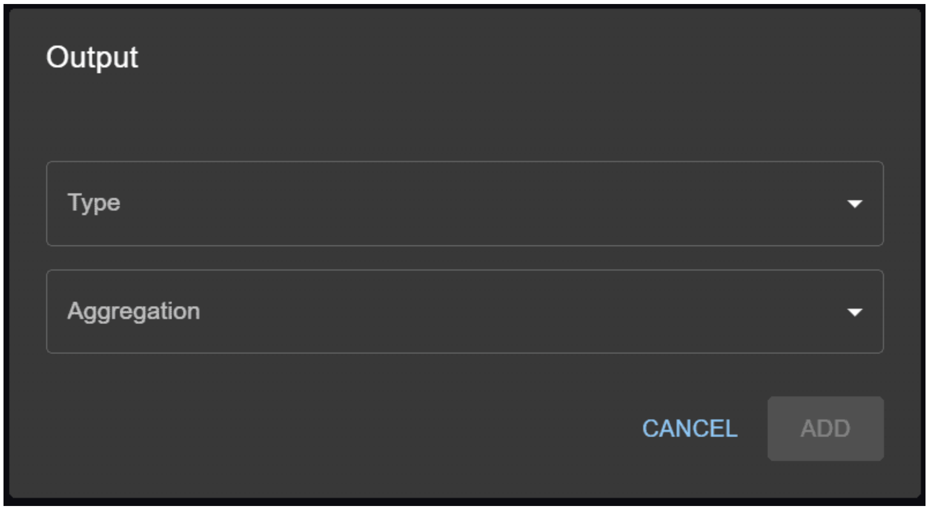


Figure 3.17: A pop-up dialog prompt that asks the user to input the required output details

visual query, it will be added to the query list component on the top-left of the map application, and after tapping on it, the query will be executed, and as queried, the cargo vessels during that time will be shown on the app (Fig 3.20). Moreover, the visual analytics component on the map will be interactively changed and show the brief analytics of the vessels that are shown on the map.

In the following paragraph, we will present a detailed query to highlight the extensive functionalities of the visual query builder interface. For instance, as shown in Figure 3.21, the query involves cargo vessels under specific conditions: 1) located within a red marked area and outside a green polygonal region 2) moving at speeds greater than 1 mph. This is alongside tanker vessels that meet the following criteria: 1) situated within the red zone and beyond the boundaries of the polygonal area 2) navigating with a course over ground ranging between 100 and 200 degrees. Additionally, the query incorporates a time-based filter, requiring that both types of vessels fall within the timeframe from ”01/01/2021 04:00:00” to ”01/02/2021 03:50:00”. It’s important to note that the specified areas are custom shapes drawn on the map application, demonstrating their applicability in our example query.

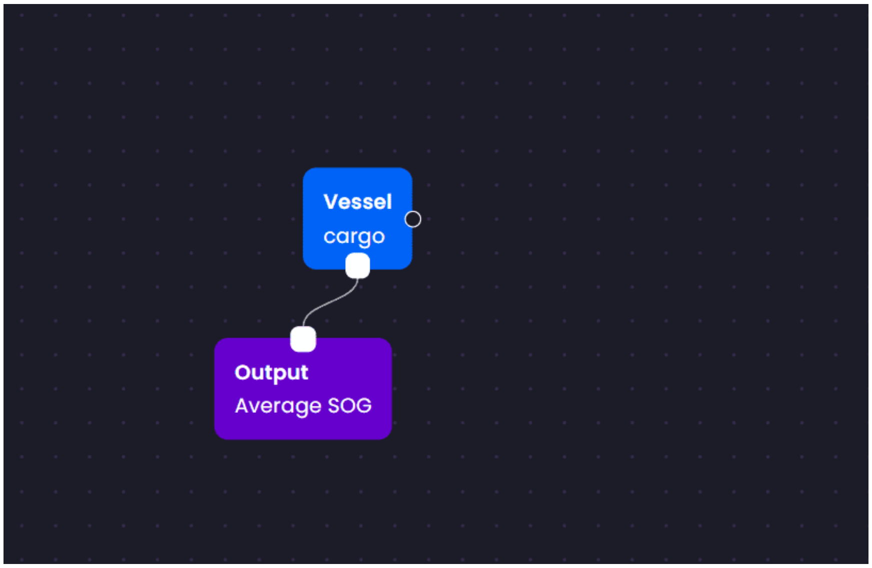
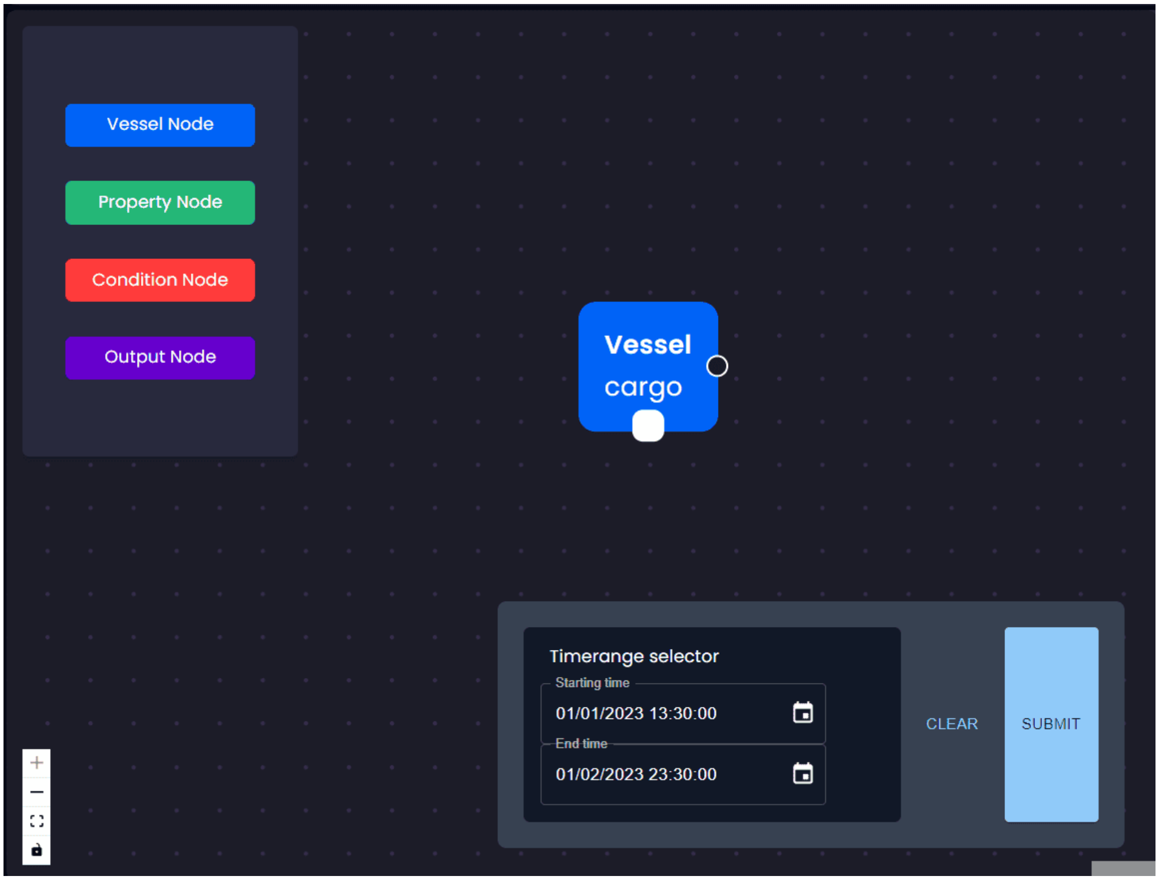


Figure 3.18: An example of the output node usage, querying average SOG of the cargo vessels

3.6 Query Parsing and Execution Process

As mentioned earlier, The AIS data that we have is stored on a MongoDB dataset. In this section, we will elaborate upon the process of parsing the visual query that is built by the user and converting it to a query that can be executed in the MongoDB database. In the visual query builder interface application, we are using two states, nodes, and edges (a.k.a. links). These are arrays of nodes and links between nodes. By interpreting these two arrays of object we are able to transform the query to a format that is readable by our database. Each node inside the array of nodes has an ID, type, and other values filled by the user. Each edge in the edge list, on the other hand, has an ID, outputNodeID, and inputNodeID. These parameters declare connections between nodes. By connecting two nodes together and linking them with an edge, a new edge object will be generated and added to the edges state of the interface, indicating the connection between these nodes. The edges state which is an array of edges, is responsible to record all the information of edges between nodes.

Our approach to parsing visual queries emphasizes a recursive methodology. Initially, upon submission of the visual query, the interface dispatches a JSON object to the server, comprising three key components: nodes, edges, and the temporal filtering



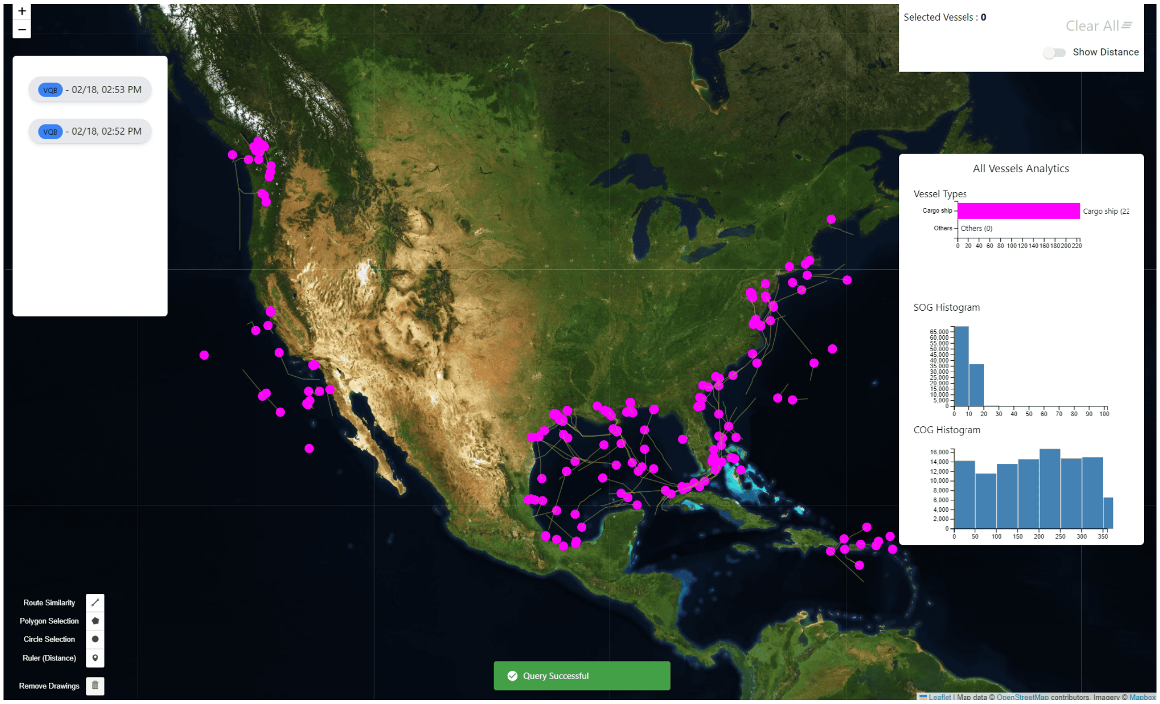
Starting time
01/01/2023 13:30:00

End time
01/02/2023 23:30:00

Figure 3.19: Formulating a simple query: querying all cargo vessels during "2023-01-01 13:30:00" to "2023-01-02 23:30:00"

value (starting date and time and ending date and time). The server employs a recursive strategy to navigate through the nodes, specifically targeting vessel nodes. It utilizes the connections defined by the edges to identify all property nodes linked to a given vessel node. For each discovered property node, the server recursively examines the edges list to locate all condition nodes associated with that property node. It then extracts the specific condition value from each node, and depending on the type of condition, will construct a query in a format interpretable by the database. For instance, a condition node specifying "greater than" with a value of "20" would be translated into a database query structure like {"$gt": 20}.

The property parsing process aggregates all conditions for a property, incorporating the property itself into the generated conditions. An example output from the property parser might resemble:



Selected Vessels : **0**

Query Successful

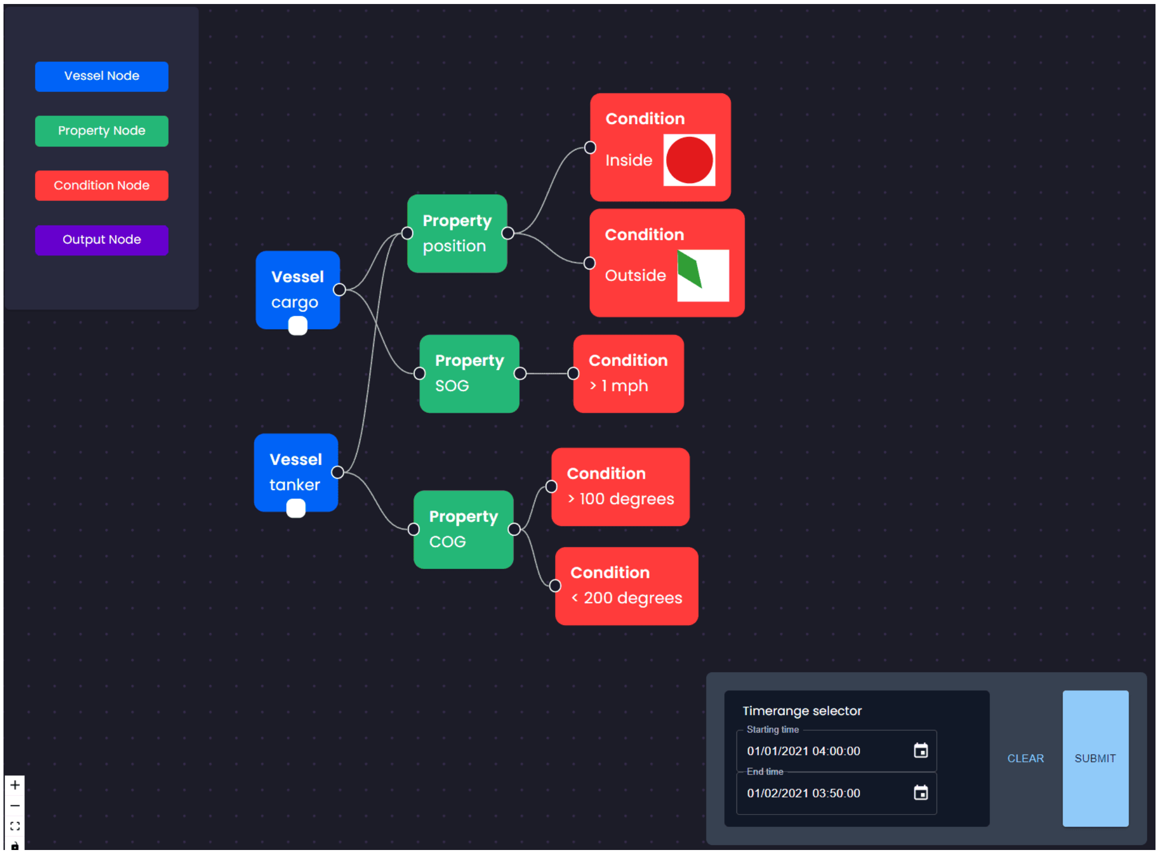
Figure 3.20: Executing the query and visualizing the result of it on the map visualization application

Listing 3.1: Example JSON Query

```
{
  ”tracks.sog”: {”$gt”: 2},
  ”tracks.cog”: {”$lt”: 200}
}
```

This process extends to parsing the properties and conditions specific to a vessel, where the vessel node parser enriches the output by adding a vesselType attribute. This addition is derived from a predefined dictionary mapping vessel type names to their numeric representations. Consequently, the output from the vessel node parser mirrors the structure previously mentioned, augmented with a vesselType: <number> attribute, where <number> is the numeric value corresponding to the vessel type. To clarify, in AIS data, each vessel type is mapped to its corresponding arbitrary number. For example, vessel type 30 indicates a fishing type vessel.

In the final step, it then combines all of the collected output objects into a single query structure using the {”$or”: [outputs]} format, just as described earlier. This approach ensures thorough and step-by-step processing from visual to database



Timerange selector

Starting time: 01/01/2021 04:00:00

End time: 01/02/2021 03:50:00

Figure 3.21: Visualizing the formulation of a more complex query

query conversion, accommodating complex queries with ease and precision. The interpretable query is ready at this point, however, the temporal filtering is not applied. For this purpose, we utilized MongoDB's aggregation pipeline feature [1]. Initially, we matched the vessel tracks according to the query generated through our recursive function. Then, we employed the $unwind operation on the vessels' $tracks property. Subsequently, we implemented a $match operation to apply the temporal constraints. The final step involved grouping the vessels by their ID, which guaranteed that the output format remained consistent with the pre-unwinding structure. The pipeline structure can be shown as follows:

Listing 3.2: The Aggregation Pipeline Used For Querying

```
pipeline = [
        // Match the initial query
        {
```

```
    $match: parsedQuery,
  },
  // Unwind the tracks array
  {
    $unwind: ”$tracks”,
  },
  {
      $match: {
        ”tracks.time”: {
          $gte: temporalFiltering[0] / 1000,
          $lte: temporalFiltering[1] / 1000,
        },
      },
  },
  // Group back if needed, and shape the output structure
  {
    $group: {
      _id: ”$_id”,
      mmsi: { $first: ”$mmsi” },
      vesselName: { $first: ”$vesselName” },
      imo: { $first: ”$imo” },
      vesselType: { $first: ”$vesselType” },
      size: { $first: ”$size” },
      tracks: { $push: ”$tracks” }, // Grouping back the tracks
    },
  },
];
```

After executing the pipeline, it would return the final result of the query as an array of AIS vessels with their tracks populated. This result will be used in the map application for further visualization.

3.7 Cognitive Walkthrough

The cognitive walkthrough (CW) is a task-based usability-inspection method that employs a cross-functional team of reviewers walking through each action within a task flow and replying to a small set of predetermined questions at each step. Its function is to identify characteristics of the interface that could be demanding to new users, and/or pose unnecessary cognitive demands [6]. The questions explore the human cognitive cycle when interacting with the system. This section discusses the methodology and results that we have taken away from conducting two CW sessions.

3.7.1 Methodology

While variations of the CW questions exist, in our work, we pose the following questions at each step:

- ”Is the effect of the action the same as the user’s goal?”
- ”Is the action visible to the user?”
- ”Will the user recognize the action as the correct one?”
- ”Will the user notice the feedback after the action is performed?”

Evaluators can answer ”Yes”, ”Yes, but”, ”Maybe”, or ”No” to each question for each action within a task. When the response is not ”Yes”, they may also provide a recommendation for addressing any related issue. We conducted two iterations of a cognitive walkthrough. The first iteration involved three independent walkthroughs: one from a novice user with no prior experience using tabletop displays or geospatial analysis software, one from a novice user with user interface design expertise, and one from an experienced user with interface design expertise. The feedback obtained from these walkthroughs was used to revise aspects of the interface (detailed below). In the second iteration, two cognitive walkthroughs were conducted on the revised interface. The first was conducted by a pair with domain expertise and experience using geospatial analysis tools. The second involved a pair with similar expertise as the first pair, plus usability expertise and experience with tabletop systems.

After an introductory session to the CW method and a brief explanation of the interface components, walkthroughs included tasks that employed direct manipulation and visual queries (see figure 3.23), each involving several discrete actions, as follows:

- **T1** Select the marked vessel using the *Circle* selection tool (4 actions).
- **T2** Obtain the vessel's Speed Over Ground at 16:00 (3 actions).
- **T3** Select all vessels in the red region using the *Polygon* selection tool (5 actions).
- **T4** Reset the state of the interface (2 actions).
- **T5** Find the distance between the two marked vessels at 20:00 using the *Ruler* tool (5 actions).
- **T6** Find vessels with a specific route shape using the *Query by Sketch* tool (4 actions).
- **T7** Create a *visual query* to identify fishing vessels located within the preselected map region (7 actions).
- **T8** Add a weight restriction to the query to identify only those fishing vessels with a load greater than 10,000kg (8 actions).

Each walkthrough took 30-45 minutes to complete. In addition to question responses, we captured observer notes. We focused on the tabletop portions of the interface for the walkthroughs. We plan to conduct walkthroughs with the full tabletop+AR interface in the future.

3.7.2 First Iteration

The first iteration of the cognitive walkthrough yielded 38/348 question responses that were not unqualified 'Yes'. While this high number of Yes responses is encouraging, several usability issues surfaced during this iteration. For example, two evaluators found the 'Circle Selection' tool's operation difficult (T1). The circle expanded by radius and not diameter, and grew more quickly than they expected. One suggested

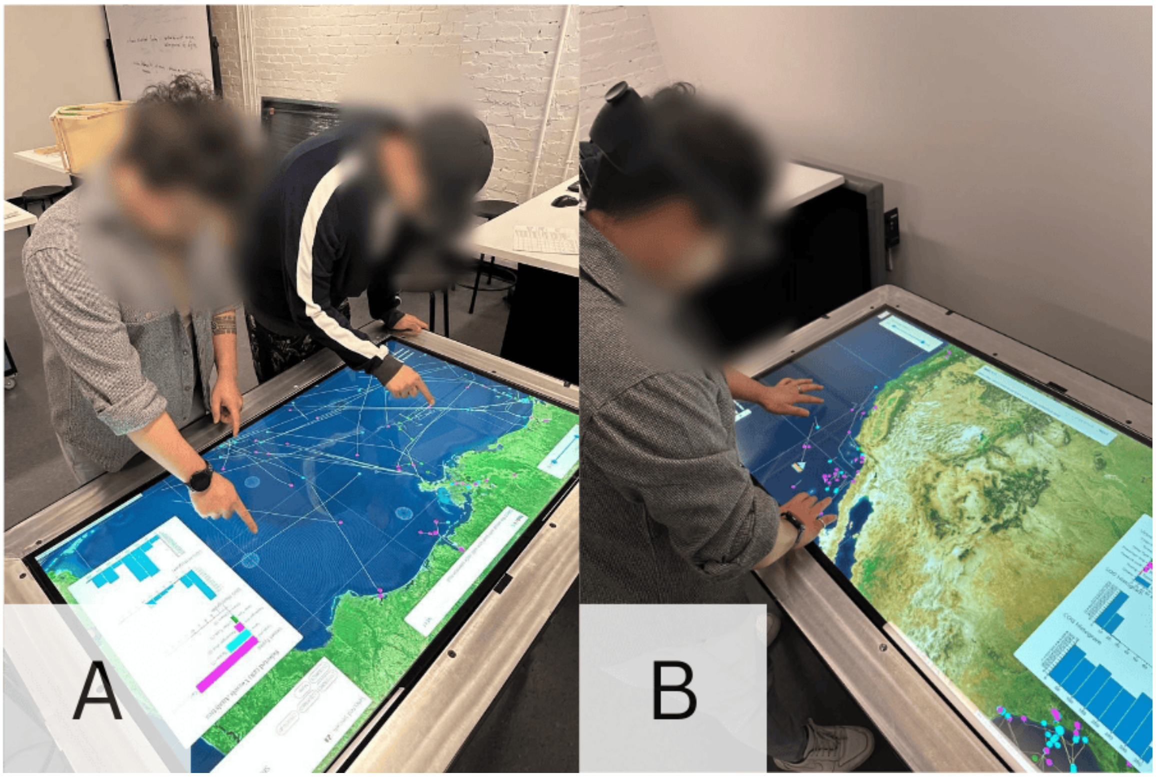


Figure 3.22: In (A) users are collaboratively interacting with the tabletop and (B), a user wears a HoloLens to view AR content while interacting with the tabletop display.

that the action wasn't apparent to users. Feedback mechanisms also needed enhancement, particularly after selecting a vessel: details about selected vessels appeared in dialogue boxes, but the vessel itself was not highlighted (T1, T2). Evaluators recommended adding instructional pop-ups for tasks such as enabling 'Polygon Selection' (T3), as well as more visible feedback for actions like deselecting vessels (T4). Several tasks were found to require action sequences that could be reduced or made more straightforward: in particular, finding distances between vessels required two 3-step selection dialogues and the use of a difficult-to-locate distance button (T5).

In response to the feedback from the first iteration, several enhancements were made to the interface. A notification system was introduced to provide immediate feedback to users after each interaction. An instruction system was added to clarify the state of certain tools and actions. Labeling was improved for better understanding, and a distance tool was integrated, as suggested by the evaluators.

3.7.3 Second Iteration

The second iteration involved two pairs of domain experts who evaluated the interface collaboratively. 24/320 question responses were other than unqualified 'Yes'.

Observation notes indicate that both pairs navigated the interface with little difficulty, but some friction was encountered in places. Some issues uncovered in the first iteration but not addressed in revisions were identified again in the second iteration, including the radius-based method of drawing a circle (T1) and the decoupling of a selected set of vessels and the encompassing shape (circle or polygon), which meant that both the shape and the selected set needed to be removed when clearing prior work in T4.

Both expert pairs faced challenges building compound queries via the AND node (T8). One pair expected the AND node to be used in the order of spoken language (i.e., $Q1 \rightarrow AND \rightarrow Q2$), while the interface required a connection of two completed sub-queries to the AND node. The other pair assumed that placing two subqueries in the same workspace implied an AND relation between them. Further issues with action ordering were encountered with the query builder. A relational operator (">" in T8) is associated with its operands (e.g., an attribute like ship weight or a value like 10,000kg) by dragging from the operand node to the relational node: one pair tried to go from the operator to the attribute.

3.7.4 Summary of Findings and Improvements

The CW method offered valuable insights into the usability of our interface. It identified areas of potential improvement, thus aiding the iterative development of the system. Despite the issues encountered, expert users found the system straightforward to pick up and use, completing all tasks with little or no instruction. This is partly due to the interface's similarity to other geospatial visualization tools they were familiar with. Broadly, evaluators successfully used direct manipulation techniques for selection, information retrieval, and query formulation, completing related tasks as expected. The visual query interface was also used successfully; however, there were several points where the mental model of the evaluators did not match the designer's model.

Future developments will concentrate on refining the interface, with an emphasis on enhancing visual cues and aligning system behaviour with user expectations. Using the insights from the CWs, we have introduced several enhancements, such as implementing a responsive notification system, an interactive instruction section (such as hint pop-ups), improved interface labeling, enhanced visualization of the drawing layer post-selection, and a simplified drawing deletion process.

3.8 Summary of the Chapter

This section describes the development and assessment of the Visual Query Builder (VQB) System, a tool designed to improve exploration and operations by visualizing and managing AIS data. The system's structure includes an API server, MongoDB dataset, React visualization application, visual query builder, tabletop interaction, and augmented reality features to deliver an engaging user experience. Emphasizing data retrieval, processing, storage, and visualization ensures effectiveness, scalability, and user-friendliness. The chapter delves into each component's functionalities, data flow process, scalability considerations, and the detailed steps involved in collecting, refining, and querying data. Additionally, it covers the development approach guided by cognitive walkthroughs that led to usability improvements based on user feedback. The incorporation of direct manipulation methods visual queries and augmented reality elements demonstrates the system strategy, in enabling users to interact with data.

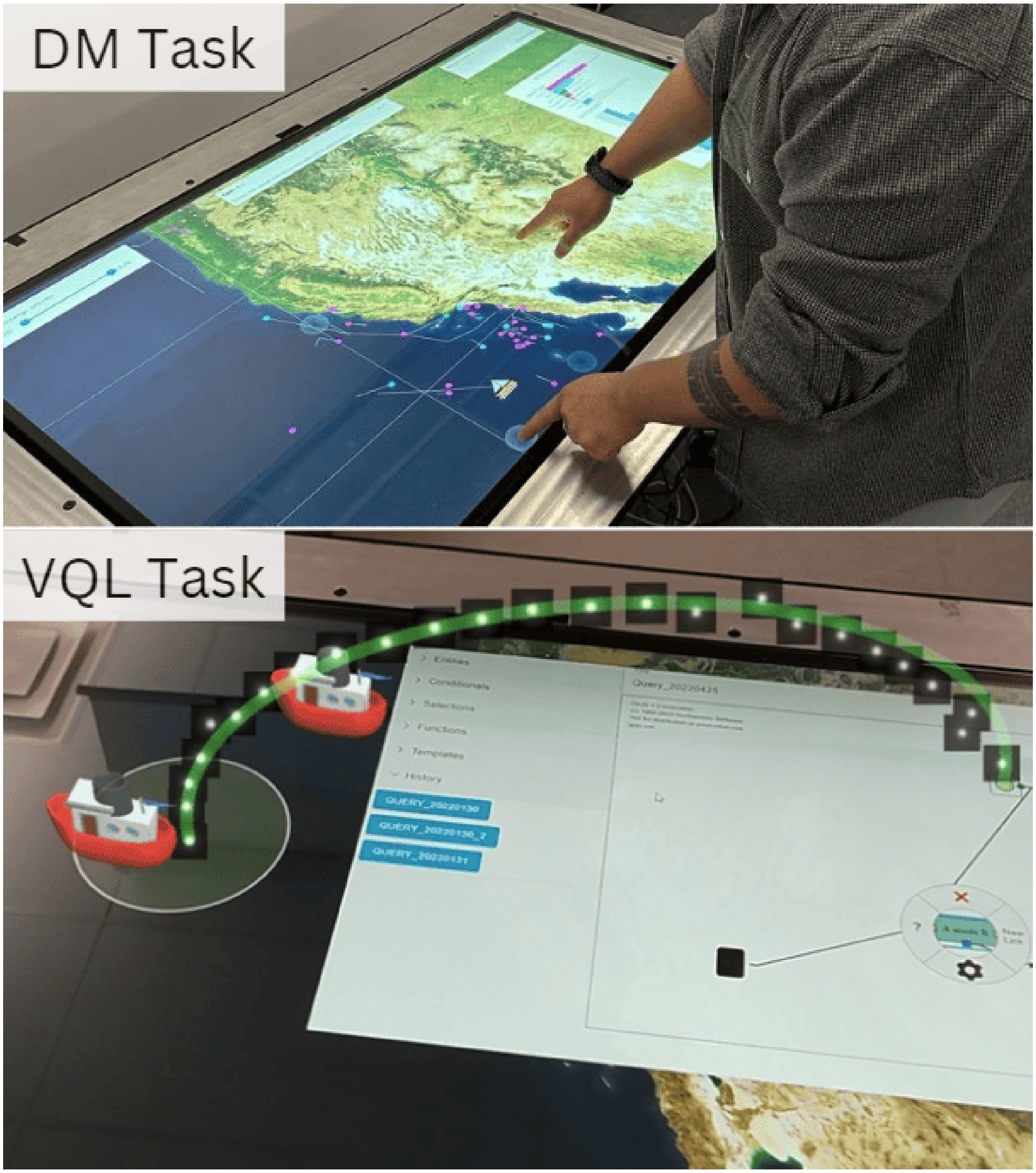


Figure 3.23: Direct Manipulation (DM) and Visual query builder (VQB) Tasks. In the VQB task image, the first-person vision of augmented reality interaction over the shared display showing the AR guide to Visual query builder (VQB) shape selection.

Chapter 4

Augmented Reality Exploration

4.1 Introduction

As described in the literature review chapter, utilizing augmented reality devices in the maritime context has grown over recent years. AR provides a wider spatial view compared to traditional interfaces that are limited, and also an immersive feeling of experience compared to traditional interfaces such as radars and tabletops. Moreover, since AR devices can detect hand motions and gestures, they offer some 3D interaction tools such as sliders and buttons, etc. Given the advantages AR offers, from enhanced spatial perception to immersive interactions, we were inspired to integrate AR into our project and experiment with its distinct features.

In this project, we used AR to augment maritime data elements on top of the tabletop in an interactive way that supports collaboration between multiple AR clients. In the AR tool we developed, we employed the third dimension (vertical axis) to represent time, drawing inspiration from the space-time cube (STC) approach discussed in the literature review chapter. Each vessel, represented by a 3D ship model, has its route visualized using a 3D cylinder. This cylinder encapsulates the vessel's track position and extends vertically by the STC approach. Consequently, older tracks are positioned higher within the cylinder.

4.2 AR Integration with Tabletop

To integrate the tabletop visualization application with the AR clients, an initial environmental sync is required to inform the AR device about the spatial position, orientation, and size of the tabletop. A prevalent method for this is using a QR code, scanned by the AR device. We've placed a printed QR marker at the center of the table. In the AR application, we've coded it to scan this QR code and then align the AR visual information based on the QR marker's orientation and height. Figure

4.1 shows this. integration of the tabletop with the environment using the QR code scanning process.

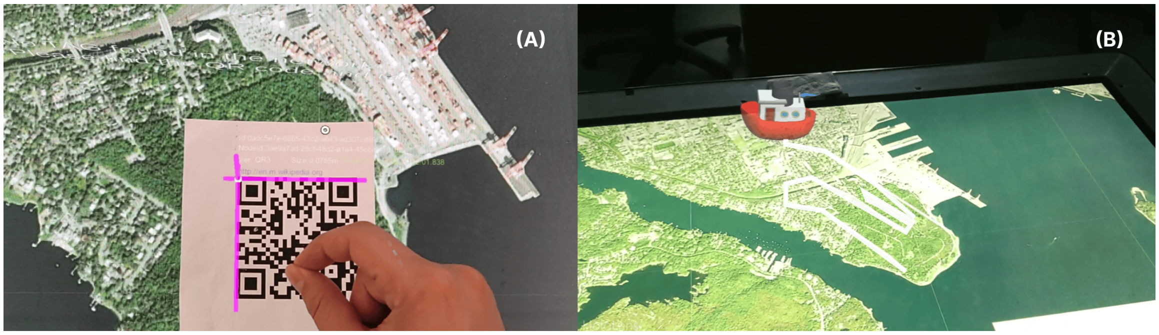


Figure 4.1: A: Pinch gesture initiates scanning of the QR code, B: A vessel and its route displayed in AR after synchronizing with the tabletop environment.

4.3 Synchronization of Augmented Visuals

In order to provide a collaborative platform for maritime data analysis, the provided platform needs to be real-time and has low latency to ensure a smooth user experience, and more importantly, multiple clients collaborating in the same environment should be synced. To synchronize visual elements in AR, we need to provide two main synchronizations:

- **AR and tabletop synchronization**: Any activity of users on the tabletop that has an effect on AR visual elements should be delivered to the AR clients. For example, when a user changes the map view, the visual elements on the AR clients should be relocated to their new position, Or when a user selects a route on the visualization application, the visual route on the AR should be highlighted as well. To provide this synchronization, we have considered a socket broadcast server that receives changes in types of key, and value pairs in a JSON format. The key indicates that what is the type of change that can be either of these: ”vesselSelect”, ”routeSelect”, ”mapViewChange”, ”VQBViewChange”, or ”connect”. After receiving it, it propagates this to all the connected AR clients in order to apply the changes visually. The reason behind choosing the socket network is to ensure a smooth user experience with quick data exchanges due to its speed and efficiency.

- **AR clients synchronization**: Besides synchronizing the tabletop application with AR clients, it's essential to maintain consistency across all AR scenes to facilitate collaborative use. To achieve this, we have decided to use the Unity Photon PUN[1] (PUN stands for Photon Unity Networking) networking tool. Photon PUN provides real-time synchronization across multiple devices, ensuring that all users see the same AR scene simultaneously.

4.4 Relation Indicator Arches

As mentioned earlier, the visual query builder is integrated with the map visualization view in a way that if a user selects an area of interest on the map, it will be added to the visual query builder's node palette. Moreover, the user can drag and drop that node to create instances on their VQB workspace. The node itself has the thumbnail of the selection with the same color on the map and the same shape to convey the relation of it to the original source on the map. However, if a user draws multiple similar shapes on the map as their areas of interest, it would be hard to tell which node in the VQB is related to which source shape. To address this issue, we have considered using AR as an assistive visualization tool to connect each area selection node on the workspace to its related source on the map in order to indicate the relation between them. Figure 4.2 shows the aforementioned arch between a node instance in the VQB workspace and its source.

The AR arch use cases are not restricted to the VQB selection node element and could be extended for relation indication of each two elements on the tabletop visualization, between two AR elements, or even the relation between any AR element and their instance on the tabletop. In the described scenario where an arch connects the VQB selection element to its source on the map, we begin by translating the shape's center position from its geospatial coordinates (latitude, longitude) into pixel coordinates on the tabletop. We then sum the relative pixel position of the selection node instance in the workspace panel with the panel's absolute pixel position (refer to Figure 4.3). Having determined the pixel positions of both elements we wish to link with an arch, it's essential to transform these tabletop positions into spatial

[1] Photon PUN networking website: https://www.photonengine.com/pun

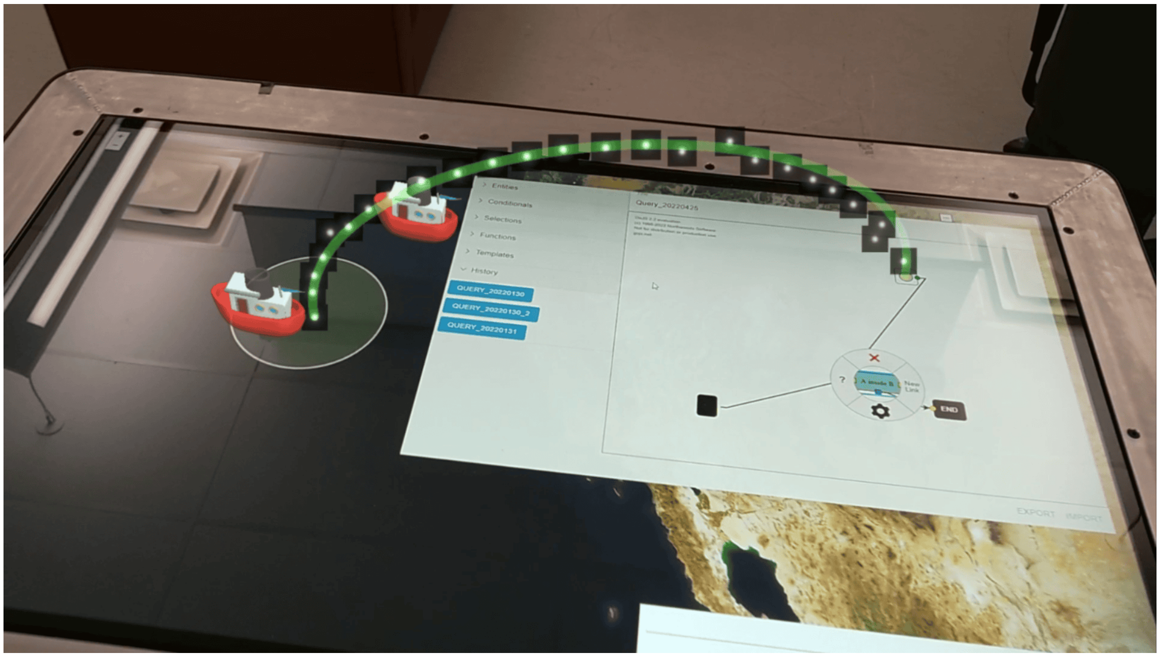


Figure 4.2: Relation indicator arch: the arch visualization in AR conveys the relation between an element in VQB and its source on the map

coordinates within the AR application. Consequently, we either instantiate a new arch or update the start and end positions of an existing one.

