UNIVERSITY OF BERGEN

MASTER THESIS

Visual Analytics for Fishing Vessel Operations

Author:
Syver STORM-FURRU

Supervisor:
Dr. Stefan BRUCKNER

A thesis submitted in fulfillment of the requirements for the degree of Masters of Informatics

in the

Visualization Group
Department of Informatics

15th October 2019

UNIVERSITY OF BERGEN

Abstract

Faculty of Mathematics and Natural Sciences

Department of Informatics

Masters of Informatics

Visual Analytics for Fishing Vessel Operations

by Syver STORM-FURRU

This thesis presents VA-TRaC, a geovisual analytics application developed in collaboration with the Norwegian Directorate of Fisheries. VA-TRaC is used for identification and verification of illegal catch operations performed by Norwegian fishing vessels. The system uses automatic methods to identify possible catch operations based on the vessel's trajectory, whose results are presented to an analyst through a web-based interface of multiple visualization views. The analyst can use these views to explore the results, verify the correctness of the automatic methods based on the analysts domain knowledge, and modify parameters of the automatic methods. The thesis discusses various aspects of the development of VA-TRaC, including its design, implementation, workflows, and how the system was validated.

Acknowledgements

I would like to thank the Directorate of Fisheries, particularly the members of the analysis group, for the insightful feedback and the enthusiasm for my work throughout the process of creating VA-TRaC. In particular, I would like to thank John Erik Ågotnes and Kine Iversen for taking so much time out of their day to answer my questions and helping validate the system, and Per Finne for the help in acquiring the data and for discussions about design and indicators suitable for the system.

I would also like to thank my supervisor, professor Stefan Bruckner, for providing insightful comments and presenting interesting papers for me to read during our biweekly meetings, and especially for the thorough readthroughs during the final months of the thesis.

Finally, I would like to thank Hilde-Sofie, my friends, and my family. Thank you for your helpful feedback, but most importantly your patience, support, and for helping reduce stress to tolerable levels.

Contents

Abstract **iii**

Acknowledgements **v**

1 Introduction **1**
1.1 The Problem 2
1.1.1 Automatic Analysis 2
1.1.2 Issues in Automatic Analysis 3
1.2 VA-TRaC, a Visualization-Based Solution 3
1.2.1 Visual Analytics 4
1.2.2 Challenges in Applying Visual Analytics 5
Working in a Governmental Agency 5
Designing Effective Visualizations 6
Validation 6
1.3 Notable Terms 6

2 State of the Art **9**
2.1 What is Visual Analytics? 9
2.1.1 Defining Works 9
2.1.2 The Components of Visual Analytics 11
Analytical Reasoning: Knowledge Discovery and Sense Making 11
Data Representations and Transformations 13
Visual Representations 14
Interaction 16
Production, Presentation and Dissemination & Moving Research into Practice 17
2.2 Visual Analytics for Fishery Management 18
2.2.1 Characteristics of Spatiotemporal Data 19
Data Attributes 20
2.3 Visualization Validation 21
2.3.1 Additional Validation Approaches 22

3 Domain Background **25**
3.1 Regulations 25
3.2 The Inspector 26

3.3 The Control Analyst 27
3.4 The Statistician 28
3.5 The Analysis Group 28
3.6 Fisheries Management in Academia 28

4 The VA-TRaC System **31**
4.1 Chapter Terminology 31
4.2 Data Abstraction 31
4.2.1 Raw data 33
4.2.2 Data Features 34
4.2.3 Indicator Algorithms 36
4.2.4 Speed Classification 36
4.2.5 kNN Gear Classification 37
4.2.6 1$ Pattern Recognition 37
Verifying Indicators 39
4.3 Task Abstraction 39
4.3.1 T1: Identify Illegal Catch Operations 41
4.3.2 T2: Explore Vessels' DCA Reports 42
T2.1: Overview Based on Reporting State 42
T2.2: Overview Based on Location 43
4.3.3 T3: Evaluate Indicated Operations 43
4.3.4 T4: Evaluate Modifications to Indicator Parameters 44
4.3.5 T5: Filter Vessels 45
4.4 Visual Abstractions and Interaction Idioms 46
4.4.1 Visual Abstraction in Practice 48
4.5 Application Details 50
4.5.1 Overview 50
4.5.2 Color Choices 51
4.5.3 The Map View 52
4.5.4 The Timeline 57
The Tabular View 58
4.5.5 Secondary Views 60
The Evaluation Viewer 61
Vessel Details 62
Template Viewer 62
Speed Indicator Settings 63
k-NN Settings 64
Template Group Viewer 66
4.6 Summary 67

5 Implementation **69**
5.1 Software Architecture 69
5.2 Technology Choices 71

6 Workflows **75**
6.1 Identifying Unreported Catch Operations 75
6.2 Building New Templates and Tuning Parameters 79

7 Validation **83**
7.1 Description of Validation Measures 83
7.1.1 Early Stages 83
7.1.2 Development Stage 84
7.1.3 Post-Development Stage 84
7.1.4 Employed Patterns 85
7.2 Threats and Validations 86
7.2.1 Domain Situation 87
7.2.2 Data/Task Abstraction 88
7.2.3 Visual Encoding/Interaction Idiom 89
7.2.4 Algorithm 90
7.3 Takeaways from Validation 92

8 Conclusion **95**
8.1 Recommendations 96

A Appendix: Interview notes and validation results **99**
A.1 Interview questions 99
A.1.1 System 99
A.1.2 Process 100
A.2 Interview with main stakeholder 100
A.2.1 Agenda and purpose 100
A.2.2 Notes 100
A.3 Interview with stakeholder from analysis and statistics department . . 101
A.3.1 Agenda and purpose 101
A.3.2 Notes 101

B Appendix: System Usability Scale questionnaire **103**
B.1 Questionnaire 103
B.2 Scoring 104

Bibliography **105**

List of Figures

1.1 Related fields of visual analytics 5

2.1 The visual analytics process 10
2.2 Pirolli and Card's sensemaking loop 12
2.3 Keim et al.'s sensemaking model 12
2.4 Sacha et al.'s knowledge discovery model 13
2.5 Haber and McNabb's visualization process 15
2.6 Chi's Data State Model 16
2.7 Munzner's Nested Model of Visualization Design 21

4.1 Munzner's datatype taxonomy 32
4.2 kNN performance 38
4.3 Initial derived templates 39
4.4 Brehmer and Munzner's task taxonomy 40
4.5 5DS: Sheet 1 49
4.6 5DS: Sheet 2 49
4.7 VA-TRaC overview 50
4.8 User workflow 51
4.9 Legend and colorscheme 52
4.10 Zoomed-out map 53
4.11 Map with trajectories 54
4.12 Trajectory toggle: All vessels 55
4.13 Trajectory toggle: Indicated operations 55
4.14 Zoomed-in map 56
4.15 Timeline 58
4.16 Tabular view 60
4.17 Indicator evaluation view 61
4.18 Template viewer 63
4.19 Speed indicator view 64
4.20 Map with DCA reports 65
4.21 Vessel group viewer 66

5.1 VA-TRaC 's architecture and data flow 70

6.1 Workflow: Selection of vessels 76
6.2 Workflow: Data quality assessment 76

6.3 Workflow: Geography-based assessment 77
6.4 Workflow: Speed assessment 78
6.5 Workflow: Template highlighting 78
6.6 Workflow: Assessment of earlier DCAs 79
6.7 Workflow: Template building 80
6.8 Workflow: Template building (cont.) 80
6.9 Workflow: Speed indicator tuning 81

7.1 Munzner's design validation framework 86
7.2 Validations for VA-TRaC 87
7.3 SUS questionnaire results 89

Chapter 1

Introduction

Seafood is one of the cornerstones of the Norwegian economy, with a long and storied history. More than one thousand years after the first trade of fish started, the seafood industry is still the second largest industry in Norway measured by gross domestic product, and serves over 10% of the global fish market. The industry consists of many different domains, including fishery, processing and aquaculture, each employing thousands of people [76].

Due to the size, impact, and complexity of the industry, multiple Norwegian governmental bodies are fully or partly dedicated to overseeing its operations. The Directorate of Fisheries is the main body fully dedicated to governance, and provides control and analysis of the most aspects of the industry. This includes governing traditional vessel fisheries, fish farming, landing facilities, and surrounding infrastructure. The Directorate of Fisheries consists of both a central office as well as multiple regional offices. The central office provides regulation, statistical analysis, and software development, and informs the industry of various events such as legislative changes. The regional offices are the operative units, and collaborate with the coast guard and police to uncover crimes.

The fleet of fishing vessels currently consists of 435 active vessels with a length over 15 meters [40]. All of these must report their operations using a system known as the Electronic Reporting System, commonly referred to as ERS. A normal fishing trip consists of three types of ERS reports: First, the vessel signals that it is commencing a fishing trip when it leaves port and notes any fish already on board. Second, it must report daily catch activities (DCAs), that include the quantities of all species caught, the duration of the fishing operations performed, and the fishing gear used. Finally, the vessel reports arrival to port. The report includes information on the total quantities of fish on board and the quantities that are intended for landing. Additional reports are needed when crossing borders to other countries' waters.

In addition to ERS reports, vessels over 15 meters long must at all times report their positions hourly. This is done through satellite-based GPS, known as the vessel monitoring system (VMS). Many also choose to use the automatic identification system (AIS), which uses land-based trackers to collect position data every five minutes.

Positions are reported both to verify that the vessel is following legislation and as a safety measure if the vessel should find itself in distress. The Directorate of Fisheries employs inspectors, who with the support of various other employees ensure that vessels report correctly. The domain and work of the Directorate's employees is further described in Chapter 3.

1.1 The Problem

ERS reporting is done manually by vessels' crews, and situations commonly arise where the vessels forget to report daily, or send delayed reports. These situations are by themselves breaches of regulation. Additionally, since ERS reporting is done manually, opportunities arise for more nefarious activities. Vessels could report fake fishing operations, deliberately report wrong values, avoid reporting, or otherwise attempt to circumvent the regulations, and it is up to the Directorate of Fisheries' employees to find out which vessels have just forgotten their daily report and which are attempting to deceive. Though currently handled well, the number of reporting vessels is rapidly increasing. As of 2018, there were 435 registered active Norwegian fishing vessels with a length over 15 meters, meaning they are legally bound to report positions and fishing operations to the directorate of Fisheries [40]. Over the coming years, this is expected to increase to an ever-larger percentage of the close to six thousand registered Norwegian fishing vessels [41]. This will strain the inspectors and analysts that attempt to identify and understand the reporting infractions, as they will need to validate the reporting of more and more vessels. There are currently few tools that increase the effectiveness of this process, and the Directorate's employees will need the help of such tools if they are to handle the influx of new reports effectively.

1.1.1 Automatic Analysis

The modern-day problem of working with the massive amounts of collected data has fostered the use of algorithms and statistical analysis as tools to gain insight from large datasets. Fields such as data mining and machine learning have successfully built algorithmic models for a variety of use cases, from identifying cancer [39], over user analysis for e-commerce [10], to automatic video captioning [89]. Using such automatic methods could be a possible solution to the problem of increased reporting, and the Directorate has recently started a working group called the analysis group to explore the possibilities found in the automatic analysis domain. The analysis group is currently building a set of so-called *indicators* for inspectors. Indicators are values based on analysis of the data collected by the directorate that provide indications of whether a vessel may have performed illegal activities and may be worth investigating. The goal is to alleviate some of the work the inspectors have to do by filtering down the possible candidates for inspections and to identify repeat offenders.

1.1.2 Issues in Automatic Analysis

Fully automatic analysis systems are not always a perfect solution, and have their own challenges. Automatic data analysis used to replace human labor in sensitive areas requires models and algorithms with special thought and consideration put into the ethical ramifications the system might have [14]. The indicator project of the Directorate's analysis group is attempting such a replacement, as the results of an inspector interpreting indicators may lead to legal action. Developers of such systems need to specifically consider how transparent and predictable the model's results are [14], as the analysis group's decision to work on indicators as supplements to inspectors exemplifies. This is a hard problem, and it may often be more suitable to use automatic analysis as a supplement to humans instead of replacing them.

Another issue with fully automatic analysis systems is that they are usually highly problem-specific, to the point where they only answer a specific question. In cases where the specifics of the problem are hard to define, it is very difficult to create fully automatic systems that function well. Further, the algorithms often require structured data, and problems where either some or all the data come from unstructured sources, are currently hard to solve using fully automatic methods. When analyzing vessel reports, unstructured data can provide a lot more insight than the reports themselves show. For example, the vessel may have hit bad weather, and thus been unable to report, or the reporting frequency may have been reduced due to technical difficulties at a base station. Though it would be possible to encode such situations into the fully-automatic model, it would quickly become very large and difficult to understand, compounding the issue of transparency.

As fishery management will need both automation and human oversight during the coming years, a combined approach must be considered instead.

1.2 VA-TRaC, a Visualization-Based Solution

For problems where automatic systems and people work in tandem, users must understand their results quickly and efficiently if the system is to provide benefits over performing the labor manually. Having established that analysis of vessel reports is best done as a collaboration between a human and a computer, one must choose how to interface between the two. A system must communicate results from the data and analysis to the user so that the user understands the data quickly and precisely. In some cases, it may also be beneficial to allow the user to steer the automatic analysis to be more effective. Data visualization, from here on known as visualization, is a good candidate for interfaces with these traits. Computer-based visualization "provides visual representations of datasets designed to help people carry out tasks more effectively"[64], and is therefore well suited when the goal is to increase the performance of human labor when collaborating with a computer.

This thesis presents VA-TRaC (Visual Analytics for Trajectory Analysis and Classification in Fishing Vessel Operations), a prototype visual analytics application for identifying irregularities in the fishing fleet's reporting of catch operations. The system has three operational goals: First, it should identify fishing operations automatically, and let a user compare these with real reports. Second, it should allow analysts to refine the automatic models. Third, it should be scalable to thousands of vessels, meeting the demands faced by the Directorate over the coming years.

VA-TRaC consists of two parts. Three indicators, in the vein of those in development by the analysis group, identify and classify vessels' catch operations automatically, even when no operations have been reported. A series of visualizations display the vessel data in a web frontend, allowing inspectors to verify and understand the indicator's results. Additionally, the system allows analysts to explore and refine the parameters of the indicators, potentially increasing the quality of indications over time.

1.2.1 Visual Analytics

VA-TRaC is a Visual Analytics system. Visual analytics is a relatively new subdomain of visualization concerning itself with the effective analysis of very large datasets by combining automatic methods, visualization, and human cognition. It consists of elements from visualization, human-computer interaction, automatic analysis, and related fields (Figure 1.1 shows related fields and how they converge to form an effective whole for data analysis). Though visual analytics started as a field focused on national intelligence, the scope has broadened, and the term "visual analytics system" is today used for applications that combine the advantages of automatic analysis and visualization to analyze large amounts of data.

Visual analytics design studies such as this thesis are the combination of applied research and validation, and seek to build a knowledge base of visual analytics capabilities based on the results of evaluations, as worded by Thomas and Cook [78]. A wide array of design studies showing the benefits of visual analytics have been performed across many different domains, data types, devices, and user groups. Examples are as far-reaching as healthcare [83], financial markets [92], social networks [46] and even music [8].

The scientific work done in the field has provided valuable insights into the development and validation of VA-TRaC. Chapter 2 provides a report on related works and the state of the art in visual analytics research.

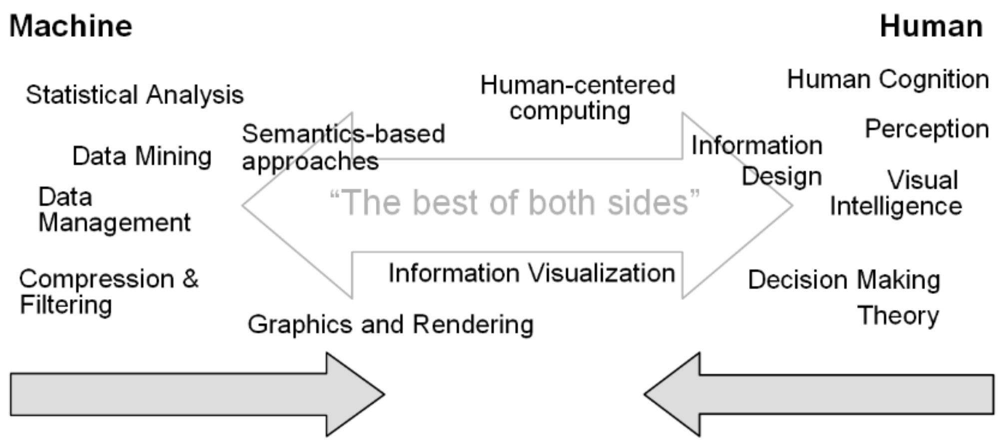

FIGURE 1.1: Related fields of visual analytics, by Keim et al. [55]

1.2.2 Challenges in Applying Visual Analytics

In domains such as fisheries management, where there are well-defined systems and standards for data acquisition and storage, a major challenge is how to gain insights using the data. As established, visual analytics systems give good results in these kinds of environments. However, they are not without challenges, and the design of visual analytics systems is an active field of research.

Each subdomain (see Figure 1.1) of a visual analytics system can be challenging on its own, and the challenges are often very context-dependent. For example, in some systems, the problem to be solved is very clearly defined, and the challenge is algorithmic, for instance in how to performantly convert complex data to visual representations on the screen. In other situations, the challenge may be designing novel techniques of user interaction, for instance in systems with multiple simultaneous users or novel display technologies such as virtual reality headsets. In the case of fisheries management at the Directorate of Fisheries, three categories of challenges were identified: Working alongside existing processes and governmental culture, designing effective and understandable visualizations, and system validation.

Working in a Governmental Agency

A large and old governmental body such as the Directorate of Fisheries develops processes for how work is performed, that over time become a deep part of the agency's culture. To successfully create analysis tools in such an environment, one needs to understand the employees' workflows. Building this understanding includes identifying the organizational structure and key stakeholders, analyzing the workflows of users, and integrating the tool so that they do not need to alter their current situation unnecessarily. Furthermore, a governmental agency is just one part of the whole governmental body, where different agencies, offices, and institutions are interlinked and often depend upon one another, adding a layer of complexity in reaching a good level of understanding. As an outsider in a large organization, this proved a daunting task. Chapter 3 further explores the domain, organization, and workflow in which VA-TRaC was built, while Chapter 6 shows how VA-TRaC is used to help effectivize this workflow.

Designing Effective Visualizations

Just as the combinations of pitch, rhythm, and instrumentation can create practically endless amounts of music, combinations of data, visual encodings, and interaction can create endless possible visualizations [64]. Therefore, designing visualizations that reduce this space into a set of effective encodings in the given context is a core challenge in any visual analytics system. Compounding on that, data visualization has traditionally not been employed much at the Directorate of Fisheries and elsewhere in fisheries management, and the Directorate's employees have varying degrees of fluency in understanding charts and other visual encodings. Designing a visual analytics system where the design space simultaneously is massive and limited proved one of the major challenges in the development of VA-TRaC. Methodology, design justifications, and a presentation of the system is found in Chapter 4, while Chapter 5 presents the system's architecture and implementation details, based on a discussion of the application's non-functional requirements.

Validation

Related to the previous challenges, validating the system is a requirement for evaluating the efficiency of a design. Offering a structured and effective approach to validation that includes the correct stakeholders, proved the final major challenge. The validation approach, as well as validation results, are presented in Chapter 7.

1.3 Notable Terms

The following terminology is used throughout the thesis. As many of these terms may have different meanings depending on context, this section clarifies their use.

- **Validation**: The act of ensuring that a given system is effective at its purpose, and solves a valid problem [64]
- **Design Study**: A design study is a research project in which the researcher attempts to solve a problem for a real-world user [64].
- **Stakeholder**: A person who needs or is interested in the visualization system [75]. This can be a user, developer, or other interested parties.