4.5 Time Range Sliders in AR

The AIS data is a spatiotemporal type of data, and as mentioned earlier, we have opted to explore the visualization of the vessels' routes using the space-time cube approach, in which the surface of the table that includes X and Z axes indicates the spatial variable and the Y axis indicates the time of each detection of AIS data. Figure 4.5 is a screenshot captured from the Unity simulation mode that shows the cubes that if we join them with a 3D line, the route of the vessel will be formed.

For highlighting a part of a route in the augmented reality client, we have used the sliders library from the HoloLens mixed reality toolkit (MRTK[2]), in a way that we combine two different sliders while removing the bar of one of them, to offer a double button slider that allows users to select the start and end time of the route portion that they want to analyze. Moreover, another slider has been dedicated to allow users

[2] Mixed Reality Tool-Kit official website: https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/

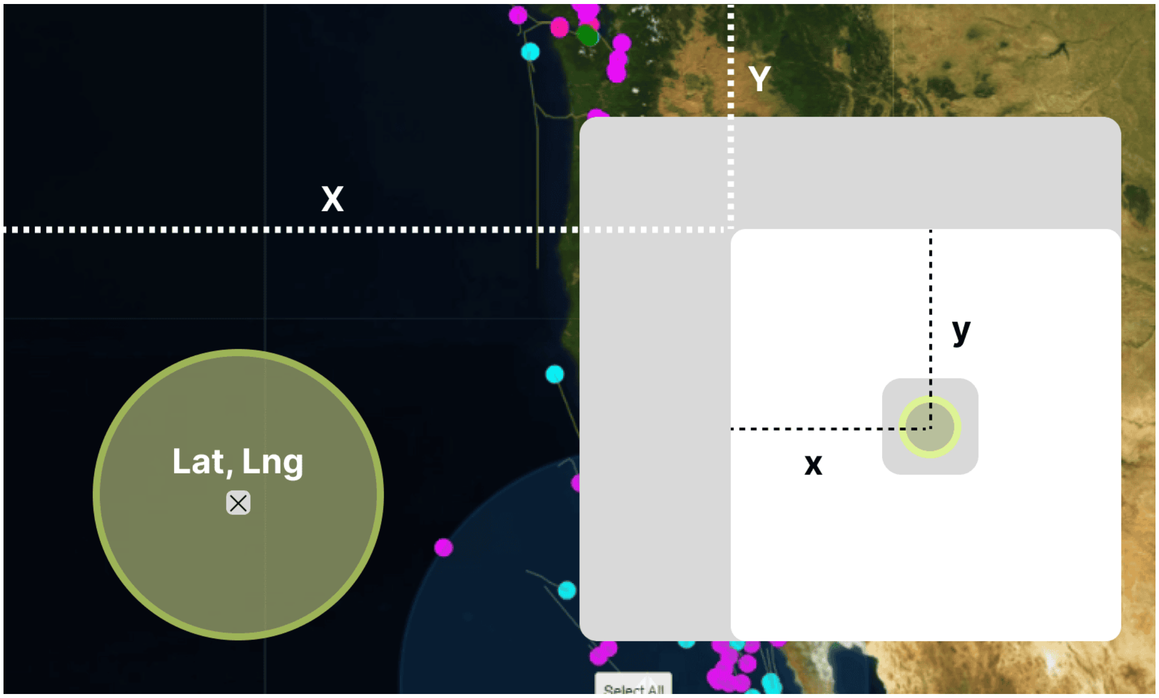


Figure 4.3: Determining the arch's initial and terminal points: X, Y represent the absolute position of the workspace relative to the tabletop's origin; x, y indicate the selection node's relative position within the workspace; lat,lng denote the geospatial coordinates of the center of the selection (highlighted as the yellow circle in this figure).

to select the current time of analysis, and after selecting the time, the vessel objects will be re-positioned at the currently selected time. Along with that slider, there is a 3D button provided to save a snapshot of the vessel object at that time. Figure 4.4 shows multiple snapshots of a vessel. The decision behind implementing this feature is to allow the user to get better insight into the comparison of vessel positions at multiple timestamps.

4.6 Challenges, Exploration Outcomes and Insights

The main challenge that we encountered while exploring augmented reality - as an assistive interactive visualization tool in a maritime context - was dealing with a large amount of data entries, and subsequently, getting stutters while using the visualization. The reason behind this issue is that after each interaction of a user that involves alteration in map view or visual query builder elements, calculation of the

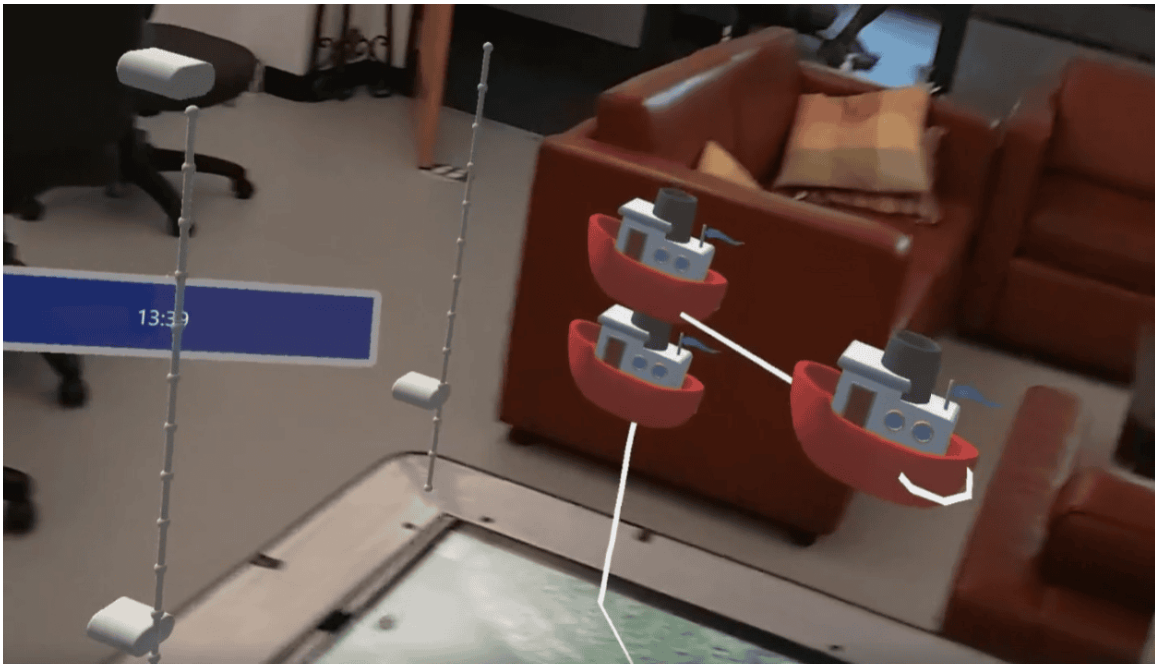


Figure 4.4: Snapshots of a vessel at different timestamps: This feature facilitates users in comparing vessel positions across various moments, deepening their analytical perspective. As depicted, three distinct snapshots capture the vessel's positions at separate times.

new positions of the augmented visual elements and re-rendering them is needed, and it causes a huge amount of processing by the Hololens device eventually causing a throttled and stuttered user experience. This issue amplifies whenever the user view contains many vessel items and the vessel routes contain many detection points due to not filtering the time range properly by the user.

In addition to a disrupted user experience when visualizing extensive data, another notable challenge is the potential for users to feel overwhelmed by an excessive amount of augmented visuals. Moreover, by visualizing large amounts of data, some visual elements could intersect and cause confusion. Another suggestion is to use data aggregation prior to the augmentation. For instance, multiple routes could be synthesized into a singular path, where the line's thickness indicates the frequency of occurrences. Alternatively, colors could be employed to represent the depth and prevalence of tracks along a route. The study [35], previously discussed in the literature review chapter, offers a prime example of maritime data aggregation in two-dimensional visualization, which could potentially be adapted effectively for augmented reality visualization.

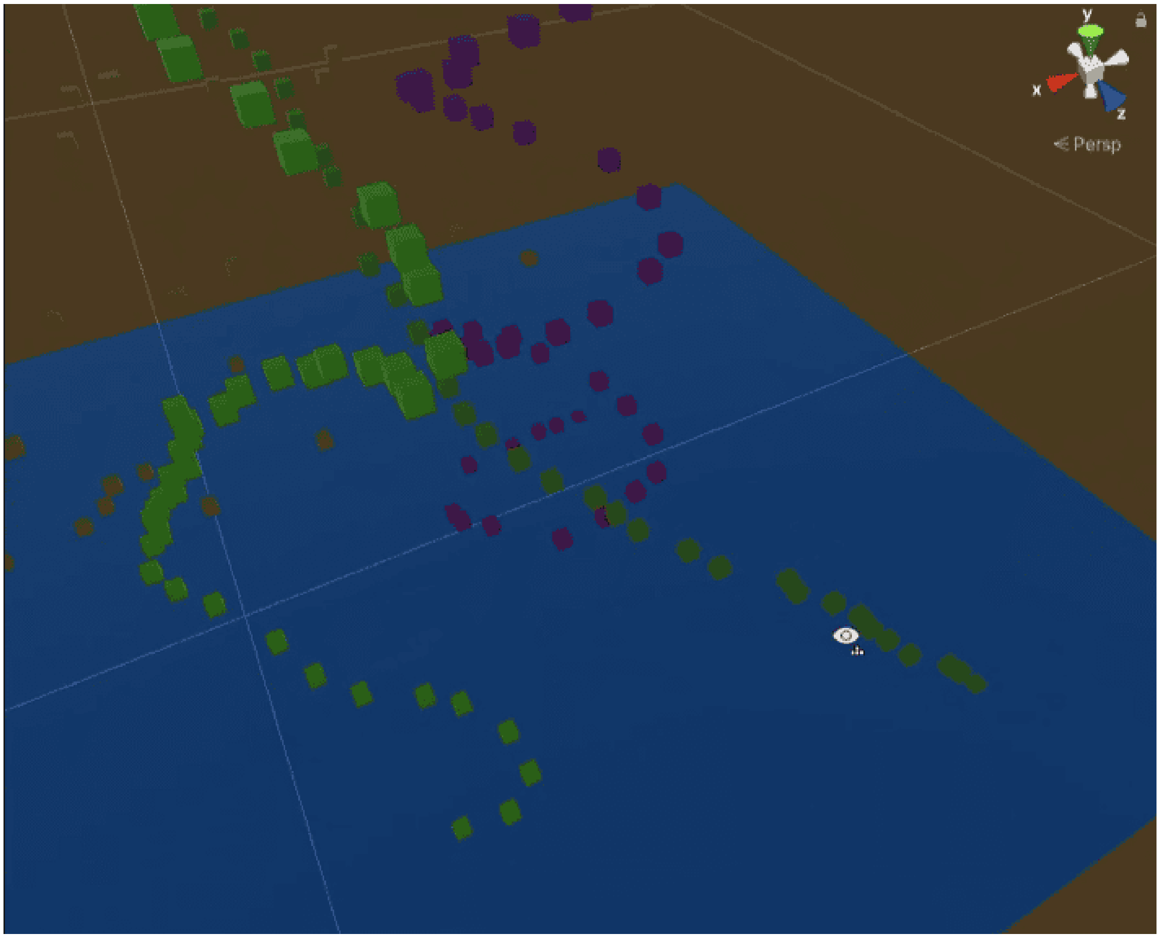


Figure 4.5: Visualizing vessels' routes using the STC approach in the Unity simulation: Cubes of identical colors represent the detections of a specific vessel. By linking these detection points, a route is established.

As an outcome of this exploration, we could infer that the best practice of implementation of an assistive AR tool for the maritime context, the proper spatial and temporal filtering would enhance the efficiency and processing time and subsequently ensure a steady and uninterrupted user experience. In addition to temporal and spatial filtering of data, it would be beneficial to let users select a handful of vessels for visual element augmentation purposes using AR. The good practice of this technique can be seen in the GeoGate [52] research they have used a tangible ring to filter the smaller portion of the view prior to augmentation.

4.7 Future Work

The utilization of augmented reality in collaborative environments has seen notable growth in recent years, making it a compelling avenue of exploration. AR presents an immersive and interactive environment, especially conducive for collaboration in visualizing spatiotemporal data, such as in the maritime context. However, due to time constraints, this thesis primarily concentrated on visual querying, leaving some intriguing aspects of AR unexplored. This section outlines potential concepts and ideas, hoping to guide and inspire future researchers in this domain. Some promising directions include:

- **Augmented panels**: The spatially expansive environment of AR significantly surpasses the limitations of traditional interfaces. This expanded view in AR can accommodate multiple panels. These panels could present cumulative data or be dedicated to visualizing specific data analytics based on user selection.

- **Evaluation of interactive tools effectiveness in the mixed reality environment**: Since the recent augmented reality tools offer various ways of user interactions (such as eye gaze, dragging and dropping, hand gestures, etc.), it provides exploring opportunities by designing direct manipulation techniques for maritime data explorations using those interaction features.

- **Evaluation of Real-time Data Streaming in Mixed Reality**: Comparing mixed reality to traditional interfaces could provide insights into the efficacy of using mixed reality environments for surveillance contexts that handle live data, versus relying on conventional display methods.

- **Spatial awareness evaluation**: Conducting a study to evaluate how maritime users perceive and interpret spatially-represented data in a mixed reality environment compared to a 2D interface could be another insightful exploration in the maritime context.

Chapter 5

Methodology for Comparative Study

5.1 Research Design

This research project conducts a comparative study of two interfaces - Visual Query Builder (VQB), and SPARQL textual query interface. We have implemented these interfaces on an interactive tabletop, providing users with a shared platform for data interaction and query execution. For the VQB interface, we have utilized direct manipulation techniques to ease the querying process. Additionally, in the SPARQL editor interface, we have implemented an auto-completion feature and provided maritime query templates and features to have a fair comparison between the two interfaces. By "fair comparison," we mean that we wanted to explore research questions without the ergonomic issues typically associated with textual editors like SPARQL. For instance, defining a polygonal shape area in SPARQL requires users to type every vertex in latitude and longitude, which can be frustrating and time-consuming. Such complexities could hinder our ability to answer our research questions effectively. Each interface and its features will be elaborated upon later in the following sections.

The study aimed to evaluate the performance, accuracy, and user preference for each interface in the context of maritime operations. Our hypothesis suggests that certain interfaces may outperform others for specific types of queries. Moreover, we hypothesize that user preference will be dictated not only by the query type but also by the individual user's comfort level and familiarity with each interface.

This research conducts a user study featuring tasks related to common query types in the maritime context, such as temporal, attribute-based, spatial, and complex queries. Each task is designed to evaluate a specific interface or a combination of interfaces, with performance and accuracy calculated based on the success rate and time taken to complete the tasks. User preferences are determined through self-reported measures and a final task during which participants can freely choose their preferred interface.

We have recruited participants who possess basic computer literacy for this study. Although the participants represent varying levels of experience with different query interfaces, all were expected to comprehend the fundamental concept of data querying. The data collected from participants has been anonymized and consists of interaction logs (such as touch inputs and drag-and-drop actions), timestamps, and performance metrics.

5.2 Interfaces for Comparison (VQB, SPARQL)

As mentioned earlier, the two interfaces that have been considered for this study are VQB and SPARQL textual editor. The overall querying platform consists of an interface a component to view the switch between the tasks, and a component that allows switching between interfaces. Hereunder, each interface, and its components will be elaborated upon.

5.2.1 VQB

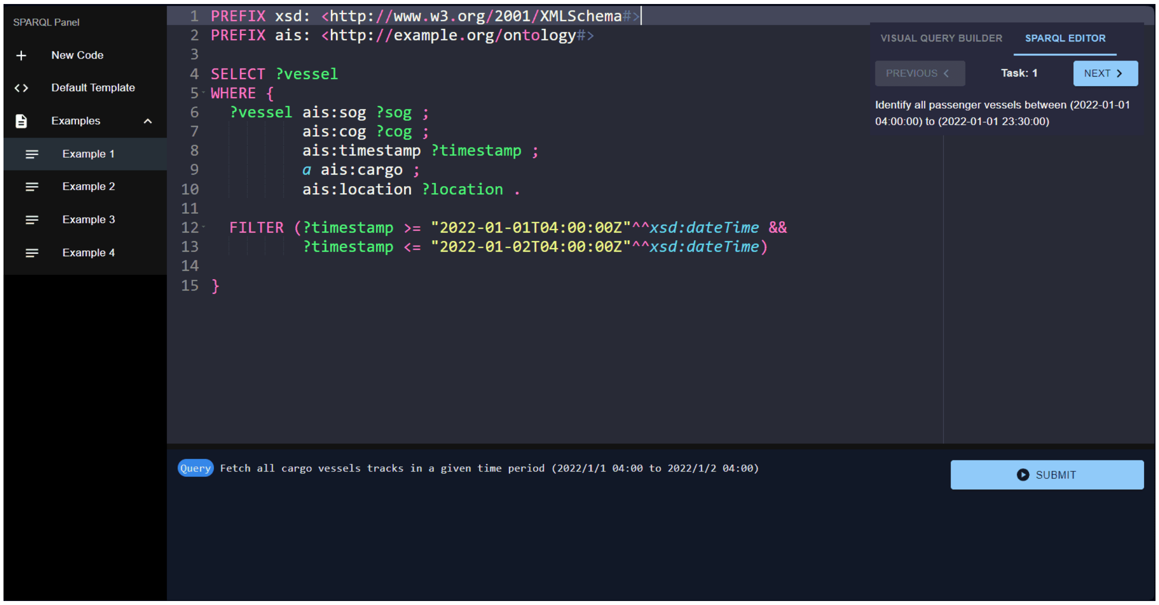
We have utilized the latest version of the visual query builder interface for this study. Chapter 3 demonstrates the latest state of the VQB and the implementation details and considerations.

5.2.2 SPARQL Editor

As mentioned in the research design, in this study we are going to compare the implemented visual query builder interface with traditional query interfaces such as textual editors. Therefore, as the study requires, we implemented a web-based textual editor that supports the SPARQL language. Following this section, the implementation of the SPARQL textual editor will be discussed. Figure 5.1 shows the implemented state of the textual SPARQL editor.

SPARQL Editor Implementation and Considerations

The SPARQL query editor consists of a navigation panel, a textual editor workspace, and a logger. The navigation panel allows users to navigate through multiple instances of the editor while offering some pre-defined example SPARQL queries as tabs, so



```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ais: <http://example.org/ontology#>

SELECT ?vessel
WHERE {
  ?vessel ais:sog ?sog ;
          ais:cog ?cog ;
          ais:timestamp ?timestamp ;
          a ais:cargo ;
          ais:location ?location .

  FILTER (?timestamp >= "2022-01-01T04:00:00Z"^^xsd:dateTime &&
          ?timestamp <= "2022-01-02T04:00:00Z"^^xsd:dateTime)

}
```

Identify all passenger vessels between (2022-01-01 04:00:00) to (2022-01-01 23:30:00)

Query Fetch all cargo vessels tracks in a given time period (2022/1/1 04:00 to 2022/1/2 04:00)

Figure 5.1: SPARQL Editor Interface

participants can view the examples during the study if needed. The textual editable area is simply for writing the SPARQL query. For this component, the AceEditor[1] library has been used. This library allows us to implement a web-based editor and handpick the features that we want to have in our study. Some of the features that this library offers and we have used in our interface are supporting the SPARQL query language, auto-completion, custom keyword suggestions, inline error detection, and theme customization. The logger component displays the syntax errors related to the SPARQL code, either given by the AceEditor library or errors after execution of the script, as the parser's output.

To have a fair comparison between the interfaces in the study, participants were given a default template. This approach was taken to avoid the need for users to spend excess time writing repetitive scripts that would be common across tasks. Listing 5.1 represents the default SPARQL code snippet that is provided for the participants.

Listing 5.1: The Default SPARQL Template

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ais: <http://example.org/ontology#>
```

[1] AceEditor Website URL: https://ace.c9.io

```
SELECT ?vessel
WHERE {
  ?vessel ais:sog ?sog ;
          ais:cog ?cog ;
          ais:timestamp ?timestamp ;
          ais:location ?location .

  FILTER (?timestamp >= ”YYYY—MM—DDThh:mm:ssZ”^^xsd:dateTime &&
          ?timestamp <= ”YYYY—MM—DDThh:mm:ssZ”^^xsd:dateTime)

}
```

In SPARQL, for defining spatial filtering, users have to define areas in GeoJSON format. For the circular areas, they have to define the center of the circle with latitude and longitude and the radius of the circle. To define a polygonal area, users have to add multiple points of the polygon in latitude and longitude. As it seems, it might get much more complex. To reduce its complexity, another feature has been implemented by defining custom spatial functions that allow users to use IDs of the areas of their interest instead of defining GeoJson areas. The functions that have been implemented are the inside and outside functions which receive the ID of any area drawn on the map application and can be used inside a FILTER.

Parsing, Compilation, and Execution

Similar to parsing queries generated by the visual query builder, queries that have been made by the SPARQL editor should be parsed and translated to the queries that our database, MongoDB can handle. For this purpose, a Javascript library called SPARQLJS[2] has been used to parse the whole query that the user writes. It generates a JSON output format of the SPARQL code containing information such as query type, an array of used variables, an array of all triples in basic graph pattern format, filters, and prefixes. This output helps us to have all the queries in an object-oriented format rather than a whole SPARQL script string format. Following this section, the methods that we have utilized to translate this JSON output to a query

[2] URL of the SPARQLJS NPM package: https://www.npmjs.com/package/sparqljs

that MongoDB can handle will be discussed.

Just as we have converted VQB queries, we employ a dictionary to map string representations of vessel types to their numeric equivalents; for instance, ”cargo” is mapped to 70. Another mapping dictionary has been used to convert the operator syntax of the SPARQL, to MongoDB-compatible operators; for example, the ”>” operator will be translated to ”$gt” (Listing 5.2). The expressions are handled recursively in this compiling process, similar to the VQB query compiling that has been explained in section 3.6. A function called parseCondition has been used to handle different types of conditions which can be ”bgp”, ”filter”, ”union”, and ”group”. This function, regarding the type of the condition, calls the related parser function by giving the extracted value of that condition as the input of that function. For example, if the condition is ”group”, then each of the expressions inside that group will be compiled separately by feeding them into the parseCondition function, and the result query of them will be gathered and concatenated in a single query object. However, as shown in listing 5.3, if the type of the conditions is ”union”, all the parsed sub-queries will be added to an array, and then will be assigned as the value of an ”$or” operator.

Listing 5.2: Operator Mapping Dictionary

```
const operatorMap = {
    ">": "$gt",
    "<": "$lt",
    ">=": "$gte",
    "<=": "$lte",
    "=": "$eq",
    "!=": "$ne",
    "&&": "$and",
    "||": "$or",
};
```

Listing 5.3: Parsing UNION Conditions