- **Domain Expert**: A person with specific domain knowledge about fishery and vessel operations. Cases with a different domain are specified explicitly.
- **Indicator**: A value based on the analysis of vessel data that gives an indication as to whether the vessel is performing illegal activities.
- **Call Sign**: An identifier for a vessel, in the form of four to six alphanumeric values.

Chapter 2

State of the Art

This chapter presents the research context in which VA-TRaC was built. It is split into three main sections: Being a visual analytics application, the first section gives an overview of the historical and current trends of visual analytics. The second section provides a deeper look at VA-TRaC 's niche, geovisual analytics for fisheries management, by discussing common data types and previous work in the domain. The third section discusses approaches to design study validation.

2.1 What is Visual Analytics?

Visual analytics is a broad field, integrating features of various other fields while at the same time having evolved into a separate entity from both information visualization and scientific visualization, the two other main subfields of visualization. The following section contains the context and general components of visual analytics research.

2.1.1 Defining Works

Visual analytics developed in the early to mid-2000s as a response to the increasing pervasiveness and quantities of digital data across society. Though there were earlier examples of publications, such as the September-October issue of IEEE Computer Graphics and Applications in 2004 [86], the field is often considered to be first described formally in the work *Illuminating the Path: The Research and Development Agenda for Visual Analytics* by authors Thomas and Cook [78]. The authors define visual analytics as the "use of visual representations and interactions to facilitate faster, more focused, analytical reasoning tasks", which is still a commonly-used definition. Evident already in this definition is the fact that visual analytics doesn't merely build on visualization, but also human-computer interaction research, research into reasoning, and analytics.

While the general focus of *Illuminating the Path* is on visual analytics for homeland security, the authors present a research agenda for the field as a whole, along with recommendations to facilitate effective use of research and applications from a variety of domains for visual analytics, including analytical reasoning, visualizations and interactions, presentations and data processing. The research agenda, recommendations and the motivating "call to action" style of writing in the book have provided ample grounds for visual analytics research, and the book continues to be one of the most-cited works in visual analytics.

The early stages of visual analytics research focused on further refining the definition of visual analytics, defining the boundaries of visual analytics, how visual analytics is distinct from related fields, and analysis of use-cases and challenges. A notable work of the period include Keim et al.'s paper *Visual Analytics: Definition, Process, and Challenges* [55]. The paper presents one of the central ideas in the field, the notion that visual analytics allows solving the problem of information overload from massive data sets by turning it into an opportunity instead of a challenge, provides a thorough overview of how visual analytics is its own sub-domain of visualization and discusses a series of related fields.

Another early breakthrough was made in the form of a general model for the visual analytics process. The model, shown in Figure 2.1, was formalized by Keim et al. in *Visual Analytics: Scope and Challenges* [56]. The formalization builds on previous descriptions of the model using less formal means and describes visual analytics as a cyclical process of going from raw data to knowledge using both visualization and models.

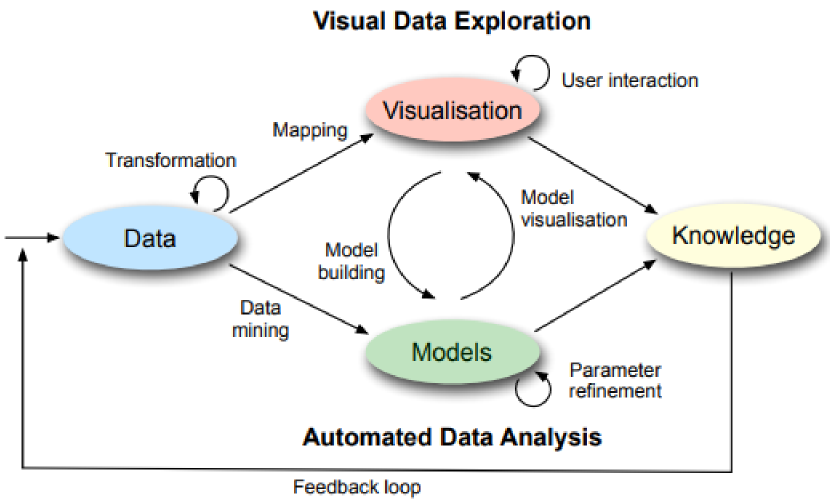

FIGURE 2.1: The Process of visual analytics by Keim et al. [57]

The paper also provides an extension of the famous information visualization mantra from Ben Shneiderman: "Overview first, zoom and filter, details on demand" [73]. The extension, "Analyse first, show the important, zoom and filter, analyze further, details on demand", gives a concise description of how visual analytics differs from information visualization, again building the case for visual analytics as a distinct field [56], and is repeated throughout the literature on visual analytics.

The early, defining works of visual analytics culminates in the 2010 book *Mastering the Information Age: Solving Problems with Visual Analytics* by Keim et al. [57], which distills the most important goals, challenges and definitions of visual analytics research of the time, presents research goals and challenges in related fields such as data mining, cognition, and infrastructure with regards to use in visual analytics, and solidifies visual analytics as its own field.

2.1.2 The Components of Visual Analytics

Each chapter of *Illuminating the Path* describes a component, that when brought together with the other components describe visual analytics as a whole. The following section describes the state of research for applying each component to visual analytics.

Analytical Reasoning: Knowledge Discovery and Sense Making

Visual analytics attempts to construct an effective way of bridging human and automated analysis, and a primary interest is therefore how humans perform their analytical reasoning.

Sensemaking is the process of how people "structure the unknown" by creating frameworks that allow them to comprehend and reason about the unknown, based on knowledge articulation [2]. Considerable visual analytics research effort has gone into the construction, evaluation, and discussion of models of knowledge discovery and sensemaking. In the early days of visual analytics, these models helped distinguish factors of visual analytics when compared to information visualization [55]. More importantly, they can help understand a fundamental aspect of visual analytics, namely the integration of human understanding in the entire analytics process, and can drive choices for interaction, analysis and visualization idioms [57].

A general framework for the sensemaking process in the context of analysis comes from Pirolli and Card [65]. The framework, illustrated in Figure 2.2, separates sensemaking into two distinct loops, foraging and sensemaking. Foraging is the process of reducing a large dataset into a manageable set of relevant information. This is done by iteratively providing higher-precision queries into the data until one or more high-precision sets of suitable size are found. The sensemaking loop, on the other hand, is the process of turning this information into mental models for decision making or hypothesis evaluation. The Pirolli-Card model is probably the most commonly-cited model in discussions of sensemaking [37].

An early and influential paper focusing on knowledge generation from the viewpoint of visualization research is van Wijk's *The Value of Visualization* [82], in which the author describes a generic model for the visualization process, later refined and concretized into the realm of visual analytics by Keim et al. [55]. Seen in Figure 2.3, "The sensemaking loop of visual analytics", focuses on the user building knowledge through the repeated process of creating a visualization from preprocessed data, extracting knowledge from the visual representation, and using the newly gained knowledge to drive further analysis and insight. The results can then be used to create new visual representations of the data, in order to avoid favoring the specific interpretation of the data provided by the first visualization.

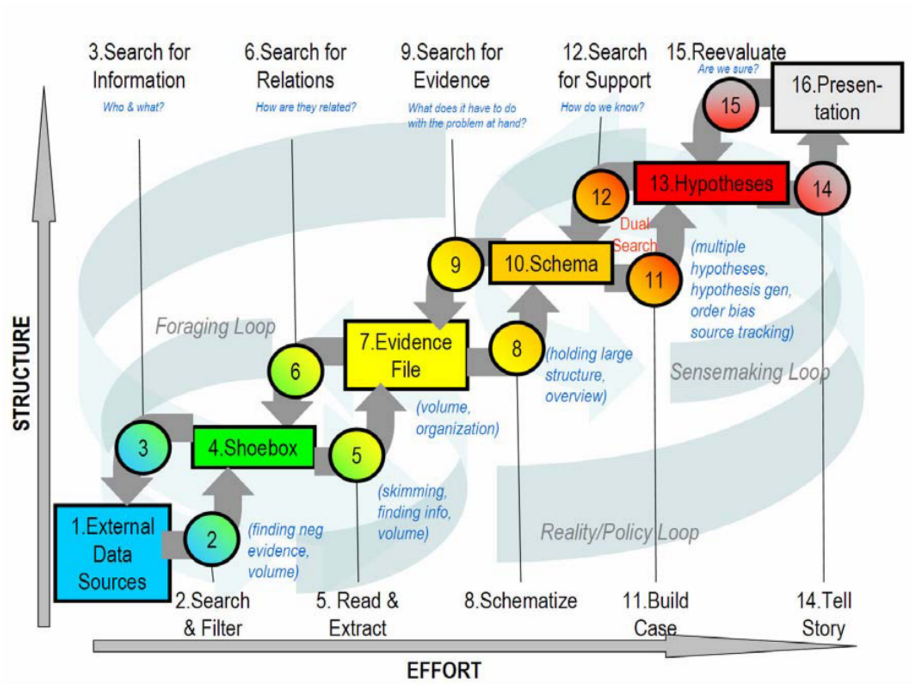

FIGURE 2.2: The Sense Making Loop by Pirolli and Card [65]

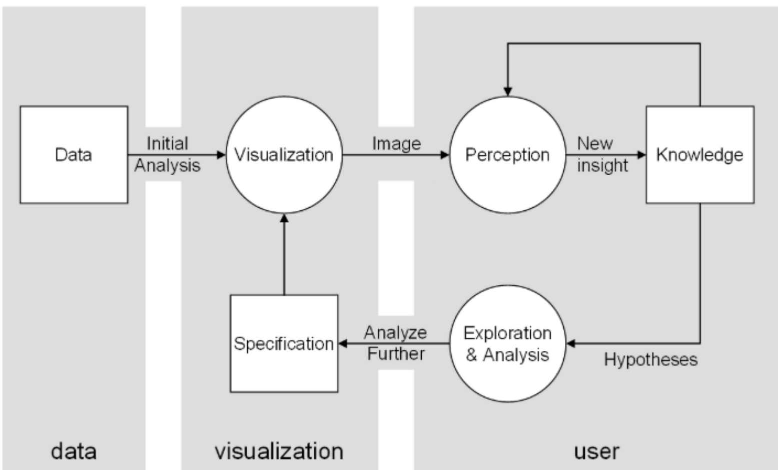

FIGURE 2.3: The Sense Making Model by Keim et al. [55]

The van Wijk-Keim sensemaking model differs from the Pirolli-Card model by focusing squarely on the system of visualization, without focusing too much on the process of human reasoning. This is the criticism made by Sacha et al. [69], who seek to marry the van Wijk/Keim sensemaking loop with Pirolli and Card's model by separating the processes performed by a human and the processes performed by a computer. Their model is illustrated in Figure 2.4. Like Pirolli and Card, The Sacha et al. model uses multiple subloops to describe the sensemaking process. The computer-based portion of the model is somewhat similar to the Keim/van Wijk model, but crucially includes the information visualization pipeline and the Knowledge Discovery and Data Mining (KDD) process. Ribarsky and Fisher have built on this work by integrating principles from cognitive science [67], adding notions of prior knowledge encoded in the computer as well as knowledge already available to the user.

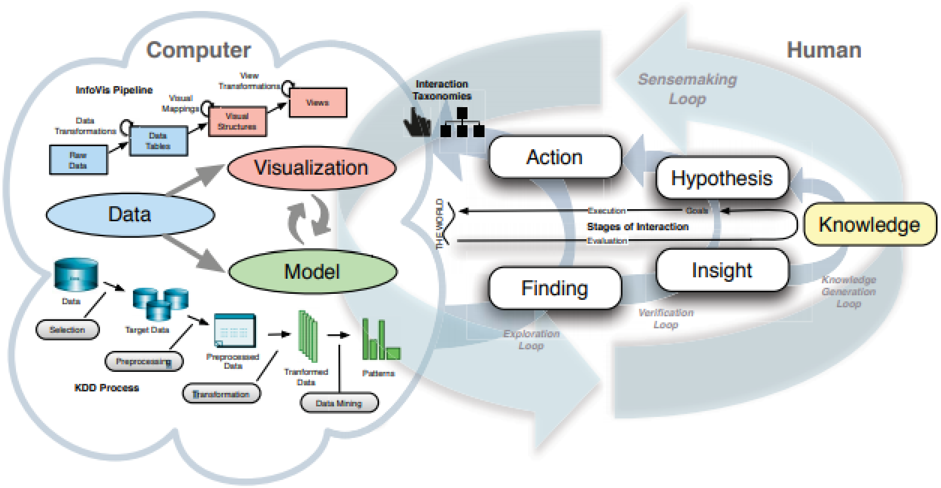

FIGURE 2.4: Sacha et al.'s knowledge discovery model [69]

As noted by Ribarsky and Fisher [67], no exisiting visual analytics system encodes all aspects of their or Sacha et al.'s model, and the authors call for further evaluation and research. However, the models provide a solid foundation for illuminating an important and distinguishing aspect of visual analytics and a roadmap for further research [37].

Data Representations and Transformations

The computational and automatic aspects of visual analytics are described by the "Data Representations and Transformations" component. It consists of two subcomponents: automatic analysis for model building, and data considerations such as storage, transformation, and transportation.

The automatic analysis performed in visual analytics is, for the most part, some form of data mining, applied using principles of KDD [57]. Due to the large variety of data mining algorithms available, visual analytics applications have employed several different, often domain-dependent, methods. In a recent state of the art report on visual analytics and machine learning, Endert et al. [37] provide a classification of different domains of data mining methods and how they have been incorporated into visual analytics tools. Examples include statistical methods such as Bayesian classification and regression models, dimensionality reduction methods such as singular value decomposition, unsupervised clustering algorithms, and classification methods such as decision trees.

When working with sufficiently large data sets, it becomes impossible to load all data into memory at the same time, and it may even become impossible to load all the data onto a single hard drive at the same time. As one of the core promises of visual analytics is the ability to effectively analyze very large data sets, challenges of data scale are both important and common. A paper by Wong et al. [23] outlines such issues, and discusses the state of approaches to mitigate them. The need for expensive, high-latency and low-throughput cloud providers to store extremely large quantities of data is one issue, the lack of databases guaranteeing atomicity, consistency, isolation and durability (so-called ACID-compliance) that support such data quantities is another. The paper also points out the challenges of passing large amounts of data over the network as a growing challenge, and discusses the need for new algorithms that can efficiently process data without keeping all the data in memory or other local storage.

Visual Representations

Visual analytics started in information and scientific visualization circles [56], and as Thomas and Cook note, visual representations are crucial because they allow exploration, and through that give the analyst insight [78]. It is also fundamentally different from automatic analysis: As noted by Keim et al. [56], visual representations don't necessarily become better or more meaningful just due to increases in computational power. Therefore, correct use of theory and empirical studies in visualization and visual representations are a key factor for visual analytics systems to be effective.

One of the most common general models of visualization is the visualization pipeline. The model consists of a step-by-step process of transforming data into a suitable visualization. It was first defined by Haber and McNabb in 1990 [47], who base their model on the following steps, each describing the type of object that is contained in the step: Raw data, derived data, abstract visualization objects, and images (see Figure 2.5). Raw data is, as the name suggests, the data as received from the outside world. Derived data contains the same data but enriched or enhanced, for instance by adding metadata or calculating attributes such as derivatives. Abstract visualization objects are imaginary objects within the extents of space and time, containing attributes such as position, geometry, texture, or reflectivity. Finally, the image is the final representation of the abstract visualization objects on the screen. A visualization is created by transforming data from one form to the next, for instance by applying traditional techniques from computer graphics to turn an abstract visualization object into an image [47].

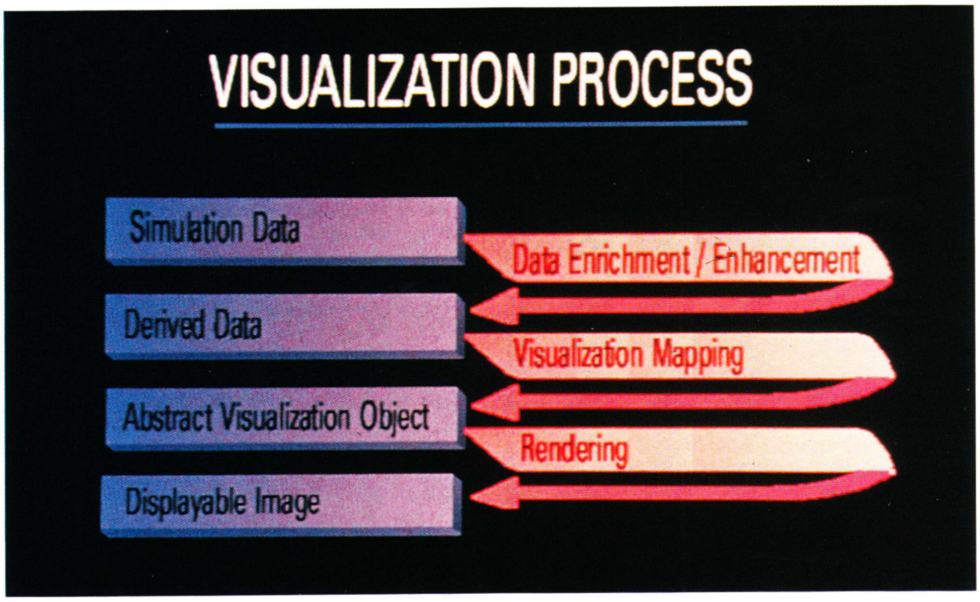

FIGURE 2.5: Haber and McNabb's visualization process [47]

Over the years, multiple extensions and refinements of the model have been proposed. A prominent example comes from Chi [21], who presents the Data State Model, shown in Figure 2.6. The model extends the work of Haber and McNabb by introducing the concept of operators. Unlike transformations, operators change values to another representation within the same level. For instance, operators on the derived data, called analytical abstraction by Chi, may change the representation from a grid to a data graph, which doesn't create any visual abstractions.

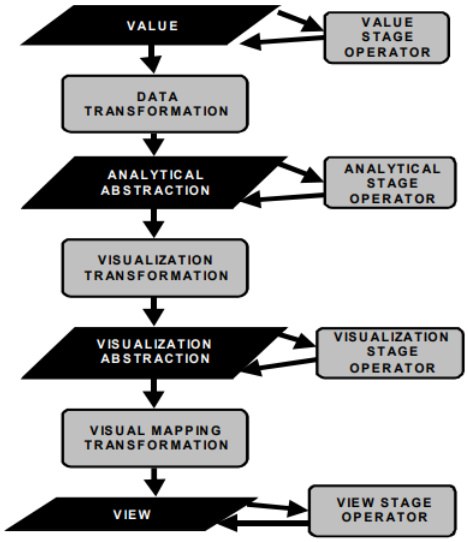

FIGURE 2.6: Chi's Data State Model [21]

The field of visualization, as distinct from other branches of computer science, has mostly focused on the last two steps of the visualization pipeline. In 1967, Bertin defined a model for abstract visualization objects which has become a standard of visualization analysis [64]. In this model, the object is described by two types of attributes: The geometric representation, called the mark, and the attributes controlling the geometric representation, called channels. Marks can encode specific items, such as points, lines, and areas, or links between items. Channels can modify the marks using for instance position, color, shape et cetera [64].

An advantage of marks and channels is that they can be analyzed and evaluated indivdually, based on how effectively they are perceived and how expressively they describe the underlying data [60]. Cleveland and McGill's seminal 1984 article *Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods* [26] evaluates the effectiveness of channels by applying a theory of perception on experiments measuring response time and precision for different chart types, and evaluates a series of common charts based on the results. The results of Cleveland and McGill have provided the groundwork for ample research into the effectiveness of visual encodings, such as Heer and Bostock's use of crowdsourcing to evaluate perception [48], Cleveland's model of visual decoding [25] and Ware's thorough analysis of visual perception [84].

Interaction

Interaction can increase the effectiveness of visual representations, especially when the data is multivariate [18], and early visualization research developed basic methods of interaction to unlock these effectiveness gains. A prominent example comes from Buja et al. [18], who describe focusing and linking as a scheme for effective multivariate data visualization using multiple, simple views that can be interacted with. Despite the early developments, visualization research has mainly focused on representation, while interaction research historically has been grounded in the field of human-computer interaction (HCI), with select techniques applied to visualization [88].

Some notable interaction techniques have found success in information visualization. Direct manipulation, in which changes the user does to the interface update the computer system state without any intermediary steps for the user [72] is a commonly used principle, while information drill-down and hyperlinks, zooming, highlighting [30], brushing [54], and the aforementioned focusing and linking [18] are well-known low-level interaction techniques. A taxonomy of such techniques and principles is presented by Yi et al. [88]. Though the sets of interaction idioms for visual analytics and information visualization have considerable overlap, Thomas and Cook request greater research focus on interaction idioms in the context of visual analytics. Notably, they recommend the development of a "science of interaction" as a means to understand the steps the analyst takes to achieve insight [78]. Their sentiment is similar to that of Yi et al. [88], who further develop the case for a science of interaction in support of visual analytics, and provide preliminary developments towards such a science, by categorizing interaction techniques based on user intent instead of user tasks or interaction operations.

In addition to the general techniques of interaction in information visualization, there have been attempts to create new interaction principles and idioms to deal with specific challenges posed by visual analytics. A notable example is semantic interaction, developed by Endert et al [36]. In semantic interaction, interactions in a visualization automatically change parameters for the automatic models generating the visualization, which notably breaks the previously discussed direct manipulation principle.

Production, Presentation and Dissemination & Moving Research into Practice

The final chapters of *Illuminating the Path* concern themselves with moving visual analytics from a conceptual and theoretical field into a practical, widely adopted one. Today, the success of visual analytics as a practical and useful field across both research and industry is evident, exemplified by the earlier mentioned conferences on visual analytics and the plethora of commercial visual analytics tools that have sprung out from both academia and industry (a comprehensive, if somewhat outdated overview can be found in a 2012 paper from Zhang et al. [90]). Systems such as Tableau [77] cover many of Thomas and Cook's recommendations, for example by letting the user select their own visual encodings and having an integrated environment for exploration and presentation of their findings [78].

Though multiple industrial-strength visual analytics solutions have emerged, there is still room for applied research in the field, for example by validating the effectiveness of solutions in specific domains. Thomas and Cook call upon researchers to validate their works as a fundamental challenge of moving research into practice [78], and approaches to validation of data, visualization, and the reasoning process have been developed for this purpose. Validation heuristics have been a part of visualization ever since Tufte's Data-to-Ink Ratio and Lie Factor [80], though approaches often have been somewhat informal. Modern heuristics are more structured, and build on work in human-computer interaction, perception and cognition, exemplified by Ware's *Information Visualization: Perception for Design* [84]. Parallel with the rise of visual analytics, wider-reaching frameworks and methodologies of visualization validation have emerged, including Tamara Munzner's *nested model of visualization design and validation* [63], and Zuk et al.'s evaluation heuristics [93], which seek to either integrate multiple previous methods of validation, extend validation to a larger part of the visualization development process, or both.

An overview of the state of validation practices can be found in *A Systematic Review on the Practice of Evaluating Visualization*. In the paper, Isenberg et al. [51] map previous visualization research to different categories of evaluation methods. They find that there has been too little focus on user experience, user performance evaluations, reports of work practices, and discussions of how new visualizations help in data analysis and reasoning, though there has been a gradual increase of such evaluations over the years. The authors also provide a set of considerations to take into account when evaluating visualization systems.

2.2 Visual Analytics for Fishery Management

One of the earliest and most well-established application domains of visual analytics is spatial and spatiotemporal data analysis, often called geovisual analytics [4]. Data of this type has become increasingly common, as exemplified by the collection of VMS, ERS and AIS data from fishing vessels. Visual analytics is well suited for the task, as it can provide more general analysis solutions for a broader range of analysts than what fully automatic solutions can [3, 5].

The earliest well-known academic exploration of geovisual analytics is *Geovisual Analytics for Spatial Decision Support: Setting the Research Agenda* by Andrienko et al. [3], published in 2007. The paper argues for understanding and taming the complexity of spatiotemporal data by visualization and interaction methods, including the use of multiple views and space-time cubes, scalability issues by path clustering, and automatic location extraction. Another early theme is the applicability of geovisual analytics to satisfy the needs of various user groups [5], and geovisual analytics for movement data analysis [6].

As elsewhere in visual analytics, domain-specific research and design studies have been common in geovisual analytics, with application domains ranging from ecology [87], to sports [52], to taxi movement [49]. Maritime vessel movement is also represented. For example, the Ph. D. thesis of Scheepens [70] presents research into geovisual analytics for maritime situational awareness, such as analysis of vessel traffic flows and glyph design for uncertainty in vessel data.

Previous works also present solutions for identification of patterns in vessel trajectory data. For instance, Enguehard et al. [38] present a system to uncover distinct and interesting patterns in fishing vessel trajectories [38]. Their approach uses the fractal dimension and velocity of trajectories to filter interesting regions in vessel movement in time periods up to a month. The approach is generic across different types of patterns, and can be used for other applications than merely identification of catch operations. However, due to how the model generalizes, it does not take into account domain-specific data useful to catch operation analysis, such as the specific movement patterns associated with different types of fishing operations and earlier catch reports. Another approach, based on the classification of so-called motifs found in the vessels' movement patterns, is presented by Li et al. [59], though their approach does not include any visualization techniques.

A commonality among existing systems for the classification of vessel movement is that they are focused on the vessel itself, not individual catch operations, and thus do not take into account data specific to the catch operations. When performing catch operation analysis, this makes evaluation at the level of individual operations difficult. Therefore, the approach of VA-TRaC may be advantageous for the specific domain of catch operation analysis, as it includes available data on catch operations and vessel metadata in the analysis, and focuses the visualization on operations instead of vessels.

2.2.1 Characteristics of Spatiotemporal Data

As noted, geovisual analytics is underpinned by spatial and spatiotemporal data. An abundance of research has been done on spatial data, exemplified by the burgeoning field of Geographical Information Systems (GIS), and though the temporal dimension often has taken a back seat [5], recent work in visualization has focused on the temporal aspect, such as Aigner et al.'s *Visualization of Time-Oriented Data* [1]. These resources provide an understanding of attributes and challenges specific to space-time. The following sections describe the important attributes of spatiotemporal data in the context of geovisual analytics, based on the terminology of Tamara Munzner's "what-why-how" framework [64].

Data Attributes

Space and time are both commonly represented as ordered, quantitative data. However, though data sets of recorded positions and times most often are quantitative, in daily speech non-quantifiable measures of space and time are often used. For instance, one cannot properly find the difference between one and two moments. In analysis scenarios for future planning, time may also be represented as branching instead of ordered [5].

The ordering of space and time varies depending on context. Space is often considered to be diverging around a point of origin, for instance in the case of latitude and longitude, a common measure of space. Space can also be cyclic: The earth is round, and by moving along the surface you will at some point end up where you started. This causes issues for instance when creating maps, and multiple schemes have been devised to project maps of the earth onto a two-dimensional surface [29]. The latitude and longitude coordinate system is also cyclic in that it specifies a point as a pair of degrees where longitude can wrap around a full circle. In the same manner as space, time can have different orderings, and it is common both to consider time as sequential and moving forward, or as cycles such as days or years [1]. Furthermore, both space and time can be represented as either absolute or relative values. Aigner et al. [1] note that time can be represented both as spans and instants, where instants are a specific moment in time and spans are a time duration that doesn't necessarily have a specific start- and endpoint. Similarly, space can be represented as both distance and locations, but due to the multidimensional nature of space, areas and volumes are also possible representations.

An important distinction between space-time and many other types of data is that both space and time are intuitively viewed at many different granularities [5]. The default scale of spatiotemporal data varies depending on context, and can greatly impact how the data is read. Moreover, users of such data often want to move between different granularities in the same data set. The dynamicity, prevalence of different granularities and domain dependence of the scales of space and time lead to specific challenges that are not immediately obvious and must be considered when visualizing such data [1]. Another distinction is the fact that both space and time are subject to spatial dependence, meaning that data in closer proximity tend to be more correlated [5]. Such dependencies are highly context-dependent, as what constitutes proximate data items varies between domains, and may change depending on various real-world occurrences [5]. For example, proximity between fishing vessel trajectories is dependent on both temporally changing weather conditions and whether the trajectory is close to land or not.

Another distinction, though a difficulty for many types of data, is that dealing with uncertainty is especially difficult when working with spatiotemporal data, due to the challenges caused by moving between granularities [1, 5]. As an example, the period from 2005 to 2009 is four years, but the number of days in the period is indeterminate, as there is no way of knowing if the measurement started and ended on the 1st of January, the 31st of December, or any other date.

2.3 Visualization Validation

The previously mentioned *nested model of visualization validation* [63] provided the base of the validation methodology employed during the development of VA-TRaC. The oft-cited paper presents a model based on the nested model of visualization design (see Figure 2.7), where each level of the model contains threats to the validity of the effectiveness claims of the visualization. Each threat is accompanied by one or more validation approaches as mitigation. The major strength of such a model is that it allows for intellectual separation of the design space of visualizations, making it easier to consider and validate each step individually [63].

The model is separated into immediate and downstream validations. Immediate validation can occur independent of other validations, while downstream validation is dependent on the validation that has been performed in previous levels. According to Munzner, immediate validation cannot offer anything more than partial evidence of success, and one must therefore employ downstream validation as well [64]. However, it is easier to perform immediate validation steps, and they may give early insights into potential issues the system faces, increasing the chance of project success. Studies in the field of software engineering have shown that up-front evaluation and analysis have a higher success rate [22], and Munzner mentions various projects that have employed immediate validation methods successfully [63].

A note on terminology: Munzner describes validation as an extension of evaluation that "is required for every level and extends beyond user studies and ethnographic observation" [63], though this is not a universal definition. For instance, Sedlmair et

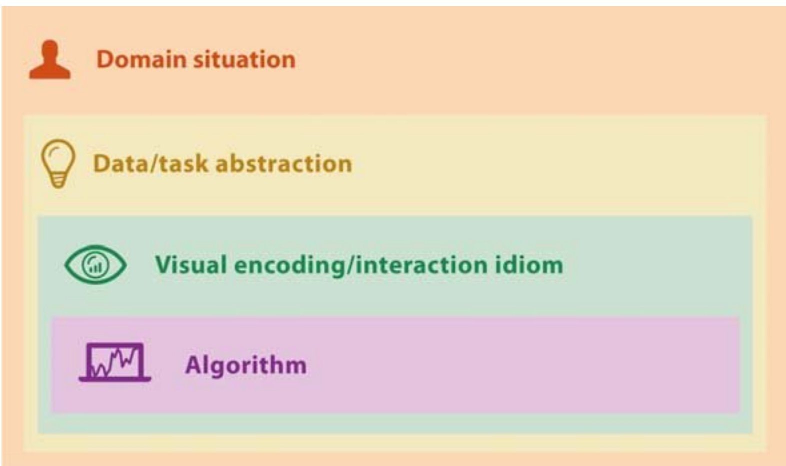

FIGURE 2.7: Munzner's nested model of visualization design [64]

al. [71] refer to "evaluation" as both "studies of already developed visualization tools as well as attempts at evaluating work practices informing the design process of a visualization tool", thus covering the same areas as Munzner's "validation". To clear up any confusion, the term validation is from this point on used in Munzner's sense, as a superset of evaluation. "Evaluation" will only be used in cases that discuss users of VA-TRaC and their process of verifying the results of fishing operation analysis.

2.3.1 Additional Validation Approaches

Although the nested model of visualization validation provides recommendations for validation approaches, the framework is general in nature, and more domain-specific recommendations may be helpful for work in their respective domains. For instance, a more domain-specific approach to validation is found in *Information Visualization Evaluation in Large Companies: Challenges, Experiences and Recommendations*, in which Sedlmair et al. [71] present challenges faced when working with applied visualization research in large companies. The paper discusses two systems developed during the author's stay at car manufacturer BMW, and provides descriptions of their evaluation methodologies as well as recommendations for other researchers working in similar domains. The Directorate of Fisheries, while not the size of the company discussed in the paper, is still a large organization filled with domain experts. The validation methodologies and recommendations of Sedlmair et al. were therefore taken into account for the validation of VA-TRaC.

The set of recommended validation approaches presented with the nested model is non-exhaustive, and how to carry out different validation approaches is intentionally left as an exercise to the reader [63]. Besides, the approaches that are presented are done so briefly, without much discussion of advantages and disadvantages. Such a discussion can instead be found in *Patterns for Visualization Evaluation* by Elmqvist and Yi [33]. The paper presents a taxonomy of patterns of downstream validation approaches, somewhat akin to software design patterns found in software engineering, that have been categorized based on the problem domain the pattern applies to, as well as advantages and disadvantages.

Visual analytics solutions present an additional set of challenges for validation, due to the interdisciplinary nature and relative youth of the field. Validation of visual analytics systems is an open research challenge, with the visualization community relying on mostly qualitative methods such as evaluation heuristics, and the automatic data analysis community mostly focusing on quantitative methods such as statistical validation [9]. Though recent work has been done to understand ways one can unify validation in the two fields [9], the domain remains relatively underexplored, leading researchers to base their validation approaches on one of the two fields.

The validation approach for VA-TRaC therefore comes from the visualization community, and is, as noted, based on Munzner's nested validation model. The nested model is supplemented by the framing of used methods in terms of the patterns of visualization evaluation found by Elmqvist and Yi [33], to provide additional context to the pros and cons of each method and to further increase comparability with other design studies that may not be based on the nested model.

Chapter 3

Domain Background

Chapter 1 introduced the Directorate of Fisheries and the work done there to ensure legal vessel activities. This chapter expands on the introduction by describing the regulations for fishing vessels, the types of employees that do work related to vessel recognition at the Directorate of Fisheries, and the end goals of this work. The descriptions are based on conversations with employees from the statistics and Control Departments at the Directorate of Fisheries' central office, as well as my own experiences working at the Directorate as an operator in the Fisheries Monitoring Centre, a department focused on monitoring of the Norwegian fishing vessel fleet.

3.1 Regulations

There are four areas of regulations that determine whether a vessel's fishing operations are illegal or not. As noted in the introduction, one of these is the duty to report ERS and VMS messages correctly and on time. In addition, the vessel must have the correct license and quota for the declared species of an operation, it must have acceptable quantities of bycatch, and it must stay within legal geographical regions.

The Directorate assigns fishing licenses to the registered Norwegian fishing vessels. The license determines where and with what equipment the vessel can fish, in what period the vessel can fish, as well as what quotas the vessel is eligible for. The quotas determine how much and what species can be fished. The vessel must conform to the limitations placed by its licenses and quotas, and as they are determined anew every year, infractions may lead the vessel to receive reduced quotas or even no license the following year.

Though the quotas specify which species the vessel is allowed to fish, it is in many cases impossible to only catch that single species. Bycatch, fish the vessel has caught that is not included in the quota, cannot exceed some percentage of the total catch amount. If the amount of bycatch is too high, it indicates that the vessel either is using an illegal equipment type for the quota, or that there have been changes in the species in the area, for example because schools of fish spawn are swimming past it. If bycatch limits are exceeded due to changes in species, the Directorate may temporarily close the area, making further fishing operations there illegal.

Additionally, licenses determine where the vessel can fish. For instance, a vessel may have a license to fish in the waters between Norway and Great Britain, but not in the northern parts of the North Sea. If the vessel then performs fishing operations north of 62 degrees, the operation will be considered illegal.

Today, these regulations are only applicable for vessels with a length over 15 meters and smaller vessels have their own set of regulations and reporting duties. The problem presented in the introduction stems from the fact that the regulations are being extended to cover smaller and smaller vessels, eventually covering all Norwegian commercial fishing vessels.

3.2 The Inspector

The inspectors are employees that control both Norwegian and foreign fishing vessels and landing facilities to verify that they have stayed within the bounds of regulations. Inspectors work directly with the vessels and facilities, and either visit the fishing vessels as they enter a port, use hired ships and submarines to visit the vessels at sea, or in some cases join a fishing vessel for an extended period.

There are three categories of regulations inspectors verify: reporting errors, illegal fishing activities, and access to specific fisheries.

Controlling reporting errors consists of verifying that all required reports have been delivered and that the values correspond to real-world values. The inspectors will often perform this at a landing facility, where they go through all reported messages from the vessel's last fishing trip. For vessels over 15 meters, this includes checking that all required ERS and VMS reports have been sent and accepted by the Directorate. They also verify that the values of the reports are correct. For instance, they will check that the quantities of different species on board correspond to the ones reported and that the ship arrived at port approximately when the vessel's master has specified. In cases where erroneous or missing reporting is found, they can check a log of contact that has occurred between the vessel and the Directorate, to see if the vessel has notified the Directorate of the problems. They will also verify that a vessel's quotas have not been exceeded.

The inspectors using hired vessels often look for illegal fishing activities. In closed areas, vessels are either not allowed to fish for specific species or using specific equipment. Inspectors will check the vessels for correct equipment types. Inspectors will also verify that the vessel's fishing operations are in accordance with their license. They also inspect quantities of bycatch, both to verify that they do not exceed the amount allowed and that the size of fish caught is sufficient. If they find that this is not the case, they can recommend closing an area [42].

Some fisheries have special regulations that extend what is normally required. For instance, fisheries of unusual species, such as fisheries for bluefin tuna, require additional reporting of quantities on board and sales locations. Such fisheries require an inspector to be on board each vessel at all times. The inspector will in these cases live on board the vessel and verify that the regulations are followed.

Inspectors may work with vessel operation recognition to find vessels that should be inspected. Due to their diverse work locations, inspectors may need to perform such analysis both while on board vessels and while at land. The process is mostly manual, and the data for each vessel's reporting is validated by going through each report. There are tools that assist in the process by displaying maps of vessel movement (an example is BarentsWatch [7]), but to my knowledge, there are no tools that automate or partially automate this process and that is available across different work environments.

3.3 The Control Analyst

The control authorities at the Directorate of Fisheries is split in two. The regional offices are the operative branch of resource control. The Control Department, housed centrally, provides professional support for the operative units in the regions. The control authorities' goal is compliance to regulations, which for the Control Department means handling regulations, creating tools that help the operative control workflow, and refining procedures and guidelines for effective regulation of fisheries. Procedures include the use of inspections, but there is also work in other directions: dialog with actors in the industry, helping develop an understanding of regulations, and otherwise making it easier for actors to comply with regulations.

One of the main tools the Control Department provides for the operative units are risk assessments of actors in the fishing industry, gathered into the National Strategic Risk Assessment. The assessment for 2019 includes main risks such as lack of or erroneous reporting, breaches of vessels' duty to land all catch, and the negative impacts of fisheries on ecosystems and ocean floor habitats [45]. Assessments are performed together with the coast guard and the different fish sales organizations and help operative units in the regions prioritize what goals and targets should be followed. They are based on various sources, including analysis of vessel operations.

The Control Department is in charge of international collaboration and works on intelligence sharing between nations. It also receives and processes complaints where there have been disagreements about the assessments made by the operatives. Finally, the department works with other institutions such as the Food Safety Authority, the Norwegian Customs, the Tax Administration and the Norwegian Metrology Service to provide compliance across the entire industry.

3.4 The Statistician

The Statistics Department also resides in the central offices and performs data management for data produced by the Directorate. The work includes facilitating easy data access, data documentation, and data quality verification. While not performing analysis of vessels themselves, the statisticians curate the data and generate metadata about the vessels, and therefore have a vested interest in the results of such analyses. A notable example of work done by the Statistics Department is the categorization of vessels into *license groups*, which group vessels with similar licenses into a set of ten categories.