```
if (condition.type === "union") {
  const unionQueries = [];
  for (let unionPattern of condition.patterns) {
    const subQuery = await parseCondition(unionPattern);
    unionQueries.push(subQuery);
  }
  queries.push({ $or: unionQueries });
  } else {
   const parsedCondition = await parseCondition(condition);
  }
}
```

5.3 Selection of Query Tasks

Based on the AIS data structure that we have in our database, there are various ways of querying are possible; querying can be temporal filtering, spatial filtering, vessel property filtering, vessel type filtering, or a combination of these types of queries that require some specific filtering for multiple types of vessels individually. For this study, five querying tasks have been designed that cover a broad range of these methods to thoroughly explore the main aspects of querying. Following are the tasks that have been considered for this study, and the table 5.1 shows the actions required for each task.

- **T1** Identify all *Passenger Vessels* between 1 Jan 2022 04:00 to 1 Jan 2022 23:30.

- **T2** Identify all vessels that were in the provided *Circular Area*, and having SOG>2 mph.

- **T3** Identify all *Cargo Vessels* that were in the provided *Polygonal Area.*

- **T4** Between 1 Jan 2022 04:00 to 1 Jan 2022 23:30, Identify all *Cargo Vessels* having SOG>2 mph and COG<140°, in addition to *Tanker Vessels* inside the provided *Circular Area*, having SOG<10 mph and COG>40°

- **T5** Identify all *Cargo, Passenger, and Tanker Vessels* inside the provided *Circular Area*

Table 5.1: Filtering Actions Required for Querying Tasks

Task	**Property**	**Temporal**	**Spatial**	**Vessel Type**	**Mixed Vessel Types**
T1		✓		✓	
T2	✓		✓		
T3			✓	✓	
T4	✓	✓	✓	✓	✓
T5			✓	✓	✓

Table 5.2: Within-subject counterbalancing

Group	A	B
Tutorial	VQB	SPARQL
	SPARQL	VQB
T1	VQB	SPARQL
	SPARQL	VQB
T2	VQB	SPARQL
	SPARQL	VQB
T3	VQB	SPARQL
	SPARQL	VQB
T4	VQB	SPARQL
	SPARQL	VQB
T5	VQB	SPARQL
	SPARQL	VQB

5.4 Participants and Sampling

For this study, 20 participants recruited, meeting two requirements: possessing a basic knowledge of SQL querying and having a computer science background. The within-subjects counterbalancing method is used for counterbalance. Participants are divided into two groups; Group A and B. Group A will complete the tasks starting with the VQB interface and then proceed to the SPARQL editor. Conversely, Group B is asked to begin the tasks using the SPARQL editor before moving to the VQB interface. The goal of this grouping is to measure the order effect. Table 5.2 shows the order of the interfaces per group for each task.

5.4.1 Demographics

All participants were in the age range of 18 to 35 comprising 16 males and 4 females. Figure 5.2 shows the age distribution, and Figure 5.3 shows the gender distribution.

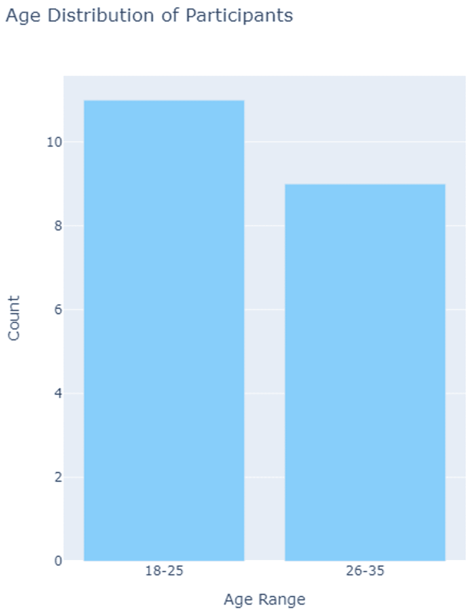


Figure 5.2: Age Distribution

One participant out of 20 was very familiar with GIS systems and maritime operation systems, two of 20 were somewhat familiar and others had no prior familiarity with it (Figure 5.5). However, all of the participants were familiar with database querying and had a basic background in at least one database querying language such as SQL. Participants are also asked to write the database querying and data management tools and interfaces that they had used before in the demographic questionnaire. Figure 5.4 shows the histogram chart displaying the occurrence of a tool among participants.

5.5 Study Procedure

Following this section, the study guideline and procedure from the starting point to the ending point will be explained. After introducing the map application and the querying interfaces briefly, participants were asked to fill out the consent form and the demographic questionnaire to get started. Once they finished filling out the consent form and demographic questionnaire, they were asked to start reading the interface introductions and documentation. As mentioned in table 5.2, the order of the documentation that they asked to read is dependent on their group; participants in group A were asked to read the documentation of the *VQB* before the *SPARQL*

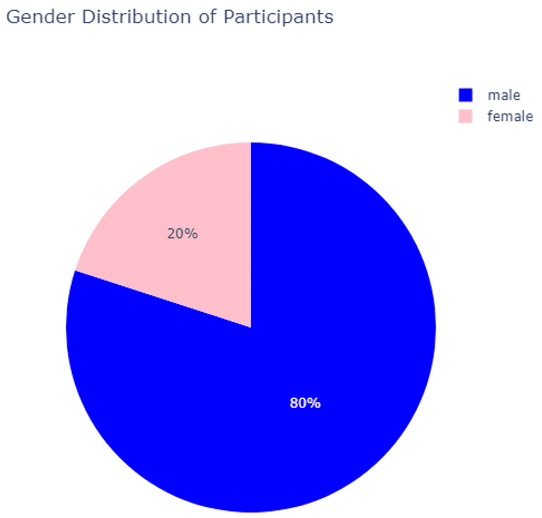


Figure 5.3: Gender Distribution

documentation. For this study, we have prepared the interface documentation for both *VQB* and *SPARQL* querying language; however, the documentation for the *SPARQL* language does not cover all aspects of the *SPARQL* querying language and only introduces the basics and required syntax for the completion of the tasks in the study. Appendix A contains the documentation and introduction for both interfaces.

Subsequently, participants were asked to complete each task (*T1*, *T2*, *T3*, *T4*) with both interfaces, and for the task *T5*, they were asked to choose their preferred interface to do the task; or do the fifth task on both interfaces if they are curious and interested. We also informed participants that they are allowed to refer back to the documentation during the study tasks or ask for hints from the study facilitator. Similar to the introduction phase, the order of the interface depends on the group that the participant is in. After completing each task with each interface, participants were asked to fill out the *NASA-TLX* questionnaire form and evaluate the load of the task completion using that interface.

After completion of all tasks by the participants, they were asked to fill out two System Usability Scale (*SUS*) questionnaires; one for each interface. The final questionnaire that they were asked to fill out was the custom questionnaire that we had

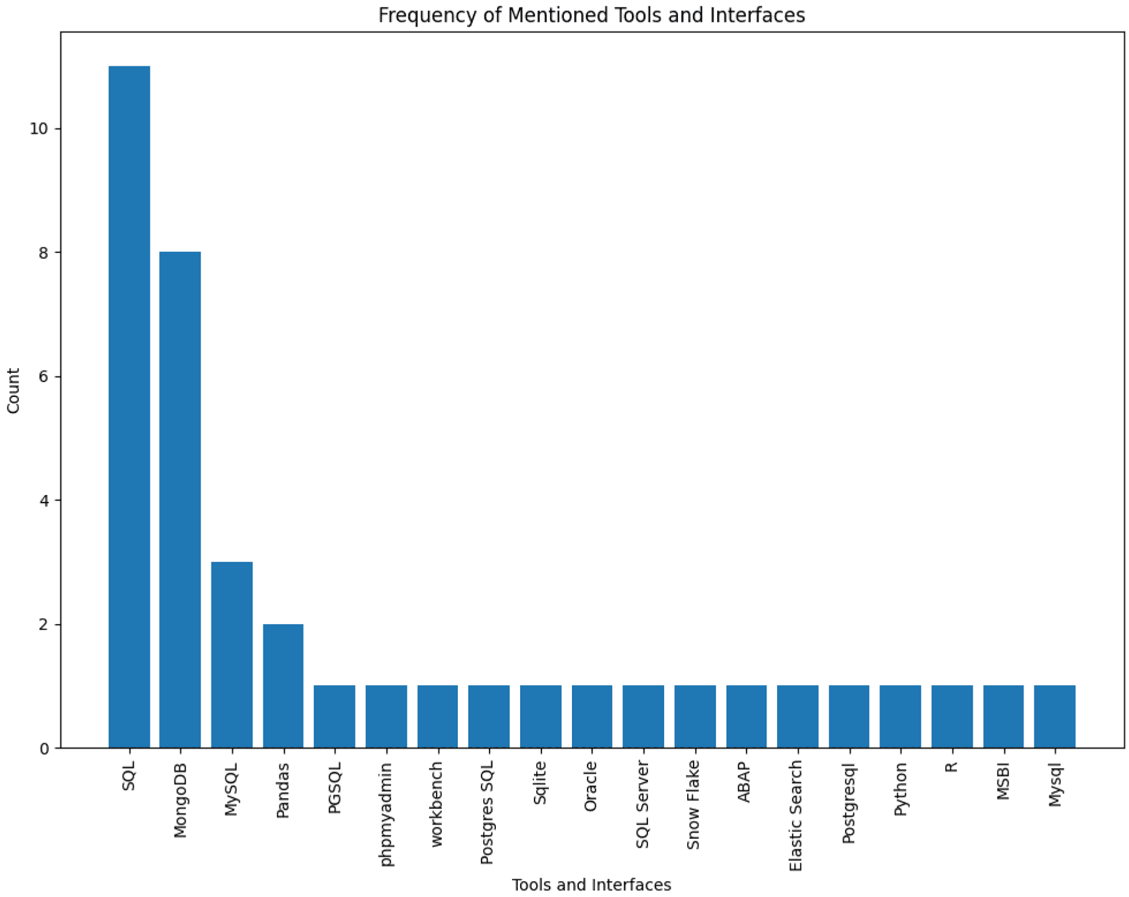


Figure 5.4: Data Management Tools and Interfaces Experience

designed. All of these questionnaires are accessible from Appendix A. Upon completing the study, participants received a cash reward as a token of appreciation for their time, effort, and participation.

5.6 Data Collection

In this section, all types of data that have been collected during the study will be explained and justified by declaring the purpose of collecting it.

5.6.1 Demographic questionnaire

In our demographic questionnaire, general questions about the participants' gender, and age have been asked to have a better understanding of our participants. Moreover, we asked participants about their related knowledge to the field of GIS, or general

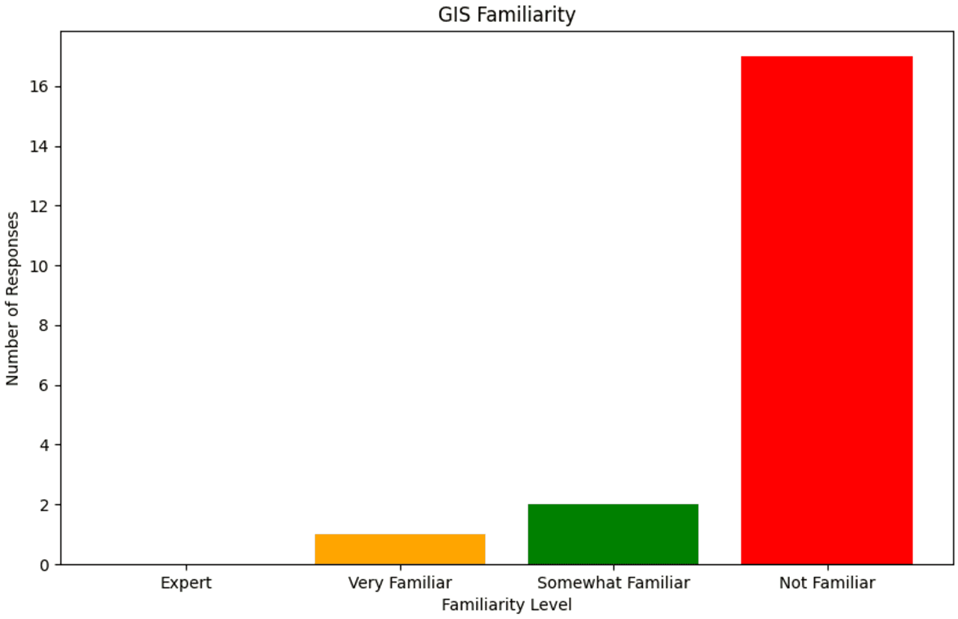


Figure 5.5: GIS Familiarity

database querying experience; possibly leading us to some findings about the relation of their prior knowledge to their performance, and accuracy in the study.

5.6.2 Screen recording of the querying tasks

Analyzing the recordings of participants' querying experience, could possibly be beneficial in detecting any unexpected interface errors, and it can highlight the most struggling aspect or component of the study. Moreover, this can help us examine how users behave when attempting to complete their tasks; for instance, several participants adopted distinctive strategies and different composition of graphs to complete their tasks.

5.6.3 Time to finish document reading

This data is basically the time that the participants spent on each documentation, the VQB interface documentation, and the SPARQL language documentation. By analyzing this data, we can get insights into how quickly participants can read and comprehend the document for each interface and start the tasks.

5.6.4 Time to finish tasks

This data is obtained by calculation of the time when the participant has started the task and the timestamp that they finished the task successfully. This data is used to compare both interfaces in terms of the performance criteria.

5.6.5 Time spent on documents during the tasks

All the timestamps that users referred back to the references and also the timestamps when they came back to the querying interface have been captured. Analyzing this data can help us to understand how much of the total time of doing the tasks was particularly spent on doing the task rather than referring to the documentation.

5.6.6 Number of referring back to the documentation during the tasks

Analyzing this data is a beneficial way to understand how participants can rely on themselves independently while doing the tasks, and compare it for each interface. This data is a beneficial performance metric for the interface comparison. Moreover, it tells us about the user-friendliness of the interface and also can highlight user experience issues related to the interface. Furthermore, it can help us to understand how effective were each interface's documentation.

5.6.7 Given hints during the tasks

Admittedly, neither of the implemented interfaces is perfectly designed, and regardless of the many improvement iterations of implementation they had on the interfaces, there remains room for more improvements. Therefore, we predicted that some users might have some issues with the interfaces (especially using them for the first time) that might be a blocking factor for them to complete their tasks. Hence, throughout the study, I, as the observer, documented the types of hints that I gave to participants and their frequency for subsequent analysis. The hint procedure involved only giving hints to participants if they explicitly asked for help and were struggling significantly, unable to finish on their own. I aimed to be as non-intrusive as possible. I aimed to be as non-intrusive as possible during the study. Analyzing highly frequent hints can lead us to some findings about issues related to the user experience of the interface. Also, it can highlight challenges inherent to the interface's concept. For example, using the "UNION" operation in SPARQL queries, had the most difficulties for the participants while the documentation for that concept was well-written and comprehensive.

5.6.8 Submitted queries and results

Both in SPARLQ and VQB, the final results of the query can be different due to the fact that the same task can be done in different ways. For instance, a property and a condition node can be shared among multiple vessel nodes, or they can be duplicated and used and both result in the same output. Similarly, in SPARQL, the variable definition can be placed in different blocks, and the conditions can be duplicated or defined globally to have the same effect. Capturing this data helps to detect patterns and the approaches that participants take to formulate their queries, and offers a valuable behavioral comparison.

5.6.9 NASA-TLX

NASA-TLX is a scale that provides a calculation for the workload of multiple actions. The scales that are involved in the calculation are mental demand, physical demand, temporal demand, performance, effort, and frustration level [23]. Each of these scales ranges from 1 to 20, and participants during the study have to mark those scales. In this study, participants were requested to complete scales for every task and interface. This data will help us to have a comparison between two interfaces in terms of the perceived workload participants during the study. Fig 5.6 shows the standard format of the NASA-TLX form, obtained from the NASA website[3]

5.6.10 System Usability Scale (SUS)

System Usability Scale is a Likert scale tailored for the subjective usability evaluation of systems and projects. It consists of 10 questions, ranging from "Strongly disagree" to "Strongly agree" as shown in Fig 5.7. This questionnaire covers a wide aspect of the usability metrics of a system, especially, complexity, effectiveness, efficiency, and satisfaction. The SUS is typically employed once the participant/user has had a chance to interact with the system under evaluation. The output of the questionnaire

[3] NASA-TLX form URL: https://humansystems.arc.nasa.gov/groups/tlx/downloads/TLXScale.pdf

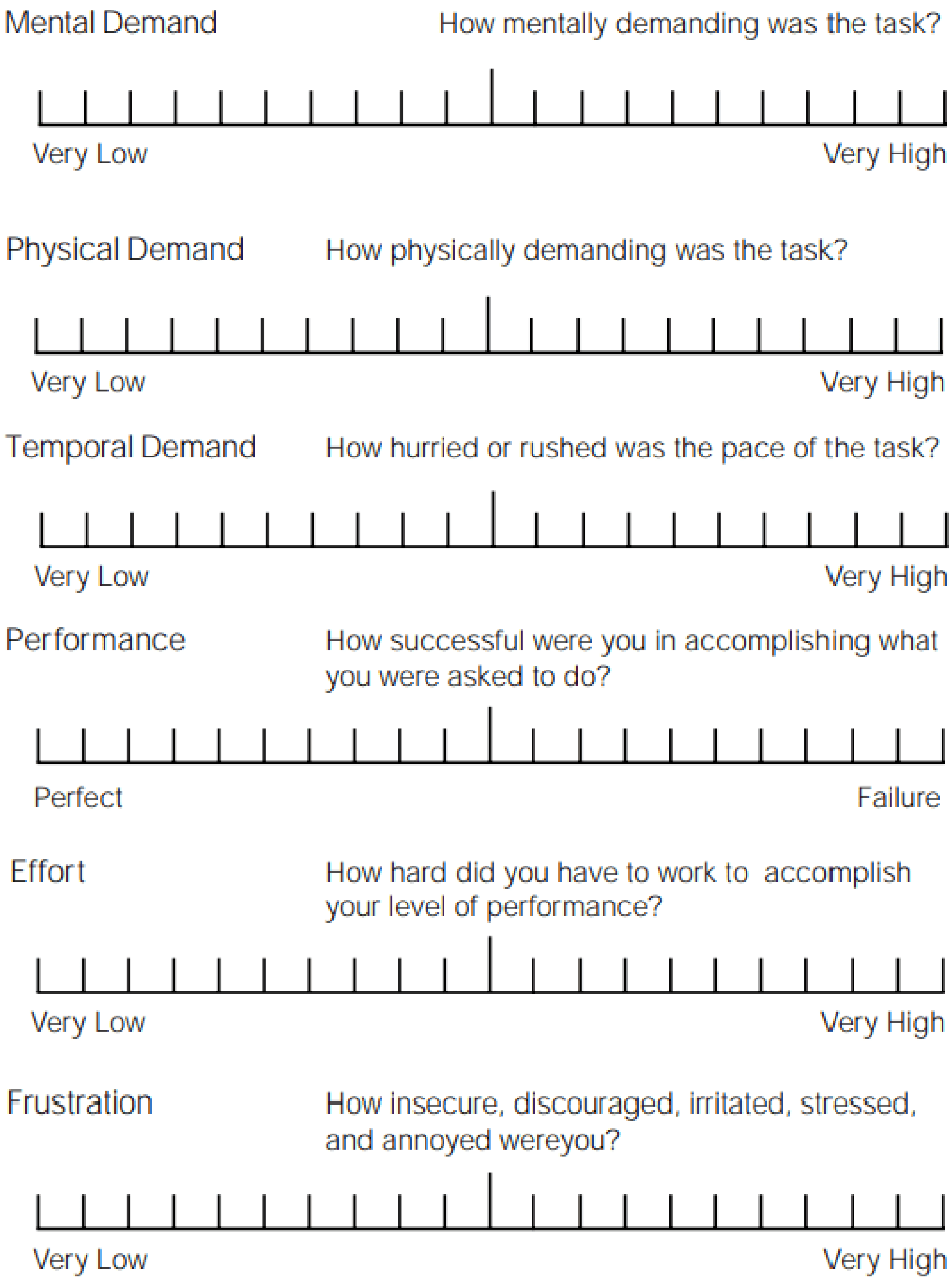
would be a number ranging from 0 to 100, demonstrating the usability score of that system [7]. As mentioned earlier, we have employed this test for both of the interfaces after the completion of all the tasks by the participant. Analyzing the results of this test would help us understand how participants perceived the usability of both interfaces and provide us with a reliable metric for comparison.

5.6.11 Custom questionnaire

We have provided a custom questionnaire to ask about the preference, task performance, task accuracy, and user experience of the interfaces from the participants. We used Likert scales for each interface to evaluate them on the categories mentioned above. Additionally, we've included open-ended questions to enable participants to describe their experiences while using the interfaces during the study. This approach will allow us to gather more relevant data that the limited closed-ended questions may not have captured. Eventually, an open-ended question regarding the overall experience of the study was added at the end of the questionnaire. The full questionnaire can be found in Appendix A.

5.6.12 Observer notes

In order to have a more accurate interface comparison, during the study, the observer took notes during the study including the hints and the type of hints given to the participants, any pause or break during the study by the participant, and the feedback given by participants. By capturing these notes, the observer aimed to gain deeper insights into the participant's interactions with the interfaces, the participant's behaviour, and their overall engagement.



Mental Demand

How mentally demanding was the task?

Very Low

Very High

Physical Demand

How physically demanding was the task?

Very Low

Very High

Temporal Demand

How hurried or rushed was the pace of the task?

Very Low

Very High

Performance

How successful were you in accomplishing what you were asked to do?

Perfect

Failure

Effort

How hard did you have to work to accomplish your level of performance?

Very Low

Very High

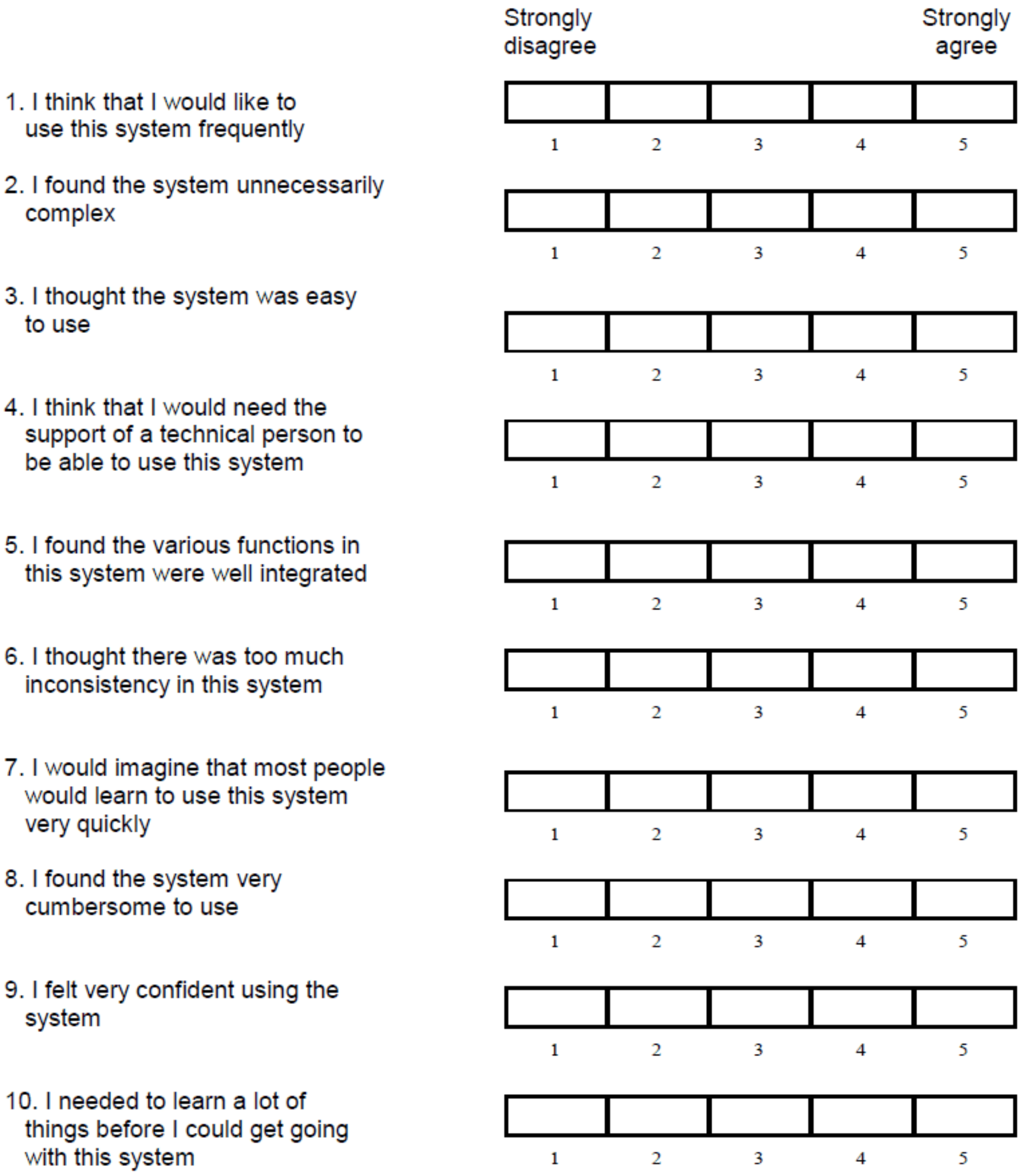
Frustration

How insecure, discouraged, irritated, stressed, and annoyed wereyou?

Very Low

Very High

Figure 5.6: NASA-TLX Workload Scale Form



Strongly disagree — Strongly agree

1. I think that I would like to use this system frequently

□ □ □ □ □
1 2 3 4 5

2. I found the system unnecessarily complex

□ □ □ □ □
1 2 3 4 5

3. I thought the system was easy to use

□ □ □ □ □
1 2 3 4 5

4. I think that I would need the support of a technical person to be able to use this system

□ □ □ □ □
1 2 3 4 5

5. I found the various functions in this system were well integrated

□ □ □ □ □
1 2 3 4 5

6. I thought there was too much inconsistency in this system

□ □ □ □ □
1 2 3 4 5

7. I would imagine that most people would learn to use this system very quickly

□ □ □ □ □
1 2 3 4 5

8. I found the system very cumbersome to use

□ □ □ □ □
1 2 3 4 5

9. I felt very confident using the system

□ □ □ □ □
1 2 3 4 5

10. I needed to learn a lot of things before I could get going with this system

□ □ □ □ □
1 2 3 4 5

Figure 5.7: System Usability Scale (SUS) Questionnaire [7]

Chapter 6

Comparative Evaluation and Analysis

As mentioned in the methodology chapter, the goal of this study is to answer three main research questions: How well can the visual querying interface perform compared to a traditional interface in terms of *performance*, *accuracy*, and *preference*? Here in this chapter, the result and analysis of the conducted user study will be presented. Additionally, the employed methodologies for data analysis will be elaborated upon, including the techniques used for statistical analysis and qualitative analysis, and the rationale behind using them.

6.1 Time Efficiency and Accuracy

Analysis of task completion times

The entire completion time is a crucial parameter for the efficiency assessment. The time gap between a participant beginning and completing a task is represented by this data. This data is captured for each task and interface during the study. The analysis of the time-captured data for 20 participants using the two interfaces shows that on average, participants finished their tasks using the VQB interface in 3 minutes and 8 seconds (M=2.95, STD=2.002), compared to the SPARQL interface which took 5 minutes and 50 seconds on average (M=4.83, STD=3.612).

The line chart in Fig 6.2 represents the time that each participant finished the task with each interface. Although this line chart represents the raw data of the participants' performance, this visualization gives us insight into how the time-to-finish of the tasks using the SPARQL interface is more volatile than the VQB interface. Moreover, Fig 6.3 shows an aggregated visualization comparing the two interfaces utilizing a box-plot chart. Analyzing this chart, it is evident that participants could do their tasks quicker using the VQB interface; with task 2 as an exception. Therefore, the effect of the task and interface is noticeable by looking at this chart, yet to be tested with the proper statistical analysis.

The violin-plot visualization in Fig 6.4 provides a deeper understanding of the density of our performance data in a way that a wider section of a violin indicates a higher density of data points which means more completion times recorded in that time interval. This visualization also suggests that the VQB interface had a denser and lower distribution, indicating faster completion times with less variation. Task 3 had particularly a wide violin for the SPARQL interface compared to the VQB interface, meaning that participants struggled more on task 3 using the textual editor interface.

Figure 6.1 represents the mean duration required to accomplish tasks with each interface for all four tasks. Despite the incremental increase in task complexity, the downtrend is visible for the SPARQL mean duration time, suggesting the apparent learning effect for the textual interface, meaning that by doing more tasks, participants became quicker using the interface. However, the same statement does not apply to the VQB interface. In other words, there were not any visible learning effects associated with that interface.

To statistically compare the two interfaces in terms of their performance (time efficiency), we have considered a two-way ANOVA test (mixed design) for the within-subjects factors (task, interface) with an additional between-subjects factor (group A and B for counterbalancing). Table 6.1 represents the result for the univariate type III repeated measures of the ANOVA test by assuming the sphericity. The ”mixed-design” ANOVA test has been used because, in this test, both within-subject and between-subject factors are present. The term ”univariate” means that this test has only one dependent variable. The repeated measures design considers that the same subject experiences all conditions; in this case, all participants, no matter what their group is, have experienced all the tasks using both interfaces. The ”sphericity” in the ANOVA test states that all combinations of related groups (i.e., conditions or time points) should have equal variances of the differences between them to have a more accurate result. To test the sphericity, we utilized the Mauchly tests for sphericity, and the result was not significant (p-value of 0.062 for task and group:task variables and the p-value of 0.078 for the task:interface and group:task:interface). Therefore, the assumption of the sphericity holds.

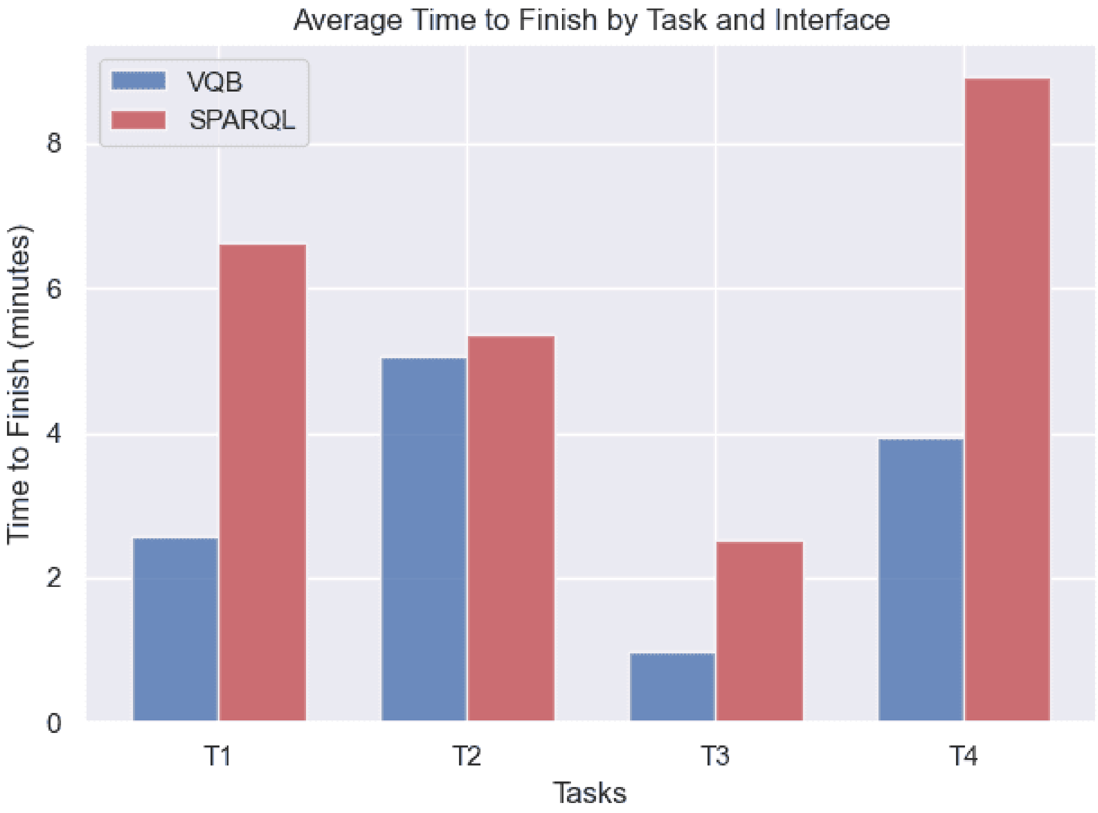


Figure 6.1: Average Time-to-finish Per Interface For Each Task

The statistical result of the ANOVA test shows that the tasks' nature significantly affects the time efficiency. This result was as expected since the different task difficulties can have an obvious effect on the time efficiency. Another significant factor was the interface. The statistical analysis result shows that the interface VQB highly outperformed the SPARQL interface in terms of performance. Another significant factor to consider is the interaction effect between the task and interface (task:interface), which was found to be significant. This indicates that the performance and time efficiency differences between the VQB and SPARQL interfaces were not consistent across all tasks, suggesting that the effectiveness of an interface can vary depending on the specific task at hand.

Table 6.1: Univariate Type III Repeated-Measures ANOVA Assuming Sphericity

Source	Sum Sq	num Df	Error SS	den Df	F value	Pr(>F)	Signif.
(Intercept)	3230.8	1	255.017	18	228.0385	1.152e-11	***
Group	0.0	1	255.017	18	0.0029	0.95760	
task	471.8	3	185.755	54	45.7201	7.619e-15	***
Group:task	15.2	3	185.755	54	1.4714	0.23260	
interface	293.2	1	90.823	18	58.1063	4.842e-07	***
Group:interface	14.6	1	90.823	18	2.8865	0.10654	
task:interface	141.1	3	152.326	54	16.6788	8.589e-08	***
Group:task:interface	21.1	3	152.326	54	2.4989	0.06928	.

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

In a nutshell, all the participants were faster using the VQB interface comparing the textual query builder on average, with one participant (P3) as an exception who completed the tasks using the SPARQL interface quicker than VQB by finishing tasks on average 28 seconds faster than with VQB. Specifically, P3’s average completion time was 2 minutes and 38 seconds for VQB, versus 2 minutes and 10 seconds for SPARQL.

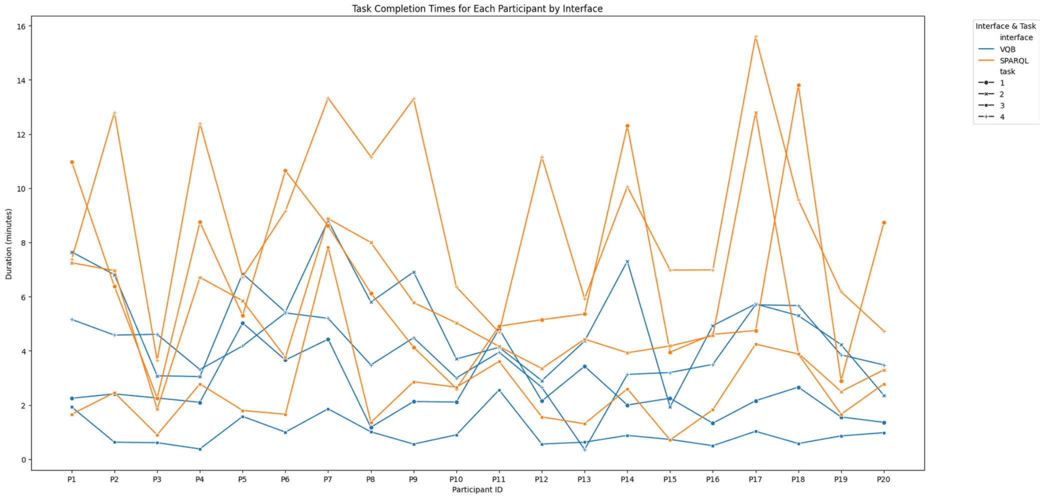


Figure 6.2: Line chart representing the time efficiency of the participants for each task, per interface

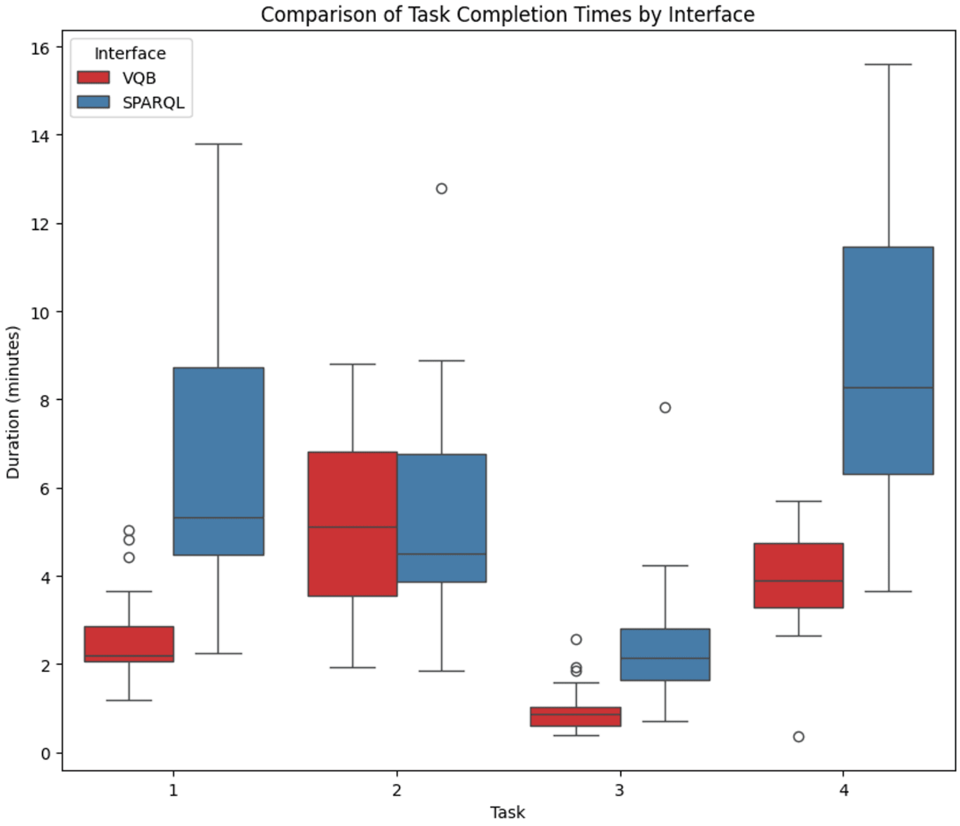


Figure 6.3: Box-plot visualization of the performance on each task

Accuracy

Here in this section, the accuracy of the two interfaces is compared, both objectively using the frequency of the errors and given hints, and subjectively by analyzing the custom questionnaire filled by the participants. Table 6.2 shows each participant's common errors and hints during the study. The bar chart on Fig 6.5 demonstrates the type of the given hints, sorted by their frequency meaning that the hints given on the top bars have been more frequent and common among the participants.

The chart in Fig 6.6, represents the total given hints to the participants during the study. On average, participants using the SPARQL interface asked for hints 4.00 times compared (std=1.974) to the VQB interface the average number of hints given was 2.10 (std=0.888). A t-test has been utilized to statistically compare the interfaces'

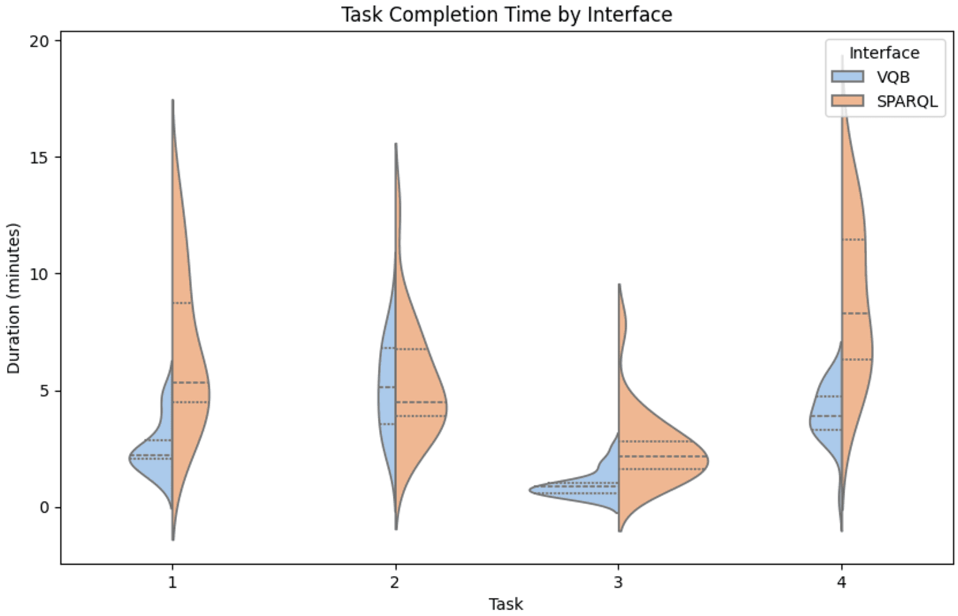


Figure 6.4: Violin-plot performance visualization

total hints given, the result shows that participants using the SPARQL interface significantly asked for more hints than the VQB interface (p-value = 0.000473 and t-statistic=-3.82422).

6.2 User Experience and Preferences

Usability and Workload

This section will demonstrate the results of the conducted usability and workload tests. Starting with the NASA-TLX workload test, the unweighted version of the NASA-TLX analysis has been considered. As mentioned earlier, this test covers the six main pillars for assessing workload, which are mental demand, physical demand, temporal demand, performance, the effort required, and frustration level, all ranging from 1 to 20. To calculate the final score for each test, we have multiplied each category score by 5 and calculated the unweighted average of it. The final results are as follows: the overall workload of the interface VQB scored 16.0/100 (mean score, i.e. average score among participants), while the SPARQL interface scored 28.7/100. A higher score on this test means that the participants, subjectively, perceived that

Table 6.2: Participant Given Hints Types Table: Indicating the Common Problems

Interface	Error Type	Participant IDs
SPARQL	Temporal Filtering	p1, p7, p10, p13,p14,p15, p16, p17, p18
	Type Definition	p1, p2, p4,p6, p7, p10, p13, p14 ,p15, p16, p18
	Spatial Filtering	p4, p16
	UNION	p4, p7, p8, p11,p13, p14, p17, p18
	Out of Block Definition	p7, p8, p9, p10, p15, p16
	Syntax Errors	p2, p3, p4, p6, p7, p8, p10, p13,p15, p17, p19
	Missing Prefixes	p8
VQB	All Vessels Selection	p1, p6, p7, p8, p10,p13, p16, p14, p17
	Spatial Filtering	p1
	Output node selection	p2
	Temporal Filtering	p3
	Panel Confusion	p4
	Drag and drop confusion	p4
	Sharing Non Sharable Nodes	p3,p5
	Wrong Connection	p5, p7, p9, p11, p16, p17, p18, p19

more workload is required for doing the tasks on that interface. Fig 6.7 shows these overall scores per task; suggesting that on average participants felt that the SPARQL interface had more required workload, while for task two the perceived workload for both interfaces was close.

The interaction graph in Fig 6.9 visualizes the interaction of the interface and task on the average workload scores. Focusing on participant groups doing the tasks on SPARQL (A.SPARQL, B.SPARQL), we can notice that group B had a relatively higher perceived workload compared to group A. This can suggest that using the VQB interface for doing a task, can give the participants a better insight into the query formulation graph and its visual representation, making them more ready to do the same task on the SPARQL.

Given that our participants were divided into two groups to analyze the potential impact of sequence (order effect), it is valuable to investigate the influence of group allocation on other variables. Figure 6.9 demonstrates the interaction between task, group, and interface type on the NASA-TLX workload scores.

In order to compare the workload score of the two interfaces statistically, we conducted a two-way ANOVA test with a mixed design. We analyzed the univariate type III repeated measures of the ANOVA test assuming sphericity, and the results

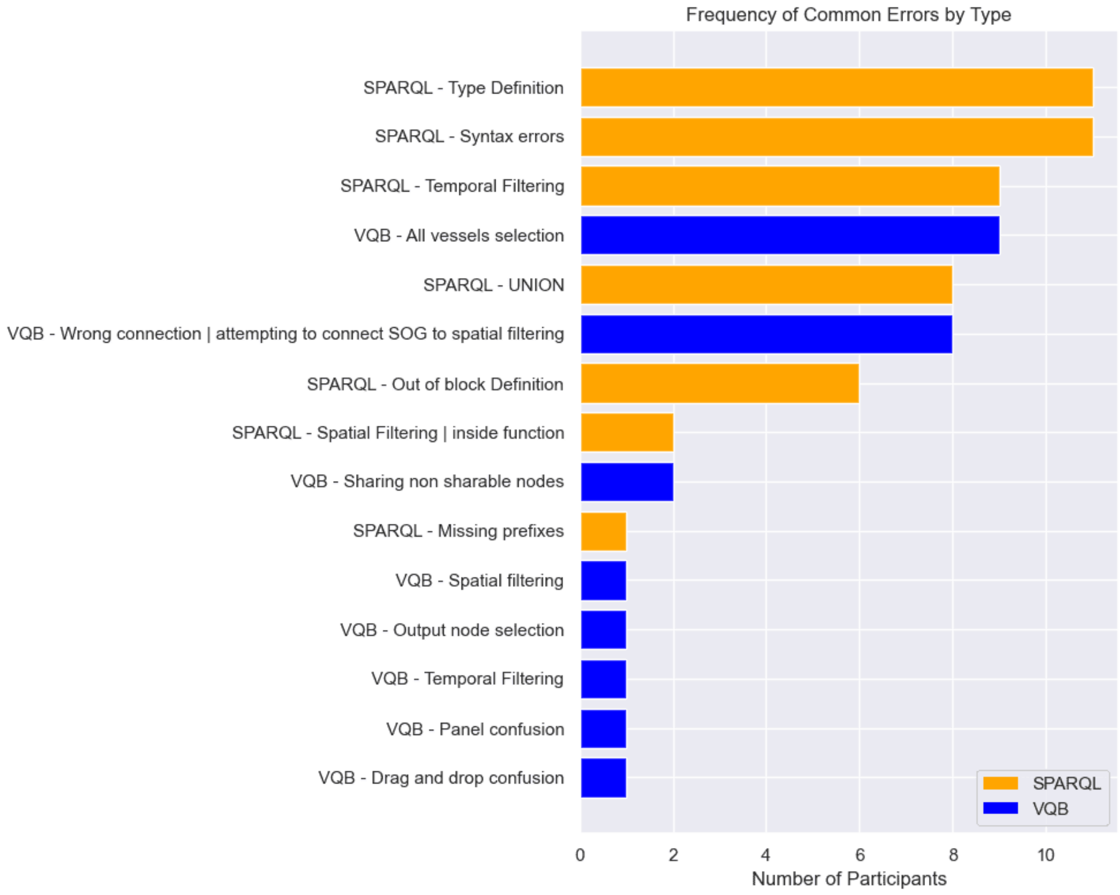


Figure 6.5: Given Hints Rate Sorted By Frequency

are presented in Table 6.3. By assuming the sphericity, the results show that the intercept, interface, task, and interface:task (the interaction effect of interface and task) have statistically significant effects since they have a P value of less than 0.05. However, the group and its interactions with other variables (task, interface) do not show a significant effect because their P value is greater than 0.05. Furthermore, to validate the sphericity consumption, Mauchly's test [38] has been used. The results didn't show any significant result, therefore, the assumption of sphericity holds. For the calculation of the represented statistical tests, the afex library[1] is used in the R programming language. Although, in this section, only the representation of the statistical test for the overall NASA-TLX score has been provided, for all questions of the NASA-TLX (mental demand, physical demand, etc.) the same statistical test

[1] URL of Afex library on Github: https://github.com/singmann/afex

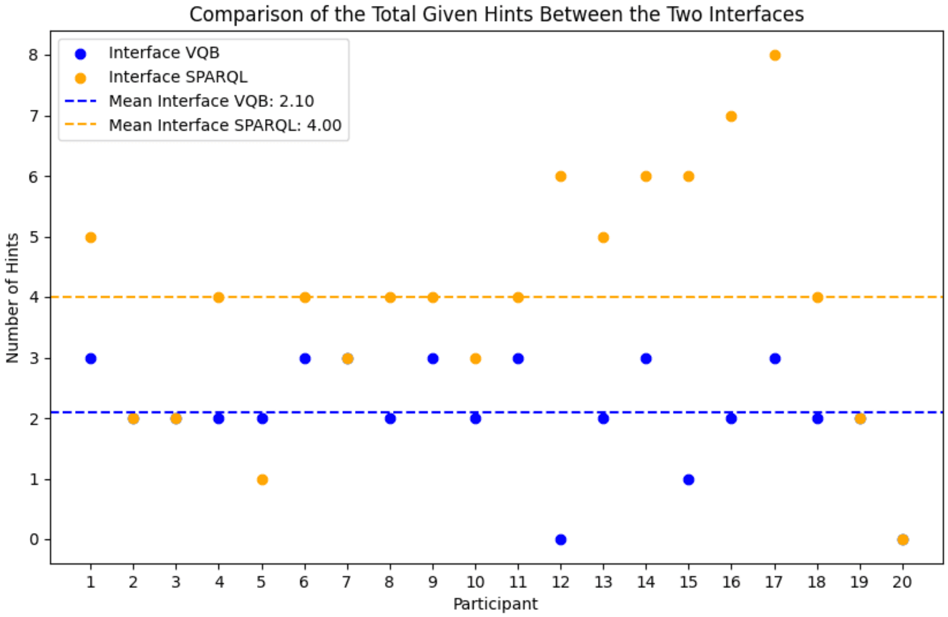


Figure 6.6: Total given hints per interface

has been applied and represented in Appendix B.

Speaking of the system usability scale (SUS) results, the mean score of the VQB interface has been determined as 84.5/100 and the SPARQL editor interface as 49.62/100. Research shows that the average score of the SUS test is 68/100 [8]. Based on the SUS results of the interfaces, and referring to Figure 6.10, it is evident that the VQB interface score qualifies as and fits into the B-tier grade scale, whereas the SPARQL editor interface ranks in the F-tier.

To compare the SUS of the interfaces, a t-test has been utilized. As the results, the t-test statistic of 6.765 with a small p-value (5.12e-08) indicates a highly significant difference between the SUS scores of the two interfaces. Fig 6.12 represents the aggregated SUS score comparison of the interfaces in a box-plot, and Fig 6.11 visualizes a scatter-plot of all participants given scores for each interface.

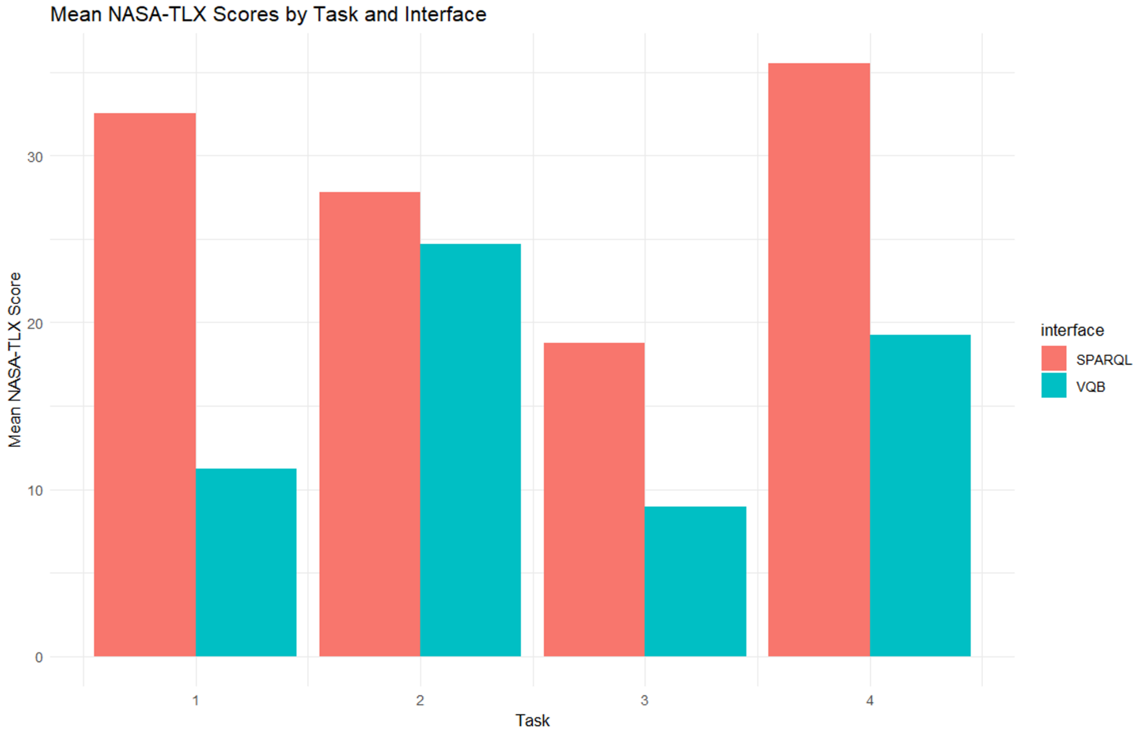


Figure 6.7: Mean NASA-TLX Scores Per Task (Tasks 1 to 4)

Table 6.3: Univariate Type III Repeated-Measures ANOVA Assuming Sphericity

Source	Sum Sq	num Df	Error SS	den Df	F value	Pr(>F)	Signif.
(Intercept)	79953	1	20353.7	18	70.7075	1.197e-07	***
group	4118	1	20353.7	18	3.6414	0.0724408	.
interface	6376	1	5697.6	18	20.1420	0.0002844	***
group:interface	688	1	5697.6	18	2.1720	0.1578160	
task	4508	3	6250.1	54	12.9831	1.689e-06	***
group:task	564	3	6250.1	54	1.6234	0.1946712	
interface:task	1874	3	4703.7	54	7.1711	0.0003870	***
group:interface:task	258	3	4703.7	54	0.9879	0.4054745	
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1							

Table 6.4: Mauchly’s Test for Sphericity

Effect	Mauchly’s W	p-value
task	0.63010	0.17286
group:task	0.63010	0.17286
interface:task	0.76739	0.49039
group:interface:task	0.76739	0.49039

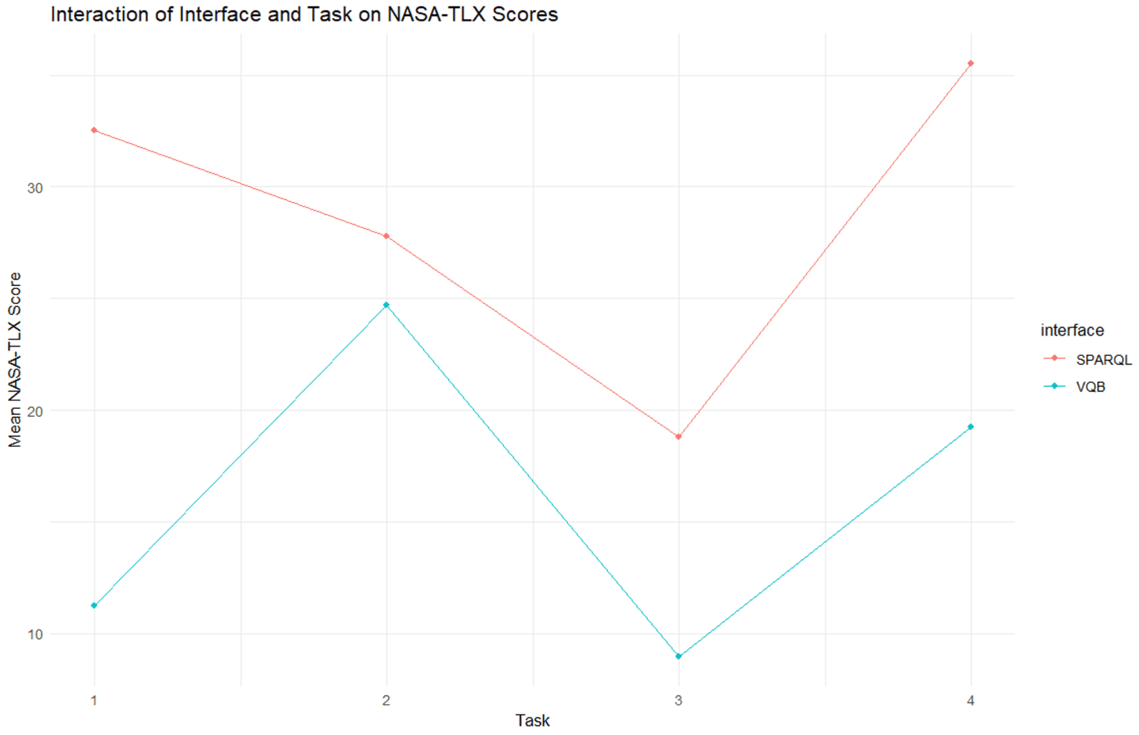


Figure 6.8: Interaction of Interface and Task on NASA-TLX workload

Preferences

For the last task in the study (T5) participants were asked to choose their preferred interface to formulate the required query on. The rationale behind this was to analyze the participants' preferences. 16 participants out of 20, chose the VQB interface, 3 out of 20 chose the SPARQL interface, and one chose both interfaces to work on since they wanted to see the equivalent representation of the result query on both interfaces.

By analyzing the custom questionnaire responses, the participants felt more accurate using VQB (Avg 4.78/5) rather than SPARQL (Avg 3.3/5). However, in some tasks, users reported that they felt they had more accuracy using the SPARQL querying approach rather than VQB. Some participants mentioned that because they didn't have prior SPARQL querying experience, they had some confusion and syntax issues using SPARQL querying. Four participants noted the limitation of the VQB interface that misled them into accurate querying such as the absence of a feature that allows users to select all vessels in their querying and adjacency of the "Clear Button" near

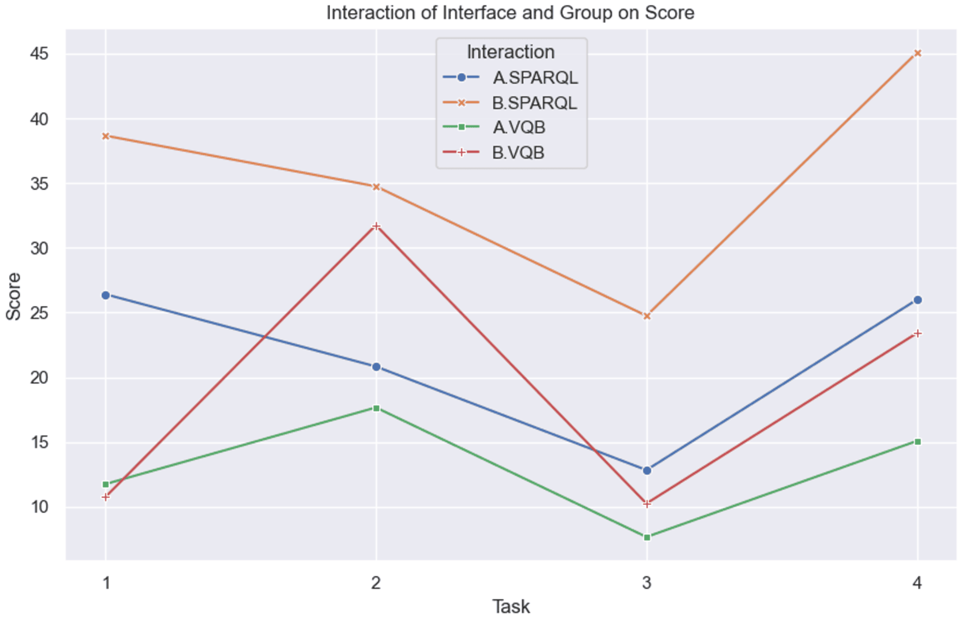


Figure 6.9: Interaction Between Tasks, Interface, and Group on the Workload Score

the "Date Picker Component" while the clear button won't affect the picked date and erased the whole graph in the working panel. One participant mentioned that at first task both interfaces were vague for them but after that, they understood the procedures, and VQB was an easier interface to understand especially for tasks that include union operation. Two participants believed that SPARQL was more robust and more capable of handling more complex queries rather than the VQB interface which has its limitations. Another participant mentioned that for using the SPARQL querying method, they need to have an assistant and cannot rely on themselves during querying.

Regarding user-friendliness, 14 participants preferred the VQB interface, and the other 6 participants didn't have any preferred interface other than others. By analyzing the open-ended question regarding the user experience, some participants mentioned that the different colors for the nodes in the VQB interface helped them to form a better mental model and helped them memorize the node functionalities better. Three participants who preferred VQB mentioned that they had picked that interface since they thought the VQB interface was more intuitive than the SPARQL editor. Although one participant mentioned that the dragging and dropping nodes

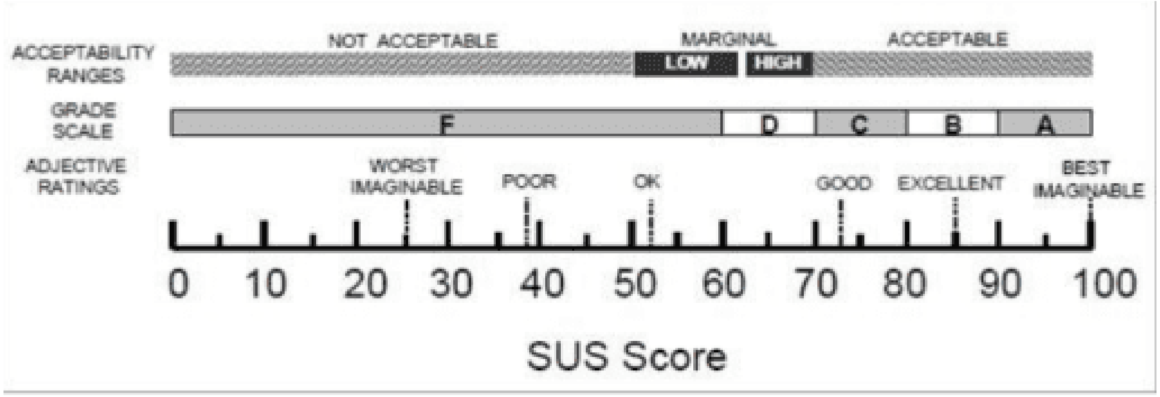


Figure 6.10: Grade rankings of SUS scores [8]

were unintuitive, three participants mentioned that they liked it. P1 while doing the tasks on the VQB interface mentioned that working with this interface feels like they are playing a game. Two participants mentioned that there was no warning message while they were trying to connect the wrong nodes, and this is an important feature that has been missing in the VQB interface.

In the same questionnaire, participants were asked to suggest any improvements for both interfaces, five participants mentioned that the duplication of the nodes is missing in the VQB interface. Regarding that, P3 mentioned “In the VQB I like to duplicate each element by double clicking on it, similar to the draw.io tool. It would be nice to generate an element somewhere in the sheet when you double-click on the creating element (instead of dragging)”. Some of the participants suggested more customization features for both interfaces, for the SPARQL editor, for example, they mentioned that changing the overall theme fonts and styling would be a good idea. For the VQB, P9 mentioned that the layout of the VQB can be changed in a way that under the node palette, we can place the temporal filtering component ”so that the users can get everything at one place”. Six participants mentioned that selecting all vessels feature with one node is missing in the VQB interface and made the querying experience complex for task 2 which they have been asked to query all types of vessels with certain conditions.

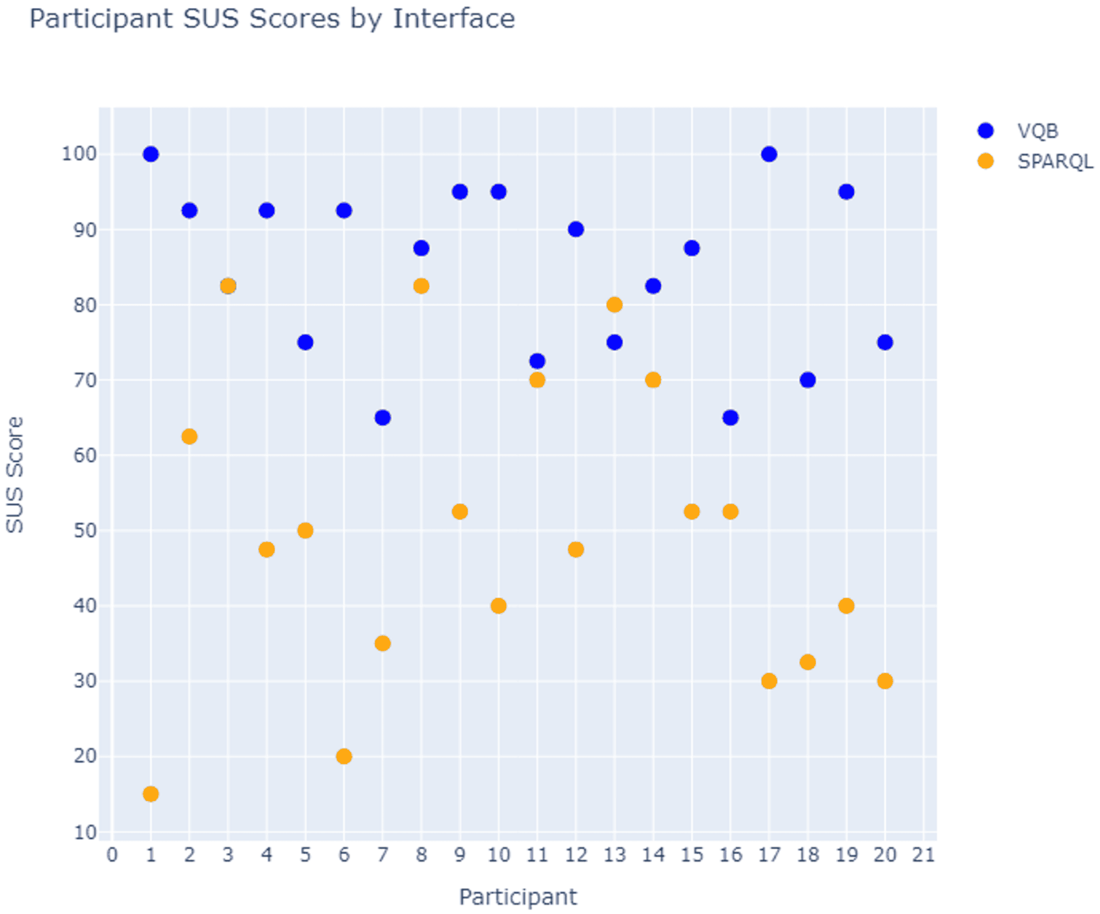


Figure 6.11: Scatter-plot visualizing of the SUS scores

Behavioral Insights

By analyzing the screen recordings of the participants while doing the tasks, it was common among participants that most of them using VQB for their first task started to explore the interface by dragging and dropping nodes for testing purposes and experimenting with all features of the VQB interface before formulating the task query. Another takeaway from the recording analysis is the number of times participants referred to references while completing the tasks. Analyzing the numbers, we can infer that participants used to go back and forth from the task environment to the documentation more while they were doing tasks using the SPARQL interface compared to the VQB interface. Table 6.5 represents the descriptive statistics for the

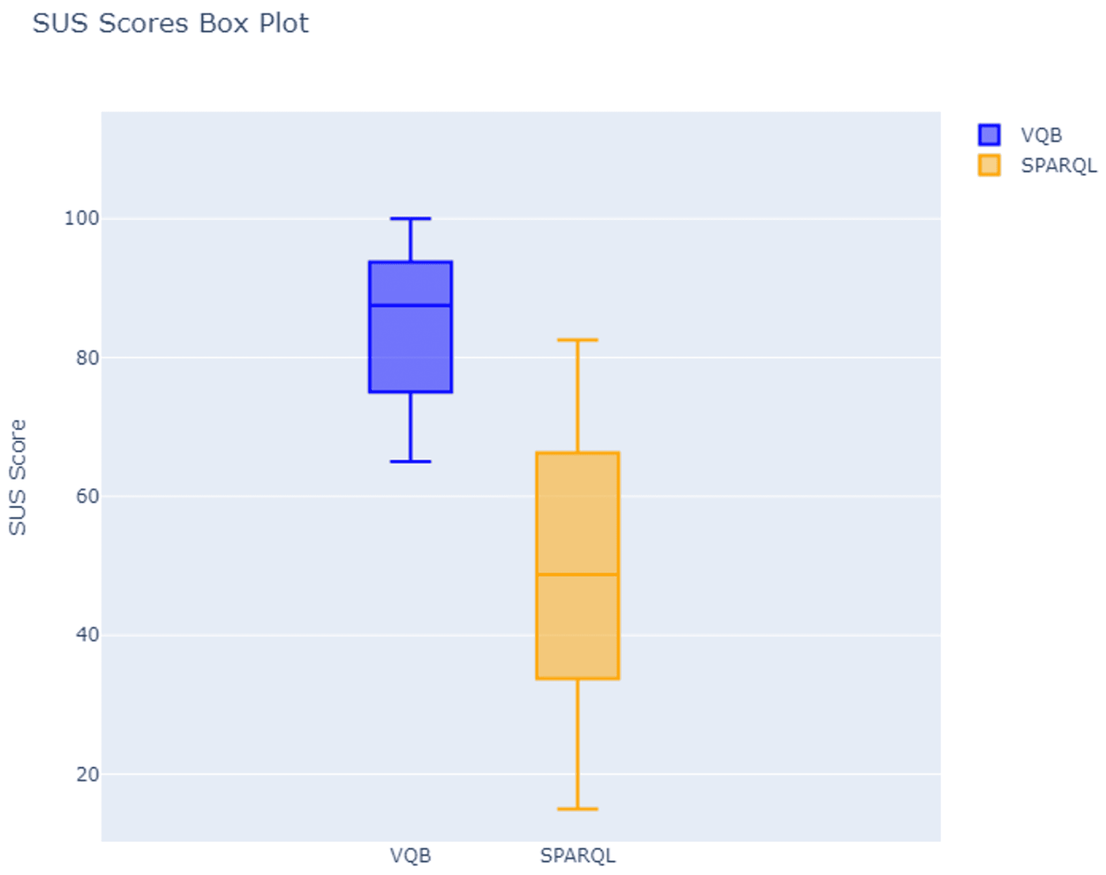


Figure 6.12: Box-plot visualization of the SUS scores

number of times that participants referred to the documentation for each task, and Fig 6.13 visualizes these reference counts in a box-plot visualization format.

Table 6.5: Descriptive Statistics for Reference Counts by Task and Interface

Task	**Interface**	**Mean**	**Median**	**SD**
T1	SPARQL	5.3	4.5	3.69
T1	VQB	1.4	1.5	1.05
T2	SPARQL	6.7	5.5	3.88
T2	VQB	2.25	2	1.89
T3	SPARQL	2.85	2.5	1.69
T3	VQB	0.1	0	0.308
T4	SPARQL	5.85	6	3.12
T4	VQB	0.2	0	0.523

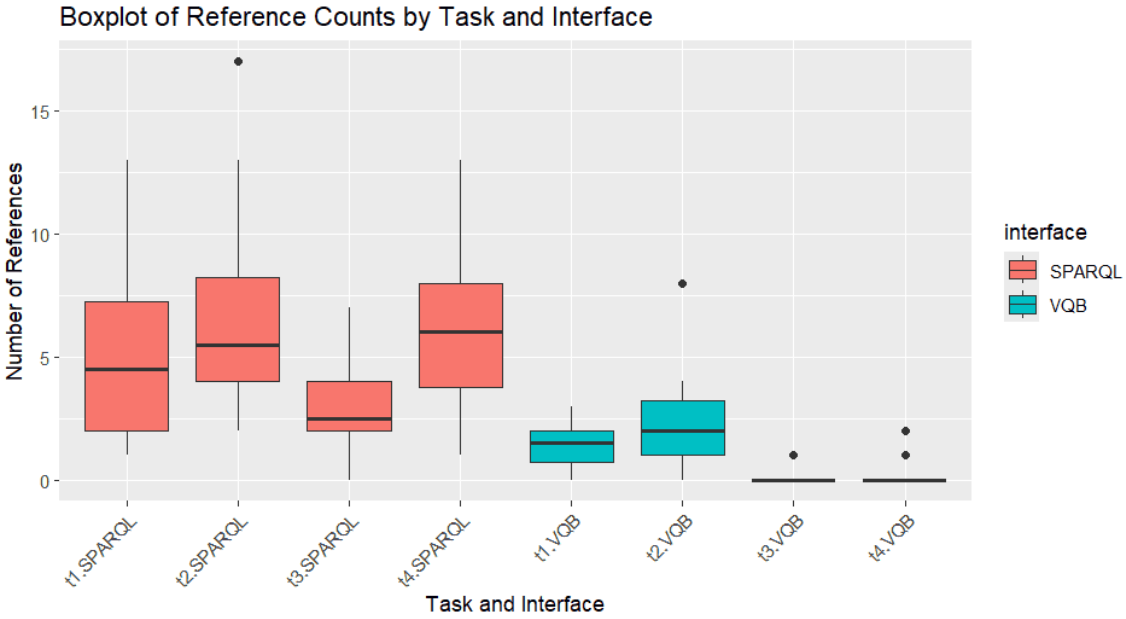


Figure 6.13: Box Plot Visualization of Reference Counts

We have again used the univariate type III repeated measures of the ANOVA test by assuming the sphericity to statistically compare the results and detect the effects' significance. Table 6.6 shows the result of the test. The model intercept is significant meaning that the overall mean is significantly different from zero. The group effect is not significant since the P value is greater than 0.05 which means there was no significant difference between the two groups used for counterbalancing (A and B). The task effect was highly significant meaning that the task's difficulty had the effect on the number of time required for the participants to refer back to the documentation. The interface has a significant effect, meaning participants using VQB referred back to references significantly less than SPARQL. The interface:task effect was significant, suggesting that one interface is more suited for certain tasks. It is worth mentioning that Mauchly's Test for Sphericity has violated the Sphericity assumption 6.7, for the task:interface and the group:task:interface. Therefore, Greenhouse-Geisser and Huynh-Feldt Correction have been used 6.8 to adjust the degrees of freedom for the F-tests in the sphericity violations, which affects the p-values. For the task-related effects, the original p-values stand (since they do not violate sphericity). However, for task:interface and Group:task:interface, the corrected p-values (GG and HF) should be used, which still show significant effects for task:interface under both corrections.

Table 6.6: Univariate Type III Repeated-Measures ANOVA Assuming Sphericity

Source	Sum Sq	num Df	Error SS	den Df	F value	Pr(>F)	Signif.
(Intercept)	1519.06	1	232.562	18	117.5727	2.534e-09	***
Group	23.26	1	232.562	18	1.8000	0.19639	
task	183.92	3	248.687	54	13.3120	1.278e-06	***
Group:task	44.02	3	248.687	54	3.1861	0.03093	*
interface	701.41	1	92.062	18	137.1385	7.468e-10	***
Group:interface	18.91	1	92.062	18	3.6965	0.07050	.
task:interface	43.57	3	199.787	54	3.9254	0.01316	*
Group:task:interface	19.77	3	199.787	54	1.7811	0.16172	

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Table 6.7: Mauchly’s Test for Sphericity

Effect	Mauchly’s W	p-value
task	0.71117	0.33735
Group:task	0.71117	0.33735
task:interface	0.49390	0.03803 *
Group:task:interface	0.49390	0.03803 *

Table 6.8: Greenhouse-Geisser and Huynh-Feldt Corrections for Departure from Sphericity

Effect	GG eps	Pr(>F[GG])	Signif.	HF eps	Pr(>F[HF])	Signif.
task	0.82213	8.7e-06	***	0.9626757	1.909998e-06	***
Group:task	0.82213	0.04127	*	0.9626757	3.284730e-02	*
task:interface	0.72717	0.02488	*	0.8312836	1.949036e-02	*
Group:task:interface	0.72717	0.17914		0.8312836	1.724412e-01	

Another metric related to user behavior that we can point out in this study is the number of times that participants asked for a break during the tasks. The result was only one out of twenty participants asked for a break while doing a task, indicating that the overall workload imposed by the study was minimal. This interpretation is also supported by the high average satisfaction score of 4.9/5 on the final custom questionnaire item, “Considering all aspects of the study, how satisfied are you with your participation? (Scale: 1-5, with 1 being very dissatisfied and 5 being very satisfied),” corroborating the minimal burden imposed by the study.

Chapter 7