3.5 The Analysis Group

A newly created, interdisciplinary group named the analysis group consists of members from both the Control and Statistics Departments. Its primary mandate is assisting the Control Department in their work, by developing models and tools that can assist the risk assessment process, both centrally and by the operative units such as inspectors. As of today, their main focus is exploring and developing various indicators of misbehavior for use by inspectors, based on automatic analysis. They are at the present the group most aligned with the work on VA-TRaC, and were therefore the primary contributors during development.

3.6 Fisheries Management in Academia

Perhaps unsurprisingly, fisheries management as a scientific field is dominated by articles about fish biology and fishing stocks [34], and works concerning data, automatic analysis, and visualization in the field are largely related to that. However, the integration of spatial data into models is recognized as an important component of fisheries analysis [20]. For instance, in the early work *Incorporating the spatial component of fisheries data into stock assessment models* Anthony J. Booth presents an example of such an integration [12].

More recently, work that focuses on the use of vessel movement data for stock assessment has become more commonplace. For instance, Duchame-Barth and Ahrens of the University of Florida have worked on the subject, such as on uncertainty in automatic classification of fishing vessel movement for the purpose of determining caught species [31], and using automatic data analysis of VMS data for evaluating fish abundance in the Gulf of Mexico [32]. Chang and Yuan discuss the combination of observations and VMS data, bringing in another data source into the mix [20]. In a more foundational work, Bonham et al. describe the standard AIS movement data format and discuss how one can use such data analyzing vessel operations in ports [11].

As stated, the main focus of the field has been on evaluation and analysis of specific or general fish stocks, and methods for identification of vessel catch operations with the goal of regulation have not been much considered in the literature, with some notable exceptions such as those presented in Chapter 2.

Chapter 4

The VA-TRaC System

This chapter presents an analysis of the algorithms, data and tasks used by VA-TRaC to augment the workflows presented in Chapter 3, as well as a rationale for the choices of visual encodings and interaction idioms. Tamara Munzner's "what-why-how" framework [64] is used for the data and task analysis.

4.1 Chapter Terminology

As the analysis is framed in terms of the "what-why-how" framework, the terminology used in this chapter is based on Munzner's terminology in *Visualization Analysis & Design* as much as possible. In cases where conflicting terminology is used, the relevant source is referenced.

- **Item**: A single, uniquely identifiable data point. For instance, in a table, the item is a single row.
- **Attribute**: A property of the dataset common across several data points. In a table, this is the column.
- **Value**: The combination of a single item and attribute. In a table, a synonym for value is pair.
- **Static and dynamic datasets**: Static datasets are available all at once when starting an analysis session. In contrast, dynamic datasets may not be fully available, and new items or attributes may emerge during execution.

4.2 Data Abstraction

Since the foundation of any visualization is data, the first step towards effective visualization is data analysis. This is the "what" of the "what-why-how" model, as illustrated in Figure 4.1. The data analysis in this phase includes the visualization designer identifying and abstracting the properties of the data, in order to find the

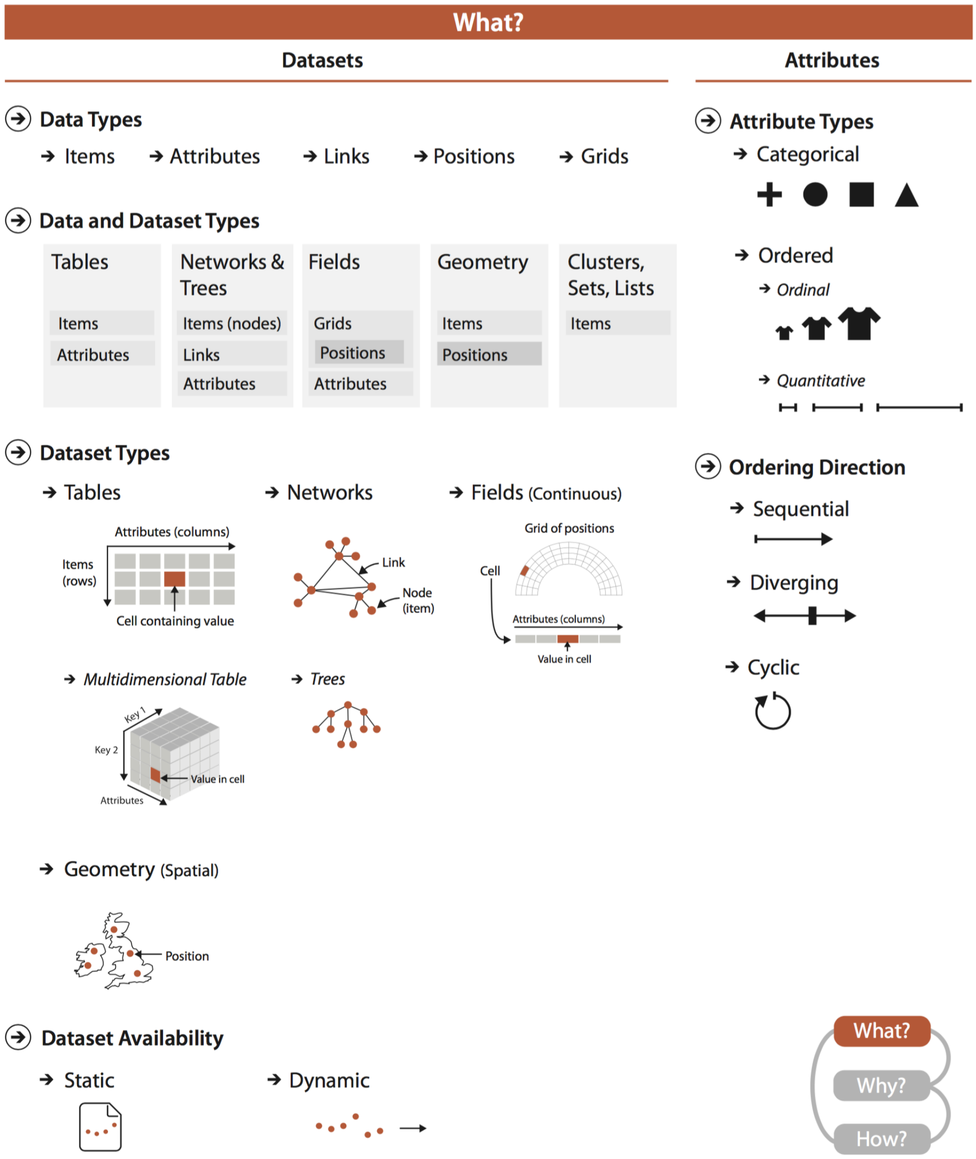

FIGURE 4.1: Munzner's categorization of datatypes and attributes [64]

attributes that should be mapped to visual encodings. This is followed by assessment of whether transforming the data into another structure may allow new ways to visualize the data that may be more suited for the particular domain the designer is working in. The following sections describe the data and data transformations used by VA-TRaC.

4.2.1 Raw data

The two main datasets used by VA-TRaC are sets of electronic catch report (ERS) and automatic identification system (AIS) data. ERS reports contain a multitude of attributes, such as the name of the vessel's master and the vessel's call sign, gear type, reported species caught and quantities on board, which port the vessel came from and is going to, the report ID, the coordinated universal time (UTC) the report was sent, and the vessel's latitude and longitude. The included attributes for each report are dependent on the type of report the item contains. For instance, DCA reports, sent to describe the vessel's daily catch activity (hence the name) may or may not include catch operations. If they do, they have a set of mandatory attributes such as start and stop of the catch operations that have occurred during the day. Similarly, port arrival reports must include estimated time of arrival and a code identifying the port [43]. Some attributes' meanings change depending on the report type. For example, in port departure reports the mandatory attributes latitude and longitude indicate the position the vessel intends to commence fishing, while in DCA reports they indicate the position where each catch operation was started [43].

Using the categorization of Figure 4.1, ERS data is sequential geometry. It is sequential in the sense that reports from a single vessel have strictly increasing IDs and contain timestamps, and geometry due to the fact that positions are available for most report types. In VA-TRaC, the ERS dataset is static, and contains the reports of all Norwegian vessels over 15 meters during the last five years. However, ERS data is usually dynamic, as new vessel reports are sent continuously. Due to the need for analyzing vessel operations as they occur, the ERS dataset used in VA-TRaC was treated as though it was dynamic for the purposes of visual abstraction. This allows for extending the system to work with dynamic ERS messages if the system is to be used outside of a prototype environment.

Since VA-TRaC is focused on analyzing fishing operations, the ERS dataset was filtered to only include daily catch activity (DCA) reports. Each report in the filtered dataset consists of a pair of items, one denoting the start of an operation and one denoting that the operation is complete. Filtering the dataset to only include reports containing fields for start-of-catch and end-of-catch removes the potential for misunderstanding when working with fields with multiple meanings, such as latitude and longitude. As mentioned in the introduction (Chapter 1), vessels oblige to send DCA reports at least once every 24 hours while they are active in a fishing trip (meaning between a port departure and a port arrival report), which results in the filtered dataset containing approximately 1.8 million items, derived from DCA reports collected between 2014 and 2019.

In addition to the ERS dataset, VA-TRaC uses position data in the form of AIS messages. In contrast to ERS data, AIS data contains a more limited set of fields, with no semantic differences depending on context. The AIS data available to VA-TRaC does not contain all the data in the AIS data specification, but the following attributes were available:

- MMSI, an identifier unique to the vessel
- Navigation status, including whether the vessel is anchored or not
- The vessel's speed in knots
- Longitude and latitude
- Course
- Timestamp

The AIS datasets available for use in VA-TRaC consist of all AIS messages sent by Norwegian fishing vessels during 2017 and the first seven months of 2018. The reports are sent in five minute intervals as long as the AIS transitter on board the vessel is activated. The datasets contains approximately 24 million items in total.

In terms of Munzner's model (Figure 4.1), the AIS data is identical to the ERS dataset, as they both can be abstracted as sequental geometry. Like the ERS data, AIS data is commonly dynamic. However, the AIS dataset in use by VA-TRaC is static, since the system the Directorate is planning to use for dynamically requesting AIS data is unfinished. Even so, like the ERS data, it was considered as dynamic during development of VA-TRaC to ease an eventual transition into using dynamic data sources.

In addition to the ERS and AIS datasets, two other datasets were used. One consists of metadata for the vessels, including the vessel's engine size, length, and license. The other contains a hierarchical structure of equipment categories, from here on referred to as the "gear hierarchy". Like the ERS and AIS datasets, they are stored on disk as flat tables.

4.2.2 Data Features

When only the spatiotemporal attributes are considered, the Munzner model shows that the ERS and AIS datasets are very similar in terms of data types, ordering, and availability, and diverge only in the granularity of time when just the spatiotemporal attributes are considered. Since both the MMSI of the AIS dataset and the call sign of the ERS dataset uniquely identify a vessel, a mapping between the two was created, allowing the use of AIS data to get the locations between each DCA start and stop for a vessel. By interpolating between these locations, one can derive the trajectories of individual fishing operations, each trajectory thus becoming a continuous, non-branching movement path.

Fishing operation trajectories have various constraints that determine their location and spatiotemporal developments. The following specific features of fishing vessel operations were identified through several discussions with members of the analysis group:

- Certain operations cause distinct spatial patterns, often based on the gear type used for the operation. An example is line fishery, which results in straight trajectories with sharp, approximately 90 degree turns.
- As vessels must slow down to perform fishing operations, the development of a vessel's speed during a fishing operation causes distinct patterns when fishery occurs.
- The habitats of various fish species are often located in specific regions, which causes vessels' fishing operation trajectories to be located in these regions. The canonical example is the spring fishery for cod in Lofoten. Likewise, quotas and licenses may contain limitations to where the vessel is allowed to fish, also limiting the locations where the vessel's fishing operations are performed.
- Fishery for certain species is limited in duration, commonly to a period of a few months. This is due to the movement of fish throughout the year, and for many species the additional constraint of regulations stating that fishery is only legal within a certain timespan. For example, Greenland halibut fishery is only legal in the spring and fall [44].
- Catch operations' durations are influenced by the gear type used, and real catch operations have a duration in tens of minutes to hours, not seconds.
- Some fisheries occur more frequently at certain times during the day. For instance, shrimp fishery often occurs early in the morning, as the vessels want to deliver fresh shrimp to customers during the day.
- The vessel's spatiotemporal patterns during a fishing vessel operation are to some degree dependent on the vessel's size, engine, hull, and other features of the vessel's construction.
- Fishery requires at least some form of movement from the vessel, even if that movement is only the vessel drifting due to sea currents and wind.

Vessels with similar characteristics, such as the aforementioned vessel size and gear type, often have similar licenses and quotas. Licenses can therefore be used to group the vessels based on the characteristics of their catch operations. Due to these characteristics being similar across similar licenses, the statistics department has created license groups, allowing vessels to be classified based on other vessels with similar attributes.

4.2.3 Indicator Algorithms

The mentioned discussions with the analysis group were held with the goal of identifying potential indicators based on the data analysis of fishing operation trajectories derived from the AIS and ERS datasets. They included an hour-long one-on-one session with a member of the analysis group resulting in the choice of using patterns in vessel speed and spatial movement as two indicators, and a so called "water cooler talk" where another member suggested using position and date of earlier DCA reports to verify whether a DCA report is plausibly correct. A presentation of these potential indicators with the entire group resulted in their choice for VA-TRaC, and further uncovered the need for different algorithm parameters for different vessels, due to differences in expected patterns for different classes of vessels. Additionally, the indicators were chosen to be based on analysis of 24 hours of data, as that matches the period in which a vessel must report a DCA. This conveniently reduces the quantity of data items needed for the analysis to an amount that can be represented in consumer computer memory without needing to be aggregated and filtered further. Even so, since fishing necessitates movement, vessels that did not move in the 24-hour period were exempt from the indicators, as they could not have been fishing.

The indicators consist of three algorithms classifying whether fishing operations have occurred for a vessel during the last 24 hours. The algorithms are as follows: A threshold-based pattern recognition algorithm classifies whether a vessel's movement matches its movement during previously received DCA reports, and returns both the gear and the duration of a similar catch operation. A k-Nearest Neighbors model based on previous fishing operations provides the gear type, and a classifier based on the vessel's speed finds durations where the vessel was moving at a speed that may indicate it was fishing. The operations found by the indicators resemble a pair of start and stop DCA reports, and can therefore be combined with the AIS data to derive the trajectory of the operation. The indicator-based operations contain three attributes: The duration of the operation, the movement path of the vessel during the operation, and the gear used. Since vessels with similar licenses often have similar characteristics for their catch operations, the parameters of the algorithms are modifiable for each license group.

4.2.4 Speed Classification

The speed classifier identifies catch operations based on the AIS dataset, in an approach similar to the classification of fishing operations done by Enguehard et al.

[38]. An operation is derived by identifying all reports from the vessel's AIS data whose speed falls below a set threshold. For each group of sequential reports, the timestamp of the first and last element forms the duration of a catch operation. The speed classifier thus returns a set of durations, each indicating a potential fishing operation.

4.2.5 kNN Gear Classification

The k-nearest neighbors classifier uses DCA messages reported over the last 5 years to classify the a vessel's gear type. Each message $M = \{m_0, m_1, ..., m_n\}$ is registered with a x and y coordinate and a gear type. For each vessel group, the threshold T is the maximal distance for which a point is considered to be within distance of the vessel v. Distance is calculated as Euclidean distance:

$$dist(x) = \sqrt{(v_x - m_x)^2 + (v_y - m_y)^2}$$

$$gear(v) = max(\forall |g| \in G), G = \sum_{m_{gear} \in M} dist(v, m) < T$$

The kNN algorithm was run early in the development phase to look for good initial settings for VA-TRaC, which resulted in two domain-specific constraints being added. Since different gear types for areas change over the course of a year, only the DCA messages that occurred in the current month are used. In addition to increasing the relative number of correct classifications, the number of data points are reduced to 1/12th of the initial size, so the algorithm runs significantly faster. The other constraint is an upper limit to the number of elements within distance the algorithm considers. As illustrated in Figure 4.2, it was found during cross-validation of the initial tests that any number of elements above 10 reduced accuracy, so the value is set to 10.

The kNN classifier returns the most commonly occurring gear type of DCA reports within the classifier's radius. If no operations fall within the radius, the classifier returns nothing.

4.2.6 1$ Pattern Recognition

The final classification algorithm is used for recognition of movement patterns. The algorithm used for this is a modification of the *1$ Recognizer* [85], created to recognize finger gestures on smartphone screens. Since single-finger gesture patterns on a phone screen have the same spatial characteristics as the movement trajectories

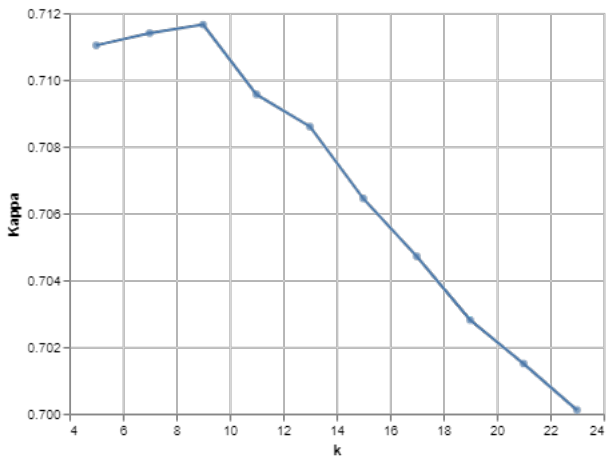

FIGURE 4.2: The X-axis denotes the number of elements that the k-NN algorithm considers (k), the Y-axis the Kappa coefficient.

made by a single vessel (a single start and endpoint, no branching paths, and continuity), the algorithm was considered to be a good fit. According to the original authors, the 1$ recognizer's accuracy is on par with other template-based pattern recognition algorithms for paths, while the algorithm is easy to understand and computationally cheap.

The modified 1$ recognizer works by comparing a set of path templates representing the catch operations of various gears with the trajectory of each vessel. By convolving the template along the vessel's trajectory, one can compare the operation to the vessel's movement throughout its trajectory. At each point of comparison, a piece of the trajectory is selected based on the duration of the catch operation in the template. The piece is then normalized, scaled, and rotated to match the template as closely as possible, after which a score is calculated based on the distance from each point in the trajectory piece to the corresponding point in the template. This is done for all templates, and each comparison score that passed a threshold is marked as a potential catch operation. The method is similar to the one proposed by Li et al. [59], whose algorithm also compares trajectory similarity with pre-set templates (which they call motifs), though their approach is based on general movement patterns and therefore focuses on the generation of templates. This was deemed unnecessary for VA-TRaC, as templates could be extracted using the method described in the Data Features subsection earlier in this chapter. Using the method, an initial set of templates were derived from all DCA reports, and assigned to the gear reported in the DCA. Some examples are shown in Figure 4.3. As can be seen, operations using gear types such as purse seine (1), line (2) and bottom trawl (3) often leave distinct patterns, though there are cases where operation trajectories are harder to classify based on shape alone (4, 5).

The pattern recognition classifier returns a set containing both the duration and gear type of all recognitions that score above the threshold.