Discussion and Implications

7.1 Interpretation of the Results

The comparative evaluation between the Visual Query Builder (VQB) and the SPARQL interfaces in this study provides insightful findings regarding performance, accuracy, and user preference. These aspects are critical in understanding how interface design influences user interaction, task efficiency, and overall satisfaction within the context of database querying.

7.1.1 Time Efficiency and Accuracy

The results from this study highlighted the visual query builder's superior time efficiency over the SPARQL interface for task completion. On average, participants completed the tasks in a shorter time using the visual query builder rather than the SPARQL editor, highlighting the effectiveness of this interface. The exception to this trend, observed in task 2 and with one participant (P3) being quicker using the SPARQL interface, suggests that there may be individual differences in how users adapt to and excel with different interface designs. Moreover, the result of the applied ANOVA test suggests a significant effect of the interface:task which means that the efficiency of the interface type on the time efficiency could vary based on the task requirements and complexity level. In this study, all of the participants did all the tasks correctly and none of the participants skipped their tasks unfinished. However, the number of errors and hints to have been measured and analyzed, and based on the analysis of the t-test, the result was significant; meaning participants using the SPARQL interface were more prone to errors and asked for more hints compared to the VQB. Using the SPARQL query editor, the common prominent errors among participants include syntax errors, type definition (filtering the vessel types), and UNION operation. These types of errors were not as common in the VQB interface;

meaning that the VQB interface mitigates these error types in the formulation of the equivalent query. However, the VQB interface was not entirely free of errors, as the most common error in the VQB was connecting the irrelevant nodes, and including all types of vessels. In a nutshell, from the results of this study, we can claim that a visual query builder can generally help novice people formulate their maritime-related queries more accurately and quickly. The study result also represented that the performance metrics exhibited less variability in the VQB interface compared to the SPARQL, mitigating the impact of external variables. To clarify and expand this claim, we can tell that we have used a population size with various levels of expertise and experiences, among other potential influencing factors, however, the analysis of metrics such as task completion time, frequency of reference consultation, and hints requested showed us that the textual editor had more variate results than the VQB. This can be interpreted as using VQB leads to less volatile results and more consistency in terms of accuracy and performance.

7.1.2 User Experience and Preferences

Workload and Usability

The statistical analysis of the NASA-TLX workload suggested that the perceived workload on the participants was highly less while they were using the VQB interface compared to the SPARQL query. This argument is supported by the result of the custom questionnaire which most of the participants mentioned the simplicity and ease of use of the interface and the implemented interactions of it. P1 particularly mentioned that working with the VQB interface feels like they are playing a game rather than querying. Another interesting finding from this study is that the perceived workload using the SPARQL editor interface in group A was significantly less than in group B. In other words, when participants completed a task using the VQB interface and then proceeded to formulate the same query in the SPARQL editor, they perceived less cognitive workload on the textual editor. However, the same argument does not apply to the VQB interface, meaning that participants who did the tasks first using the SPARQL interface, did not have a less perceived workload using the VQB. The line chart in Figure 6.9 represents this claim by comparing the *B.SPARQL* (the orange line) and the *A.SPARLQ* (the blue line). By analyzing the result of the t-test usability score of the interfaces, the VQB interface achieved a significantly higher score than the SPARQL interface. The VQB interface was placed in the B-tier (excellent range) and the SPARQL interface was placed in the F-tier (OK range). However, there is a key aspect to consider about the score of the system usability score. The system usability score is usually conducted to analyze the usability performance of a system, and then the score is compared with the given metrics published in the other research. For example, the average SUS score mentioned in the [8] study has been determined as 68/100. In our research, since we have collected the SUS questionnaire for both interfaces, it conveyed the comparison between interfaces and it possibly followed by the novelty effect; meaning that the participant might have given inflated scores for the newer interface (VQB), while potentially undervaluing the traditional textual interface. Such inflated scores for the newer interface and deflated scores for the traditional one suggest that the responses might be affected by the freshness of the experience rather than a measured evaluation of usability. The result of the custom questionnaire can support this argument since there were not many reports of usability problems with the SPARQL, while we have received positive feedback for the intuitiveness of both interfaces. Supporting this claim further, some participants attributed a low physical workload score to tasks performed with the textual interface (i.e. they perceived a high physical workload), even though these tasks did not require significant physical effort.

User Preferences

The fifth task of this user study was designed to determine what interface participants would choose to do their tasks on. 17 participants out of 20 preferred the VQB interface over the SPARQL interface. This result was aligned with the custom questionnaire result that the majority of the participants mentioned the VQB interface was an intuitive and visually appealing interface. Some participants explicitly mentioned that the shape and colors of the nodes in the VQB interface helped them to recognize the query and speed up their querying process.

Those who preferred the SPARQL interface over VQB mentioned that the SPARQL interface can handle more complex queries and is more robust. Moreover, they have mentioned that the auto-completion feature enhanced their querying process. Interestingly, one of the participants who preferred the SPARQL interface also performed better with this interface. It is worth mentioning that none of the participants in this study were familiar with the SPARQL querying and they were non-experts in maritime querying. This observation leads us to propose a new research question: assessing the effectiveness of the VQB interface for maritime querying among experts. By involving experts, we can further evaluate the VQB interface and compare it to the SPARQL baseline.

Although participants in the open-ended questionnaire mentioned that both interfaces were user-friendly and intuitive, they reported plenty of usability issues and missing features that might help enhance both interfaces. The suggestions were around changing the layout of the elements in the interface, the ability to query all vessel types in VQB instead of manually adding them one by one, the shortcuts such as duplication of nodes for the VQB interface, and Vim textual editor's shortcuts for the SPARQL interface.

7.2 Implications of the Findings

The result of this study highlighted the effectiveness of the visual query builder interface enriched with direct manipulation techniques and tailored for maritime querying. This study also showed that this interface can boost performance and accuracy while reducing the variability of time efficiency, accuracy, and perceived workload compared to a traditional textual query editor. This decreased fluctuation results in a more uniform and predictable performance, particularly for novice users of the system. Following this section, the implications of the findings from the study will be elaborated upon in different aspects and categories.

Interface Design and HCI

Based on this research, we can convey that the performance and user satisfaction of the interface designs can be boosted by leveraging proper visual elements and direct manipulation; especially when the target users are novices and new to specific technical domains of the system. Therefore, this study advocates designers to provide intuitive interactions to ensure ease of use, time efficiency, accuracy, and consistency.

The less perceived workload of the VQB also encourages the proper and not overwhelming way to use visual elements in the design of query systems. This can help novice users to have a better transition to a more complex query system.

Education and Training

While this research emphasized the effectiveness of the visual query builders, we do not think that these interfaces can fully and completely substitute the traditional, and more complex interfaces such as textual querying systems. Several considerations inform this perspective:

Firstly, experienced users of traditional and more complex interfaces such as textual querying systems may find it difficult to switch their main querying interface since they might be more comfortable with the interface with which they have gained experience over time, regardless of the potential benefits of the visual interfaces that they can offer. This tendency was observed in this study and can support this argument. Some of the participants preferred the textual querying interface while they performed generally better with the visual interface.

Secondly, the implementation of the visual query systems has its own difficulties and challenges; since they need compilers to be implemented that translate visual queries to a query that is interpretable by the database system. Any changes to the visual query builder or adding querying features to it should be reflected on the compiler as well, therefore we can imply that scaling the visual query builder systems is time-consuming and is not challenge-free.

Despite these challenges, this research highlights a noteworthy advantage of integrating visual query interfaces into the query formulation process. Especially, users who engaged with a visual query builder interface (VQB) before using a textual interface demonstrated a marked improvement in workload scores; i.e. resulting in less perceived workload. This improvement can be attributed to the visual interface's ability to make the user's mental model ready and engage their visual memory, resulting in a smoother transition to the textual interface and enhancing overall efficiency. Therefore, based on these findings, we strongly advocate for the integration of graph-based visual querying systems into educational and training programs.

7.3 Limitations and Future Research Directions

One of the key limitations of this study is that the participants with prior familiarity with SQL querying were recruited, and none of the participants had prior knowledge and experience with SPARQL querying methods. Therefore, the study results cannot be generalized, and we cannot infer that the VQB interface will always outperform textual editors in terms of accuracy, performance, and preferences. Therefore, conducting a comparative study with tighter controls on participants' recent experience with SPARQL querying is recommended. Such a study would allow for a more detailed analysis of the impact this variable has on the comparative metrics of interface effectiveness.

Another consideration of this research is that in the result of this comparative study, we have observed a novelty effect, especially for the subjective metrics; resulting in lower scores in workload, system usability, and preference tests. However, the novelty effect is not permanent and can be reduced over time [50]. To better understand the long-term effects of using visual query builders like VQB, longitudinal studies could be conducted to assess how user preference rates evolve over time as the novelty effect reduces.

In this study, the designated querying tasks were relatively straightforward and didn't require much hard work. The fact that only one participant out of 20 asked for a break, and on average, participants have completed their tasks in 4.48 minutes would support this claim. Therefore, one suggestion for future study designs is to consider some overwhelming tasks to comparatively measure the effectiveness of the interfaces as the task's difficulty increases significantly.

Regarding the VQB interface, it is worth mentioning that the current version does not support all types of queries for maritime data, such as queries involving the relationship between multiple areas (e.g., intersect, contain) or other types of compound queries that SPARQL supports, such as NOT or XOR operands. While our results demonstrate the effectiveness of the VQB for non-experts, it would not be advisable to substitute textual querying interfaces due to the limitations of the VQB for constructing complex queries. However, this study advocates the complementary usage of text and VQB interfaces, benefiting from the VQB's engagement of the visual memory of participants and the capability of textual querying interfaces to handle complex queries.

Lastly, in this research, we have highlighted the effectiveness of the visual query builder interface and the role of engaging the visual memory. A question that emerged from our research concerns the impact of integrating visual aids: how do additive visuals and informative visuals to textual interfaces increase accuracy, performance, and user satisfaction? To explore this, future research could investigate a hybrid interface that combines traditional textual query editors with visual components, like a live graph-based visualization acting as a mini-map alongside the textual interface, or some visual components that visualize some part of a querying text as the user hovers their mouse on, such as visual thumbnail of the geoAreas and temporal values.

Chapter 8

Conclusion, Contributions, and Future Work

8.1 Summary of Contributions

In this research, we have presented and implemented a system architecture that allows users to interact collaboratively with the system to explore maritime data, with novel user interactions and direct manipulation techniques. This system architecture includes a tabletop device for displaying the map as the core of the visualization. It uses a tangible cube, equipped with a QR code underneath, which the tabletop can read to determine the cube's position and orientation. This allows the visualization map to display a ring menu exactly where the user places the cube. This system also allows multiple AR clients to connect to it and augment maritime visualizations in a space-time cube format. For query formulation, any device that supports a web browser can connect to the system and run our prototype visual query builder. This architecture has been published as a conference paper titled "Design of an Extended Reality Collaboration Architecture for Mixed Immersive and Multi-surface Interaction" [43]. The architecture and system prototype published in this article will assist mixed reality designers and researchers in developing innovative visualizations within immersive environments. By leveraging multiple devices, these visualizations can enhance decision-making processes across various domains, including simulation, education, gaming, and analysis.

During this research, the proposed architecture system has been implemented in a modular way that helps with the scalability of this project, as we have modeled, designed, and developed various function components that are required for maritime data exploration, maintenance, and observation. These components include back-end services and APIs, that seamlessly work with the database, with a universal format.

These components range from features facilitating spatial querying and temporal filtering, as well as various data retrieval functionalities, to specialized maritime operations such as detecting route similarities, simplifying distance calculations, streamlining data processing, AR and table-top integration components, etc. The components developed in this research can be used as building blocks of other maritime-related studies and contribute to future research, as at the time of writing this thesis, one graduate student is utilizing these components in their research.

Subsequently, this research focused on implementing and evaluating visual query builders for the maritime context. We have proposed a graph-based visual query builder, tailored for maritime operations, designed to allow non-experts to formulate their maritime queries without the necessity of learning a query language. Following the implementation of the earlier version of the visual query builder, two cognitive walk-through sessions were conducted by novice users with different levels of interface design expertise, and two pairs of expertise in the maritime domain. Although the sessions yielded encouraging outcomes and positive feedback, cognitive walkthrough facilitators identified several user experience and usability issues. These concerns were systematically addressed and improved through iterative refinements of the querying model, as well as enhancements to the visual components and user interactions.

We conducted a user study with 20 novice participants in order to evaluate the effectiveness of the visual query builder on maritime query formulation, and compare it with the SPARQL query formulation as an example of a traditional querying method for maritime querying. A web-based textual query editor has been implemented for this study. This interface has been enriched with auto-completion features, helper tools and functions such as custom geospatial filtering functions, and SPARQL query templates to facilitate the querying experience of the participants. These enhancements to the editor were implemented specifically to ensure a fair comparison between the interfaces. In this user study, we have compared the performance, accuracy, and user satisfaction of using the visual query builder over the SPARQL textual editor interface. The study's findings indicated that participants learned to use the visual query builder interface more swiftly and completed their tasks more quickly compared to their performance with the SPARQL interface. Participants had less difficulty working with the visual query builder and this led to requiring less help from the study facilitator compared to the SPARQL interface. Furthermore, the data on accuracy and time efficiency revealed that the visual query builder provided participants with a more stable and consistent experience, characterized by fewer variations in results.

The findings, on accuracy and efficiency when using the visual query builder emphasize its role in ensuring a more reliable user experience. This stability is particularly important for beginners as it helps reduce the learning curve and decrease the need for help. Additionally, assessments like NASA-TLX and SUS indicate a perceived workload and higher usability score, with the visual querying interface. Participants who initially used the visual query builder before the textual interface reported less perceived workload, implying that this interface aids in building users' mental models and leveraging their memory effectively. As a result, users are better prepared to navigate querying interfaces and their language; making it easier to transition to complex querying systems smoothly.

By analyzing the user satisfaction questionnaire results we can infer that although more participants chose the VQB interface over the SPARQL interface, the results were mixed and some participants preferred the SPARQL querying system. Additionally, most of the participants expressed their ideas and recommendations for improving both interfaces, which is valuable information and can be considered either for enhancing the interfaces or as a consideration for the future implementation of similar systems. At the time of writing this thesis, we are in the process of preparing a paper to publish the results of this study as well as the insights we have obtained through the journey of the design, implementation, and iterative enhancements of the visual query builder. We believe that this publication would considerably help other researchers in the visual querying domain and system designers who want to refine and improve their existing interfaces, moreover, it serves as a valuable resource for guiding future developments and innovations in visual query systems.

8.2 Recommendations for Future Research

Although there are numerous studies in the field of maritime data visualization and querying including this research, various aspects of further exploration and development still remain in this topic. This section outlines ideas and recommendations for future studies. These suggestions aim to address the current study's limitations and provide some research questions that arose during the research.

The popularity of mixed reality has surged in recent years. Additionally, the tools and equipment supporting augmented reality (AR) and virtual reality (VR) have seen significant enhancements, offering more features than ever before, such as precise hand-tracking, eye-tracking, and spatial recognition. Therefore, exploration of the direct manipulation techniques in mixed reality for maritime querying is highly recommended. It is worth investigating the effectiveness of the tools and features that mixed reality offers for maritime querying since they provide an immersive environment and also, maritime querying context relates directly to spatial querying and filtering. However, based on what we experienced in this research, it is highly recommended to refine and simplify the data, such as vessel routes polyline simplification, or any boundaries on the map such as restricted areas before projection in a virtual reality or augmented reality environment; since the AR/VR devices have a limited computational power since having a limited battery capacity. In addition to the simplification, drilling-down, and filtering techniques greatly help in this matter. For instance, instead of visualizing numerous vessels and their routes, limiting the number of vessels, and the time range of their routes will significantly reduce the stutters during the mixed reality experience, as well as enhance the visibility by focusing attention on key data points and reducing visual clutter. This selective display approach can lead to a more streamlined and effective user experience, where critical information is highlighted and easier to interpret within the immersive environment.

8.3 Final Remarks

To conclude my master's thesis, the study that we conducted showed that the implemented VQB interface can greatly enhance maritime querying for novices. The majority of the participants favored the VQB interface over the SPARQL interface due to its ease of use, and intuitiveness. However, some participants preferred the SPARQL interface due to its capabilities of handling complex queries and its robustness. Future user studies can be conducted to analyze the effectiveness of the VQB interfaces for maritime experts. Overall, this thesis establishes a foundation for future developments in visual data querying, particularly in specialized areas like maritime data analysis.

Bibliography

[1] Aggregation pipeline - mongodb manual. https://www.mongodb.com/docs/manual/core/aggregation-pipeline/. (Accessed on 02/18/2024).

[2] Protected Seas. https://protectedseas.net/, 2023. Accessed on March 31, 2023.

[3] Vessel Traffic Data. https://marinecadastre.gov/ais/, 2023. Accessed on March 31, 2023.

[4] Oszkár Ambrus, Knud Möller, Siegfried Handschuh, et al. Konduit vqb: a visual query builder for sparql on the social semantic desktop. In *Workshop on visual interfaces to the social and semantic web*, 2010.

[5] Natalia Andrienko and Gennady Andrienko. Visual analytics of vessel movement. *Guide to Maritime Informatics*, pages 149–170, 2021.

[6] Marilyn Hughes Blackmon, Peter G Polson, Muneo Kitajima, and Clayton Lewis. Cognitive walkthrough for the web. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 463–470, 2002.

[7] John Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.

[8] John Brooke. Sus: a retrospective. *Journal of Usability Studies*, 8:29–40, 01 2013.

[9] Howard Butler, Martin Daly, Allan Doyle, Sean Gillies, Stefan Hagen, and Tim Schaub. The geojson format. Technical report, 2016.

[10] D Calcinelli and Michel Mainguenaud. Cigales, a visual query language for a geographical information system: the user interface. *Journal of Visual Languages & Computing*, 5(2):113–132, 1994.

[11] Valéria MB Cavalcanti, Ulrich Schiel, and Cláudio de Souza Baptista. Querying spatio-temporal databases using a visual environment. In *Proceedings of the working conference on advanced visual interfaces*, pages 412–419, 2006.

[12] Kārlis Čerāns, Agris Šostaks, Uldis Bojārs, Juris Bārzdiņš, Jūlija Ovčiņņikova, Lelde Lāce, Mikus Grasmanis, and Artūrs Sproģis. Viziquer: a visual notation for rdf data analysis queries. In *Metadata and Semantic Research: 12th International Conference, MTSR 2018, Limassol, Cyprus, October 23-26, 2018, Revised Selected Papers 12*, pages 50–62. Springer, 2019.

[13] C. Claramunt, C. Ray, L. Salmon, E. Camossi, M. Hadzagic, A-L. Jousselme, G. Andrienko, N. Andrienko, Y. Theodoridis, and G. Vouros. Maritime data integration and analysis: Recent progress and research challenges. *Advances in Database Technology - EDBT*, 2017-M:192–197, January 2017.

[14] Alex Endert, Lauren Bradel, and Chris North. Beyond control panels: Direct manipulation for visual analytics. *IEEE Computer Graphics and Applications*, 33(4):6–13, 2013.

[15] Barrett Ens, Sarah Goodwin, Arnaud Prouzeau, Fraser Anderson, Florence Y. Wang, Samuel Gratzl, Zac Lucarelli, Brendan Moyle, Jim Smiley, and Tim Dwyer. Uplift: A tangible and immersive tabletop system for casual collaborative visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1193–1203, 2021.

[16] Michele Fiorini, Andrea Capata, and Domenico D Bloisi. Ais data visualization for maritime spatial planning (msp). *International Journal of e-Navigation and Maritime Economy*, 5:45–60, 2016.

[17] Kenneth P. Fishkin, Anuj Gujar, Beverly L. Harrison, Thomas P. Moran, and Roy Want. Embodied user interfaces for really direct manipulation. *Commun. ACM*, 43(9):74–80, sep 2000.

[18] Furkan Goz, Alev Mutlu, and Orhan Akbulut. Analysis of ramer-douglas-peucker algorithm as a discretization method. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, 2018.

[19] F. Guerroudji-Meddah, H. Belbachir, and R. Laurini. Envis: A visual environment for gis querying. In *2009 International Conference on Multimedia Computing and Systems*, pages 68–71, 2009.

[20] Florian Haag, Steffen Lohmann, Stephan Siek, and Thomas Ertl. Queryvowl: A visual query notation for linked data. In Fabien Gandon, Christophe Guéret, Serena Villata, John Breslin, Catherine Faron-Zucker, and Antoine Zimmermann, editors, *The Semantic Web: ESWC 2015 Satellite Events*, pages 387–402, Cham, 2015. Springer International Publishing.

[21] Asaad Hakeem, Mun Wai Lee, Omar Javed, and Niels Haering. Semantic video search using natural language queries. In *Proceedings of the 17th ACM International Conference on Multimedia*, MM '09, page 605–608, New York, NY, USA, 2009. Association for Computing Machinery.

[22] Abbas Harati-Mokhtari, Alan Wall, Philip Brooks, and Jin Wang. Automatic identification system (ais): Data reliability and human error implications. *The Journal of Navigation*, 60(3):373–389, 2007.

[23] Sandra G Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.

[24] Wei He, Jinyu Lei, Xiumin Chu, Shuo Xie, Cheng Zhong, and Zhixiong Li. A visual analysis approach to understand and explore quality problems of ais data. *Journal of Marine Science and Engineering*, 9(2), 2021.

[25] Wei He, Jinyu Lei, Xiumin Chu, Shuo Xie, Cheng Zhong, and Zhixiong Li. A visual analysis approach to understand and explore quality problems of ais data. *Journal of Marine Science and Engineering*, 9(2), 2021.

[26] Kevin Patrick Helzel, Alexander Klaus, and Mathias Jahnke. Mixed reality maps to help convey disaster information. *Advances in Cartography and GIScience of the ICA*, 3:1–7, 2021.

[27] Zainul Hidayah. The importance and utilization of big data for indonesian maritime information. *Journal Asro*, 11(04):61–71, 2020.

[28] Mohaiminul Islam and Shangzhu Jin. An overview of data visualization. In *2019 International Conference on Information Science and Communications Technologies (ICISCT)*, pages 1–7. IEEE, 2019.

[29] Pan Jiacai, Jiang Qingshan, Hu Jinxing, and Shao Zheping. An ais data visualization model for assessing maritime traffic situation and its applications. *Procedia Engineering*, 29:365–369, 2012. 2012 International Workshop on Information and Electronics Engineering.

[30] Xander Keijser, Malena Ripken, Igor Mayer, Harald Warmelink, Lodewijk Abspoel, Rhona Fairgrieve, and Crawford Paris. Stakeholder engagement in maritime spatial planning: The efficacy of a serious game approach. *Water*, 10(6), 2018.

[31] Meriana Kobeissi, Nour Assy, Walid Gaaloul, Bruno Defude, Boualem Benatallah, and Bassem Haidar. Natural language querying of process execution data. *Information Systems*, 116:102227, 2023.

[32] Richard O. Lane, David. A. Nevell, Steven D. Hayward, and Thomas W. Beaney. Maritime anomaly detection and threat assessment. In *2010 13th International Conference on Information Fusion*, pages 1–8, 2010.

[33] Joohee Lee, Wan Chul Yoon, and Hyun Chung. Formal or informal human collaboration approach to maritime safety using fram. *Cognition, Technology & Work*, 22(4):861–875, 2020.

[34] Bao Lei. The data visualization of large scale ais trajectories data on hadoop. In *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pages 758–761, 2020.

[35] Yan Li, Ryan Wen Liu, Jingxian Liu, Yu Huang, Bin Hu, and Kai Wang. Trajectory compression-guided visualization of spatio-temporal ais vessel density. In *2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–5, 2016.

[36] Ryan Wen Liu, Yu Guo, Jiangtian Nie, Qin Hu, Zehui Xiong, Han Yu, and Mohsen Guizani. Intelligent edge-enabled efficient multi-source data fusion for autonomous surface vehicles in maritime internet of things. *IEEE Transactions on Green Communications and Networking*, 6(3):1574–1587, 2022.

[37] Steven C Mallam, Salman Nazir, and Sathiya Kumar Renganayagalu. Rethinking maritime education, training, and operations in the digital era: Applications for emerging immersive technologies. *Journal of Marine Science and Engineering*, 7(12):428, 2019.

[38] John W. Mauchly. Significance test for sphericity of a normal n-variate distribution. *The Annals of Mathematical Statistics*, 11(2):204–209, 1940.

[39] Fabio Mazzarella, Michele Vespe, Alfredo Alessandrini, Dario Tarchi, Giuseppe Aulicino, and Antonio Vollero. A novel anomaly detection approach to identify intentional ais on-off switching. *Expert Systems with Applications*, 78:110–123, 2017.

[40] Daphne Miedema and George Fletcher. Sqlvis: Visual query representations for supporting sql learners. In *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–9, 2021.

[41] Jun Min Mou, Cees van der Tak, and Han Ligteringen. Study on collision avoidance in busy waterways by using ais data. *Ocean Engineering*, 37(5):483–490, 2010.

[42] Ricardo Cardoso Pereira, Pedro Henriques Abreu, Evgheni Polisciuc, and Penousal Machado. A data visualization approach for intersection analysis using ais data. In *VISIGRAPP (3: IVAPP)*, pages 208–215, 2019.

[43] Thiago Porcino, Seyed Adel Ghaeinian, Juliano Franz, Joseph Malloch, and Derek Reilly. Design of an extended reality collaboration architecture for mixed immersive and multi-surface interaction. In Barbara Göbl, Erik van der Spek, Jannicke Baalsrud Hauge, and Rod McCall, editors, *Entertainment Computing – ICEC 2022*, pages 112–122, Cham, 2022. Springer International Publishing.

[44] Ruslan Rydvanskiy and Nick Hedley. Mixed reality flood visualizations: reflections on development and usability of current systems. *ISPRS International Journal of Geo-Information*, 10(2):82, 2021.

[45] Bahador Saket, Samuel Huron, Charles Perin, and Alex Endert. Investigating direct manipulation of graphical encodings as a method for user interaction. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):482–491, 2020.

[46] Ana Carolina Salgado. Data engineering: Modeling and integration issues. 05 2008.

[47] Stacey D Scott, Antoine Allavena, Katherine Cerar, Glenn Franck, Mark Hazen, Ted Shuter, Chris Colliver, and SD Scott. Investigating tabletop interfaces to support collaborative decision-making in maritime operations. In *Proceedings of ICCRTS 2010: International Command and Control Research and Technology Symposium*, pages 22–24, 2010.

[48] Stacey D Scott, Antoine Allavena, Katherine Cerar, Glenn Franck, Mark Hazen, Ted Shuter, Chris Colliver, and SD Scott. Investigating tabletop interfaces to support collaborative decision-making in maritime operations. In *Proceedings of ICCRTS 2010: International Command and Control Research and Technology Symposium*, pages 22–24, 2010.

[49] Stacey D. Scott, M. Sheelagh T. Carpendale, and Kori Inkpen. Territoriality in collaborative tabletop workspaces. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, CSCW '04, page 294–303, New York, NY, USA, 2004. Association for Computing Machinery.

[50] Grace Shin, Yuanyuan Feng, Mohammad Hossein Jarrahi, and Nicci Gafinowitz. Beyond novelty effect: a mixed-methods exploration into the motivation for long-term activity tracker use. *JAMIA open*, 2(1):62–72, 2019.

[51] Sandeep Kumar Singh and Frank Heymann. Machine learning-assisted anomaly detection in maritime navigation using ais data. In *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 832–838, 2020.

[52] Seung Youb Ssin, James A. Walsh, Ross T. Smith, Andrew Cunningham, and Bruce H. Thomas. Geogate: Correlating geo-temporal datasets using an augmented reality space-time cube and tangible interactions. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 210–219, 2019.

[53] Zoran Stojanovic, Slobodanka Djordjevic-Kajan, and Dragan Stojanovic. Visual query and analysis tool of the object-relational gis framework. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 328–335, 2000.

[54] Tanja Svarre and Tony Russell-Rose. Think outside the search box: A comparative study of visual and form-based query builders. *Journal of Information Science*, 0(0):01655515221138536, 0.

[55] Floris van den Oever, Morten Fjeld, and Bjørn Sætrevik. A systematic literature review of augmented reality for maritime collaboration. *International Journal of Human–Computer Interaction*, pages 1–16, 2023.

[56] J.J. van Wijk. The value of visualization. In *VIS 05. IEEE Visualization, 2005.*, pages 79–86, 2005.

[57] Hernán Vargas, Carlos Buil-Aranda, Aidan Hogan, and Claudia López. Rdf explorer: A visual sparql query builder. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I 18*, pages 647–663. Springer, 2019.

[58] Guillermo Vega-Gorgojo, Laura Slaughter, Martin Giese, Simen Heggestøyl, Ahmet Soylu, and Arild Waaler. Visual query interfaces for semantic datasets: An evaluation study. *Journal of Web Semantics*, 39:81–96, 2016.

[59] Kai Wang, Maohan Liang, Yan Li, Jingxian Liu, and Ryan Wen Liu. Maritime traffic data visualization: A brief review. In *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*, pages 67–72, 2019.

[60] Niels Willems, Huub Van De Wetering, and Jarke J Van Wijk. Visualization of vessel movements. In *Computer Graphics Forum*, volume 28, pages 959–966. Wiley Online Library, 2009.

[61] Paweł WoŹniak, Margareta Lützhöft, Thomas Porathe, Asim Evren Yantac, and Morten Fjeld. Examining interactive surfaces for maritime operations. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, page 341–344, New York, NY, USA, 2013. Association for Computing Machinery.

[62] Mengtian Yin, Llewellyn Tang, Chris Webster, Shen Xu, Xiongyi Li, and Huaquan Ying. An ontology-aided, natural language-based approach for multi-constraint bim model querying. *Journal of Building Engineering*, 76:107066, 2023.

Appendix A

User study materials and instructions

A.1 Introduction to SPARQL Querying In Maritime Context

SPARQL Introduction

SPARQL (SPARQL Protocol and RDF Query Language) is used to query databases that store data in RDF (Resource Description Framework) format. RDF data consists of triples with a subject-predicate-object structure, representing how entities (subjects) are related to each other (predicate) with specific values or other entities (objects).

1. PREFIX:

The

`PREFIX` keyword in SPARQL is used to define a shorthand for longer URIs (Uniform Resource Identifiers), making queries easier to read and write.

Example:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

Here, `foaf:` is a prefix for the namespace `http://xmlns.com/foaf/0.1/`.

2. SELECT:

The

`SELECT` keyword specifies which variables to return in the results. Variables are indicated with a question mark `?`.

Example:

```
SELECT ?name ?email
```

This selects the values for `?name` and `?email` from the data.

3. WHERE:

The `WHERE` clause defines the patterns that SPARQL should match in the data. These patterns are written as triples.

Example:

```
WHERE {
  ?person foaf:name ?name .
  ?person foaf:mbox ?email .
}
```

This pattern matches any `?person` with a `foaf:name` and `foaf:mbox`.

Defining the Type of a Subject:

To specify the type of a subject, use the shorthand 'a', which is equivalent to `rdf:type`.

Example:

```
WHERE {
  ?fruit a example:apple .
}
```

This pattern matches any `?fruit` that is of type `example:apple`.

Using 'a' to Define Types:

The 'a' keyword can be used within a pattern to specify that the subject is a certain type of entity, as defined in an ontology or schema.

Example:

```
WHERE {
  ?person a foaf:Person .
  ?person foaf:name ?name .
  ?person foaf:mbox ?email .
}
```

In this example, the query will return results where `?person` is of type `foaf:Person`, and for each person, their name and email are returned.

4. FILTER: `FILTER` is used to restrict the results based on given criteria.

Example:

```
FILTER (?age > 18)
```

This filters the results to include only those where `?age` is greater than 18.

5. UNION: `UNION` allows for the combination of multiple query patterns. The results will include matches for either pattern.

Example:

```
{
  ?person foaf:name ?name .
} UNION {
  ?person foaf:mbox ?email .
}
```

This will match `?person` with either a `name` or an `email`.

Operators:

- `&&` (AND): Combines multiple conditions, all of which must be true.
- `||` (OR): Combines conditions, at least one of which must be true.

Use of Variables:
Variables are placeholders for values and are prefixed with a `?`. They are used to select and manipulate data.

Example:

```
?person foaf:age ?age
```

This matches a person's age to the variable `?age`.

Combining PREFIX and WHERE:

By combining `PREFIX` with `WHERE`, you can simplify the pattern matching in your query.

Example:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name ?email
WHERE {
  ?person foaf:name ?name .
  ?person foaf:mbox ?email .
}
```

This selects the name and email of every person in the dataset.

Example 1: Selecting a Person Named Alex Over 5 Years Old

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?person

WHERE {

  ?person foaf:age ?age ;
                  foaf:name ?name .

  FILTER (?age > 5 && ?name = "Alex")
}
```

In this query, both conditions — the person's name being "Alex" and their age being greater than 5 — must be met for a result to be returned.

Example 2: Combining People Named Alex Over 5 and Samantha Over 20

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?person
WHERE {
  ?person foaf:age ?age .
  ?person foaf:name ?name .

  {
    FILTER (?age > 5 && ?name = "Alex")
  }
  UNION
  {
    FILTER (?age > 20 && ?name = "Samantha")
  }
}
```

Result: Return persons with one of the two conditions: 1) having the name of Alex and also having the age more than 5. 2) having the name Samantha and having the age more than 20 years old.

In this query, the `FILTER` within each block of the `UNION` now checks both the name and age conditions. Only if both conditions are true will the individuals be selected in the result set. This demonstrates the use of the `&&` operator to require multiple conditions to be true simultaneously.

SPARQL in Maritime Context

Here we will introduce some glossaries and examples for writing SPARQL queries for maritime querying, especially, to use for the study.

ais Prefix

As a prefix, we will use an example ontology URI for this study.

```
PREFIX ais: http://example.org/ontology#
```

Resources that can be used using this URI are as follows:

Term	Description	Notes
ais:cargo	cargo vessel(s)	
ais:passenger	passenger vessel(s)	
ais:tanker	tanker vessel(s)	
ais:fishing	fishing vessels(s)	
ais:sog	Speed Over Ground of the vessel(s)	
ais:cog	Course Over Ground of the vessel(s)	
ais:timestamp	Timestamp of the vessel(s) tracks	Each ais track has a timestamp that indicates the time of the track.
ais:location	GeoLocation of the vessel(s) tracks	
ais:inside(?location , "Area ID")	Inside function	Checks if the ?location variable is inside the area or not. The first input is the variable that indicates the ais:location and the second input is ID of the area of interest in string format.
ais:outside(?location , "Area ID")	Outside function	Same as above, but checks if the tracks are outside the area or not.

xsd Prefix

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

Term	Description	Notes
“timestamp”^^xsd:dateTime	timestamp in string format readable by SPARQL	“YYYY-MM-DDTHH:mm:ssZ” → This is the right format for the times that we use in the string. Also known as ISO 8601 datetime format

How to use:

```
FILTER (?timestamp >= "2022-01-01T04:00:00Z"^^xsd:dateTime &&
          ?timestamp <= "2022-01-02T04:00:00Z"^^xsd:dateTime)
```

This filters the timestamps that are between two given date-times: `2022-01-01 04:00:00` and `2022-01-02 04:00:00`.

Vessel Types

To indicate vessel types, we can use `a` to indicate types.

```
WHERE {
  ?vessel a ais:cargo .
}
```

This pattern matches any `?vessel` that is of type `ais:cargo`.

Of course, this was merely a basic introduction to SPARQL querying, designed to provide participants with sufficient knowledge to engage in the study. Now, let's assemble the pieces of this puzzle and explore some examples of querying using SPARQL in a maritime context.

Example 1) select all fishing vessels and (tanker vessels with a weight > 5000 Kg)

```
PREFIX ais: <http://example.org/ontology#>

SELECT ?vessel
WHERE {

  ?vessel ais:weight ?MyWeightVariable .

  {
    ?vessel a ais:fishing .
  }
  UNION
  {
    ?vessel a ais:tanker .
    FILTER (?MyWeightVariable > 5000)
  }
}
```

Example 2) Select all cargo vessels between these times: `(2021-02-05 05:30:50)` and `(2023-05-04 15:45:00)` that have `SOG > 5 mph` and `COG < 20 degrees`.

```
PREFIX ais: <http://example.org/ontology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?vessel

WHERE {

  ?vessel ais:timestamp ?timestamp ;
          ais:sog ?MySOGVariable ;
                    ais:cog ?MyCOGVariable .
```

```
     FILTER (?timestamp >= "2021-02-05T05:30:50Z"^^xsd:dateTime &
               ?timestamp <= "2023-05-04T15:45:00Z"^^xsd:dateTime

  ?vessel a ais:cargo .
  FILTER (?MySOGVariable > 5  && ?MyCOGVariable < 20 )

 }
```

Example 3) select any vessels inside the area with ID: 2167

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ais: <http://example.org/ontology#>

SELECT ?vessel

WHERE {

  ?vessel ais:location ?vesselTrackGeoLocations .

  FILTER (ais:inside(?vesselTrackGeoLocations, "2167"))