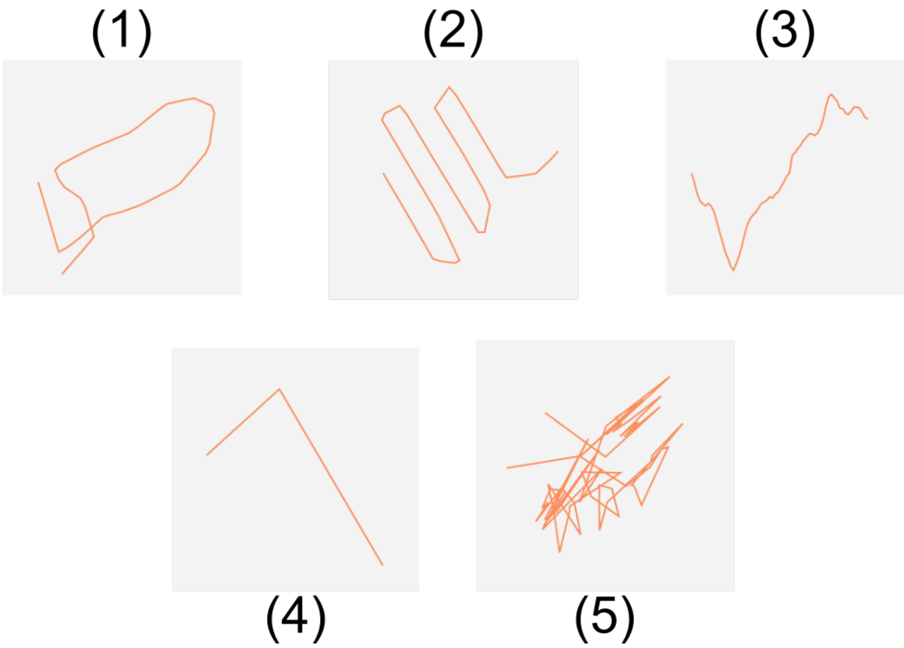

FIGURE 4.3: Initial templates derived from DCA reports. (1) is a purse seine operation. (2) is a line operation. (3) is a bottom trawl operation. (4) is a bottom trawl operation with a short duration. (5) is a trawl operation where the trawl type is undefined.

Verifying Indicators

In order to verify the performance of the indicators, sensitivity and fall-out is used. The calculations are based on the temporal similarity of DCA reports and the potential catch operations from the indicators. Sensitivity for a single vessel is calculated as $\frac{M_{overlap}}{M_{DCA}}$, where $M_{overlap}$ is the total number of minutes from the indicated catch operations that overlap DCAs, and M_{DCA} is the total number of minutes in the DCA reports. Fall-out is calculated as $\frac{M_{DCA}}{M_{all}}$, where M_{all} is the total number of minutes in the indicated catch operations. These values are averaged across all vessels to get a general sense of the indicators' performance. High sensitivity is preferable, as is low fall-out, though no fall-out is a sign of the indicators overfitting the data and therefore not finding any potential operations outside the ones reported.

Since the k-Nearest Neighbors indicator does not return a duration, its results are not used for the calculation of sensitivity and fall-out. Even so, calculating indicator performance based on the durations of indicated operations was deemed a good choice, as it is simple to gain an intuitive understanding of how the results change due to parameter changes, even for inspectors not trained in statistical analysis.

4.3 Task Abstraction

As noted, Munzer's framework (first presented by Brehmer and Munzer [15]), was used for task abstraction. An overview of the framework of task abstractions is

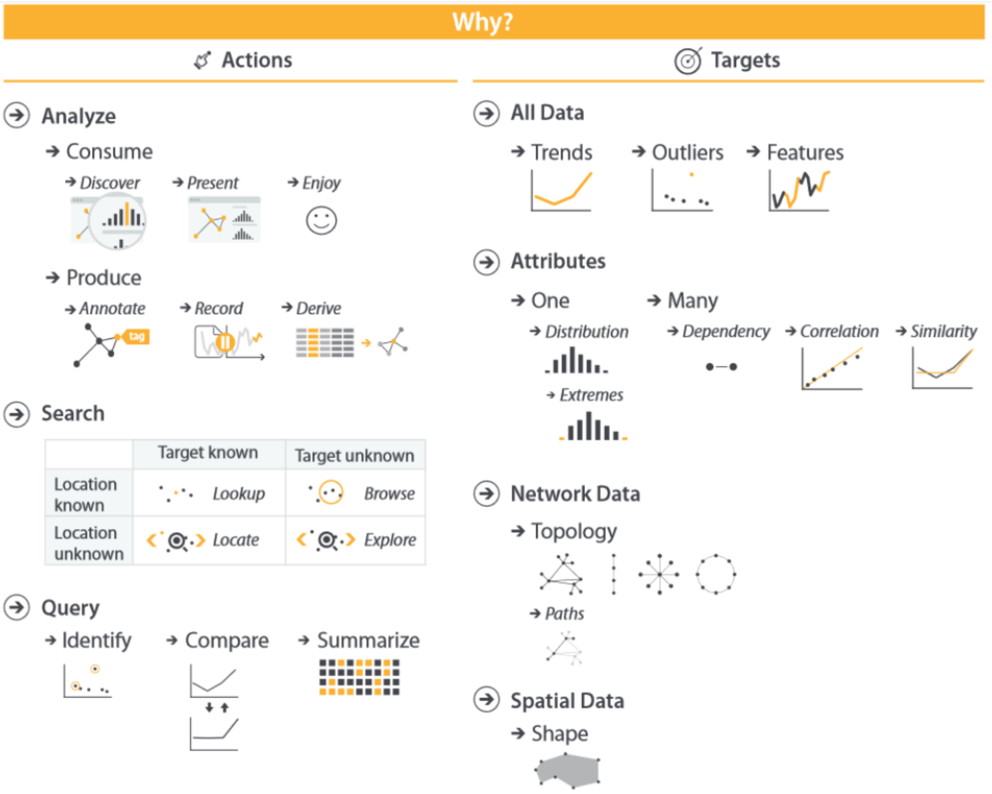

FIGURE 4.4: Brehmer and Munzner's multi-level framework of task abstractions [64]

found in Figure 4.4. In order to use the framework successfully, one must find user needs that can be abstracted in the manner presented by the framework. As common in software engineering [62], a traditional approach to requirement analysis based on functional and non-functional requirements [24] was chosen for VA-TRaC, with the functional requirements serving as the base for task abstraction.

The functional requirements were collected during the earliest days of the development process, through a group interview with eight employees of the Directorate, coming from the analysis group, the statistics department, the control department, and the IT department. Unfortunately, the inspectors were not represented as they are not located in the same city as the Directorate headquarters, though the inclusion of members of the analysis group helped alleviate this somewhat, as their work includes collaboration with the inspectors. The interview was conducted in an open manner, where the question of how the participants envisioned either real or imagined workflows for identifying illegal catch operations. As the question was discussed, talking points repeated by multiple participants were noted, and became the basis for the functional requirements. The requirements were gradually refined in conjunction with the identification of the indicator algorithms, through the individual interviews and informal discussions with members of the analysis group discussed during the section on indicators earlier in this chapter. The following sections describe each task abstraction. The first two tasks are based on existing workflows for catch operation analysis, while the last three are based on requirements to VA-TRaC's workflow.

4.3.1 T1: Identify Illegal Catch Operations

As discussed in Chapter 3, a catch operation is considered illegal if it isn't reported in a DCA report, if the DCA report lacks some of its required fields, if the operation takes place using non-licensed equipment or in a non-licensed location, or if it occurs outside the allowed period of fishery for a species. DCA reports that lack required fields are not accepted, and are therefore easy to identify using the existing systems at the Directorate. For operations that are contained in a DCA report, the equipment, location, and period of an operation can be analyzed by deriving the trajectory of individual operation in the manner discussed in the Data Features section of this chapter, and the content of the report can be considered as more or less plausible based on the features of fishing operation trajectories described in the same section. Inspectors will currently analyze vessels in this manner to look for infractions. However, for cases when the DCA report is missing, the inspector must explore the reports and trajectories of vessels looking for the features specific to fishing operation trajectories discussed in the Data Features section. If one is found, the inspector must evaluate that the operation plausibly took place by using tacit knowledge such as whether the location of the operation is popular for fishing. Only then do they know which vessels to target for inspections. This is a time-consuming and tedious process, and inspectors are therefore often reduced to either patrolling the ocean or responding to tips from the public to find illegal catch operations from vessels that do not report.

Therefore, the core functional requirement of VA-TRaC was the automatic identification of illegal catch operations where no DCA reports have been sent. Given the central position of this requirement, the indicator algorithms were chosen based on their ability to solve this task. By using the indicators to automatically derive the trajectories of fishing operations, analysis group analysts and inspectors can analyze operations without a corresponding DCA report in a similar manner to those that do have a report. In terms of the terminology from Figure 4.4, the system needs to *derive* what it thinks is a fishing operation from the data by applying the previously discussed indicator algorithms to the ERS and AIS datasets, as illustrated in Table 4.1. The result is, for each vessel, a set of trajectories that have been identified as a potential catch operation. The data derivation step is done without visual feedback to the user, as is common in many visualization applications generally [64], and visual analytics in particular [56]. Once the derived data is ready, the user wants to *explore* the operations as described above, to *identify* the *features* (location, spatiotemporal patterns, gear) of each potential operation and compare it to tacit and explicit knowledge of the vessel, to assess whether the operation is likely to actually have happened.

	Actions
Analyze	Derive
Search	N/A
Query	N/A

	Actions
Analyze	Discover
Search	Explore
Query	Identify

Targets
Features

TABLE 4.1: Task 1: Identify Illegal Catch Operations.

4.3.2 T2: Explore Vessels' DCA Reports

To find vessels that lack reporting, inspectors explore the reporting state of vessels to find those that lack DCA reports for a given day, and then assess whether the movement of those vessels correspond to a catch operation. Vessels that have sent a DCA report in a given day have, at first glance, performed their required reporting, and the inspector can use this as a filter to find vessels that seem to be fishing without sending reports. In the initial group interview discussions, the first thing that was brought up was the possibility of seeing which vessels have reported catch operations and which have not, to be able to filter vessels in this manner. It was also mentioned that an overview of the vessels in a specific geographical region would be useful as well, for instance if an inspector is in the area and wants to find the reporting state of all vessels in their vicinity. Since these two types of overview have different characteristics when abstracted into tasks, they have been separated into two subtasks.

T2.1: Overview Based on Reporting State

The task abstraction of getting an overview based on reporting state is shown in Table 4.2. Here, the user wants to *discover* which vessels have reported catch operations and which have not, in order to filter the set of vessels for which to further analyze catch operations. The user is in this case looking for a known target, a vessel with absent DCA reports, though the reporting state of each vessel is unknown, and must be *located*. The goal is to reduce the number of vessels that are considered for further analysis by creating a *summary* of the *outlying* vessels where no reports have been received and where the indicators have found a potential catch operation. The task may also involve filtering away vessels that have not reported, in cases where the user wants to assess the legality of catch operations performed by vessels that have sent DCA reports (see Chapter 6 for an example). In this case the outliers are vessels that have reported instead of those that have not.

	Actions
Analyze	Discover
Search	Locate
Query	Summarize

Targets
Outliers

TABLE 4.2: Task 2.1: Overview Based on Reporting State.

T2.2: Overview Based on Location

The task abstraction of seeing vessels in a specific region (see Table 4.3) is somewhat different. In this case, the user still wants to *discover* vessels' reporting states, but they know the geographical location of the vessel and are looking for vessels in the region that have not reported. They are therefore *looking up* specific vessels that they consider suspicious based on the fact that the vessels are not reporting. The goal of the task is to *identify* a specific vessel. The targets are once again outliers, vessels that haven't reported catch operations.

	Actions
Analyze	Discover
Search	Lookup
Query	Identify

Targets
Outliers

TABLE 4.3: Task 2.2: Overview Based on Location.

4.3.3 T3: Evaluate Indicated Operations

When a possible catch operation is found in **T1**, the user must evaluate the likelihood of it being a real catch operation or not. In practice, this means comparing the results of the indicators to knowledge kept internally and externally by the user (such as knowledge of good fishing areas and active regulations), and will therefore necessarily involve varying queries and searches depending on the user's knowledge. For every potential catch operation found by the indicators, all the features from the discussion of data features earlier in this chapter can be evaluated by the user in an attempt to verify whether the operation actually took place.

1. Distinct spatial patterns in a catch operation trajectory are evaluated by *identifying* the *shape* of the catch operation as a plausible shape for the gear type reported by the indicator. Such plausible shapes are found as tacit knowledge among several members of the analysis group and elsewhere in the Directorate. Equivalently, temporal patterns are *identified* by the *shape* of the changes in the vessel's speed during the course of the operation.

2. The likelihood that fishery has occurred in the location of the catch operation is evaluated by *comparing* the catch operation to earlier reports *located* close to

the vessel, or by *comparing* the position to it's geographical surroundings. The likelihood that fishery has occurred based on date is assessed by *comparing* the date with knowledge of regulations active at that date.

3. Assessing whether the duration of the operation is plausible is done by *looking up* and *identifying* its duration. Similarly, the likelihood of an operation being performed at a certain timepoint can be assessed by *looking up* and *comparing* the start and end timepoint of the duration to the real catch operations of *similar* vessels or real reports from the same vessel.

As seen in Table 4.4, the plausible methods of comparison differ in both search and query actions, as well as targets. However, they have several commonalities. When evaluating an indication, the vessel in question has already been identified and located, and the user is interested in validating it's catch operations. Thus the target for search actions is known, leaving *locate* and *lookup* as the possible search actions. The user either wants to *identify* or *compare* the operation's attributes (note: comparison with the user's knowledge is the goal even in identification tasks). The targets are either *shape* when identifying spatial and temporal patterns or the vessel's geographical region, *features* (duration, date and timepoint of the operation) when evaluating other temporal aspects of the operation, or *similarity* when comparing spatiotemporal patterns of operations to those of other vessels.

	Actions
Analyze	Discover
Search	Locate, Lookup
Query	Compare

Targets
Similarity, Shape, Features

TABLE 4.4: Task 3: Evaluate Indicated Operations.

4.3.4 T4: Evaluate Modifications to Indicator Parameters

Indicators are, as the name suggests, merely an indication of what can potentially be a catch operation. Further, as discussed earlier in this chapter, the spatiotemporal trajectory of a correct indication is dependent on various external attributes of the vessels, such as license and size. The indicators take this dependency into consideration by being applied with different parameters depending on the license group. Due to changes (for instance gear changes, number of vessels, fish movement) in active vessels for different fisheries, licenses and quotas, the parameters for a specific group may need to be modified periodically. The user therefore needs to see whether such a modification increases the performance of the indicator for the current vessels, which can be done by using the indicator evaluation scheme presented earlier in this chapter.

As seen in Table 4.5, the first step when evaluating changes to indicator parameters is performing aggregation over vessels and calculating the effect on model accuracy caused by a parameter change, using the comparison of DCA reports and algorithms. As with deriving the indicated catch operations, this is a step that needs to happen before the user can get meaningful feedback, and is therefore done without visual representation. Thus, the task is a chain starting with a *derive* task. After deriving new information about model accuracy, the user wants to *compare* the result with previous results to see if the *trends* in model accuracy are positive.

	Actions
Analyze	Derive
Search	N/A
Query	N/A

	Actions
Analyze	Discover
Search	Lookup (Browse)
Query	Compare

Targets
Trends

TABLE 4.5: Task 4: Evaluate Modifications of Indicator Parameters

4.3.5 T5: Filter Vessels

When identifying illegal catch operations, the user is rarely interested in going through potential catch operations for all vessels in the same analysis session. As an example, an inspector may want to go straight to a vessel that previously has been caught performing illegal operations, or they may want to only look at vessels using specific gears that fish in highly regulated fisheries.

As seen in Table 4.6, the goal of all such filtering operations is to *derive* a smaller *summary* of the original dataset that includes the vessels considered as relevant for further analysis. Such a summary can include both a group of vessels with a common attribute, or a single vessel. The goal when filtering differs depending on the filtering operation: When filtering by geographical region the goal is to *browse* unknown vessels in a specific region. In the other cases, the goal is *locating* vessels of unknown location based on a known attribute. The *features* the user is interested in is highly-context dependent:

- Geographical region
- Whether the vessel is anchored
- Vessel license group
- Vessel name or call sign
- Reported or indicated gear types
- Reporting status

A commonality between all filtering operations is that they are primarily based on interaction and aren't directly visualized. The already existing visual representations are used by the user to decide upon the filtering operation to perform, and the operation shrinks the dataset, which is displayed in the same manner as before. To be useful to the analyst, the filtering task must therefore be followed by other tasks involving visual feedback, and the chosen interaction idioms should therefore facilitate this. Additionally, each filtering operation must be reversible, so the analyst can start another series of analysis tasks on another subset of the vessels.

	Actions
Analyze	Derive
Search	Locate, Browse
Query	Summarize

Targets
Features

TABLE 4.6: Task 5: Filter Vessels

4.4 Visual Abstractions and Interaction Idioms

The final step of the "what-why-how" is the mapping of the abstract data and tasks to effective visual encodings and interaction idioms. The choices for these encodings and idioms were informed by a variety of sources. Obviously, Munzners *Visualization Analysis & Design* was used, as it provides a comprehensive guide for which encodings are effective for specific combinations of data and tasks. Additionally, more informal heuristics, principles, and recommendations from the visualization literature was used. Zuk's *Heuristics for Information Visualization Evaluation* [93] provided multiple general heuristics, including Shneiderman's famous mantra ("Overview first, filter and zoom, then details on demand"). The seminal works of visual analytics provided more specialized principles and recommendations suited to the data quantities used for this project [57, 78, 5]. The following paragraphs describe the general guidelines that were followed when choosing visual abstractions.

The ERS and especially AIS datasets are reasonably large (with 1,8 million and 24 million items respectively, and both consisting of 6 or more attributes), meaning effective visualization of individual elements in the complete datasets is very difficult. Furthermore, the user is interested in evaluating the catch operations of individual vessels, so techniques for aggregation of visual elements is not a good solution. Instead, the visual analytics approach of preprocessing and filtering as a means to reduce data quantity before anything is displayed is a valid approach [3, 5]. A good way to go about this is to reduce the number of loaded vessels by filtering on certain features, as recommended by Andrienko et al. [5]. For VA-TRaC, this is done through the mentioned domain-dependent filters, including only using AIS messages for a single day and ERS reports for a single month, and removing vessels that have not moved during the 24-hour period of the AIS data. Since VA-TRaC is focused on illegal catch operations, removing vessels that have reported DCAs was also considered, but since **T2.1**, overview based on reporting state, requires both vessels with and without reports, this idea was discarded.

No single visualization can effectively display several datasets with a large number of attributes and attribute types simultaneously. Some, such as parallel coordinates [50] and generalized scatterplot matrices [35], provide an opportunity to display multivariate datasets within a single visualization but are unable to deal effectively with spatiotemporal data. Another tempting approach is the use of a space-time cube, which is presented as a useful visualization in early works on geovisual analytics [6]. The cube represents time and space as the X, Y and Z axis of a cube, integrating time as an equally important dimension to space. In a domain where time is often presented as of secondary importance this is a valuable idea [5]. However, space-time cubes suffer from the same problems as any other 3D visualization: It increases the potential for occlusion, and may warp the perception of the user [64]. For task **T3**, in which the user compares personal knowledge of catch operations with visual representations, on-screen comparisons are impossible. It is therefore particularly important that the perceived attributes of the visualized catch operation correspond to the underlying data and how the users envision the attributes. Since the space-time cube is a less common visualization than regular maps and timelines, it is less likely that inspectors have an intuitive understanding of how to read such a cube. They will therefore have more difficulty comparing the displayed trajectories with their personal domain knowledge.

Since no single visualization can effectively encode all the attributes of the data and the user must be able to intuitively compare the data to their own knowledge, a setup using multiple views was applied instead. When employing multiple views, the designer gets the opportunity to select visual encodings that are effective at showing only parts of the data, and therefore displaying multiple datasets with a wide variety of attributes becomes easier. Multiple views have become a common technique to address issues with visualizing diverse data and is shown to be more effective than individual visualizations in such cases [27]. The chosen views were all conservative in the sense that they are commonly used visual representations of data. Users therefore have a higher chance of having encountered the representations before, which increases their willingness to understand what the representation shows and how it compares to what they already know [81]. Finally, since the system might be used with dynamic data in the future, none of the chosen views feature visual encodings that are degraded by the dynamic addition and removal of new items.

As mentioned, one of the preprocessing steps includes a filter that restricts AIS data to 24 hours, bounded by the period from 00:00 to 23:59 in a single day. By restricting the user from changing temporal granularities and showing all durations as linear and sequential, one avoids the issues that can come from changing temporal granularities, discussed in Chapter 2. The choice is further justified by the fact that no tasks are based on seeing trends in the spatiotemporal data outside of the 24-hour resolution.

4.4.1 Visual Abstraction in Practice

The "five-design sheet" methodology [68] was used to create the initial design of VA-TRaC. In the methodology, the designer creates three very different potential designs as paper mockups which are reviewed by domain experts. The reviews are then used as the basis of a final design for the system.

The design of sheet 1 is based on the user selecting various linked views to display at once, where no single view is more prominent than the others. This approach is flexible, as the user can specify which views are most suited for the task at hand, but requires the user to know which combinations of views are effective for solving a particular task. Sheet 2 contains a more conservative design, where the visualizations are static and a map view is prominently featured. This is more similar to the existing mapping systems in use at the Directorate than sheet 1. Both sheets draw inspiration from various analysis systems for analysis of movement data of both fishing vessels and other sources [11, 38, 87]. Common between the systems is the use of a prominent map view overlaid with spatial trajectories, augmented by charts and tables showing detail views of related data. Other commonalities include a timeline view and widgets for tweaking model parameters.

The design sheets were shown to two domain experts at the Directorate as they were created. After discussing the first two designs, it became clear that one of the design sheets (seen in Figure 4.6) was preferred. The design from sheet 2 was therefore chosen as the base for initial design of VA-TRaC.

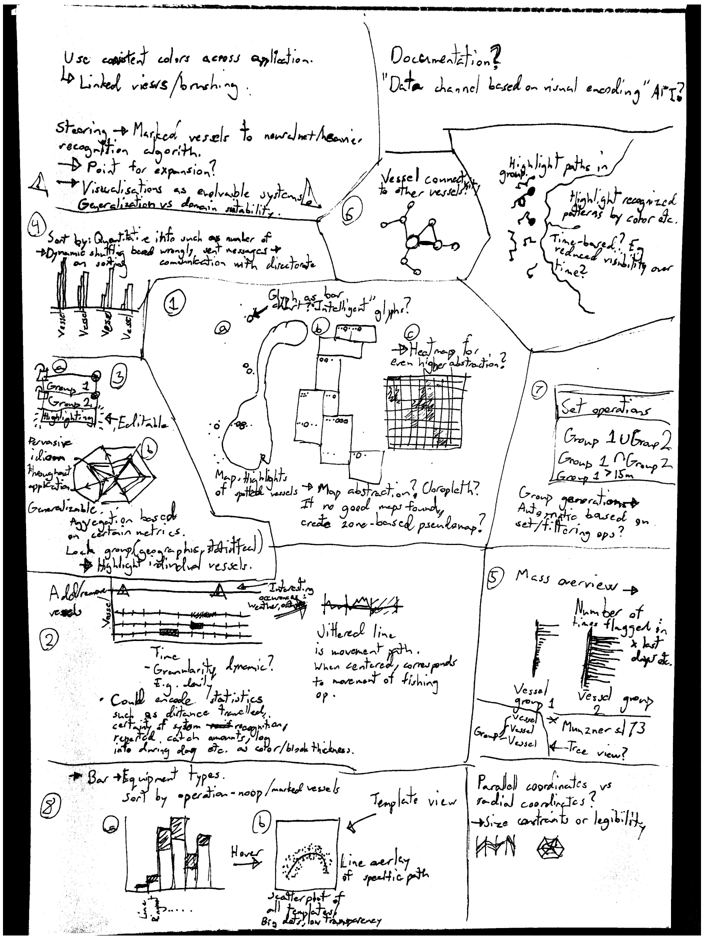

FIGURE 4.5: Design sheet 1 from the five-design sheet process

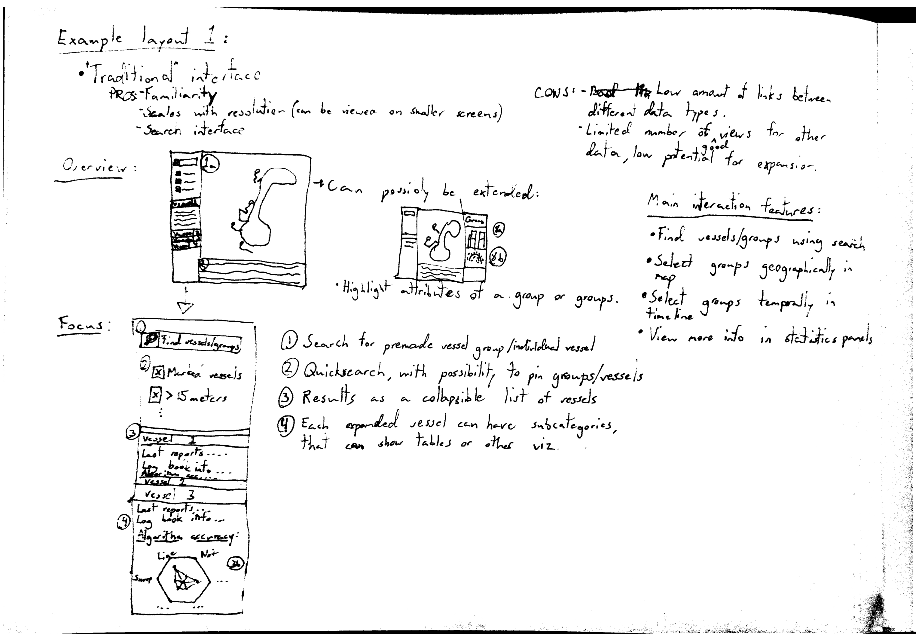

FIGURE 4.6: Design sheet 2 from the five-design sheet process

4.5 Application Details

This section presents an overview of VA-TRaC, then presents an analysis of each visual component by discussing how it addresses both the abstract tasks and more concrete workflows given the available data. Additionally, interaction idioms and the component's relation to other components are discussed. As the names and call signs of vessels are confidential, their occurrences in figures have been blurred.

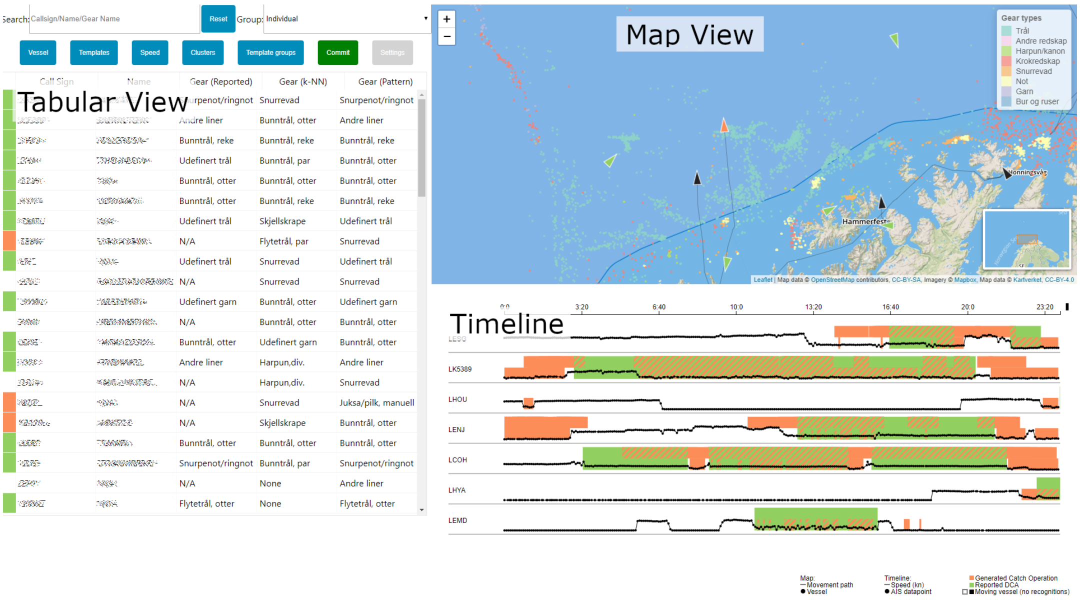

FIGURE 4.7: Overview of VA-TRaC, with the Map View, Tabular View and Timeline annotated.

4.5.1 Overview

Upon starting the application, the user is presented with the "main view", shown in Figure 4.7. It consists of three visualizations, including a map, a timeline and a tabular view. Additionally, there is a toolbar that lets the user filter the dataset in various ways (**T5**), and access additional functionality. The additional functionality is used to validate the individual indicators, as well as to modify their parameters. An additional visualization in the main view, used for seeing trends of indicator performance, can be enabled for situations where the user is performing **T4**, and therefore modifying indicator parameters to increase performance.

Figure 4.8 illustrates the typical workflow of VA-TRaC in terms of the task abstractions. The workflow starts after the initial automatic analysis wherein indicators are run and stationary vessels are filtered out, with the user identifying vessels that may have been fishing without sending a DCA reports in the tabular view, possibly filtering the vessels by name, gear or license group. The vessels are then selected sequentially to analyze whether the indicators are correct for that vessel. The currently selected vessel is located in the map, which shows the trajectory the vessel has taken, and on the timeline, which shows the duration of catch operations and the speed of the vessel over time. The time and position where the vessel is categorized as fishing are highlighted using a consistent colorscheme in both the map and the timeline, and superimposed on the displays of the vessel's trajectory. Seeing the actual events of the vessel together with the results of the automatic analysis, the user can form an opinion on whether the vessel may have avoided reporting a real catch operation, or whether the indicators were mistaken in their assessment. More in-depth presentations of typical workflows are found in Chapter 6.

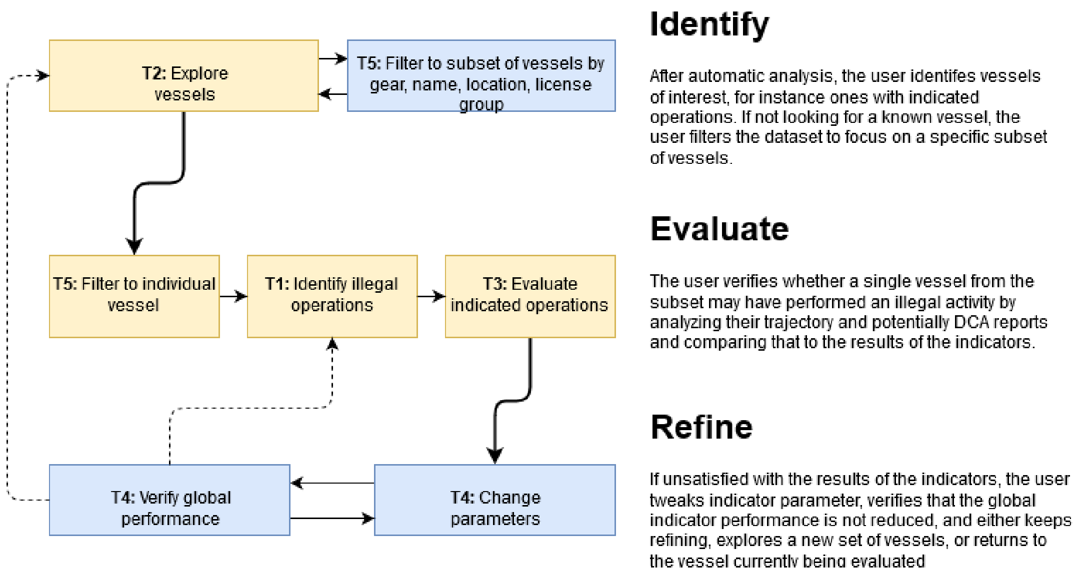

FIGURE 4.8: The user's workflow based on abstract tasks. Yellow tasks are mandatory, blue are optional. The flow of user operations is denoted by arrows.

4.5.2 Color Choices

A recurring theme throughout the main views is the use of a categorical, red-green-black colormap to represent the reporting state of a vessel. Red encodes vessels and operations that have been identified by the indicators, green vessels that have reported a DCA and operations from DCAs, and black/white encodes vessels and trajectories where neither of the above are true. Encoding the reporting state in the color hue channel provides a large degree of popout, the phenomenon where individual items stand out among others immediately [84]. This makes discerning important vessels and operations, such as the ones mentioned in the previously described workflow, a quicker process. Additionally, color hue can be combined with multiple other channels, notably the ones encoding spatial regions or positions, which are the most effective channels for both ordered and categorical attributes [64]. Finally, the color hue is in itself an effective channel for categorical attributes. The mapping is found in the right-most part of Figure 4.9, which shows the legend visible to users of VA-TRaC.

An important note is that though the colormap is categorical, the colors are ordered by importance. This comes into play when more than one attribute of the vessel is present at the same time. For instance, the vessel may have reported a DCA, but is still categorized as fishing by the indicators. In this case, the green, representing vessels that have reported DCAs, overrides both the other colors unless it is possible to display both. In general, knowing whether a vessel has sent one or more DCA reports is more important than whether the vessel falls within the other two categories encoded by the colormap, since vessels that have reported DCAs are within the law. Likewise, both the green and the red overrides the black, which are vessels that are merely moving. In cases where the user may want to understand where the categories of reported DCAs and indicated fishing operations overlap, a hatched pattern of red and green is used to show that both have occurred.

Green and red can be hard to separate by people with dichromatic color blindness [84]. Even so, a red-green colormap was selected, based on Ware's recommendation of the unique hues yellow, green, blue and red for encodings of small symbols [84]. Of these colors, the green and red are distinct from the other colors of the map view, notably the blue of the water. Additionally, the color scheme of green as something positive and red as an alert is very common in Western culture, and is seen for instance in traffic lights, so the choice makes it easier for users to identify the hues' meanings. Finally, yellow and green are also difficult to separate for many dichromatically color blind (those that suffer from deuteranopia) [28], and lack the contextual significance of red and green, while both green/yellow and red/yellow color combinations have less contrast than red/green and are thus harder to distinguish [84].

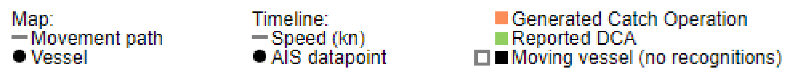

FIGURE 4.9: The legend. To the right, the green and red colormap.

4.5.3 The Map View

The map view, seen in Figure 4.10, shows the spatial dimension of the vessels superimposed over a geographical map. Each vessel is marked by a dynamic glyph, which when zoomed out is a dot, and when zoomed in is a triangle pointing in the direction the vessel is currently going (seen in Figure 4.11). As shown by (1) in Figure 4.10, the glyphs are color-coded according to the colormap in Figure 4.9. The glyphs use the ordering of importance in the colormap, with a white outline as a luminance contrast boundary, increasing popout [84]. The green/red colormap on the glyphs allow the user to visually filter vessels that have reported DCAs, those that have been reported by the indicators, and those who are moving without any real or perceived operation. Hovering a glyph shows a popup of the vessel's call sign ((3) in Figure 4.14), allowing the identification of vessels. Being able to visually identify the vessels' categories and finding their identity solves **T2.2** (vessel identification), where the user looks for the reporting state of vessels with known locations.

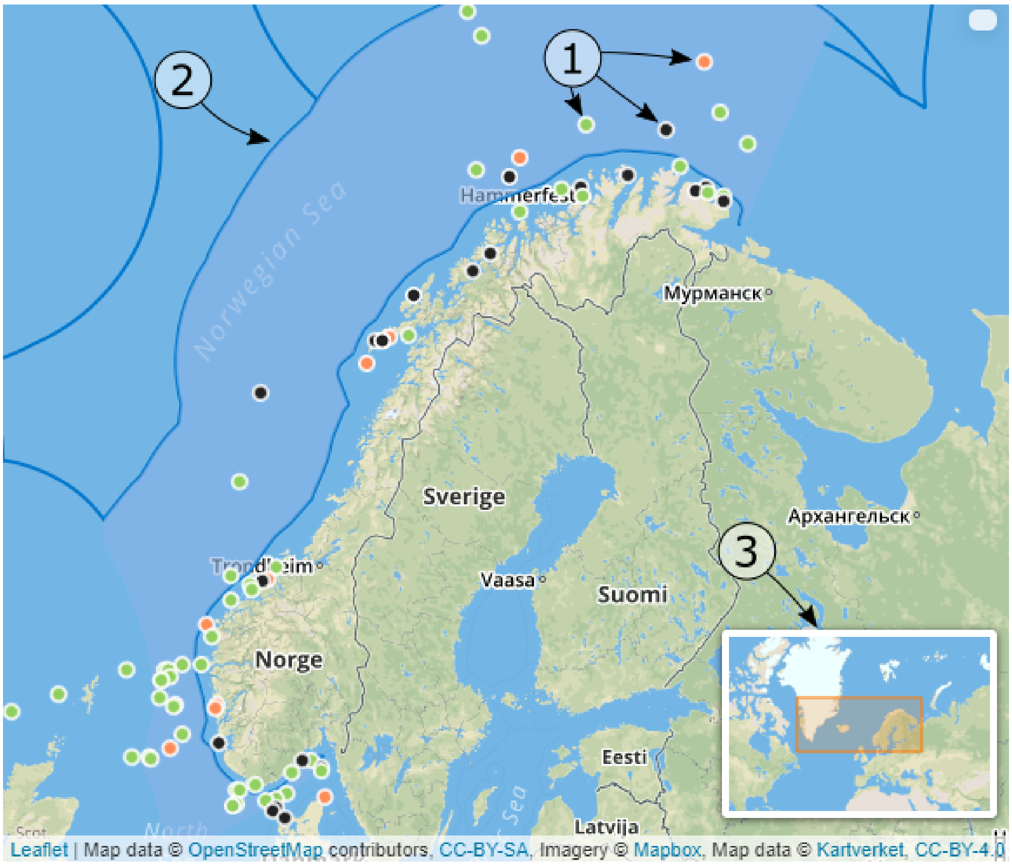

FIGURE 4.10: Zoomed out view of the map. (1) shows vessels of different reporting states. (2) shows the borders of various ocean regions. (3) shows the minimap.

As discussed in Chapter 2, a major challenge when working with spatial data is dealing with navigation between different granularities, and zooming in a map is one type of such navigation. This type of interaction occurs a lot in VA-TRaC. To alleviate occlusion and information overload, different glyphs are used depending on zoom level. This increases the user's perception of vessels, as the dot glyph is horizontally symmetric regardless of rotation and is therefore easier to perceive even at small sizes [13]. This comes at a cost of not being able to encode the direction of the vessels, but when zoomed out this is not a concern, as one is interested in getting an overview of the vessels' reporting situations based on location (**T2.2**). Additionally, the use of smaller glyphs alleviates occlusion, which would hinder the task of getting an overview through the map. As the scale increases, fewer items are visible at the same time and their positions are spaced further apart on the screen, reducing the need for glyph symmetry, so additional data in the form of the vessel's current course is encoded by switching to an elongated triangle glyph. The "Overview first,

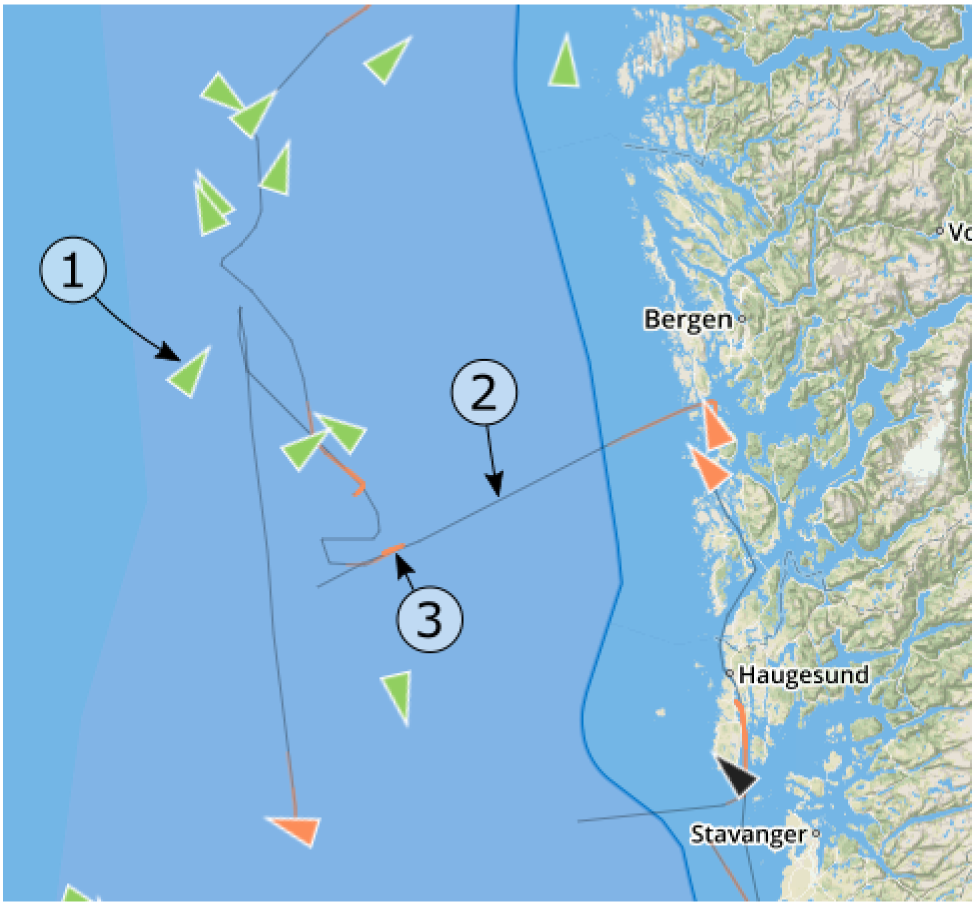

FIGURE 4.11: A zoomed-in view of the map. (1) shows the triangular glyph used for vessels at this zoom level. (2) shows a vessel trajectory. (3) shows a catch operation.

filter and zoom, then details on demand" mantra encourages similar behavior, where more attributes are shown as the number of visible items is reduced.