}
```

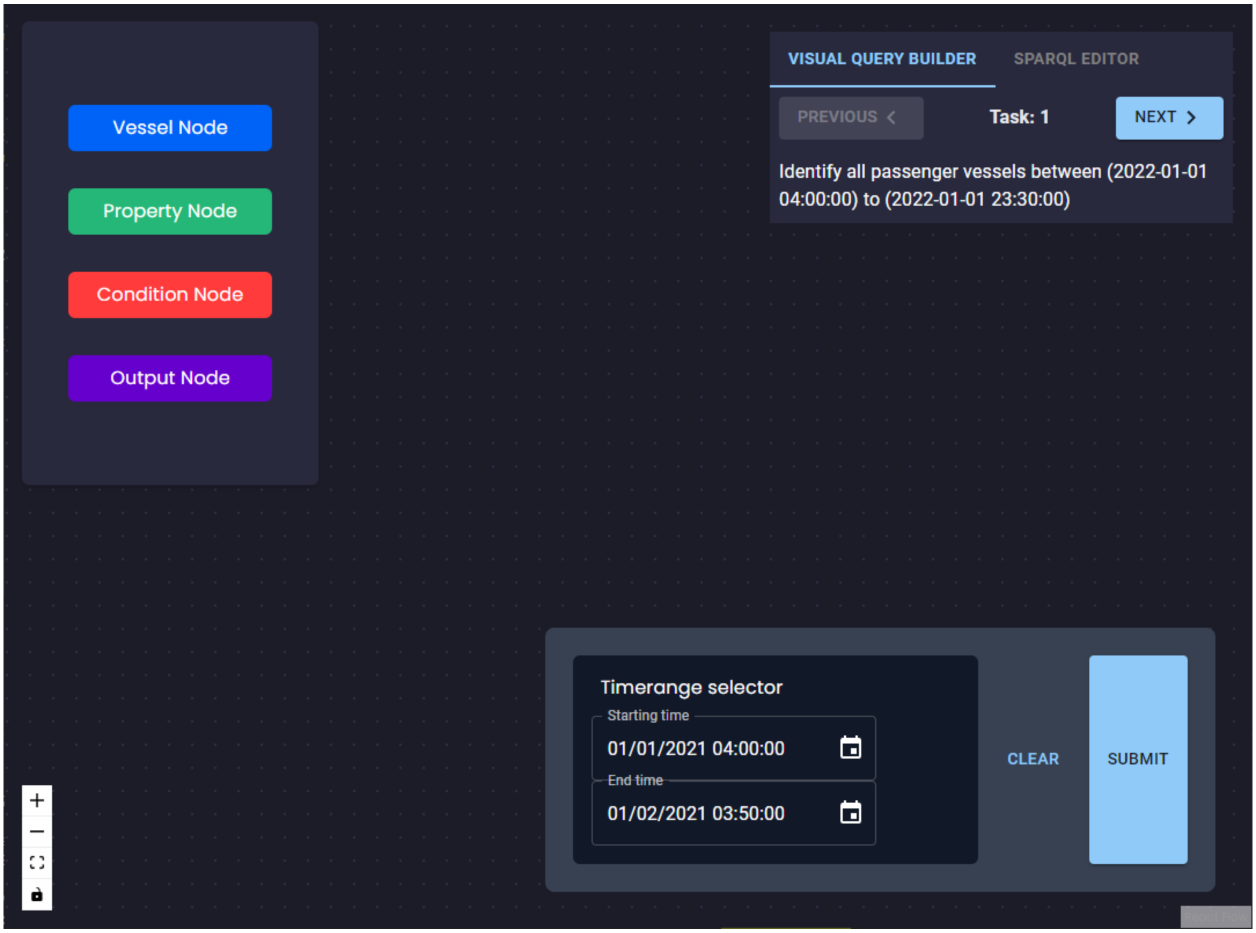
A.2 Introduction to Visual Query Builder (VQB) Interface

Visual Query Builder Introduction

This visual query builder is a prototype designed to form and generate maritime queries using visual elements such as sliders, nodes, and links.

Components

The figure bellow shows the default components of the prototype.



Identify all passenger vessels between (2022-01-01 04:00:00) to (2022-01-01 23:30:00)

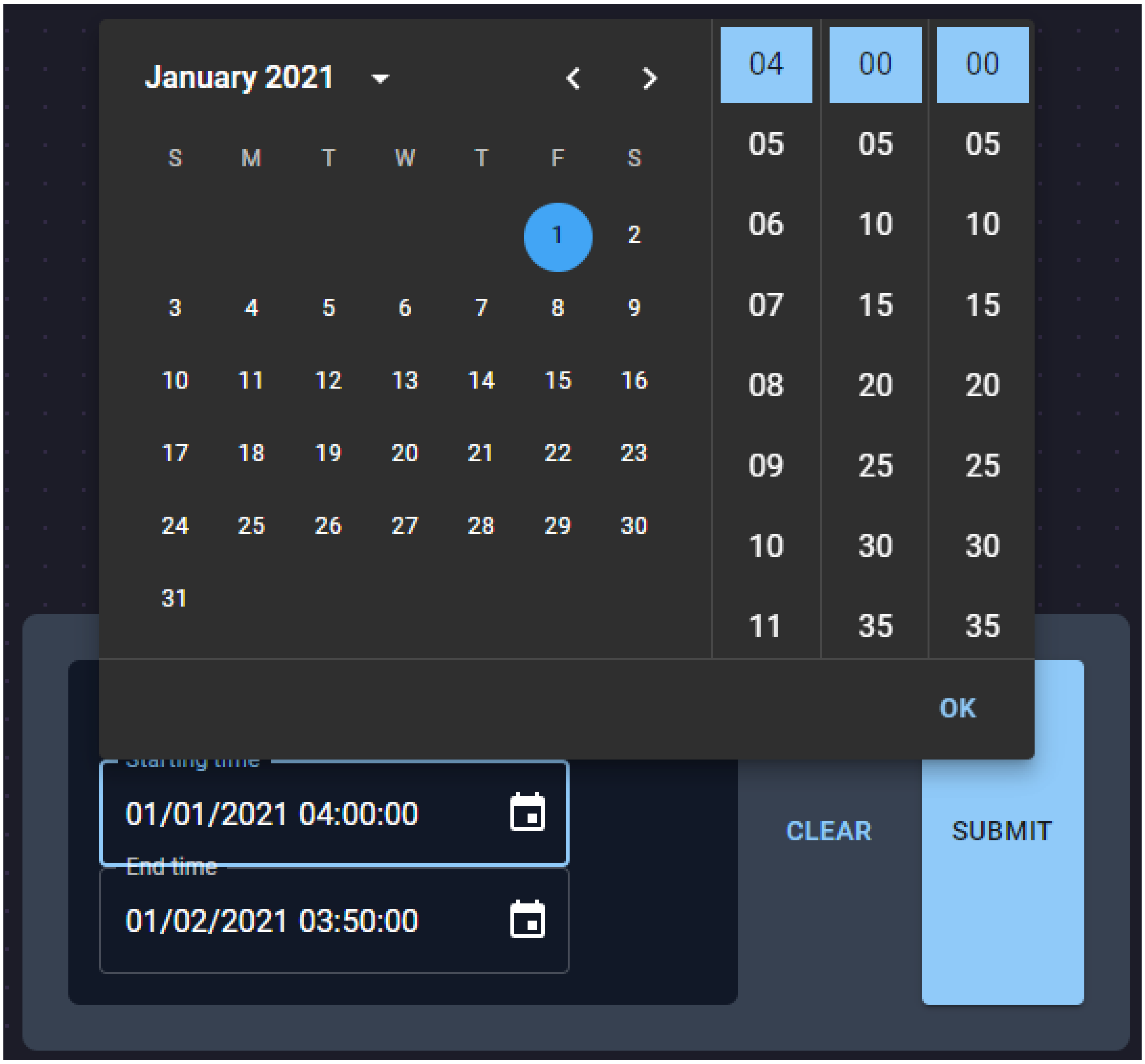
Timerange selector

Starting time: 01/01/2021 04:00:00

End time: 01/02/2021 03:50:00

Node Palette component) The Node Palette component is located at the top-left of the visual query builder. It allows you to create an instance of a node by simply dragging and dropping it onto the central working canvas.

Time Range Selector) This section, found at the bottom right of the visual query builder, includes two inputs: 1) starting time and 2) ending time. You can either directly input numbers by typing, or click on the calendar icon to the right of the input field to open the visual datetime selector (refer to the figure below).



January 2021

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

04	00	00
05	05	05
06	10	10
07	15	15
08	20	20
09	25	25
10	30	30
11	35	35

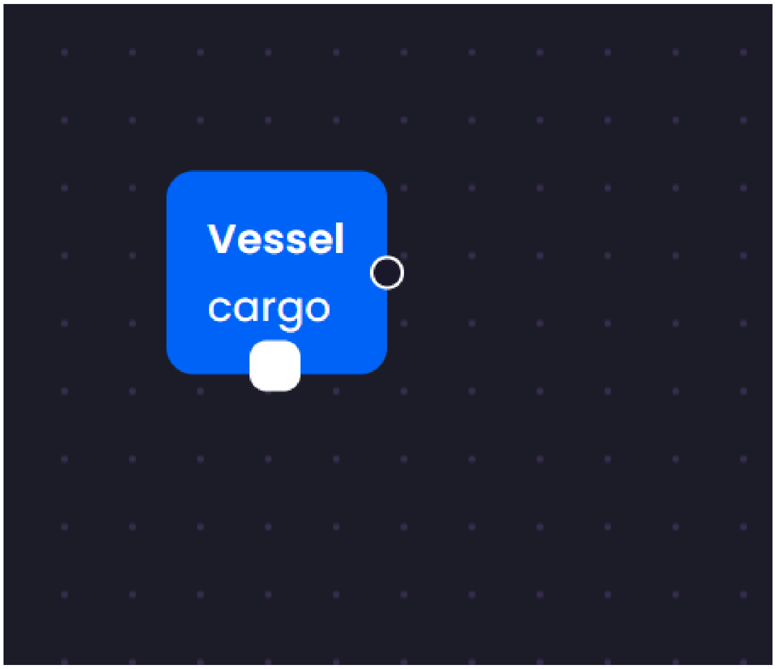
Starting time: 01/01/2021 04:00:00

End time: 01/02/2021 03:50:00

How It Works?

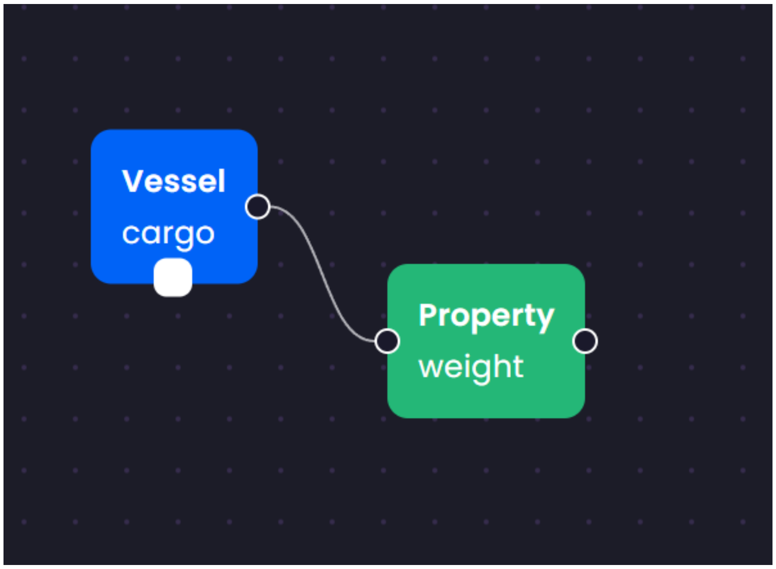
Vessel Node)

This visual query builder is specifically designed for maritime queries. As such, there is a vessel node in the nodes palette. When you drag and drop this node onto the canvas, you'll be asked to specify the vessel type for your query, such as 'Cargo vessels'.



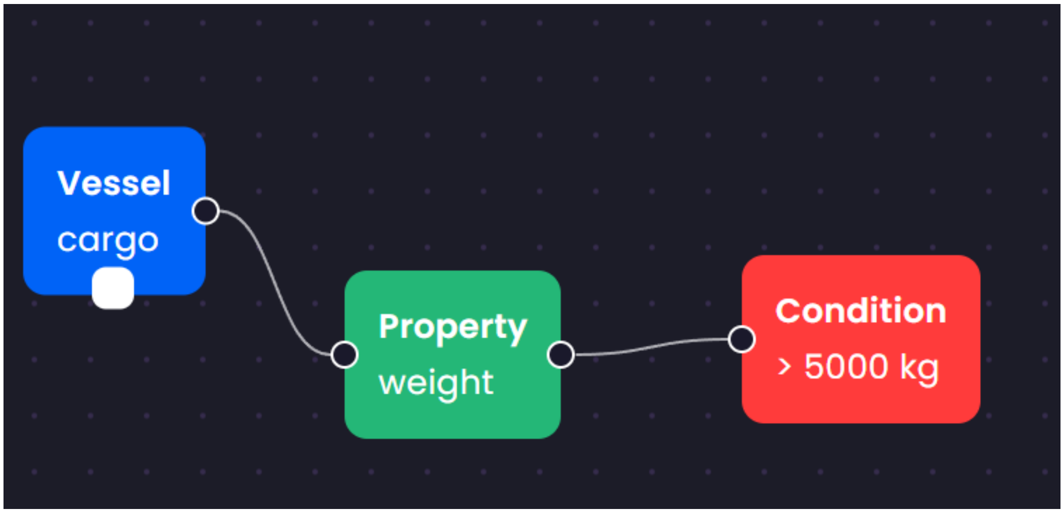
Property Node)

When you drag and drop a property node onto your canvas, you'll be prompted to select the property of the vessel tracks you're interested in. Options include position, SOG (speed over ground), COG (course over ground), weight, height, and route. Property nodes can be linked to vessel nodes to indicate the specific vessel property you want to explore. As an example, in the figure below, we're querying the weight of cargo vessels.



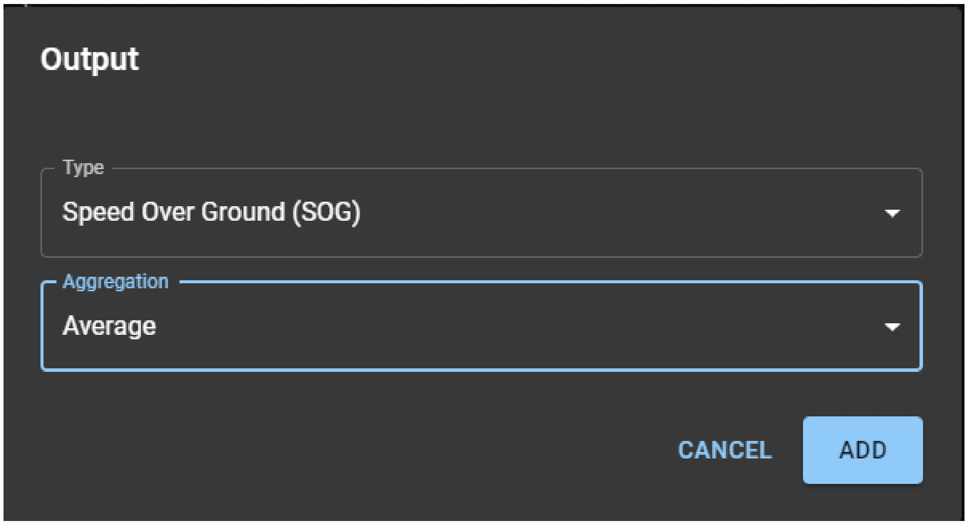
Condition Node)

Condition nodes are used to specify conditions for property nodes. For instance, if the property node is set to 'weight', you can add a condition that only includes cargo vessels weighing more than 5000 kg. Once you drag a condition node from the palette to the canvas, you can set the condition by adjusting the parameters in the pop-up. Condition nodes can be connected to property nodes.



Output Node)

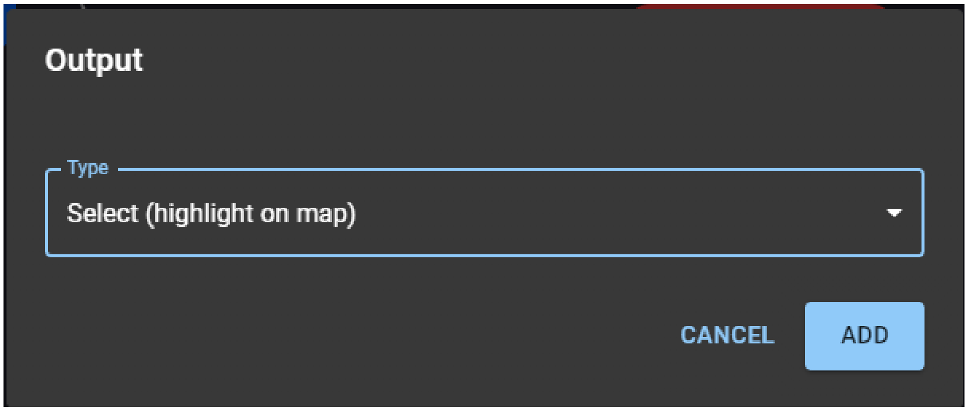
The output node, is for choosing the output type of the query that you have created. Options are: select (Highlight on the map) or you can select a property such as SOG (speed over ground), if the selected option is a property, you will be prompted what aggregation you want (average, min, max).


Output

Type: Speed Over Ground (SOG)

Aggregation: Average

Average SOG (speed over ground)

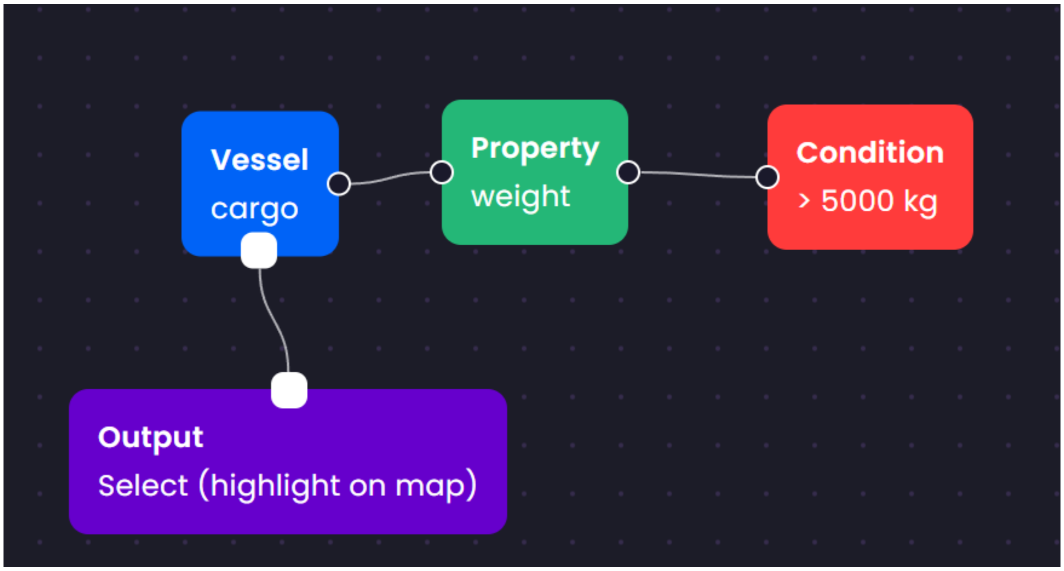


Output

Type: Select (highlight on map)

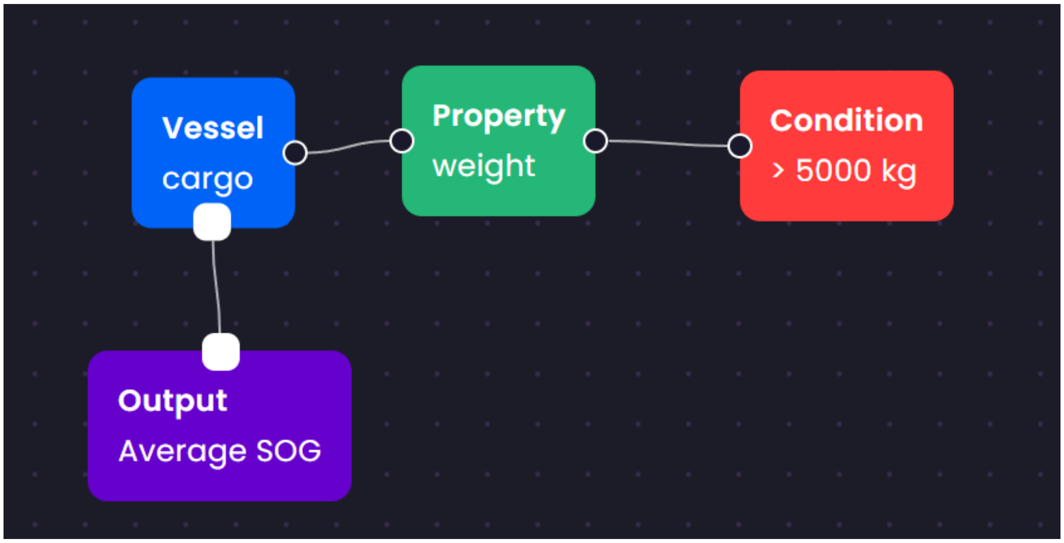
Selecting a vessel option. This output type allows you to see the highlighted selected vessel on he map visualization application

Output nodes can be linked to a vessel node, but only one output node can be connected at a time. The example below illustrates a query for selecting (highlighting on the map) cargo vessels that weigh more than 5000 kg.

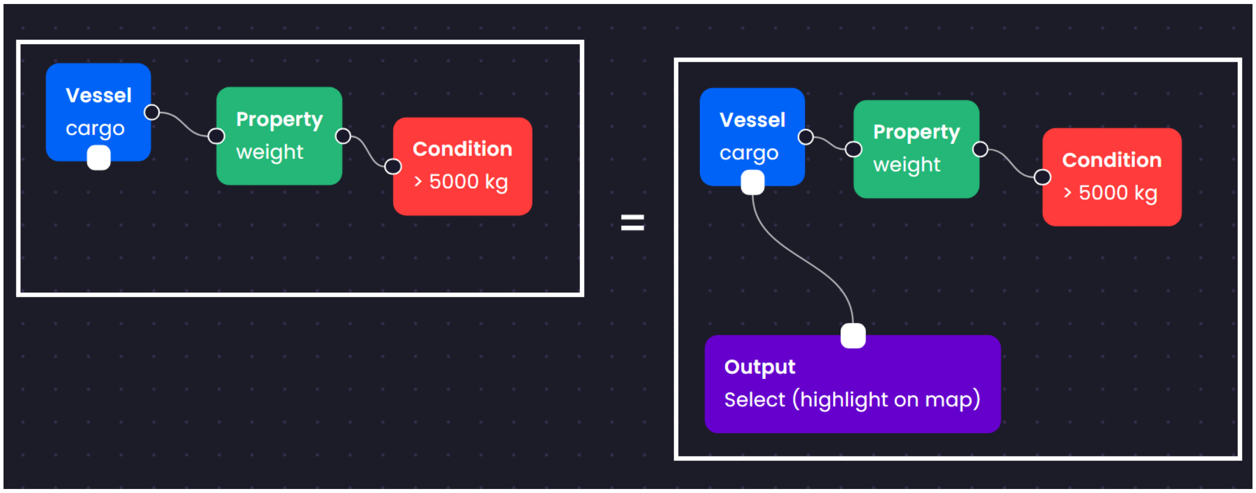


The example below demonstrates a query to calculate the average speed over ground (SOG) of cargo vessels that meet the condition of weighing more than

5000 kg. In this scenario, the query yields only a numerical result, without causing any visual updates in the map application.



Note: For simplicity, if you leave the output type of a vessel node unlinked, we consider it as a selection on the map. Therefore these two figures bellow will have the same result and both will highlight the cargo vessels having weight>5000 kg on the map application.

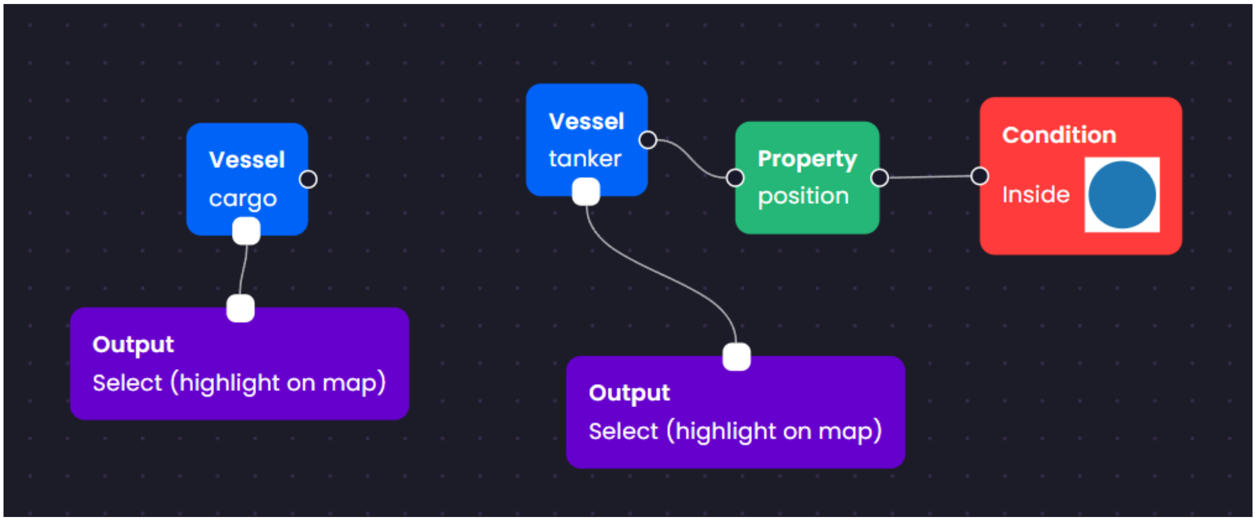


Updating/Deleting a Node

If you want to change any properties of a node already on your canvas, tap on the node. A pop-up will appear asking for the new properties. After you've made your changes, press UPDATE button. For deleting a node, after clicking on the node, press DELETE button.

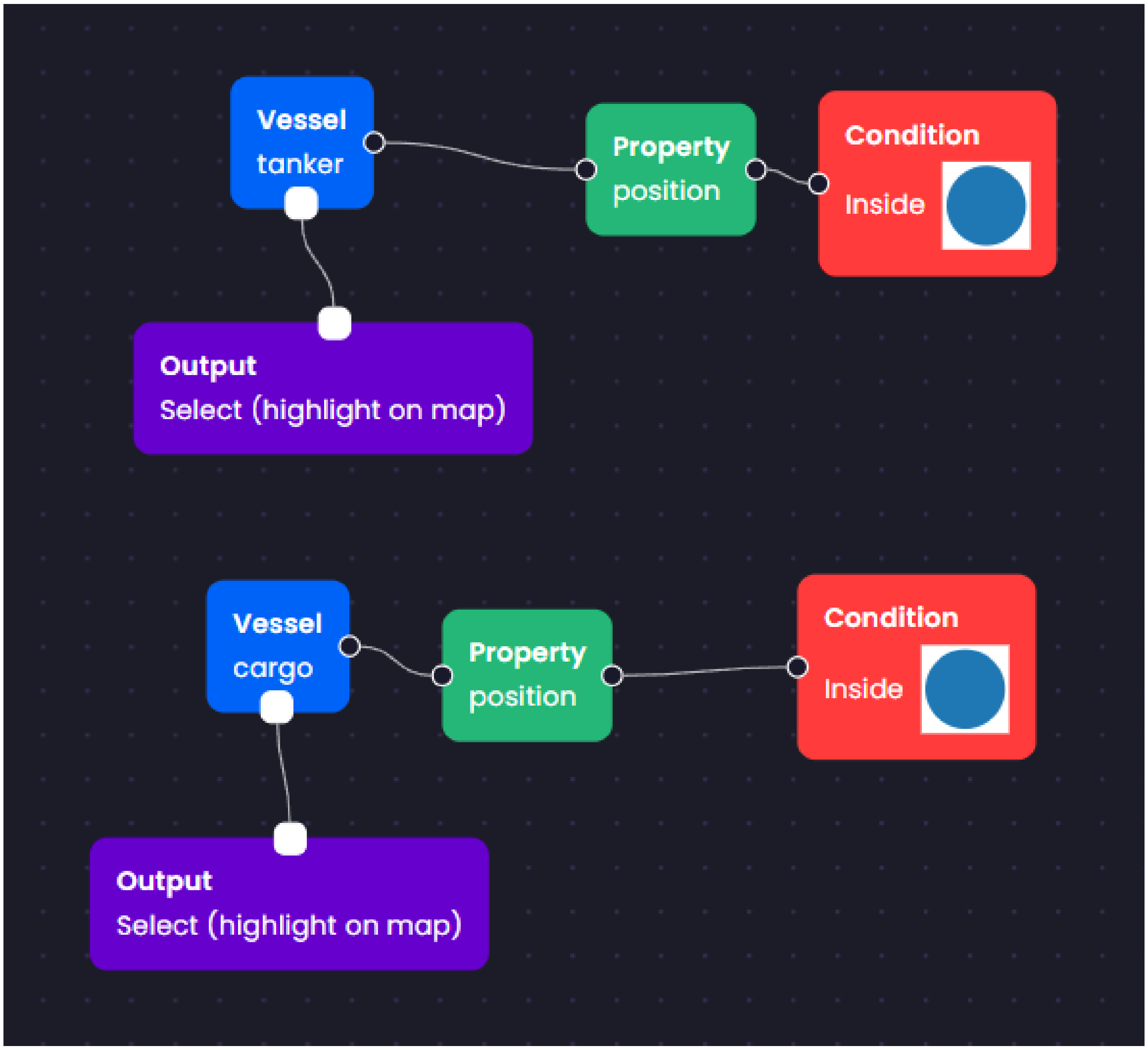
Querying Multiple Vessel Types

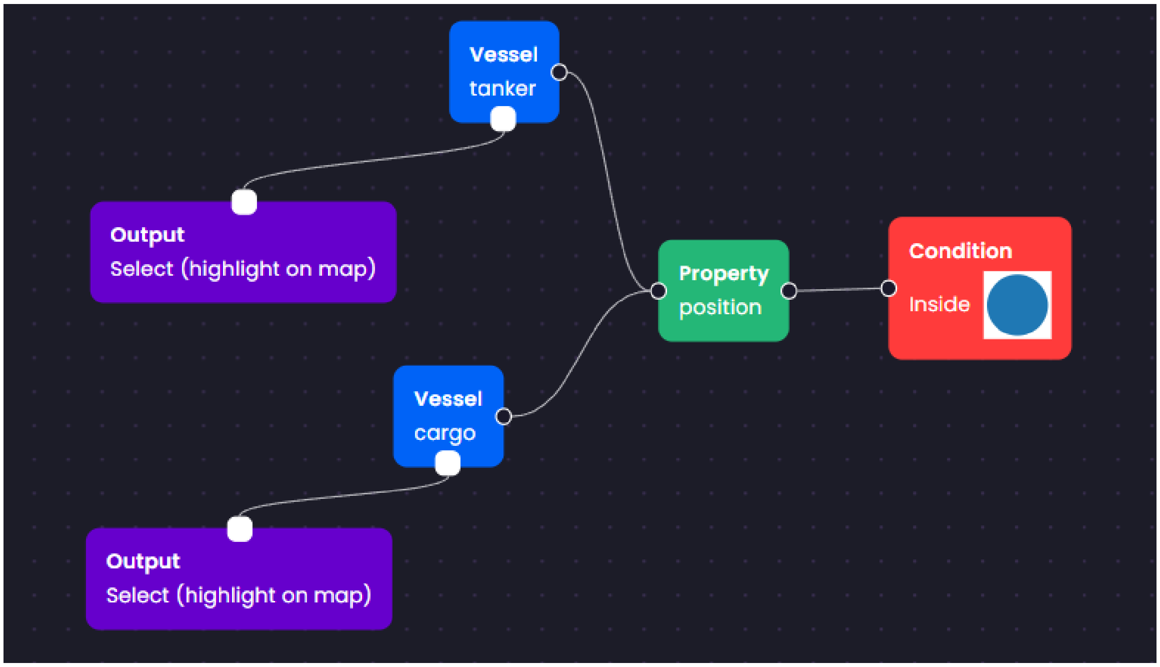
You can have multiple vessel nodes and assign them different, (or same) conditions. For example, Figure bellow shows how to query all cargo vessels in addition to tanker vessels that are in the blue circular area of interest



Sharing Same Property Filtering

If your query applies the same filter to different vessel types, you can link multiple vessel types to a single property node. For instance, the two figures below will yield identical results.


Vessel
tanker
Property
position
Condition
Inside
Output
Select (highlight on map)
Vessel
cargo
Property
position
Condition
Inside
Output
Select (highlight on map)


Vessel
tanker
Output
Select (highlight on map)
Property
position
Condition
Inside
Vessel
cargo
Output
Select (highlight on map)

A.3 Research Ethics Board (REB) Application

DALHOUSIE UNIVERSITY

RESEARCH ETHICS BOARDS APPLICATION FORM

Prospective Research

This form should only be used if new data will be collected. For research involving only secondary use of existing information (such as health records, student records, survey data or biological materials), use the *REB Application Form – Secondary Use of Information for Research.*

This form should be completed using the *Guidance for Submitting an Application for Research Ethics Review*.

SECTION 1. ADMINISTRATIVE INFORMATION

[File No: office only]

Indicate the preferred Research Ethics Board to review this research:

[ ] Health Sciences OR [X] Social Sciences and Humanities

Project Title:

Comparative Evaluation of Query Interfaces for Maritime Data Analysis

1.1 Research team information

Lead researcher (at Dalhousie)	Name	**Seyed Adel Ghaeinian**		
	Email (@dal)	**adelghaeinian@dal.ca**	Phone	**+17823217323**
	Banner #	**B00869723**	Academic Unit	**Faculty of Computer Science**
Co-investigator names, affiliations, and email addresses	Dr. Thiago Porcino, Dalhousie University, thiago@dal.ca Dr. Derek Reilly, Dalhousie University, reilly@cs.dal.ca Dr. Joseph Malloch, Dalhousie University, jmalloch@dal.ca Zixiang Yu, Dalhousie University, zx412506@dal.ca Yeaminur Rahman, Dalhousie University, yeaminur.rahman@dal.ca			
Contact person for this submission (if	Name			
	Email		Phone	

not lead researcher)				
Study start date	15 July 2023	Study end date	31 August 2023	

1.2 For student submissions (including medical residents and postdoctoral fellows)			
Degree program	M.Sc. Computer Science		
Supervisor name and department	Dr. Joseph Malloch Faculty of Computer Science		
Supervisor Email (@dal)	jmalloch@dal.ca	Phone	902-222-8468

Department/unit ethics review (if applicable). **Undergraduate minimal risk research only**.

Attestation: [ ] I am responsible for the unit-level research ethics review of this project and it has been approved.

Authorizing name:

Date:

1.3 Other reviews		
Other ethics review (if any) for this research	Where?	
	Status?	
Scholarly/scientific peer review (if any)		
Is this a variation on, or extension of, a previously approved Dal REB submission?	[X] No [ ] Yes Dal REB file #_________________	

If yes, describe which components of the current submission are the same as the previously approved submission (list section numbers), and which components are different from the previously approved submission (list section numbers). You may also use highlighting to clearly indicate revised text.

1.4 Funding		[ ] Not Applicable
	Agency	NSERC

Funding (list on consent form)	Award Number	550722
	Institution where funds are/will be held	[X] Dalhousie University [ ] Other: ____________________
Was a Dal release of funds agreement issued for this award?		[ ] Yes Date of RoF Agreement: ___________

1.5 Attestation(s). The appropriate boxes *must* be checked for the submission to be accepted by the REB

[X] I am the **lead researcher** (at Dalhousie) named in section 1.1. I agree to conduct this research following the principles of the Tri-Council Policy Statement *Ethical Conduct for Research Involving Humans* (TCPS) and consistent with the University *Policy on the Ethical Conduct of Research Involving Humans*.

I have completed the TCPS Course on Research Ethics (CORE) online tutorial.

[X Yes [ ] No

For Supervisors (of student / learner research projects):

[X] I am the **supervisor** named in section 1.2. I have reviewed this submission, including the scholarly merit of the research, and believe it is sound and appropriate. I take responsibility for ensuring this research is conducted following the principles of the TCPS and University Policy.

I have completed the TCPS Course on Research Ethics (CORE) online tutorial.

[X] Yes [ ] No

SECTION 2. PROJECT DESCRIPTION

2.1 Lay summary

2.1.1 In **plain language**, describe the rationale, purpose, study population and methods to be used. Include a summary of background information or literature to contextualize the study. What new knowledge, or public or scientific benefit is anticipated? [maximum 500 words]

In this digital age, the maritime industry has seen a surge in the generation of extensive geospatial and attribute data. Efficient and accurate querying of this data is pivotal for enhancing operational efficiency, security, and environmental sustainability. Traditional textual query languages like SQL and SPARQL are often used for this purpose [1], but require a high level of expertise and may not be inherently user-friendly. To address this issue, more intuitive interfaces, such as visual query builders (VQB) and direct manipulation (DM) interfaces, have emerged. These interfaces provide a more graphical, interactive approach to query formulation and can potentially allow a broader range of users to effectively engage with the data.

This research project sets out to conduct a comparative study of two distinct interfaces that have been implemented: 1) an interactive Visual Query Builder enhanced with direct manipulation techniques, and 2) a SPARQL interface. Providing users with a shared platform for data interaction and query execution. In addition to offering a larger and collaborative workspace, the interactive tabletop allows the interfaces to be more flexible and interconnected, meaning users can initiate a query in one interface and modify it in another. This adds a layer of dynamism and interactivity to the query process.

The study aims to evaluate the performance, accuracy, and user preference for each interface in the context of maritime operations. Our hypothesis suggests that certain interfaces may outperform others for specific types of queries. Moreover, our hypothesis is that user preference will be dictated not only by the query type but also by the individual user's comfort level and familiarity with each interface.

The research will conduct a user study featuring tasks related to common query types in the maritime context, such as temporal, attribute-based, spatial, and complex queries. Each task will be designed to evaluate a specific interface or a combination of interfaces, with performance and accuracy calculated based on the success rate and time taken to complete the tasks. User preferences will be determined through self-reported measures and a final task during which participants can freely choose their preferred interface.

We seek participants who possess basic computer literacy for this study. Although the participants will represent varying levels of experience with different query interfaces, all are expected to comprehend the fundamental concept of data querying. The data collected from participants will be anonymized and will consist of interaction logs (such as touch inputs and drag-and-drop actions), timestamps, and performance metrics.

To evaluate the user experience precisely, we will utilize the NASA Task Load Index (NASA-TLX) [2] to assess perceived workload and the System Usability Scale (SUS) [3] to measure overall usability. We will also use a custom questionnaire to capture feedback specific to our context and research questions.

This study hopes to show how different search tools work for maritime-related data. The results could help improve data querying systems for the maritime domain and for other fields that use location and property data. We will also gain insight into which search tools work best and are most preferred in general computer use.

[ ] This is a pilot study.

[X] This is a fully developed study.

2.1.2 Phased review. If a phased review is being requested, describe why this is appropriate for this study, and which phase(s) are included for approval in this application. Refer to the guidance document before requesting a phased review.

[X] Not applicable

2.2 Research question

State the research question(s) or research objective(s).

Research Objectives:

- To compare the performance and accuracy of a Visual Query Builders (VQB) interface enhanced with Direct Manipulation (DM) techniques, and the SPARQL interface in executing various types of queries in a maritime context.
- To determine user preferences among these three types of interfaces when performing different query tasks.

Research Questions (RQs):

RQ1: How does the performance (time and success rate) of an interactive Visual Query Builder (VQB) interface enhanced with Direct Manipulation (DM) techniques compare with the SPARQL interface when executing temporal, attribute-based, spatial, and complex queries in a maritime database?

Hypothesis 1: We expect that for temporal and spatial queries, the interactive VQB interface will outperform the SPARQL interface due to its intuitive visual design. Conversely, for attribute-based and complex queries, we anticipate that the SPARQL interface may provide better performance, as it is well-suited to handle specific attribute conditions and complex, multi-variable queries.

RQ2: How does the accuracy of the results generated by the interactive VQB interface compare with the SPARQL interface for each type of query task?

Hypothesis 2: We predict that the accuracy of results may be higher for SPARQL for complex queries due to its granular level of control, while the interactive VQB interface might produce more accurate results for spatial and temporal queries, given their more visual and intuitive approach to these types of tasks.

RQ3: Which interface (the interactive VQB or SPARQL) do users prefer for different types of query tasks, and what are the factors influencing this preference?

Hypothesis 3: We hypothesize that user preference will likely lean towards the interfaces that are more intuitive and user-friendly for each task type. Thus, we expect the interactive VQB to be preferred for spatial and temporal queries, while SPARQL might be the preferred choice for complex and attribute-based queries due to its comprehensive nature, despite its potential steep learning curve.

2.3 Recruitment

2.3.1 Identify the study population. Describe and justify any inclusion / exclusion criteria. Also describe how many participants are needed and how this was determined.

The study population for this research will be individuals who have basic computer literacy and can comprehend the concept of data querying. Prior familiarity with maritime operations or GIS systems is beneficial but not necessary, as one of our objectives is to assess the user-friendliness of the interfaces for potentially novice users.

Inclusion criteria:

- Adult participants aged 18 and above
- Possess basic computer literacy

Exclusion criteria:

- Individuals who do not meet the above inclusion criteria
- Individuals with visual impairments that could significantly impact their ability to interact with and evaluate the user interfaces

We aim to recruit 30 participants for this study. This number is based on a 2x15 counterbalancing design, ensuring that each participant uses both the Interactive VQB and the SPARQL interface, with an equal number of participants starting with each interface. This design helps to control for order effects, such as learning and fatigue. The bonus task, in which users can choose their preferred interface, does not require counterbalancing as it is not tied to a specific interface.

2.3.2 Describe recruitment plans and append recruitment instruments. Describe who will be doing the recruitment and what actions they will take, including any screening procedures.

We plan to recruit students from the Dalhousie Faculty of Computer Science. We will recruit them by initially sending out an email regarding the study notice through Dalhousie internal email server cs-jobs@kil-lsv-2.its.dal.ca and csgrads-bounces@cs.dal.ca. The students who are willing can respond using the provided email address and we will follow up with them. The email draft (recruitment notice) is attached in Appendix A. Investigators listed in this protocol will conduct the recruitment.

Prospective participants will receive an initial contact message explaining the purpose and nature of the study. This message will include information about the study tasks, time commitment, and compensation.

Participants will be screened based on their familiarity with maritime operations or GIS systems. There will be no strict exclusion criteria, as the study aims to include participants with varying levels of experience with different query interfaces.

2.3.3 If you require permission, cooperation, or participation from a community, organization or company to recruit your participants, describe the agreement obtained from the relevant group(s). Attach correspondence indicating their cooperation and/or support (required). Describe any other community consent or support needed to conduct this research. (If the research involves Indigenous communities complete section 2.11).

[X] Not applicable

2.4 Informed consent process

2.4.1 Describe the informed consent process:

A) How, when and by whom will the study information be conveyed to prospective participants? How will the researcher ensure prospective participants are fully informed?

The study information will be conveyed to prospective participants through a detailed written consent document. This document will be provided to participants prior to their involvement in the study in person. The researcher will ensure that prospective participants are fully informed by explaining the purpose of the study, the procedures involved, any potential risks or benefits, the voluntary nature of participation, and the right to withdraw at any time without any consequences.

B) Describe how consent will be documented (e.g. written signature, audio-recorded, etc).

Consent will be documented through written signature. Participants will be asked to sign the consent form to indicate their voluntary agreement to participate in the study. The signed consent forms will be securely stored to ensure confidentiality and compliance with ethical guidelines.

[X] Append copies of all consent information that will be used (e.g. written consent document, oral consent script, assent document/script, etc.).

The consent form is attached as an appendix.

Note: If the research will involve third party consent (with or without participant assent), and/or ongoing consent, ensure these are described above.co

2.4.2 Discuss how participants will be given the opportunity to withdraw their participation (and/or their data) and any time (or content) limitations on this. If participants will not have opportunity to withdraw their participation and/or their data explain why.

Participants will be given the opportunity to withdraw their participation at any time during the study. They will be informed of this right during the informed consent process and reminded of it throughout the study. Participants can choose to withdraw from the study without providing a reason, and their decision to withdraw will have no negative consequences.

Should a participant choose not to participate or to withdraw their participation, any data collected from them up to that point will be deleted and will not be used in the final analysis. When data is used in publications, it's essential to note that while non-aggregated data might be included, all such data will be rigorously anonymized. This ensures that individual participants cannot be identified from the published data.

Hence, withdrawal can take place any time before the data analysis phase (i.e., approximately 6 weeks after the last participant has completed the study).

The participants will receive compensation even if they choose to withdraw from the study.

2.4.3 If an alteration/exception to the requirement to seek prior informed consent is sought, address the criteria in TCPS article 3.7A. If the alteration involves deception or nondisclosure, also complete section 2.4.4.

[X] Not applicable

2.4.4 Describe and justify any use of deception or nondisclosure and explain how participants will be debriefed.

[X] Not applicable

2.5 Methods, data collection and analysis

2.5.1

A) Where will the research be conducted?

GEM lab, 1577 Barrington St Halifax, NS B3J 1Z8

B) What will participants be asked to do?

1) Fill out the demographic questionnaire. This questionnaire will ask participants about their gender, age, and previous experience with querying interfaces.

2) Read the provided SPARQL querying tutorial document and examples for 20 minutes

3) Participate in the maritime data querying experience for up to 60 minutes and answer researchers during the experimentation, such as:

T1) Identify all passenger vessels between two timestamps.

T2) Identify all vessels that were in a specific circular area, and SOG bigger than 2 mph.

T3) Identify all cargo vessels that passed through a defined polygonal area.

T4) Identify all cargo vessels having sog more than 2 and cog less than 140, in addition to tanker vessels inside a circular area of interest, having sog less than 10 and cog more than 40 (in a certain timestamp)

T5) Identify all cargo, passenger, and tanker vessels, that all of them are inside a pre-defined circular area.

Note: The initial four participants will initially engage with the visual query builder, followed by the SPARQL editor for these four tasks. Conversely, the subsequent participants will start with the SPARQL editor, proceeding to the visual query builder. This approach ensures counterbalancing across all participants. The participants can choose their preferred interface to do T5.

-- first 4) T1,T2,T3,T4 -> interface 1 first then interface 2.

-- second 4) T1,T2,T3,T4 -> interface 2 first then interface 1.

4) Fill out the interface evaluation, NASA TLX, and System Usability Scale (SUS) questionnaires. These questionnaires can be found in the appendices of this document.

C) What data will be collected using what research instruments? *(Note that privacy and confidentiality of data will be covered in section 2.6)*

- Demographic questionnaire: This questionnaire will collect information about the participants' gender, age, and previous experience with querying interfaces.
- Task performance data: Data related to the participants' performance on each task will be collected, such as task completion time and success rate.
- Interface evaluation questionnaires: Participants will be asked to fill out questionnaires assessing their experience with each interface, including aspects like ease of use, satisfaction, and overall usability. These questionnaires include the NASA TLX and System Usability Scale (SUS) questionnaires, and our custom questionnaire.
- Screen recordings: The study will involve recording the contents of the screen during participants' interactions with the interfaces during the task execution. This will enable researchers to analyze participants' decision-making processes, navigation patterns, and overall user behavior. Screen recordings will be securely stored and used solely for research purposes, ensuring participant privacy and confidentiality.
- Logs of user interactions: Detailed logs of participants' interactions, such as taps, clicks, and drag-and-drop actions will be recorded. These logs provide insights into participants' behavior and usage patterns within the interfaces.
- Queries created with the tool: The study will capture the queries formulated by participants using the different interfaces. This includes the query structures, parameters, and any modifications made during the query formulation process. Analyzing these queries can provide valuable information about participants' query formulation strategies, preferences, and accuracy (correctness).

D) How much of the participant’s time will participation in the study require?

Around 90 minutes

[X] Append copies of all research instruments (questionnaires, focus group questions, standardized measures, etc)

[ ] This is a clinical trial (physical or mental health intervention) – ensure section 2.12 is completed

2.5.2 Briefly describe the data analysis plan. Indicate how the proposed data analyses address the study’s primary objectives or research questions.

- **Descriptive Analysis**: Demographic data collected through the questionnaire will be analyzed using descriptive statistics. This analysis will provide an overview of the participants' characteristics and their previous experience with querying interfaces.
- **Task Performance Analysis**: Task completion times and success rates for each task and interface combination will be analyzed. This analysis will assess the performance of each interface in terms of efficiency and effectiveness for specific query types. Comparative analysis between interfaces will help identify any significant differences in task performance.
- **Interface Evaluation Analysis**: The data collected from interface evaluation questionnaires, such as the NASA TLX and SUS, will be analyzed. Mean scores and standard deviations will be calculated to assess participants' subjective experiences with each interface. Comparative analysis will reveal user preferences, ease of use, satisfaction levels, and overall usability of the interfaces.
- **Interaction & Screen Recording Analysis**: The recordings capturing participants' interactions will be reviewed and analyzed. This analysis will provide insights into participants' decision-making processes, navigation patterns, and overall usage behavior. It will help identify usability issues, error types, and cognitive processes during task execution.
- **Qualitative Analysis**: Open-ended responses from the questionnaires and post-task interviews will be analyzed using thematic analysis. This analysis will identify common themes and patterns in participants' feedback regarding their experiences with each interface.

The proposed data analyses directly address the primary objectives and research questions by providing insights into the performance, accuracy, and user preference for each interface in the maritime data querying context. The task performance analysis evaluates interface efficiency and effectiveness, while the interface evaluation analysis assesses user preferences and satisfaction. The screen recording analysis provides a deeper understanding of participants' interactions and cognitive processes. Collectively, these data analyses will generate empirical evidence to support conclusions about the effectiveness of different interfaces and guide the design of more efficient and user-friendly systems for maritime data querying.

2.5.3 Describe any compensation that will be given to participants and how this will be handled for participants who do not complete the study. Discuss any expenses participants are likely to incur and whether/how these will be reimbursed.

Participants are not expected to incur any expenses as a result of their participation in the study. Participants will not be required to purchase or provide any additional materials or resources for the study.

Participants in this study will receive compensation for their time and participation. The compensation amount, that is 20CAD, will be outlined in the study's informed consent form, which participants will review and agree to prior to their involvement in the study.

Participants will be compensated irrespective of their decision to continue or withdraw during the task.

The compensation will be paid in cash. Participants will be required to sign a receipt.

2.6 Privacy and confidentiality

2.6.1

A) Describe who will have knowledge of participants' identities.

Only Seyed Adel Ghaeinian will have direct knowledge of participants' identities as he will be communicating with participants to schedule and run the study sessions. Dr. Joseph Malloch will have access to consent forms containing the names of participants since the forms will be securely stored in his office.

B) Describe the level of identifiability of the study data (anonymous, anonymized, de-identified/coded, identifying) (see TCPS Chapter 5A – types of information for definitions).

anonymized data. We will assign a unique number as participant's ID to each participant. All the recordings – paper, and digital will not contain the names or any other personal details of the participants.

level of identifiability

Paper documents: The questionnaires distributed to the participants during the study will have only the participant ID.

Screen Recording: We screen record all the participants' interactions with the software applications. We then bind these recordings to the participants' ID.

Results and interaction logs: These will only be referred using participants' ID

C) Specify which members of the research team (or others) will have access to participants' data and for what purpose.

The members of the research team, including Seyed Adel Ghaeinian, Dr. Thiago Porcino, Dr. Joseph Malloch, Dr. Derek Reilly, Zixiang Yu, and Yeaminur Rahman will have access to participants' data. They will have access to the data for the purpose of conducting data analysis, evaluating the research findings, and generating insights related to the study objectives and research questions.

D) Describe measures to ensure privacy and confidentiality of study documents and participant data during the data collection and analysis phase. *[Note that plans for long term storage will be covered in 2.6.2*]

- Address: handling of documents/data during data collection; transportation or transfer of documents/data; storage of documents/data (during the study).
- If a key-code will be maintained, describe how it will be kept secure.

- For electronic data, describe electronic data security measures, including file encryption and/or password protection as applicable.
- For hard copy documents, describe physical security measures (specify location).

To ensure privacy and confidentiality of study documents and participant data during both the data collection and analysis phases, we have implemented the following measures:

Handling of Documents/Data During Data Collection:
Data, if collected in a non-digital format, will be promptly digitized. Digital versions will be stored in a secure location with restricted access to uphold the confidentiality of the participants' information.

Transportation or Transfer of Documents/Data: Data will be uploaded directly to a dedicated OneDrive folder provided by Dalhousie University, eliminating the need for physical transfer and reducing the risk of potential breaches.

Storage of Documents/Data (During the Study): All study data will be stored on the OneDrive platform within a shared folder that's only accessible to the co-investigators of this study. This folder will be secured with restricted access controls.

Key-Code Security: We will not employ a key-code system. Instead, all participant data will undergo a thorough anonymization process before storage or analysis, ensuring that individual participants' identities remain confidential.

Electronic Data Security Measures: All electronic data will utilize the security features of OneDrive, including file encryption. Additionally, particularly sensitive documents within this environment will be password-protected.

Physical Security Measures: Any physical documents, except the consent forms, will be digitized and immediately shredded, ensuring no tangible copies can be accessed unauthorizedly. The retained consent forms will be securely stored in a locked cabinet within the lab.

[ ] This research involves personal health records (ensure section 2.13 is completed)

2.6.2 Describe plans for data retention and long-term storage (i.e. how long data will be retained, in what form and where). Will the data eventually be destroyed or irreversibly anonymized? If so, what procedures will be used for this? Discuss any plans for future use of the data or materials beyond the study currently being reviewed.

[ ] This research will be deposited in a data repository (ensure section 2.14 is completed)

Our data retention and long-term storage plan are as follows:

Duration of Retention: All study data will be retained for a period of 3 years following the conclusion of the research.

Form and Location: The data, which will have undergone a rigorous anonymization process before storage, will be stored electronically in a secure OneDrive folder provided by Dalhousie University.

This folder will have restricted access controls, ensuring only the co-investigators and authorized personnel can access the stored data.

Destruction: As the stored data is already anonymized, upon reaching the end of the retention period, all data will be permanently deleted from our storage solutions, ensuring no remnants remain.

Plans for Future Use: There are no plans for future use of the data beyond the scope of the current study.

2.6.3

Describe if/how participant confidentiality will be protected when research results are reported:

A) For quantitative results - In what form will study data be disseminated?

[ ] Only aggregate data will be presented

[X] Individual de-identified, anonymized or anonymous data will be presented

[ ] Other. If "other", briefly describe dissemination plans with regard to identifiability of data.

[ ] Not applicable, only qualitative data will be presented

Only aggregate data will be presented for task completion statistics, NASA-TLX and SUS results. Individual anonymized data may be presented to exemplify differences in queries produced using the different interfaces, or when presenting analysis of participants' preferences during task 4.

B) For qualitative results - Will identifiable data be used in research presentations/publications? If participants will be quoted, address consent for this and indicate whether quotes will be identifiable or attributed.

[X] Not applicable, only quantitative data will be presented

2.6.4 Address any limits on confidentiality, such as a legal duty to report abuse or neglect of a child or adult in need of protection, and how these will be handled. Ensure these are clear in the consent documents. (See the guidance document for more information on legal duties and professional codes of ethics).

[X] Not applicable

2.6.5 Will any information that may reasonably be expected to identify an individual (alone or in combination with other available information) be accessible outside Canada? And/or, will you be using any electronic tool (e.g. survey company, software, data repository) to help you collect, manage, store, share, or analyze personally identifiable data that makes the data accessible from outside Canada?

[X] No

[ ] Yes. If yes, refer to the University *Policy for the Protection of Personal Information from Access Outside Canada*, and describe how you comply with the policy (such as securing participant consent and/or securing approval from the Vice President Research and Innovation).

2.7 Risk and benefit analysis

2.7.1 Discuss what risks or discomforts are anticipated for participants, how likely risks are and how risks will be mitigated. Address any particular ethical vulnerability of your study population. Risks to privacy from use of identifying information should be addressed. If applicable, address third party or community risk. (If the research involves Indigenous communities also complete section 2.11)

Participants may experience mild frustration or stress while performing the study tasks or interacting with the interfaces. To ensure their well-being, participants will have the option to take a break or to withdraw from the study at any time without consequences. The research team will provide a supportive environment, address concerns, and provide help to make them feel more comfortable. Participant comfort and safety are paramount throughout the study.

2.7.2 Identify any direct benefits of participation to participants (other than compensation), and any indirect benefits of the study (e.g. contribution to new knowledge).

By participating in this study, individuals will have the opportunity to gain practical experience with different query interfaces used in the maritime domain, enhancing their technological skills and knowledge. They will contribute to the advancement of research in human-computer interaction and the development of more user-friendly and efficient query interfaces for maritime data. The study's findings will benefit industries reliant on spatial and attribute data, leading to improved systems and better decision-making processes. Overall, participants' involvement offers them a chance to engage with cutting-edge technology, contribute to new knowledge, and shape the future of interface design.

2.8 Provision of results to participants and dissemination plans.

2.8.1 The TCPS encourages researchers to share study results with participants in appropriate formats. Describe your plans to share study results with participants and discuss the process and format.

This research will be presented at or published in journals and conferences focused on computer science and human-computer interaction. This study's findings will all be presented in publications that are published. Publications will be linked from the lab website.

2.8.2 If applicable, describe how participants will be informed of any material incidental findings – a discovery about a participant made in the course of research (screening or data collection) that is outside the objectives of the study, that has implications for participant welfare (health, psychological or social). See TCPS Article 3.4 for more information.

[X] Not applicable

2.8.3 Describe plans for dissemination of the research findings (e.g. conference presentations, journal articles, public lectures etc.).

This research will be published in conferences, journals, or other suitable scholarly venues.

2.9 Research Team

2.9.1 Describe the role and duties of all research team members (including students, RA's and supervisors) in relation to the overall study.

Dr. Reilly is advising the master's student and collaborates with paper writing, suggesting adequate conferences to publish the work.

Dr. Malloch is the main supervisor of the master's student, collaborating with paper writing, suggesting, and directing the research goals.

Dr. Porcino is the postdoctoral researcher, helping the master's student in paper writing, co-advising with Dr. Reilly and Dr. Malloch.

Mr. Ghaeinian is a master's student researching XR and visual queries, writing research papers, prototyping and implementing functionalities, and designing user studies.

2.9.2 Briefly identify any previous experience or special qualifications represented on the team relevant to the proposed study (e.g. professional or clinical expertise, research methods, experience with the study population, statistics expertise, etc.).

Dr. Reilly codirects the Graphics and Experiential Media (GEM) Lab at Dalhousie University as an associate professor in the Faculty of Computer Science. He has extensive experience designing and running human-participant studies.

Dr Malloch is an Assistant Professor in the Faculty of Computer Science at Dalhousie University where he codirects the Graphics and Experiential Media (GEM) Lab. He has 20 years of experience designing and evaluating interactive computer systems with human participants.

Dr. Porcino is a postdoctoral fellow at Dalhousie University and Graphics and Experimental Media (GEM) Lab, focused in Human-Computer Interaction, Virtual/Augmented Reality, Cybersickness, and Agile methodologies. He holds a Ph.D. and Master's degree in computer science from Fluminense Federal University, Brazil.

Mr. Ghaeinian is a Master's student under Dr. Malloch's supervision working as a research assistant. He is currently working on data visualization and HCI at Graphics and Experiential Media (GEM) Lab at Dalhousie University.

2.10 Conflict of interest

Describe whether any dual role or conflict of interest exists for any member of the research team in relation to potential study participants (e.g. TA, fellow student, teaching or clinical relationship), and/or study sponsors, and how this will be handled.

[X] Not applicable

No conflicts of interest exist for this study, as investigator (Adel) is not currently teaching or employing anyone who could potentially participate in the study.

Blind Review Process: The identities of participants will not be revealed during data analysis. This ensures unbiased assessment and analysis of the data.

Voluntary Participation: Participation is entirely voluntary. Fellow students or acquaintances will be informed that their decision to participate or not will not have any repercussions on their relationship with the researcher.

Transparent Communication: All participants, including acquaintances or fellow students, will be clearly informed about the study's objectives, and any potential risks or benefits. This ensures that they make an informed decision about their participation.

Anonymized Data: All data will be anonymized before analysis, ensuring that individual identities are not recognizable in the dataset.

By implementing these measures, we aim to mitigate any potential conflicts of interest and ensure the integrity of the research process.

2.11 Research involving Indigenous peoples

Consult TCPS Articles 9.1 and 9.2 in determining whether this section is applicable to your research.

[X] Not applicable – go to 2.12

2.11.1 If the proposed research is expected to involve people who are Indigenous, describe the plan for community engagement (per TCPS Articles 9.1 and 9.2). If community engagement is not sought, explain why the research does not require it, referencing TCPS article 9.2.

2.11.2 State whether ethical approval has been or will be sought from Mi'kmaw Ethics Watch and if not, why the research does not fall under their purview. If the research falls under the purview of other Indigenous ethics groups, state whether ethical approval has been or will be sought.

2.11.3 Describe plans for returning results to the community and any intellectual property rights agreements negotiated with the community with regard to data ownership (see also 2.11.4 if applicable). Append applicable research agreements.

2.11.4 Does this research incorporate OCAP (Ownership, Control, Access, and Possession) principles as described in TCPS Article 9.8?

[ ] Yes. Explain how.

[ ] No. Explain why not.

2.12 Clinical trials

[X] Not applicable – go to 2.13

2.12.1 Will the proposed clinical trial be registered?

[ ] No. Explain why not.

[ ] Yes. Indicate where it was/will be registered and provide the registration number.

2.12.2 If a novel intervention or treatment is being examined, describe standard treatment or intervention, to indicate a situation of clinical equipoise exists (TCPS Chapter 11). If placebo is used with a control group rather than standard treatment, please justify.

2.12.3 Clearly identify the known effects of any product or device under investigation, approved uses, safety information and possible contraindications. Indicate how the proposed study use differs from approved uses.

[ ] Not applicable

2.12.4 Discuss any plans for blinding/randomization.

2.12.5 What plans are in place for safety monitoring and reporting of new information to participants, the REB, other team members, sponsors, and the clinical trial registry (refer to TCPS Articles 11.6, 11.7, 11.8)? These should address plans for removing participants for safety reasons, and early stopping/unblinding/amendment of the trial. What risks may arise for participants through early trial closure, and how will these be addressed? Are there any options for continued access to interventions shown to be beneficial?

2.13 Use of personal health information

[X] Not applicable

2.13.1 Research using health information may be subject to Nova Scotia's *Personal Health Information Act*. Describe the personal health information (definition explained in the guidance document) required and the information sources, and explain why the research cannot reasonably be accomplished without the use of that information. Describe how the personal health information will be used, and in the most de-identified form possible.

2.13.2 Will there be any linking of separate health data sets as part of this research?

[X] No

[ ] Yes

If yes:

A) Why is the linkage necessary?

B) Describe how the linkage will be conducted (it is helpful to append a flow diagram)

C) Does that linkage increase the identifiability of the participants?

2.13.3 Describe reasonably foreseeable risks to privacy due to the use of personal health information and how these will be mitigated.

2.14 Data Repositories

[X] Not applicable

2.14.1 Identify and describe the data repository in which the research data will be deposited. What is its focus, who are its target users, who can access deposited data and under what circumstances? For how long will the data be kept in the repository?

2.14.2 Describe the data set to be released to the repository. If there is personal and/or sensitive information in the data, describe how you will prepare the data for submission to the repository and mitigate risks to privacy. Identify all fields that will be included in the final data set (include as an appendix).

2.14.3 Is agreeing to have one's data deposited a requirement for participation in the study? If yes, provide a justification. If no, indicate how participants can opt in or out.

SECTION 3. APPENDICES

Appendices Checklist. Append all relevant material to this application in the order they will be used. This may include:

[X] Reference list

[ ] Permission or support/cooperation letters (e.g. Indigenous Band Council, School Board, Director of a long-term care facility, anyone whose permission you need to conduct recruit participants or conduct research)

[ ] Research agreements (required for research involving Indigenous communities)

[X] Recruitment documents (posters, oral scripts, online postings, invitations to participate, etc.)

[ ] Screening documents

[X] Consent/assent documents or scripts

[X] Research instruments (questionnaires, interview or focus group questions, etc.)

[ ] Debriefing and/or study results templates

[ ] List of data fields included in data repository

[ ] Confidentiality agreements

Consent Form Templates

Sample consent forms are provided on the Research Ethics website and may be used in conjunction with the information in the *Guidance* document to help you develop your consent form.

Appendix A: Recruitment Notice:

Dear potential participants,

We invite you to participate in a research study evaluating different query interfaces for maritime data analysis. This study aims to compare the effectiveness, efficiency, and user experience of Visual Query Builder (VQB) and SPARQL interfaces within a maritime context.

The study involves performing tasks using these interfaces, providing feedback through questionnaires and interviews. We are seeking individuals with basic computer usage skills and an interest in maritime operations or geographic information systems (GIS). No prior expertise or experience with query interfaces is required.

Your participation will take approximately 90 minutes, and as a token of appreciation, you will be compensated $20 for your time and contribution.

All information collected will be treated confidentially, and your participation is voluntary, and by participating, you will contribute to research in human-computer interaction and maritime data analysis, helping to shape the design of more user-friendly interfaces for maritime applications.

If you are interested or have any questions, please contact us at adelghaeinian@dal.ca . Your valuable contribution will greatly enhance the success of this study.

Sincerely,

Seyed Adel Ghaeinian
Dalhousie University

Appendix B: Demographic Questionnaire:

User ID:

Age:

[ ] 18-25

[ ] 26-35

[ ] 36-45

[ ] 46+

Gender:

[ ] Man

[ ] Woman

[ ] Prefer to self-describe: ____________________________________

[ ] Prefer to not answer

Experience with Query Interfaces or Data Analysis / Querying Tools (such as MongoDB, SQL, etc.):

[ ] Yes, frequently

[ ] Yes, occasionally

[ ] No

If yes, please specify which tools or interfaces you have used:

____

Familiarity with Maritime Operations or GIS Systems:

[ ] Expert (I work/have worked in this domain or have formal training)

[ ] Very familiar (I've used GIS systems or have knowledge about maritime operations)

[ ] Somewhat familiar (I have some knowledge or have used related systems/tools)

[ ] Not familiar

Programming Languages Experience:

Which programming languages are you familiar with? (Please list):

Level of Experience:

How would you describe your overall programming expertise?

[ ] Beginner

[ ] Intermediate

[ ] Advanced

Additional Background:

Please describe any additional experiences or background information you feel might be relevant to this study, such as courses taken, jobs held, or hobbies related to maritime operations, GIS systems, or data querying. (Optional)

Appendix C: NASA-TLX Questionnaire [2]

How mentally demanding was the task?

□□□□□□□□□□□□□□□□□□□□

Very Low Very High

How physically demanding was the task?

□□□□□□□□□□□□□□□□□□□□

Very Low Very High

How hurried or rushed was the pace of the task?

□□□□□□□□□□□□□□□□□□□□

Very Low Very High

How successful were you in accomplishing what you were asked to do?

□□□□□□□□□□□□□□□□□□□□

Very Low Very High

How hard did you have to work to accomplish your level of performance?

□□□□□□□□□□□□□□□□□□□□

Very Low Very High

How insecure, discouraged, irritated, stressed, and annoyed were you?

□□□□□□□□□□□□□□□□□□□□

Very Low Very High

Appendix D: System Usability Scale Questionnaire [3]

1. I think that I would like to use this system frequently

Strongly disagree				Strongly agree

2. I found the system unnecessarily complex

Strongly disagree				Strongly agree

3. I thought the system was easy to use

Strongly disagree				Strongly agree

4. I think that I would need the support of a technical person to be able to use this system

Strongly disagree				Strongly agree

5. I found the various functions in this system were well integrated

Strongly disagree				Strongly agree

6. I thought there was too much inconsistency in this system

Strongly disagree				Strongly agree

7. I would imagine that most people would learn to use this system very quickly

Strongly disagree				Strongly agree

8. I found the system very cumbersome to use

Strongly disagree				Strongly agree

9. I felt very confident using the system

Strongly disagree				Strongly agree

10. I needed to learn a lot of things before I could get going with this system

Strongly disagree				Strongly agree

Participant ID:

Maritime Data Querying Interface Feedback and User Experience Assessment

Interface Preferences:

Which interface did you find most intuitive and user-friendly for querying maritime data?

[   ] Visual Query Builder

[   ] SPARQL Editor

Please explain why you preferred this interface over the others.

Were there any specific features or functionalities of the preferred interface that stood out to you? If yes, please describe.

Task Performance and Accuracy:

How confident were you in your ability to accurately complete the given tasks using each interface? (Scale: 1-5, with 1 being not confident at all and 5 being very confident)

Visual Query Builder

1	2	3	4	5

SPARQL Editor

1	2	3	4	5

Which interface allowed you to complete the tasks with the highest level of accuracy?

Were there any particular challenges or difficulties you encountered while using any of the interfaces? If yes, please explain.

User Experience:
How would you rate the overall usability and user-friendliness of each interface? (Scale: 1-10, with 1 being very poor and 10 being excellent)

Visual Query Builder

1	2	3	4	5	6	7	8	9	10

SPARQL

1	2	3	4	5	6	7	8	9	10

Did you find any interface to be more visually appealing or aesthetically pleasing than the others? If yes, please specify (SPARQL, Visual Query Builder).

Were there any specific aspects of the interfaces (e.g., layout, navigation, interaction) that you particularly liked or disliked? Please provide details.

Improvement Suggestions:

Based on your experience with the interfaces, are there any specific improvements or enhancements you would recommend? This could include additional features, interface modifications, or any other suggestions to make the interfaces more effective or user-friendly.

Overall Impression:

Considering all aspects of the study, how satisfied are you with your participation? (Scale: 1-5, with 1 being very dissatisfied and 5 being very satisfied)

1	2	3	4	5

Do you have any final comments or feedback regarding the study, the interfaces, or any other aspect you would like to share?

Appendix F: CONSENT FORM

Project Title: Comparative Study of Querying Methods for Maritime Datasets

Lead Researchers: Seyed Adel Ghaeinian, Faculty of Computer Science, adelghaeinian@dal.ca
Dr. Thiago Porcino, Faculty of Computer Science, Thiago@dal.ca

Other Researchers: Dr. Joseph Malloch, Faculty of Computer Science, jmalloch@dal.ca
Dr. Derek Reilly, Faculty of Computer Science, reilly@cs.dal.ca
Zixiang Yu, Faculty of Computer Science, zx412506@dal.ca
Yeaminur Rahman, Faculty of Computer Science, yeaminur.rahman@dal.ca

Introduction: You are invited to take part in a research study conducted by Seyed Adel Ghaeinian and Dr. Thiago Porcino from the Faculty of Computer Science at Dalhousie University. Your participation in this study is entirely voluntary, and you have the right to withdraw at any time without any consequences. The purpose of this study is to conduct a comparative evaluation of different querying methods for maritime datasets, SPARQL interfaces, and visual query builders (VQB). The study aims to assess the performance, accuracy, and user preference of these interfaces.

Procedure: If you decide to participate in this research, you will be asked to perform a series of tasks that involve querying maritime datasets using the SPARQL, and VQB interfaces. You have the option to take breaks during the task, and you may stop at any point. If you choose to withdraw from the study, your data will be deleted and will not be used for analysis. If you choose to withdraw before completing the study you will still receive compensation. You can also choose to withdraw your data after completing the study by contacting the researchers up until data analysis has been completed approximately six weeks after your participation.

The study will take place on the 4th floor of the 1577 Barrington Street building in Halifax. Your participation will take approximately 90 minutes, and you will be compensated with cash amount of $20 for your time. We will ask you to sign a receipt acknowledging receipt of payment.

Benefits and Risks: There are no direct benefits for participating in this study, but your involvement will contribute to understanding the effectiveness of different querying methods for maritime datasets. The study involves minimal risks, such as potential mild frustration or stress during task completion. However, the research team will be available to provide assistance and address any concerns that may arise.

Confidentiality and Data Handling: Your participation and any data collected during the study will be treated with strict confidentiality. All electronic data will be stored on a secure OneDrive platform provided by Dalhousie University, with access restricted only to the research team. Physical documents, with the exception of this consent form, will be promptly digitized and then shredded to ensure there are no physical copies that could be misplaced or accessed by unauthorized individuals. This consent form will be stored in a locked cabinet in the lab.

All collected data will undergo a de-identification process to ensure that your personal identity remains confidential. This means that direct identifiers, such as your name and email, will be removed or replaced with pseudonyms. Your original data will be retained for a duration of 3 years, after which it will be permanently deleted. The data will only be used for the purposes of this study and will not be utilized for any other research unless you provide explicit consent. Engagement with the software, including keystrokes and touches, will be recorded, to assess the effectiveness of the querying methods. This recording will only be accessible to the research team and will be stored with the same precautions as other study data.

Contact Information If you have any questions, concerns, or difficulties during the study, you may contact Seyed Adel Ghaeinian at adelghaeinian@dal.ca or Dr. Thiago Porcino at thiago@dal.ca. If you have any ethical concerns regarding this study, please contact the Office of Research Ethics at ethics@dal.ca.

Consent

By signing below, you acknowledge that you have read and understood the information provided in this consent form. You freely and voluntarily agree to participate in this study. Your participation is entirely voluntary, and you may withdraw at any time without penalty or loss of benefits. Your signature confirms your consent to participate in the study.

Participant's Name ______________________

Participant's Signature ______________________

Date _________________________

Please provide your email address if you would like to receive a summary of the study results (optional):

Email address: _________________________

References

[1] D. Kolas, “Supporting Spatial Semantics with SPARQL,” *Transactions in GIS*, vol. 12, no. SUPPL. 1, pp. 5–18, 2008, doi: 10.1111/J.1467-9671.2008.01132.X.

[2] S. G. Hart, “Nasa-Task Load Index (NASA-TLX); 20 Years Later,” *http://dx.doi.org/10.1177/154193120605000909*, pp. 904–908, Oct. 2006, doi: 10.1177/154193120605000909.

[3] “SUS: A ‘Quick and Dirty’ Usability Scale,” *Usability Evaluation In Industry*, pp. 207–212, Jun. 1996, doi: 10.1201/9781498710411-35.

[4] R. E. Kirk, “Latin Square Design,” *The Corsini Encyclopedia of Psychology*, pp. 1–2, Jan. 2010, doi: 10.1002/9780470479216.CORPSY0499.

Appendix B

Statistical Analysis of NASA-TLX Questions Individually

NASA-TLX Two-way ANOVA (Mixed-Design)

This section provides the statistical analysis for each parameter (question) in the NASA-TLX individually.

▼ Q1 - How mentally demanding was the task?

Results:

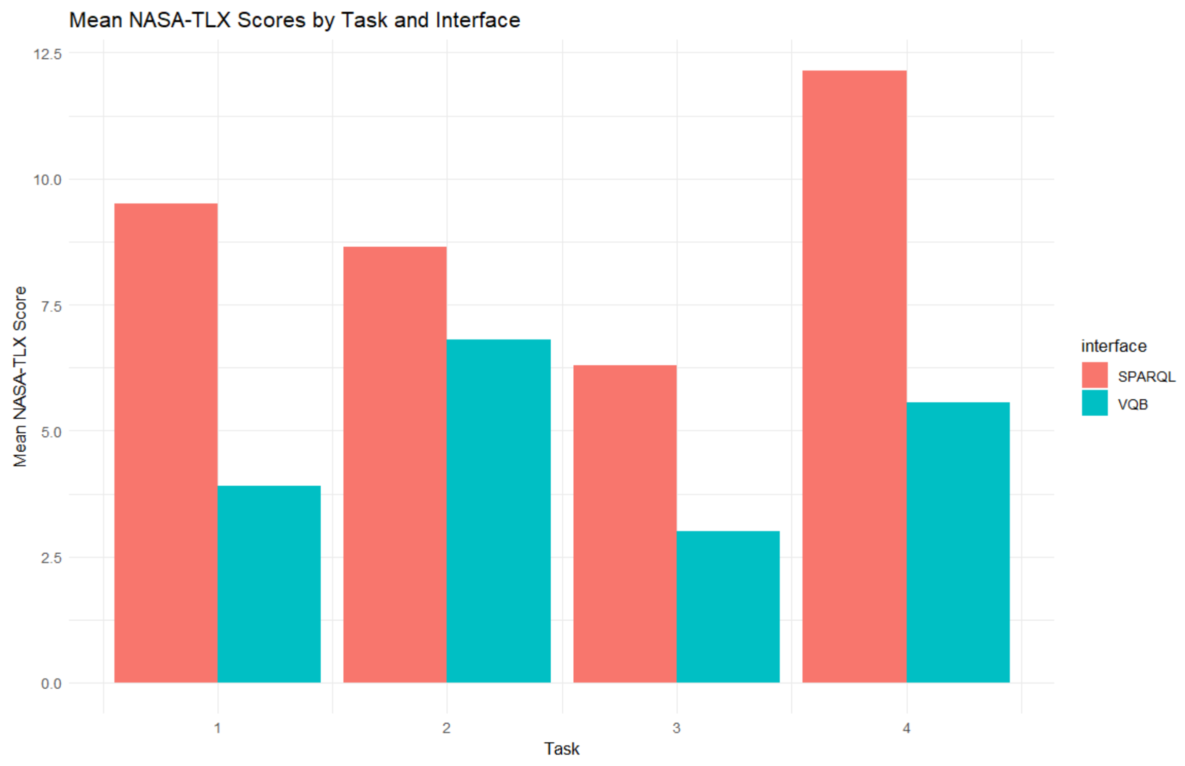
Interface	Mean
VQB	4.81
SPARQL	9.15

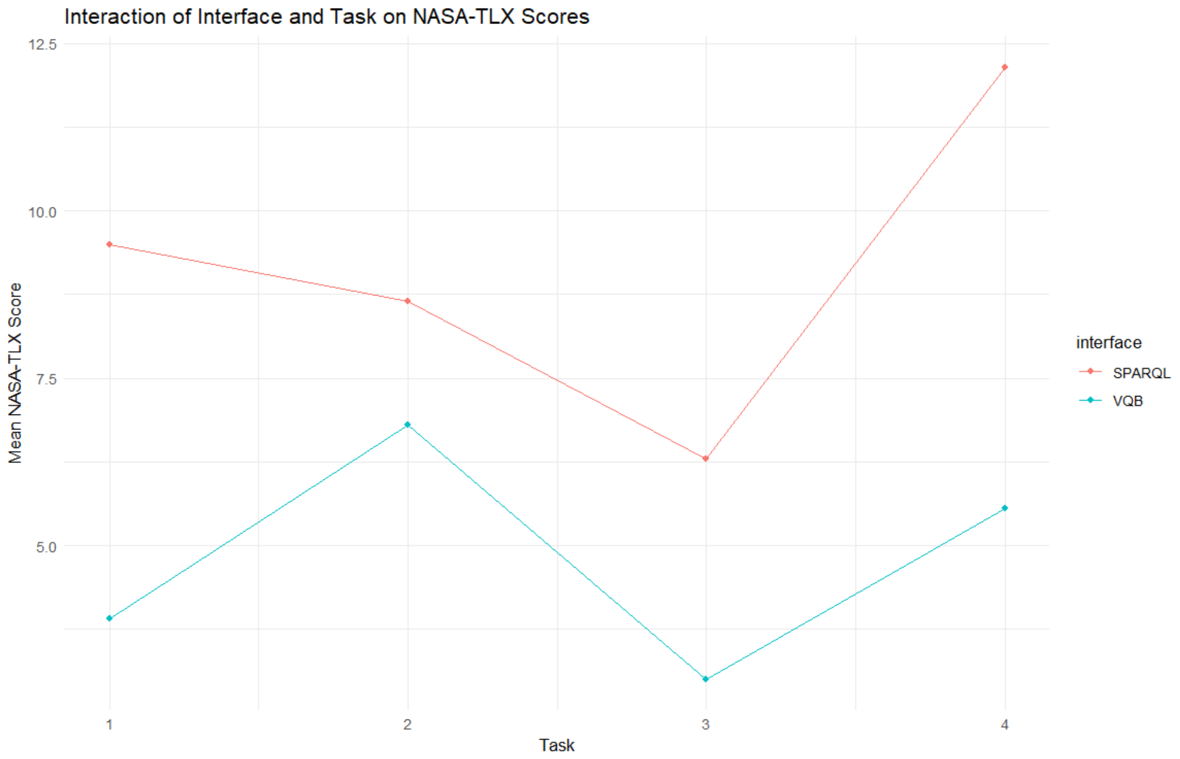
```
Univariate Type III Repeated-Measures ANOVA Assuming Spher
icity