The map imagery comes from the company Mapbox, who provides map tiles that are created by combining data from many sources, and that in their words give precise and comprehensive maps all over the globe [61]. Since maps may lack relevant information depending on the granularity level set by the user [3], using map tiles from a provider that advertises their maps as comprehensive and precise was considered a good choice.

When zoomed in, each moving vessel's trajectory becomes visible on the map. Overlaid on the trajectories are vessels' movements during fishing operations, colored according to the color map 4.9, letting the user identify the spatial component of real and indicated catch operations. The hatched pattern shows overlaps of DCA and indicator-identified trajectories, which is useful when the user wants to evaluate the overlap of indicator classifications and reported DCAs to see how well the indicators match the real world. Other mapping solutions in use at the Directorate encode the movement path as continuous, and the DCA reports as discrete points along that path. Though the data in the DCA dataset consists of discrete datapoints, the underlying activity is continuous, and the catch operations derived from the DCA reports are therefore encoded as such.

In contrast to Enguehard et al.'s visual analytics system for fishing vessels [38], which uses higher-order spline interpolation in trajectories, VA-TRaC uses linear interpolation. Enguehard et al. use spline interpolation as it can increase accuracy of trajectory displays for data with low temporal resolution [79]. However, their approach is based on VMS, which is a vessel position reporting system with a much lower reporting rate than the AIS data VA-TRaC is based on. For Norwegian vessels, VMS reports are sent every hour, while AIS reports are sent every five minutes. The benefits to accuracy from using higher-order spline interpolation is therefore reduced substantially by using AIS instead of VMS reports, to an extent where the cost of performing higher-order spline interpolation for every trajectory was deemed to outweigh the benefits.

Akin to the display of vessel glyphs, the display of trajectories depends on the zoom level. When too many vessels are displayed at the same time, the trajectories stop being useful as they occlude each other. Likewise, when evaluating an indicated catch operation in the map, one is interested in the shape of the path, which becomes difficult to discern when zoomed far out [18]. The choice of displaying trajectories or not is domain-driven: The interesting parts of a vessel's trajectory in the described use case is when it is fishing, which occurs in limited geographical regions. Furthermore, the user is most interested in seeing the vessels that have not reported their daily catch. Therefore, trajectories of vessels that have reported a DCA are hidden by default and must be explicitly enabled, to not occlude the trajectories for indicated operations that are evaluated in task **T3**. As evident by comparing Figure 4.12 to Figure 4.13, occlusion from the trajectories of even two additional vessels can be quite high, and seeing the trajectories of the vessels of interest becomes more difficult.

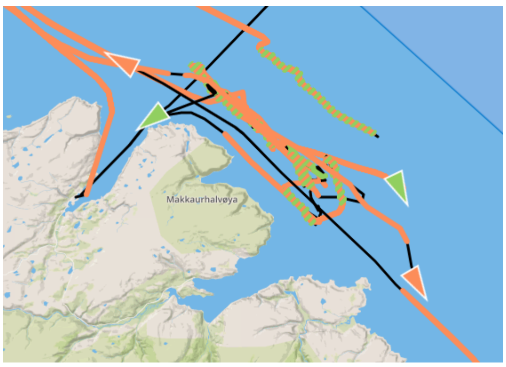

FIGURE 4.12: Displaying all vessel trajectories.

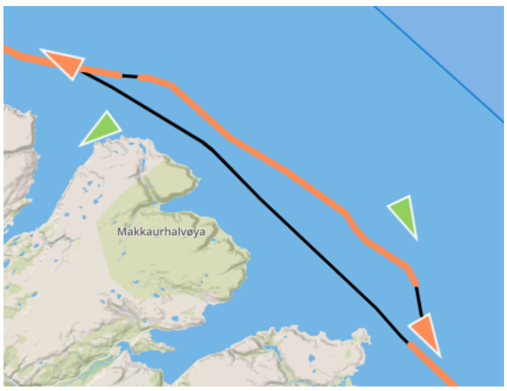

FIGURE 4.13: Hiding trajectories of vessels with DCA reports.

Figure 4.14 shows the map at a very close zoom level. At such a level, the thickness of the trajectory lines is increased to make both real and indicated fishing operations more visible. It thus becomes slightly easier to understand the movement patterns caused by indications of a specific vessel, and to solve the trajectory- and geography-based subtasks of **T3**. In Figure 4.14, one sees that (1) a the vessel's reported catch operation overlaps the operation found by the indicators, and (2) that the indicators have identified more than the reported operation as potential fishing. As illustrated in Figure 4.8, **T3** is completed after the user has filtered the dataset down to a single vessel, and the user therefore avoids occluded paths even when the line thickness increases. Since only a single vessel is rendered, the user will often enable rendering trajectories for all vessels when at this stage of analysis, to be able to analyze trajectories even when the vessel has reported a DCA.

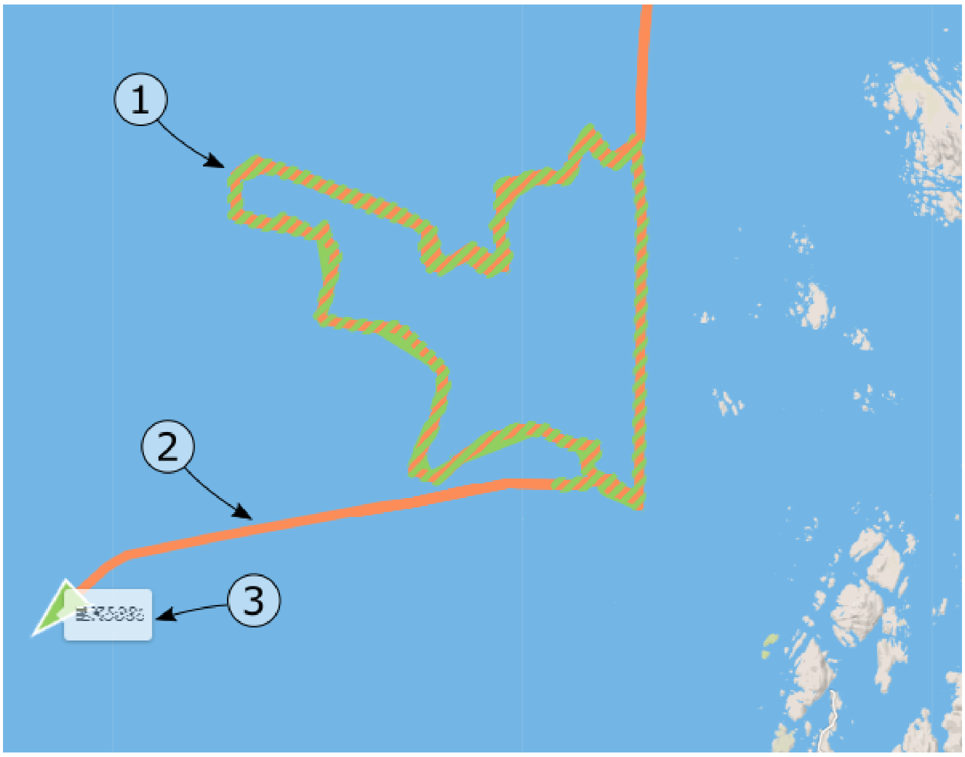

FIGURE 4.14: A closely zoomed view of a single vessel. (1) shows the overlapping DCA-reported and indicated catch operation trajectories, (2) the indicated catch operation trajectory and (3) the tooltip of the vessel's callsign (blurred to anonymize the vessel).

The map view uses publicly available map data from the Norwegian Mapping Authority to display borders of Norwegian waters (annotated as (2) in Figure 4.10. A blue line represents the border between the territorial coast waters of Norway and the ocean regions controlled by Norway. The ocean regions are overlaid with a translucent geometric shape, which is shaded slightly orange. The border view was added at the request of domain experts at the Directorate of Fisheries, as many licenses and quotas are very dependent on these borders, as discussed earlier in this chapter. For example, some vessels are not allowed to fish outside coastal waters, and many licenses disallow fishing in foreign waters.

Finally, the map view displays a minimap (annotation (3) in Figure 4.10). According to Cockburn et al. [27], *overview and detail* views such as minimaps allow users to better orient themselves in a larger view. Combining *overview and detail* with zooming, the user can navigate to multiple granularities based on preference, while still keeping the spatial orientation of the whole. Since a typical workflow in VA-TRaC sees a lot of jumps in space, keeping the user oriented is helpful. Plumlee and Ware [66] has showed that overview+detail outperforms zooming in cases where the user must expend a lot of visual memory elsewhere, which may be the case in the map view due to the number of marks and channels used to encode all the displayed attributes.

4.5.4 The Timeline

The timeline (Figure 4.15) shows the duration of every vessel's reported and indicated catch operations in the 24-hour window being analyzed. It also shows the vessel's speed over time, and the timestamps of each AIS report received during the period. It has two dimensions: The Y-axis is categorical, each category being a vessel (annotation (5) in Figure 4.15). The X-axis is quantitative, displaying the time of day. When the user wants to evaluate the temporal aspects of the indicated reports as part of **T3**, they already know what vessel and attribute they are looking for and are thus performing a lookup with the goal of comparison between different vessels or between different points in time. According to Munzner, the canonical example of an effective chart that fits such data and lookup tasks is the bar chart, which is used as the base of the visualizations [64]. In addition to using the most effective available channels, the bar chart is ubiquitous and is readily understood by the domain experts at the Directorate of Fisheries. The temporal axis in the timeline is horizontally oriented to maximize the resolution devoted to it, as identification and analysis of the temporal aspects of catch operations are more important tasks than comparisons between vessels.

Annotations (1) and (2) in Figure 4.15 show bars displaying indicated catch operations. In contrast to a regular bar chart, the timeline positions each bar according to the duration of a catch operation. The bars use the same colormap as elsewhere in the application, and like the trajectories in the map view, use the red and green hatched pattern to display periods where both algorithmic and reported catch operations have occurred, as shown by annotation (3). Also, operations found by the speed indicator and the 1$ pattern indicator are separated spatially, to allow the user to discern how the system identified the vessel as fishing. Annotation (1) shows results from speed indications, which are placed in the bottom part of the vessel's timeline. Annotation (2) shows results from 1$ pattern indication, which is placed in the top part. DCA reports cover both regions, as they are not tied to either indicator. The placement of the indicated operations is based on the gestalt principle that

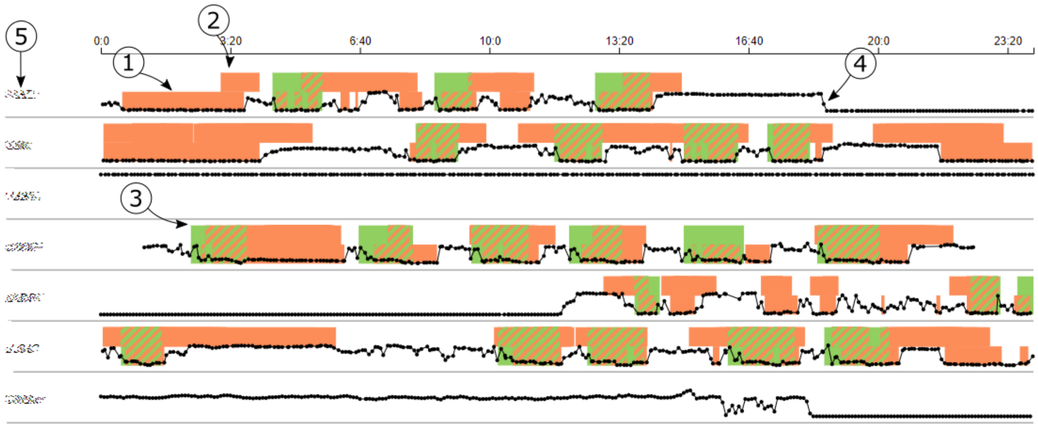

FIGURE 4.15: The timeline view. (1), (2) and (3) show bars of the various catch operations. (4) shows the line chart showing vessel speed. (5) shows the Y axis displaying vessel names (blurred to anonymize the vessels).

marks in close proximity are seen as grouped [84], making it more obvious that a catch operation can be seen as singular when it is indicated by both indicators in the same period.

Annotation 4 in Figure 4.15 shows the line chart of the vessel's speed during the day, superimposed on the bar chart. The dots encode the timepoint and speed of the vessel's individual AIS data points. Time is encoded as position along the X-axis using the same scale and labels as the catch operation bars, while speed is encoded as position along the Y-axis. The goal of the line chart is three-fold. First, reductions in speed can be caused because of fishing operations, so seeing the changes in speed during the day can be used to verify that the indicated fishing operations are within reasonable bounds. Second, it allows the user to get a deeper understanding of the vessel's entire trajectory, as it shows the rate of change in the vessel's position over time. Finally, the density of dots allows the user to understand the uncertainty of the automatic analysis, as they can see the number of data points used as inputs to the algorithms.

The Tabular View

The tabular view, shown in Figure 4.16, is a text table where each row is a vessel. The attributes shown are the vessels' reporting state, names and call signs, the gear the vessel has reported using, and the ones recognized by the k-NN and 1$ pattern indicators. The vessels' states are encoded using the same colormap as in the previous views, though in this case the precedence rules are applied. The view is alphabetically sortable on a per-column basis. This allows for flexible on-the-fly pseudo-filtering, aiding task **T5**. For instance, the user can sort by reported gear type if vessels using a specific gear is of interest, or they may sort by reporting state.

The use of a table to list the vessels was suggested by a domain expert as an intuitive way to quickly identify vessels that have been recognized by the automatic analysis (**T1**). Additionally, the table is used for **T2.1**, as the reporting state of each vessel is one attribute visible in the chart. Text tables are good at precise information lookup, and are familiar to many audiences [53]. A text table to encode the textual data was therefore found suitable for vessel identification tasks where the user wishes to locate the vessel spatially, but where the vessel's reporting state is known. Further, the analysis group members that may become users of VA-TRaC have an intuitive understanding of table views, having used Excel as part of their daily work for a long time.

The gear names presented in the view come from the gear hierarchy, briefly discussed earlier in this chapter. There are no channels for categorical data not already in use in either the map or the timeline that can encode 43 categories efficiently, so text was chosen as the most suitable encoding for the names. A new visualization for the gear categorization was considered, but the idea was ultimately rejected due to lack of screen real estate in the main view. As the table already supports text data well, it was chosen as the most suitable location to place the results of the gear recognition. Another advantage of using text as the encoding is that the attribute becomes searchable, using a standard search interface familiar to most people. The search field is located in the top left corner of the main view.

Call Sign	Name	Gear (Reported)	Gear (k-NN)	Gear (Pattern)
[illegible]	[illegible]	Andre liner	Bunntrål, otter	Andre liner
[illegible]	[illegible]	Bunntrål, reke	Bunntrål, reke	Bunntrål, reke
[illegible]	[illegible]	N/A	Snurrevad	Snurrevad
[illegible]	[illegible]	N/A	None	Snurrevad
[illegible]	[illegible]	N/A	Bunntrål, otter	Snurrevad
[illegible]	[illegible]	N/A	Bunntrål, otter	Teiner
[illegible]	[illegible]	N/A	Snurrevad	Snurpenot/ringnot
[illegible]	[illegible]	Snurrevad	Bunntrål, otter	Snurrevad
[illegible]	[illegible]	Snurpenot/ringnot	None	Snurpenot/ringnot
[illegible]	[illegible]	Snurpenot/ringnot	Bunntrål, otter	Snurpenot/ringnot
[illegible]	[illegible]	N/A	None	No match.
[illegible]	[illegible]	Snurrevad	Snurrevad	Snurrevad
[illegible]	[illegible]	Snurpenot/ringnot	Snurpenot/ringnot	Snurpenot/ringnot
[illegible]	[illegible]	Snurrevad	Snurrevad	Snurrevad
[illegible]	[illegible]	N/A	None	Snurrevad
[illegible]	[illegible]	N/A	None	Snurrevad
[illegible]	[illegible]	Snurrevad	None	Snurrevad
[illegible]	[illegible]	Snurrevad	Andre liner	Snurrevad

FIGURE 4.16: The tabular view. Vessel names and call signs have been anonymized.

4.5.5 Secondary Views

The main views are focused on the core use case of VA-TRaC: investigating and analyzing trajectories and the results of automatic classification algorithms. As time progressed and it became clear that model building also would be an important aspect of VA-TRaC, various extra views were added to give the user tools to control parameters of the indicators. Tuning of indicators is mostly envisioned to be performed by members of the analysis group, who work using desktop workstations with multiple screens. To take advantage of this, all but one of the secondary views are available as pop-up screens from the main menu. This gives the user control over view positions, which views are open at once, and the size of the views, while retaining the window handling experience that users are familiar with from working with other desktop applications. The secondary views include settings and template selection for the 1$ pattern indicator, settings for the speed indicator and the k-NN indicator, and a view for displaying additional, constructed templates available for the 1$ algorithm, as well as a parameter evaluation viewer.

The Evaluation Viewer

When modifying parameters of the indicators, the user can enable the evaluation viewer (seen in Figure 4.17), which places an additional chart in the main view. The evaluation viewer uses the output from the indicator evaluation discussed earlier in this chapter, by showing the sensitivity and fall-out as a line chart. The viewer is thus VA-TRaC 's solution to **T4**, evaluating modifications to indicator parameters. The Y-axis encodes the relative sensitivity and fall-out, the X-axis encodes user interactions. For each modification to a model parameter, sensitivity and fall-out are calculated and added to the chart, allowing the user to see the global development of sensitivity and fall-out as they tune the parameters. Vertical lines highlight the discrete nature of the X-axis and hovering one of these lines will display the precise sensitivity and fall-out at that change. To not crowd the line chart, changes that are more than 20 steps back in time are removed. The viewer is placed in the main view to highlight the fact that the sensitivity and fall-out are not tied to a single indicator.