                     Sum Sq num Df Error SS den Df  F valu
e    Pr(>F)
(Intercept)          7798.1      1  1276.56     18 109.955
5 4.282e-09 ***
group                 273.0      1  1276.56     18   3.849
5 0.0654125 .
interface             752.6      1   704.01     18  19.241
2 0.0003560 ***
group:interface        43.1      1   704.01     18   1.100
9 0.3079636
task                  382.4      3   494.39     54  13.921
5 7.686e-07 ***
group:task             13.1      3   494.39     54   0.477
6 0.6991729
interface:task        139.8      3   335.54     54   7.497
```

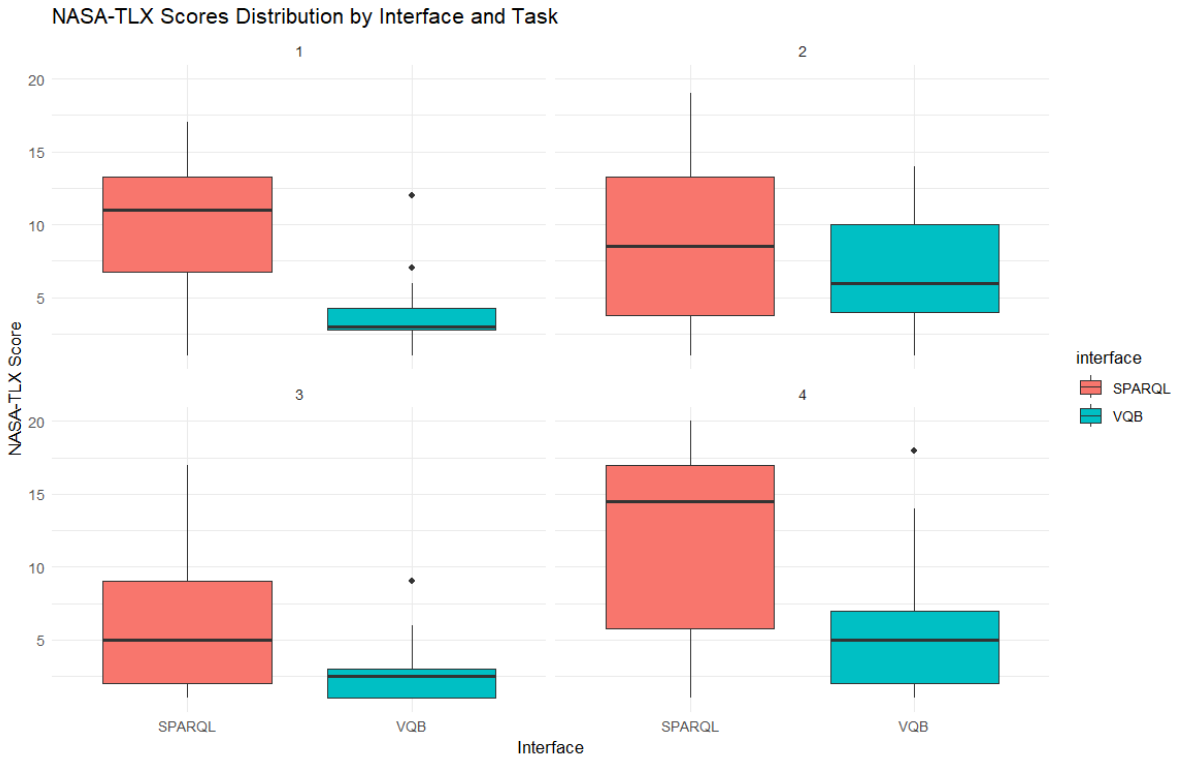
```
9 0.0002771 ***
group:interface:task   26.6      3   335.54     54   1.425
3 0.2454524
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1


Mauchly Tests for Sphericity

                     Test statistic p-value
task                        0.77124 0.50178
group:task                  0.77124 0.50178
interface:task              0.77650 0.51749
group:interface:task        0.77650 0.51749


Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

                      GG eps Pr(>F[GG])
task                 0.85015  4.155e-06 ***
group:task           0.85015  0.6691119
interface:task       0.86322  0.0006075 ***
group:interface:task 0.86322  0.2494470
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

                       HF eps   Pr(>F[HF])
task                 1.002365 7.685943e-07
group:task           1.002365 6.991729e-01
interface:task       1.021037 2.770783e-04
group:interface:task 1.021037 2.454524e-01
```

Result Interpretation

- **Intercept** Highly significant (p < 0.001) → overall model is significant.
- **group** Non-significant (p = 0.094) → grouping (the order of tasks) may not have a significant effect on the mental demanding scores.
- **interface** Significant (p = 0.0017), indicating a significant effect of interface type on the scores. Based on this result and the mean difference between two interfaces, we can interpret that VQB is perceived as less demanding than SPARQL
- **group|interface**: Not significant → no interaction effect between group and interface.
- **task** Highly significant (p < 0.001) → significant differences across tasks. This suggests that different tasks have significantly different mental demanding
- **group|task**, **interface|task**, **group|interface|task**: The interactions involving tasks are not all significant, with only **interface|task** being significant (p < 0.001), suggesting that the effect of the interface differs depending on the task.







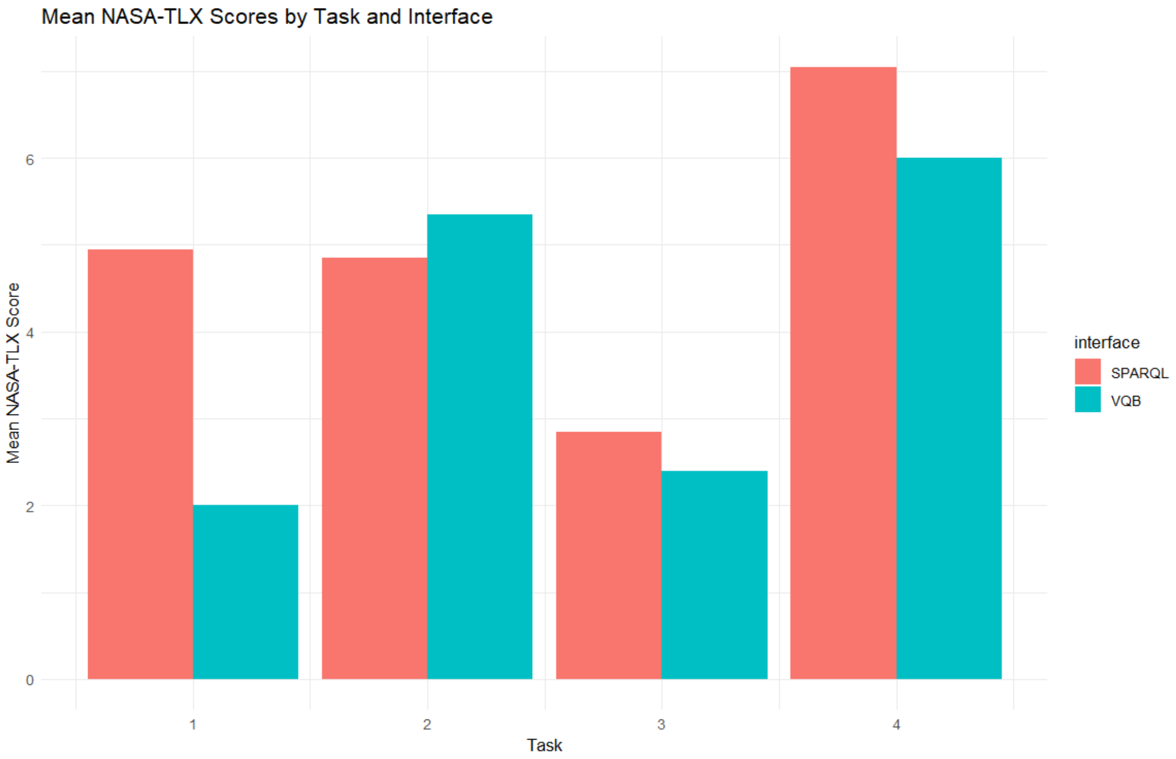
▼ Q2 - How physically demanding was the task?

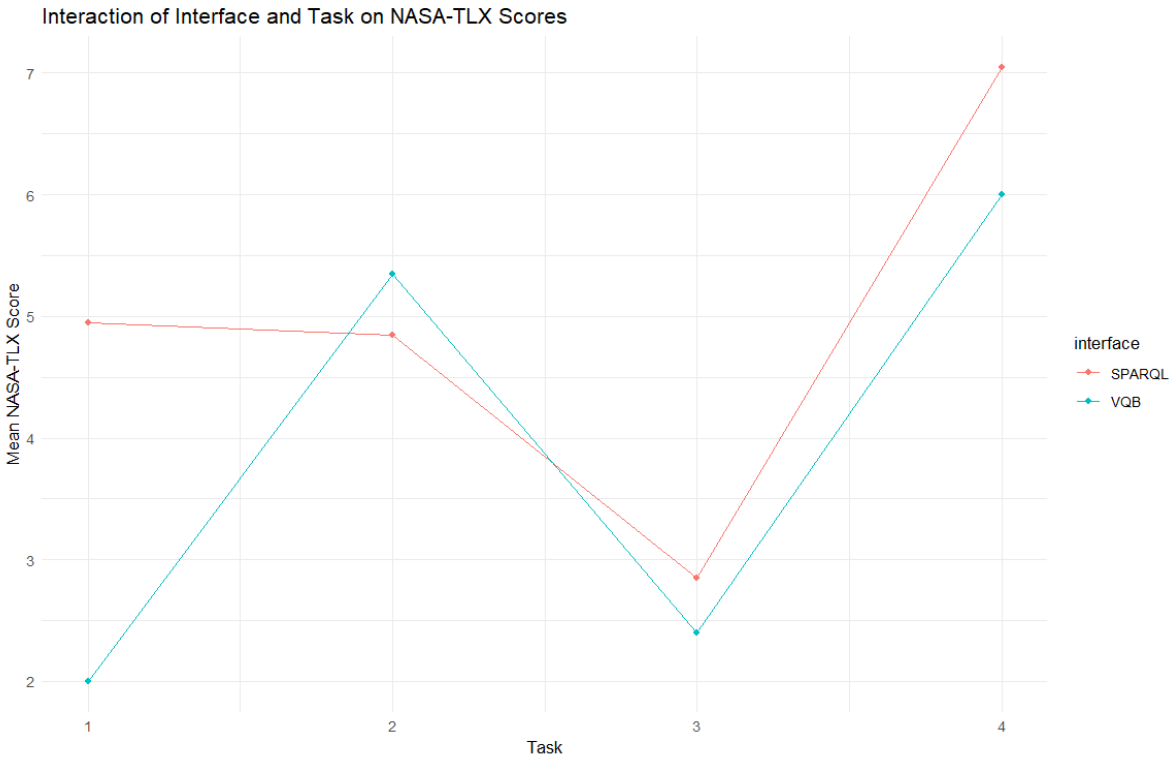
Results

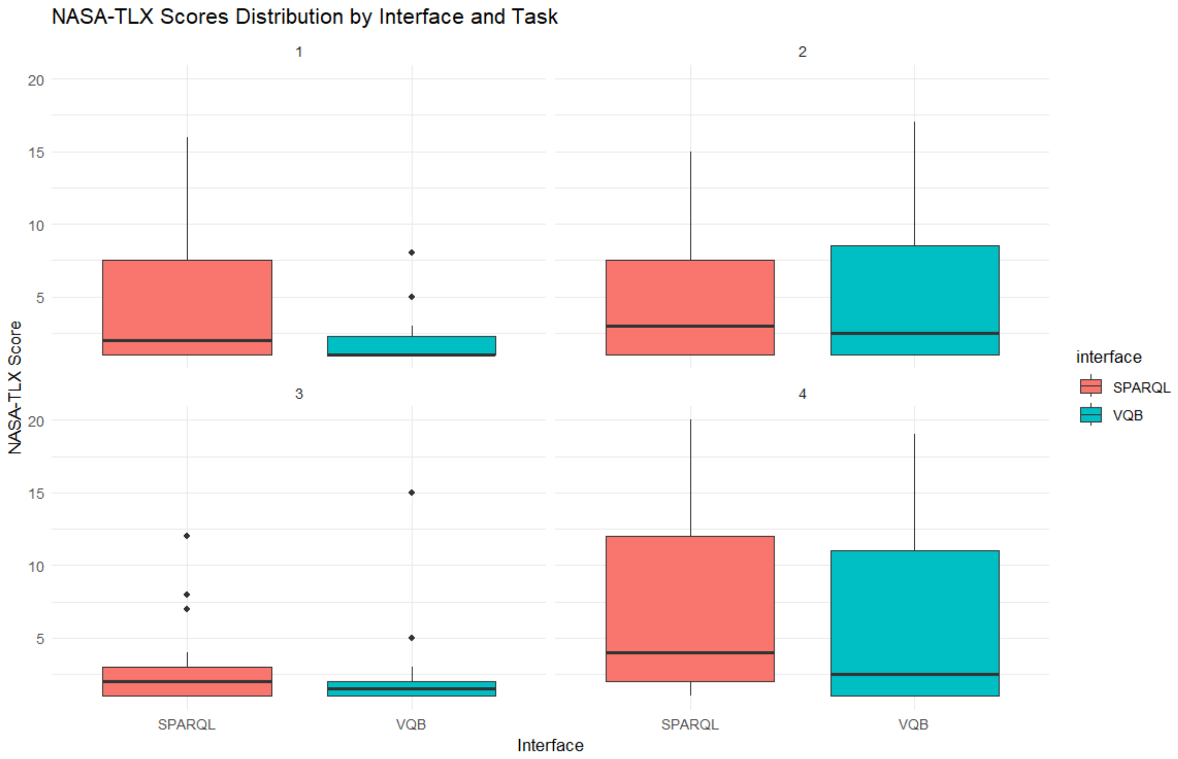
Interface	Mean
VQB	3.94
SPARQL	4.92

```
Univariate Type III Repeated-Measures ANOVA Assuming Spher
icity

                      Sum Sq num Df Error SS den Df F valu
e    Pr(>F)
(Intercept)          3141.76      1  1924.31     18 29.388
0 3.767e-05 ***
group                  28.06      1  1924.31     18  0.262
4   0.61468
interface              39.01      1   177.21     18  3.962
0   0.06194 .
group:interface         3.91      1   177.21     18  0.396
8   0.53668
task                  360.32      3   682.24     54  9.506
6 3.846e-05 ***
group:task             19.82      3   682.24     54  0.522
9   0.66838
interface:task         63.57      3   597.34     54  1.915
6   0.13800
group:interface:task   29.47      3   597.34     54  0.888
0   0.45327
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1


Mauchly Tests for Sphericity

                     Test statistic    p-value
task                        0.25937 0.00042103
group:task                  0.25937 0.00042103
```

```
interface:task              0.25330 0.00035418
group:interface:task        0.25330 0.00035418


Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

                      GG eps Pr(>F[GG])
task                 0.61066  0.0007443 ***
group:task           0.61066  0.5819916
interface:task       0.65072  0.1631545
group:interface:task 0.65072  0.4182608
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

                        HF eps   Pr(>F[HF])
task                 0.6763933 0.0004494572
group:task           0.6763933 0.5997589037
interface:task       0.7288888 0.1573540558
group:interface:task 0.7288888 0.4277321101
```


Mean NASA-TLX Scores by Task and Interface
Mean NASA-TLX Score
0
2
4
6
1
2
3
4
Task
interface
SPARQL
VQB


Interaction of Interface and Task on NASA-TLX Scores
Mean NASA-TLX Score
2
3
4
5
6
7
1
2
3
4
Task
interface
SPARQL
VQB



Q3 - How hurried or rushed was the pace of the task?

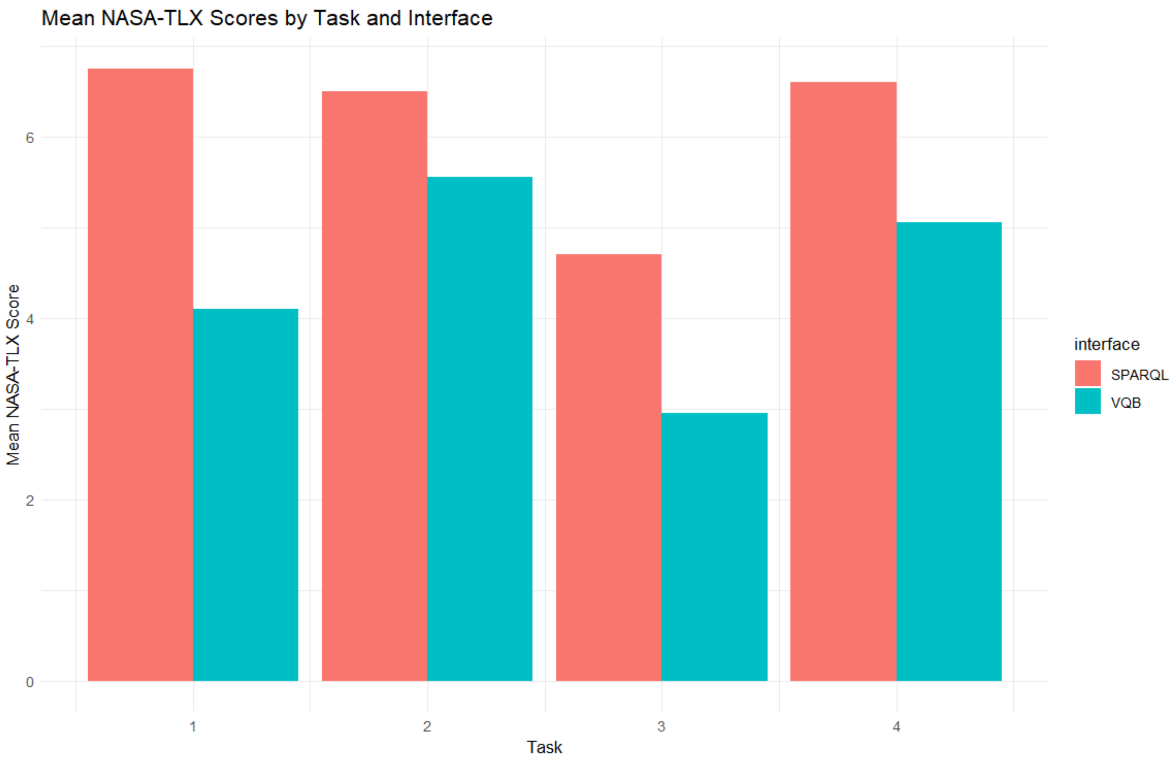
Results

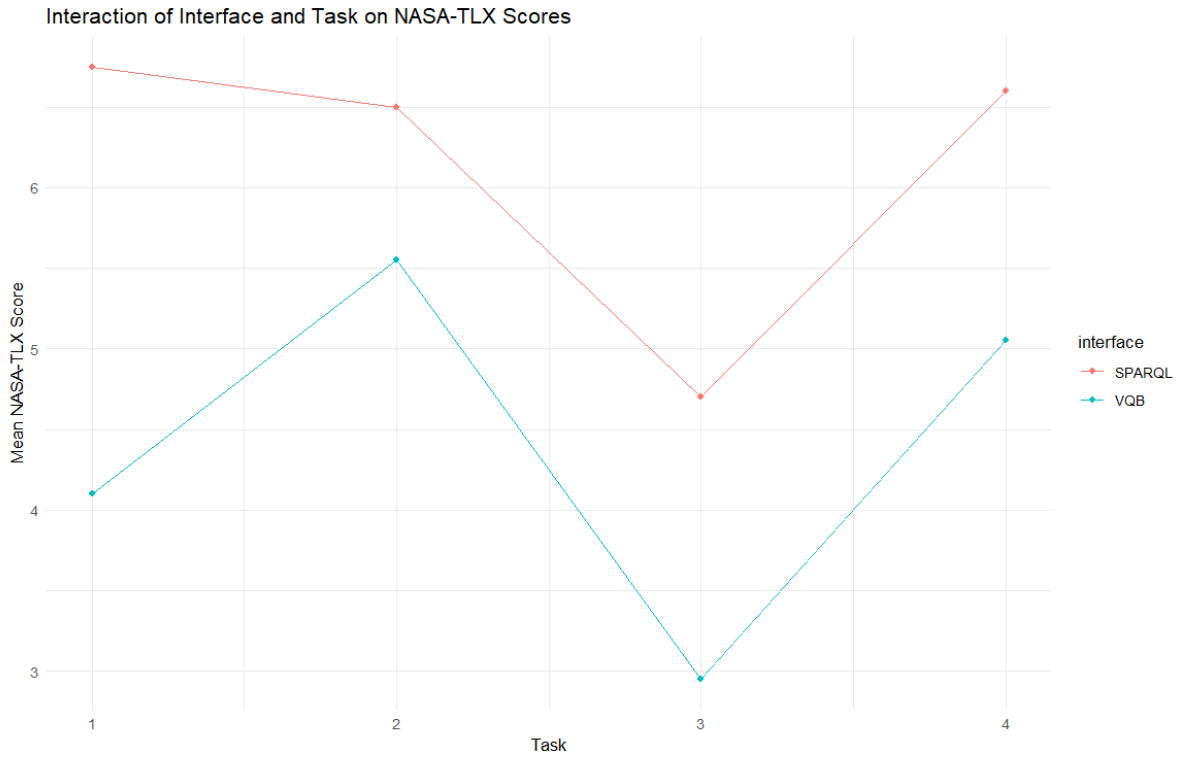
Interface	Mean
VQB	4.41
SPARQL	6.14

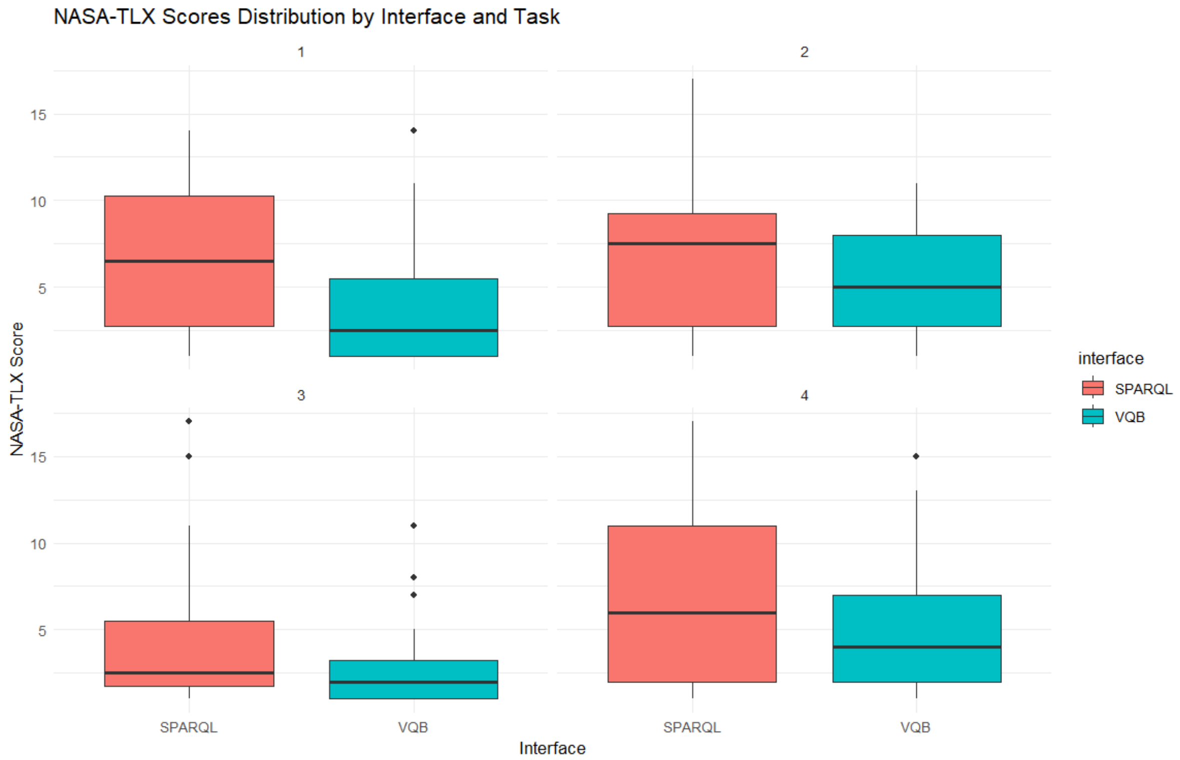
```
Univariate Type III Repeated-Measures ANOVA Assuming Spher
icity

                     Sum Sq num Df Error SS den Df F value
Pr(>F)
(Intercept)          4452.1      1  1352.90     18 59.2341
4.233e-07 ***
group                  62.5      1  1352.90     18  0.8315
0.37388
interface             119.0      1   304.95     18  7.0256
0.01627 *
group:interface        38.0      1   304.95     18  2.2445
```

```
0.15143
task                  119.6      3   524.30     54  4.1060
0.01071 *
group:task             66.6      3   524.30     54  2.2865
0.08903 .
interface:task         14.9      3   262.45     54  1.0202
0.39097
group:interface:task    2.7      3   262.45     54  0.1835
0.90719
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1


Mauchly Tests for Sphericity

                     Test statistic  p-value
task                        0.46637 0.026057
group:task                  0.46637 0.026057
interface:task              0.61247 0.146324
group:interface:task        0.61247 0.146324


Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

                      GG eps Pr(>F[GG])
task                 0.73167     0.0210 *
group:task           0.73167     0.1104
interface:task       0.76691     0.3782
group:interface:task 0.76691     0.8609
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

                        HF eps Pr(>F[HF])
```

```
task                   0.8374036 0.01608316
group:task             0.8374036 0.10141829
interface:task         0.8856839 0.38536362
group:interface:task 0.8856839 0.88716047
```







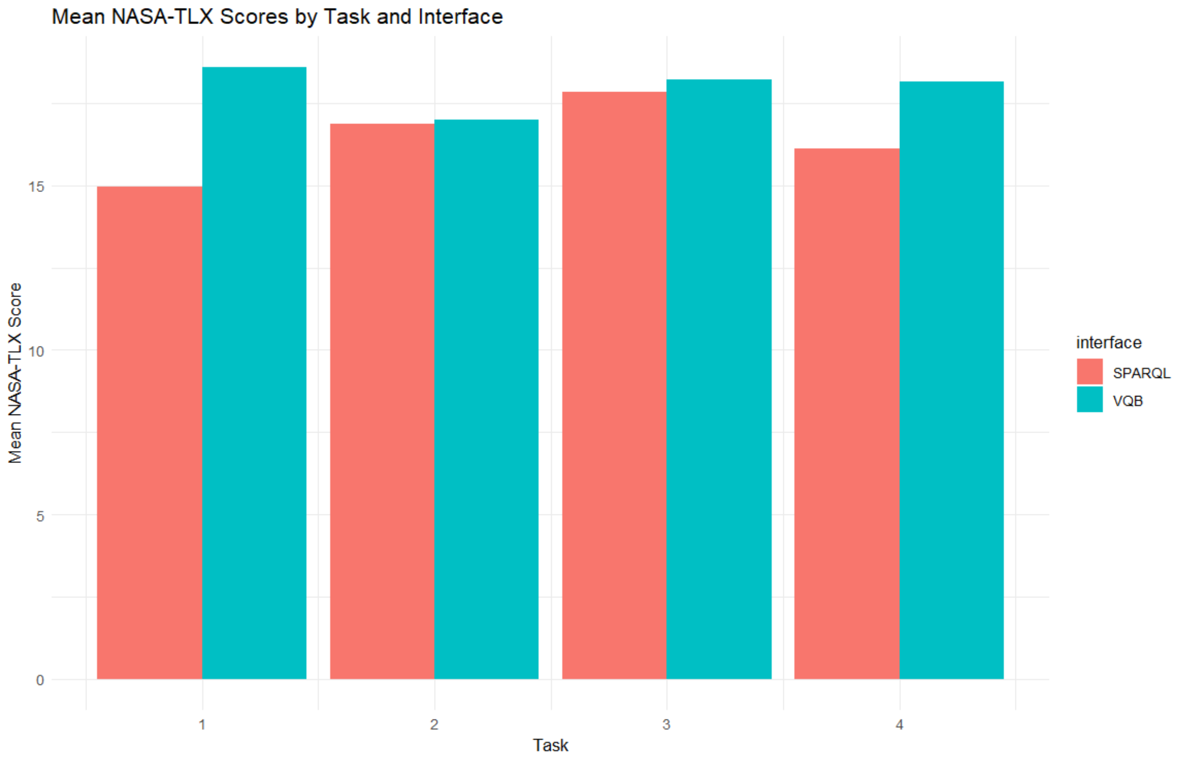
▼ Q4 - How Successful were you in accomplishing what you were asked to do?

Results

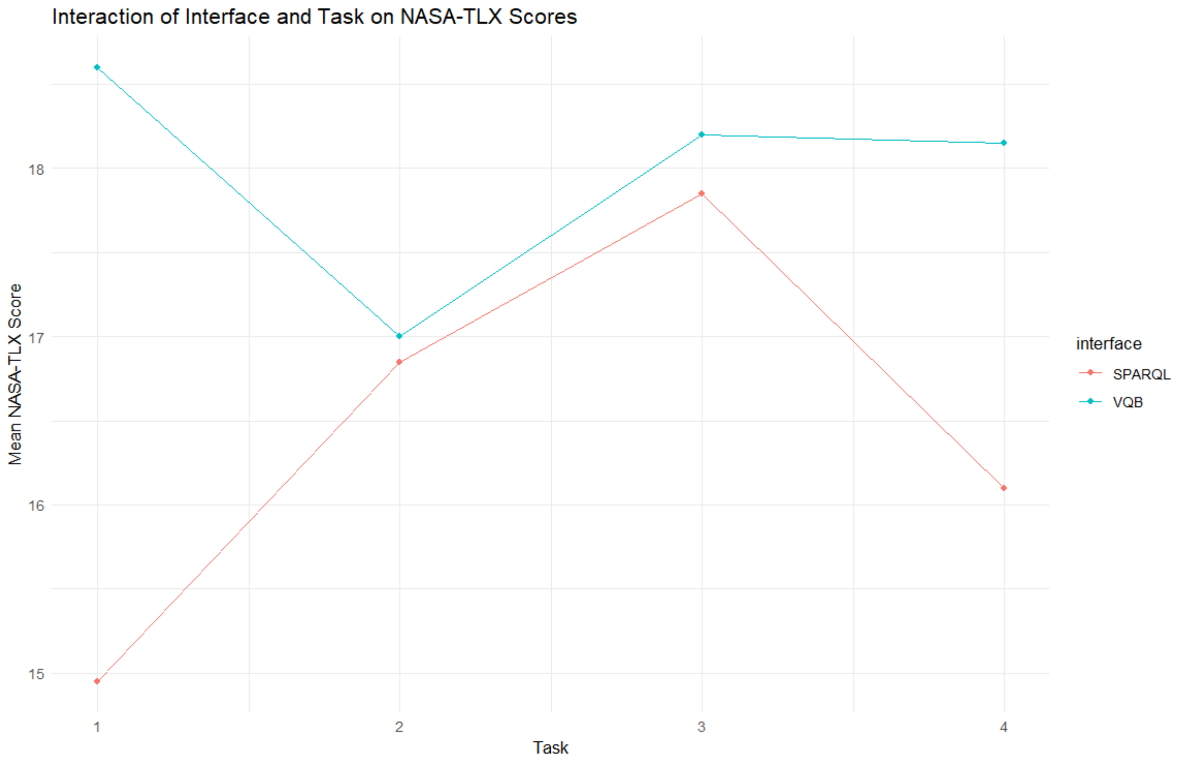
Interface	Mean
VQB	18.0
SPARQL	16.4

```
Univariate Type III Repeated-Measures ANOVA Assuming Spher
icity

                     Sum Sq num Df Error SS den Df   F val
ue    Pr(>F)
(Intercept)           47403      1   704.80     18 1210.63
85 < 2.2e-16 ***
group                    18      1   704.80     18    0.46
55  0.503773
interface                96      1   150.65     18   11.48
22  0.003273 **
group:interface          23      1   150.65     18    2.68
84  0.118444
task                     38      3   308.60     54    2.19
75  0.098897 .
group:task                3      3   308.60     54    0.20
27  0.894088
interface:task           81      3   245.55     54    5.90
84  0.001459 **
group:interface:task     19      3   245.55     54    1.36
35  0.263752
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1


Mauchly Tests for Sphericity

                     Test statistic  p-value
```

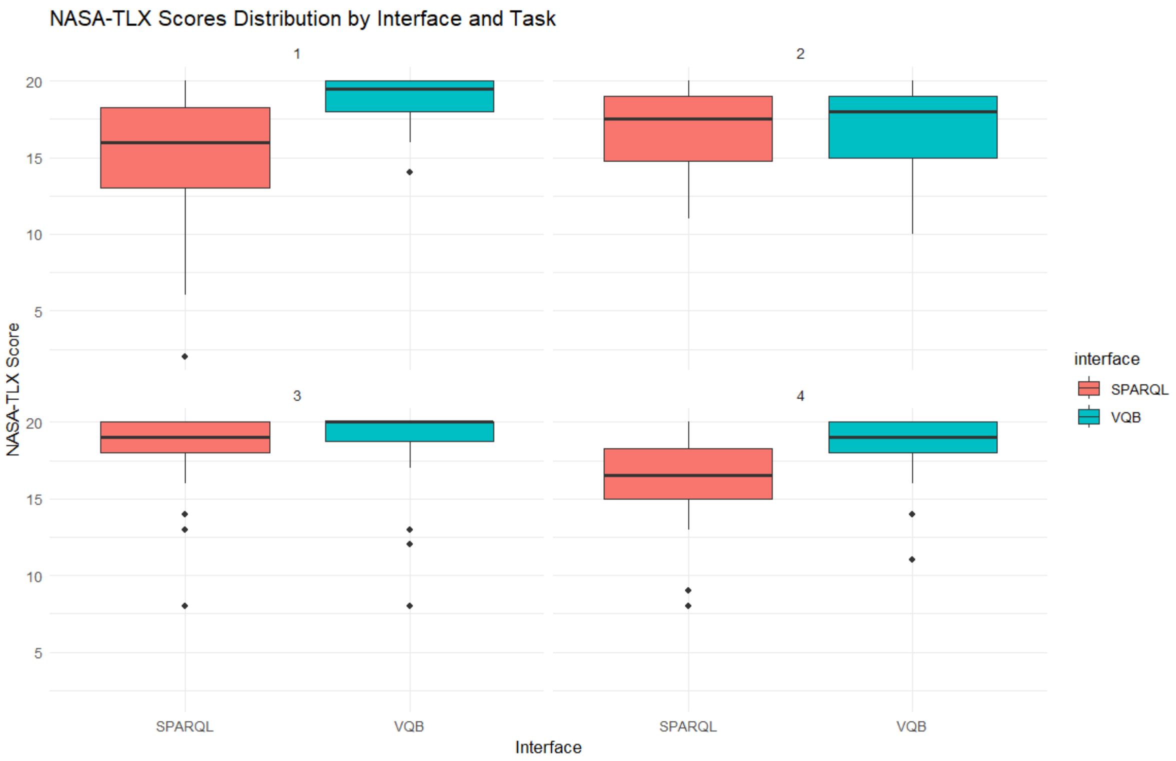
```
task                        0.26539 0.000498
group:task                  0.26539 0.000498
interface:task              0.67292 0.250905
group:interface:task        0.67292 0.250905


Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

                      GG eps Pr(>F[GG])
task                 0.53704   0.137590
group:task           0.53704   0.770330
interface:task       0.78062   0.003704 **
group:interface:task 0.78062   0.267752
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

                        HF eps Pr(>F[HF])
task                 0.5819215 0.13344262
group:task           0.5819215 0.78821989
interface:task       0.9046532 0.00218416
group:interface:task 0.9046532 0.26574955
```



Mean NASA-TLX Scores by Task and Interface



Interaction of Interface and Task on NASA-TLX Scores



▼ Q5 - How hard did u have to work to accomplish your level of performance?

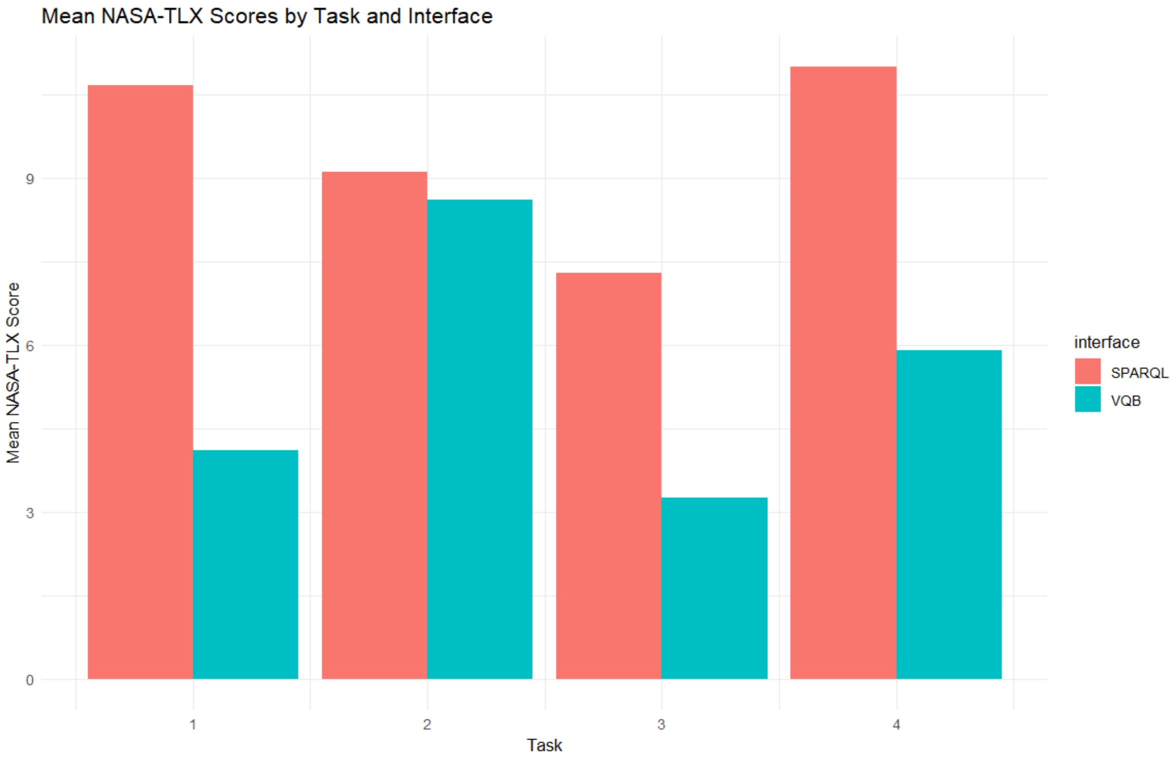
Results

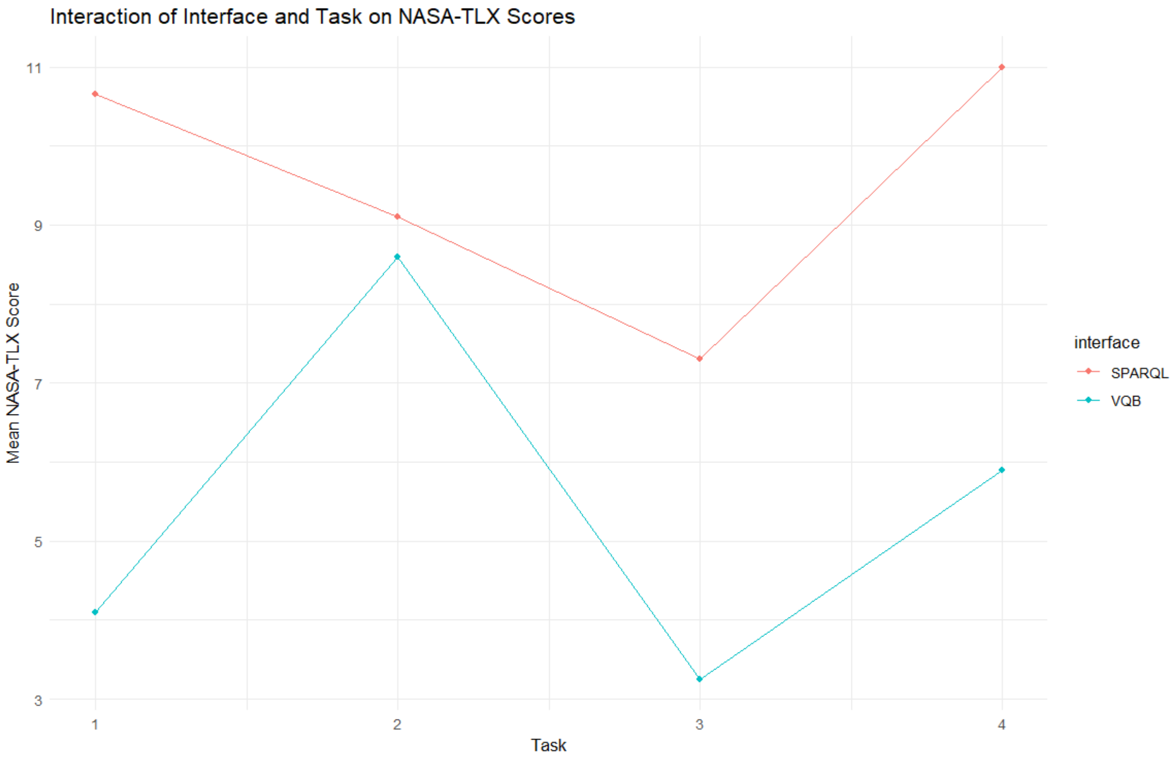
Interface	Mean
VQB	5.46
SPARQL	9.51

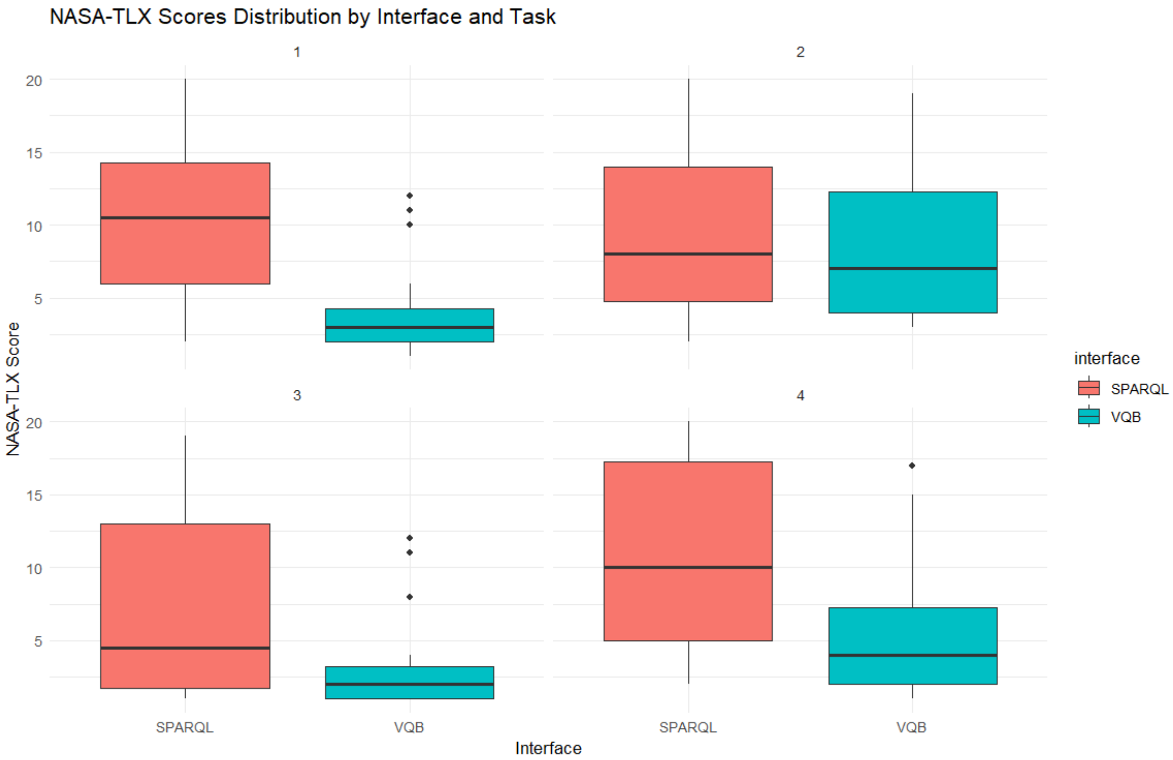
```
Univariate Type III Repeated-Measures ANOVA Assuming Spher
icity

                    Sum Sq num Df Error SS den Df F value
Pr(>F)
(Intercept)         8970.0      1  1623.47     18 99.4536
9.315e-09 ***
group                562.5      1  1623.47     18  6.2366
0.0224325 *
interface            656.1      1   687.78     18 17.1710
0.0006098 ***
group:interface       50.6      1   687.78     18  1.3249
```

```
0.2647716
task                  307.6      3   474.12     54 11.6789
5.240e-06 ***
group:task            140.2      3   474.12     54  5.3245
0.0027475 **
interface:task        199.6      3   615.42     54  5.8365
0.0015762 **
group:interface:task   30.5      3   615.42     54  0.8928
0.4508651
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1


Mauchly Tests for Sphericity

                     Test statistic p-value
task                        0.36428 0.00480
group:task                  0.36428 0.00480
interface:task              0.85160 0.74865
group:interface:task        0.85160 0.74865


Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

                      GG eps Pr(>F[GG])
task                 0.70816  8.299e-05 ***
group:task           0.70816   0.008072 **
interface:task       0.90072   0.002383 **
group:interface:task 0.90072   0.442685
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

                        HF eps   Pr(>F[HF])
```

```
task                 0.8055402 3.289695e-05
group:task           0.8055402 5.621846e-03
interface:task       1.0751237 1.576189e-03
group:interface:task 1.0751237 4.508651e-01
```







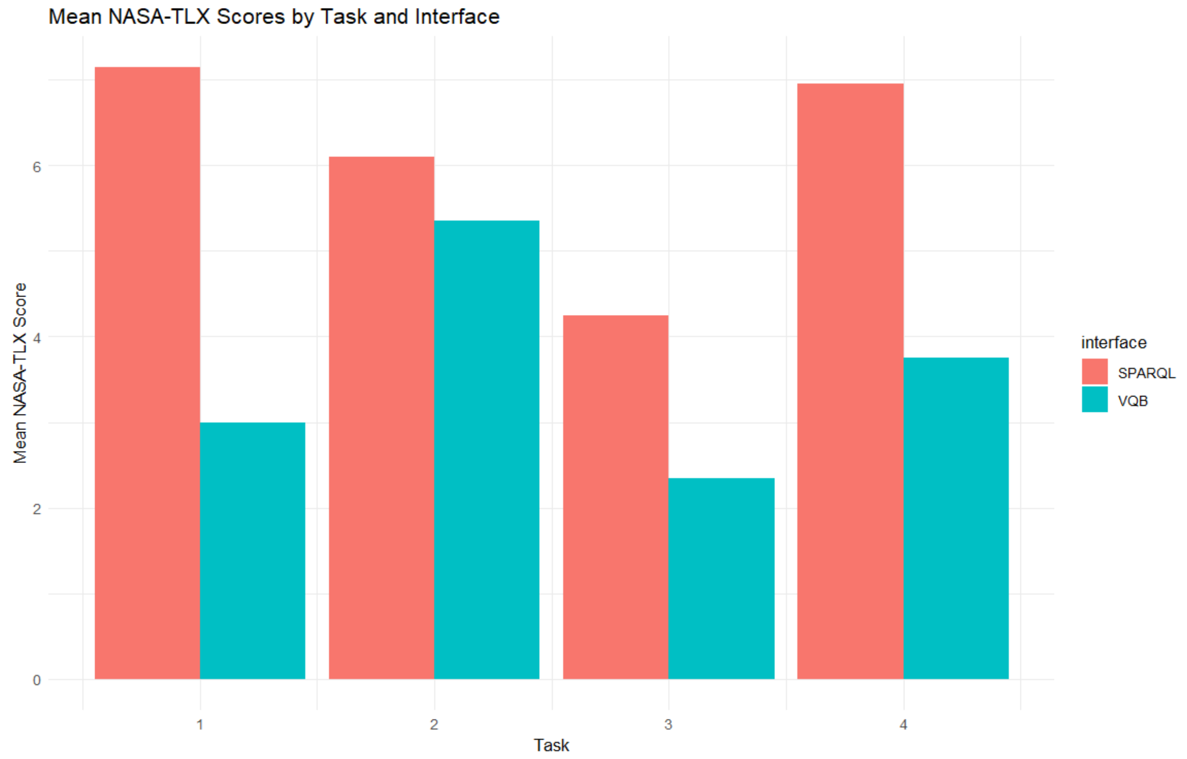
▼ Q6 - How insecure, discouraged, irritated, stressed, and annoyed were u?

Results

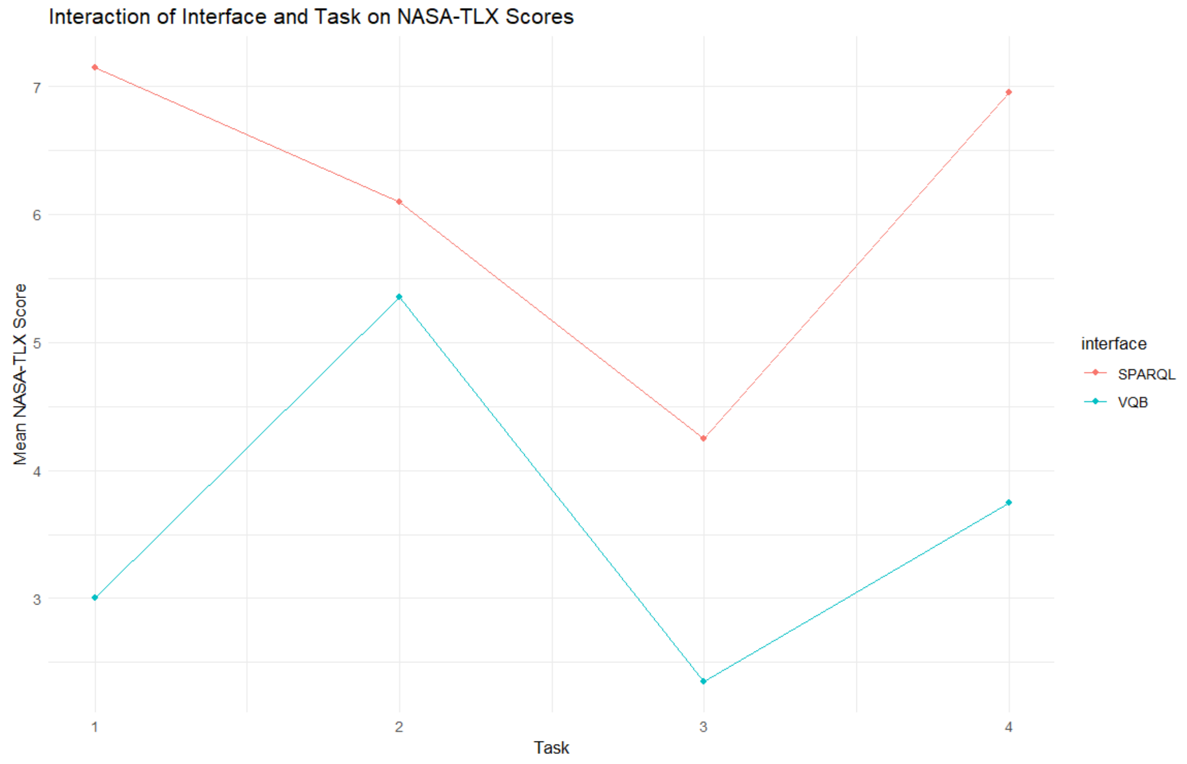
Interface	Mean
VQB	3.61
SPARQL	6.11

```
Univariate Type III Repeated-Measures ANOVA Assuming Spher
icity

                     Sum Sq num Df Error SS den Df F value
Pr(>F)
(Intercept)          3783.0      1  1532.62      18 44.4299
2.969e-06 ***
group                 372.1      1  1532.62      18  4.3701
0.0510315 .
interface             250.0      1   255.72      18 17.5970
0.0005444 ***
group:interface        24.0      1   255.72      18  1.6911
0.2098638
task                  138.7      3   561.88      54  4.4441
0.0073049 **
group:task             40.6      3   561.88      54  1.3022
0.2831494
interface:task         66.4      3   331.18      54  3.6063
0.0189942 *
group:interface:task   25.7      3   331.18      54  1.3982
0.2533154
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1


Mauchly Tests for Sphericity

                     Test statistic p-value
task                        0.85998 0.77349
group:task                  0.85998 0.77349
```

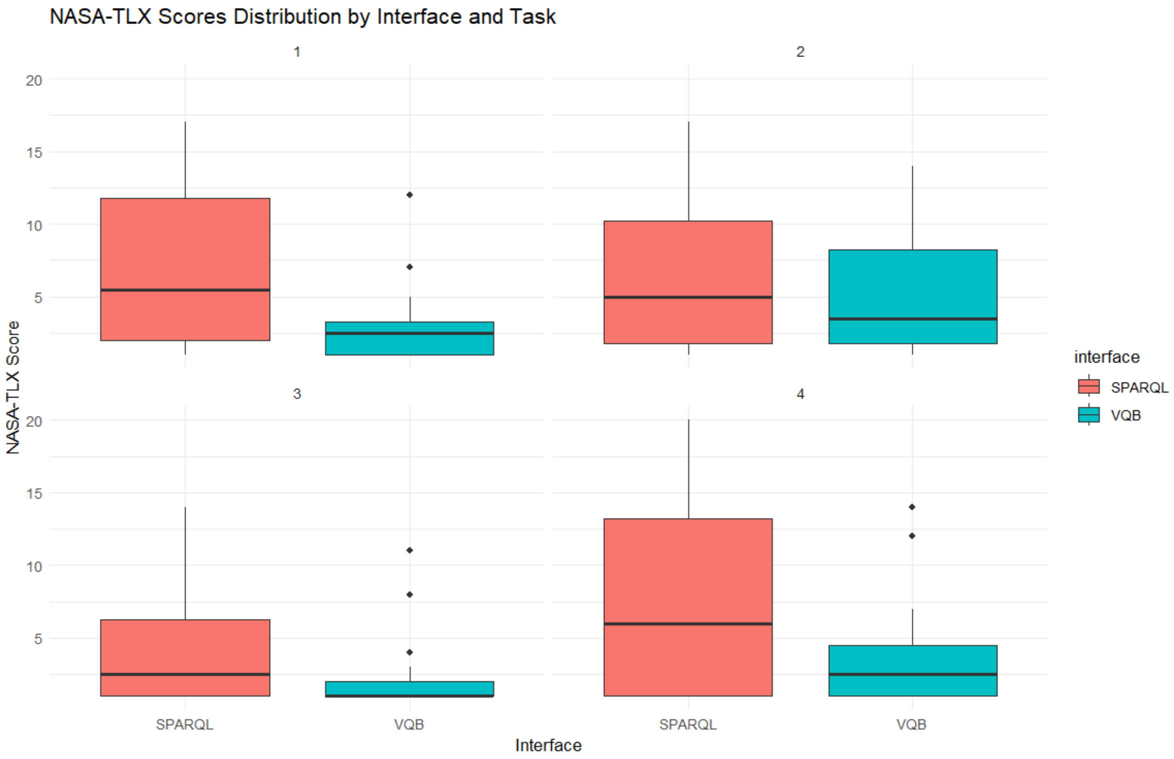
```
interface:task              0.79328 0.56856
group:interface:task        0.79328 0.56856


Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

                      GG eps Pr(>F[GG])
task                 0.91685   0.009235 **
group:task           0.91685   0.284095
interface:task       0.85408   0.025500 *
group:interface:task 0.85408   0.256992
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

                       HF eps  Pr(>F[HF])
task                 1.098636 0.007304873
group:task           1.098636 0.283149392
interface:task       1.007969 0.018994172
group:interface:task 1.007969 0.253315376
```



Mean NASA-TLX Scores by Task and Interface



Interaction of Interface and Task on NASA-TLX Scores


NASA-TLX Scores Distribution by Interface and Task
1
2
3
4
20
15
10
5
NASA-TLX Score
SPARQL
VQB
Interface
interface
SPARQL
VQB