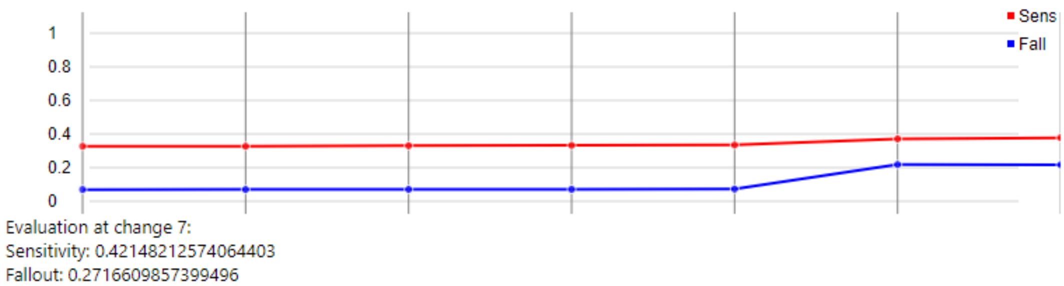

FIGURE 4.17: Sensitivity and fall-out developments as speed and pattern indicator thresholds are modified.

The displayed line helps guide the user toward parameters changes that do not degrade indicator performance for the current AIS and ERS data. Keeping historical context allows the user to evaluate the trends of parameter settings by verifying that new changes are an improvement over ones that were done sometime earlier. In the example session shown in Figure 4.17, the user has modified the thresholds of the speed classifier. Most modifications haven't caused much change to the overall indicator performance, but the second-to-last change, an increase in the maximum speed considered as a fishing operation for purse seine vessels, caused a marked increase in the model's fall-out as well as a visible increase in sensitivity. The user can compare the modification with previous modifications, and thus see that the sensitivity has not been this high earlier.

Vessel Details

The vessel details view shows metadata about the currently selected vessel, including engine size, length, and the license group of the vessel, and supplements other views based on the principle that such details are not always needed. The view is purely textual, as it only shows static values associated with a vessel. Additionally, the view contains a button to flag the vessel, which is the mechanism used for noting that the vessel may be worth further investigating.

Template Viewer

Seen in Figure 4.18, the template viewer is the main view for handling the manipulation of templates and parameters for the 1$ pattern indicator. Exemplified by annotation (1), the view shows all templates the indicator is comparing to the vessel's trajectory, and highlights the templates that have been recognized. The view also allows the user to select individual templates to build new, averaged templates, and commit these to a database. Finally, there is a small toolbar with tools for easier handling of multiple templates, a slider to tune the indicator's threshold, and a checkbox to swap between pre-calculated and constructed template sets.

The two most important uses of the view are to see which templates have been found in the vessel's trajectory and building new trajectories. Templates that have been recognized are identified by a dot glyph in a yellow hue. As with the vessel glyphs in the map, this increases popout by using a previously unused hue in a separate spatial region [84]. Hovering over recognized templates will highlight them using a yellow border, with matching colored highlighting in the timeline and the map, allowing the user to assess whether the indicated operation is plausible, in line with task **T3**. This workflow is shown in Chapter 6.

To build new patterns, the user can click on the templates he wants to include. Shown by annotation (2) in Figure 4.18, this will highlight the template by changing the hue to green. A double-click gives the template additional weight when computing the average path, and highlights the template in a blue hue. The hue choices in the view are again based on Ware's [84] recommendation of using the unique hues if possible, and the fact that hue is the most effective unused categorical channel [64]. The average template is displayed in a larger view (annotation (3) in Figure 4.18), which shows translucent versions of paths of all selected templates, with the average path highlighted in red.

Tuning the indicator's threshold changes the score a template must reach to be considered as recognized. The threshold is changed using the slider in the toolbar of the template viewer. Slider adjustments follow the principle of direct manipulation [72], and immediately recalculate indications from the 1$ pattern recognizer.

Finally, the view allows for changing template datasets. Once a sufficient number of averaged templates have been created, using the averaged datasets will avoid the overfitting of templates to the data that happens due to the pre-processed templates being based on DCA reports individual vessels have submitted for the gear type.

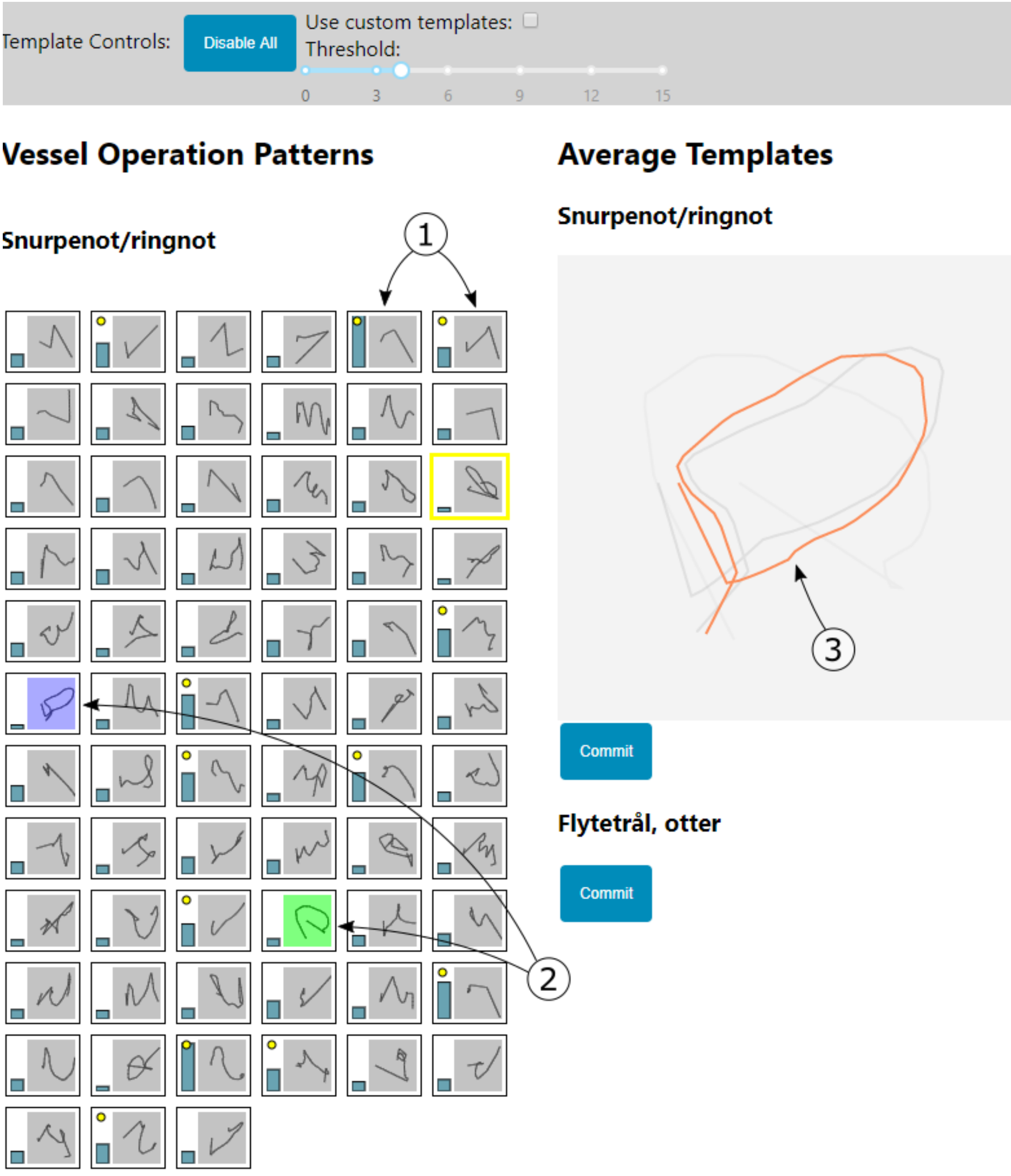

FIGURE 4.18: The template viewer. (1) shows two example templates. (2) shows templates used to build the average. (3) shows the average template.

Speed Indicator Settings

The view for tuning the speed indicator parameters, seen in Figure 4.19, consists of a table where each row is a license group. For each group, a histogram shows the speed of the group's vessels in the current dataset during fishing operations, the current maximum speed a vessel in the group may have to be considered fishing, and a slider to change the threshold for that group. The histogram bar where the currently selected vessel resides is highlighted using a hue and saturation shift. This causes popout, and lets the user quickly identify the license group the vessel is in.

The histograms allow the user to get an idea of the distribution of speed in the fishing operations for that license group, helping guide the user when setting the speed threshold for that group, and allows comparing the currently selected vessel with others in the group to assess whether it is an outlier. Since they are stacked vertically, the histograms also allow for comparison of speed distributions between various groups.

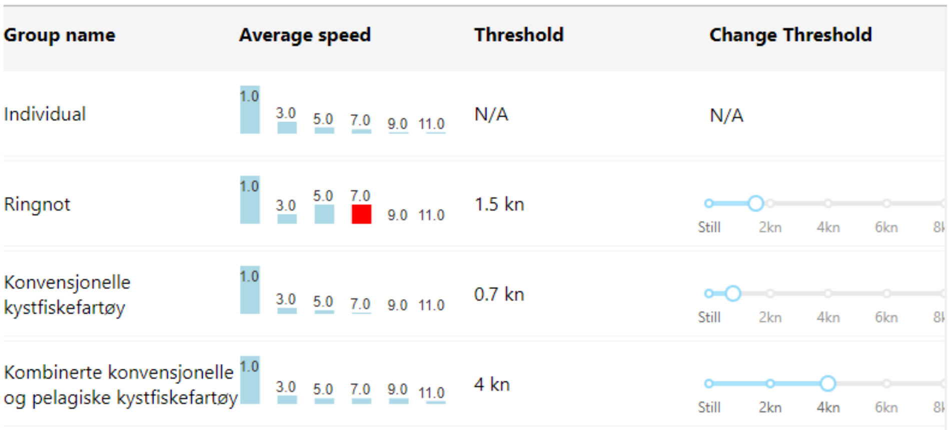

FIGURE 4.19: Speed indicator view

k-NN Settings

The k-Nearest Neighbors parameter window lets the user toggle visual display and filtering of the data used by the k-NN identifier, as well as tweaking and displaying the radius in which k-NN is applied to each license group. The view consists of a tabular view of group names and sliders for adjusting the maximum radius of the k-NN algorithm, and a toolbar. The toolbar contains radio buttons that toggle the display of an overlay showing all reports used by the k-NN algorithm in the map view, and a button to toggle displaying the radius of each vessel as defined by the maximum distance for that license group.

Figure 4.20 shows a part of the map view with k-NN overlays enabled. As DCA reports have inherent spatial information, they are displayed as dots on the map, shown by annotation 2. Color hue encodes the gear type used for each report. The color map chosen for encoding the gears was created using a twelve-bin categorical color map from ColorBrewer [16], a tool which creates well-defined color maps for cartography, and displayed in a legend overlaying the map (annotation 3 in Figure 4.20). The categorical color hues let the user evaluate whether the gear identified by the indicator is likely by comparing the location of the vessel with the gears used during previous operations within the vessel's proximity.

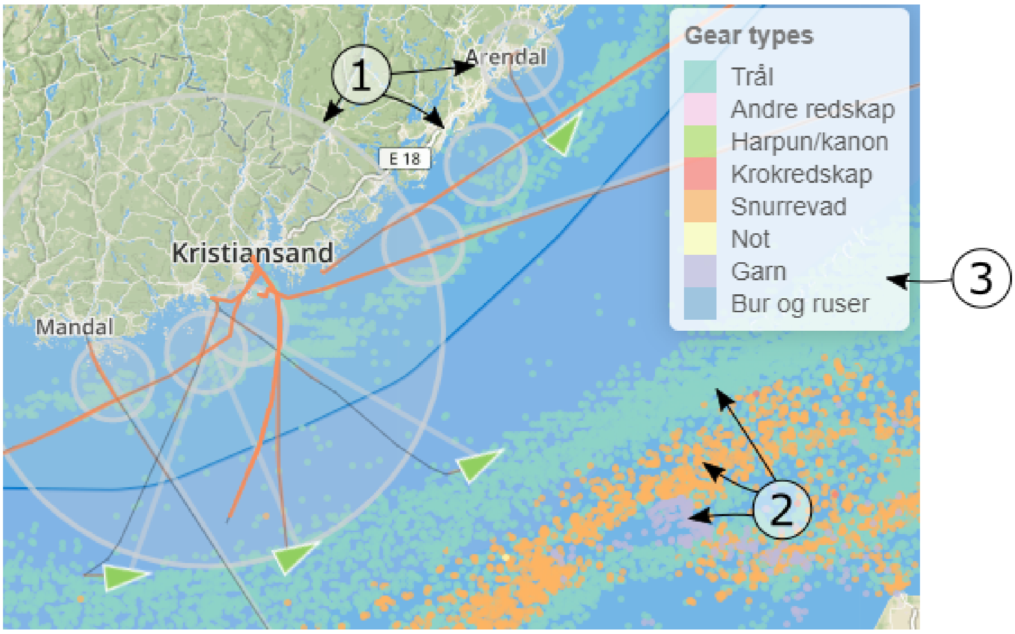

FIGURE 4.20: Visual representations of previous DCA reports and k-NN distance radii in the map. Currently groups of gear types are displayed.

Due to limitations to how many hues are discernable when encoding categorical data [64], glyphs were considered for visual encoding. However, glyphs suffer from the same issue of being difficult to discriminate, especially at small sizes, and do not scale to more than roughly twelve categories unless their size increases [13]. Additionally, shapes are a less effective channel than color hue for categorical data [64]. Finally, overlapping glyphs are prone to erroneous identification, as the combination of two glyphs' shapes may correspond to that of another glyph [13]. This is an issue with sufficiently large and spatially tightly packed datasets encoded by glyphs, such as the DCA report dataset. Hue was therefore used in lieu of glyphs.

As it is infeasible to encode more than twelve categories using color hue [64], the user can switch between different sets of categories by toggling between displaying the reports encoded using the different groupings of the gear hierarchy. At the lowest level, only gears within one high-level grouping are shown. This lets the user get a full overview of the gears used in the area around a vessel, then drill down to see more detailed information as desired.

The radius displays are shown as transparent circles around the center of each vessel's movement over the last day, with a line from the center point to the vessel. In some cases, the vessel has moved quite far away from the average position, and this line helps guide the user to see which circle corresponds to which vessel. As the data the circles were based on are radii, the natural encoding is a circle. The circles are used as a complement to the DCA report dots. To avoid occluding the DCA dots and to avoid confusion with other color choices, a translucent, non-attention grabbing light grey, not used elsewhere in the map view, was chosen as the circle's fill.

Template Group Viewer

The template group viewer shows a small version of the saved average templates for each equipment type and each license group, similar to the larger versions displayed in the template builder. The versions are laid out in a matrix, the X-axis consisting of gear types and the Y-axis of license groups. The template group viewer helps the user decide if they should consider using average templates instead of templates from individual vessels. If the user is satisfied with the number of templates available for a license group, they can switch to the averaged template dataset in the template builder. Additionally, the view helps the user decide which new templates may be worth constructing, by finding combinations of license groups and templates that haven't been identified before. A common workflow would be to identify a license group/gear pair one is dissatisfied with, open the template view on a second monitor, filter the dataset to only include vessels of the chosen license group, and go through each vessel in the tabular view until one find a suitable candidate for constructing a template.

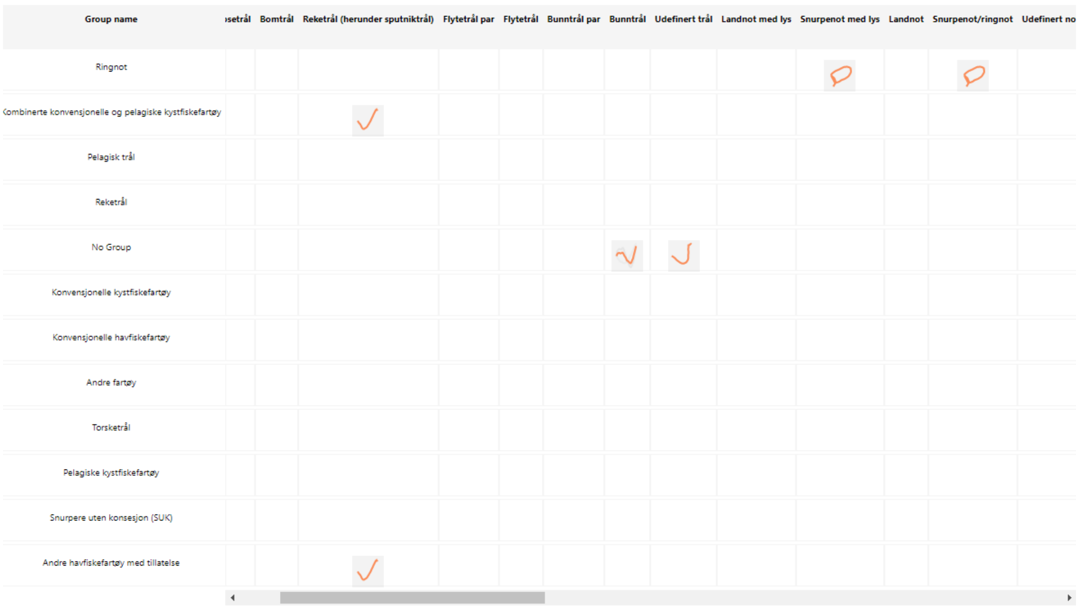

FIGURE 4.21: The Vessel Group viewer. The small red patterns show the template assigned to that license group/gear combination.

4.6 Summary

This chapter has presented the data used by VA-TRaC, and the tasks a user undertakes in order to perform analysis of potentially illegal fishing operations. Additionally, the views present in VA-TRaC have been presented, and for each a justification for the choices of the visual encodings used has been provided. As shown, VA-TRaC gives the user the ability to quickly assess the legality of fishing operations across all active vessels, through the use of automatic analysis and multiple, linked visualizations. This allows inspectors and analysts at the Directorate of Fisheries to perform such assessments in a timely manner, and thus improves on a task that has required tedious and time-consuming manual labor. Instead of providing a generic solution for vessel movement analysis, which existing visual analytics do, VA-TRaC focuses solely on catch operation analysis, and can therefore use contextual data such as catch reports and license information to provide indicators that are intuitive for the Directorate's employees, and that provide additional, relevant feedback such as the gear type used. Additionally the specialized approach lets VA-TRaC provide tools for changing the indicators' parameters and evaluating their effectiveness. This allows the system to be modified to be applicable across situations in a dynamic field influenced by changing regulations and technological advancements, the weather, and the movement of fish throughout the year.

Chapter 5

Implementation

This chapter provides a description of the architecture and technologies employed in VA-TRaC, and discusses the reasoning behind the choices.

5.1 Software Architecture

The web is a suitable platform for geospatial visual analytics systems [57]. Users can access the system without needing to download additional software, and across operating systems and devices. More importantly, visual analytics often requires strong compute capabilities to process the data. Using the server/client architecture of the web, one can offload heavy computations to the server, making the system more responsive and available even on devices with low computational power. The server can also handle large data sets without the user needing to download all the data onto their hard drive, which can be a problem for sufficiently large data sets and devices such as tablets [91].

A useful way to build a software architecture is to consider the flow the data should take throughout the application. In the field of visualization, dataflow is often modeled using the visualization pipeline, here exemplified by Chi's data state model [21], described in Chapter 2 and illustrated in Figure 2.6. An effective architecture should follow a similar model for unidirectional data flow, as it makes reasoning about the system simpler and more structured. In a sense, this is a generalization of Munzner's "what-why-how" framework, allowing the designer to focus on each stage's input and output data, tasks, and development idioms separately.

Figure 5.1 illustrates the architecture employed by VA-TRaC. It is a three-tier architecture, commonly used for web applications. In a three-tier architecture, a database tier stores the data, a server tier, often called the application logic tier, performs computations, and a client tier displays the results of the computations. The architecture allows for separation of concerns, and maps nicely to the data state model. The database tier stores values (with the value stage operator being any external data processing). The server performs the data transformation, and the client performs

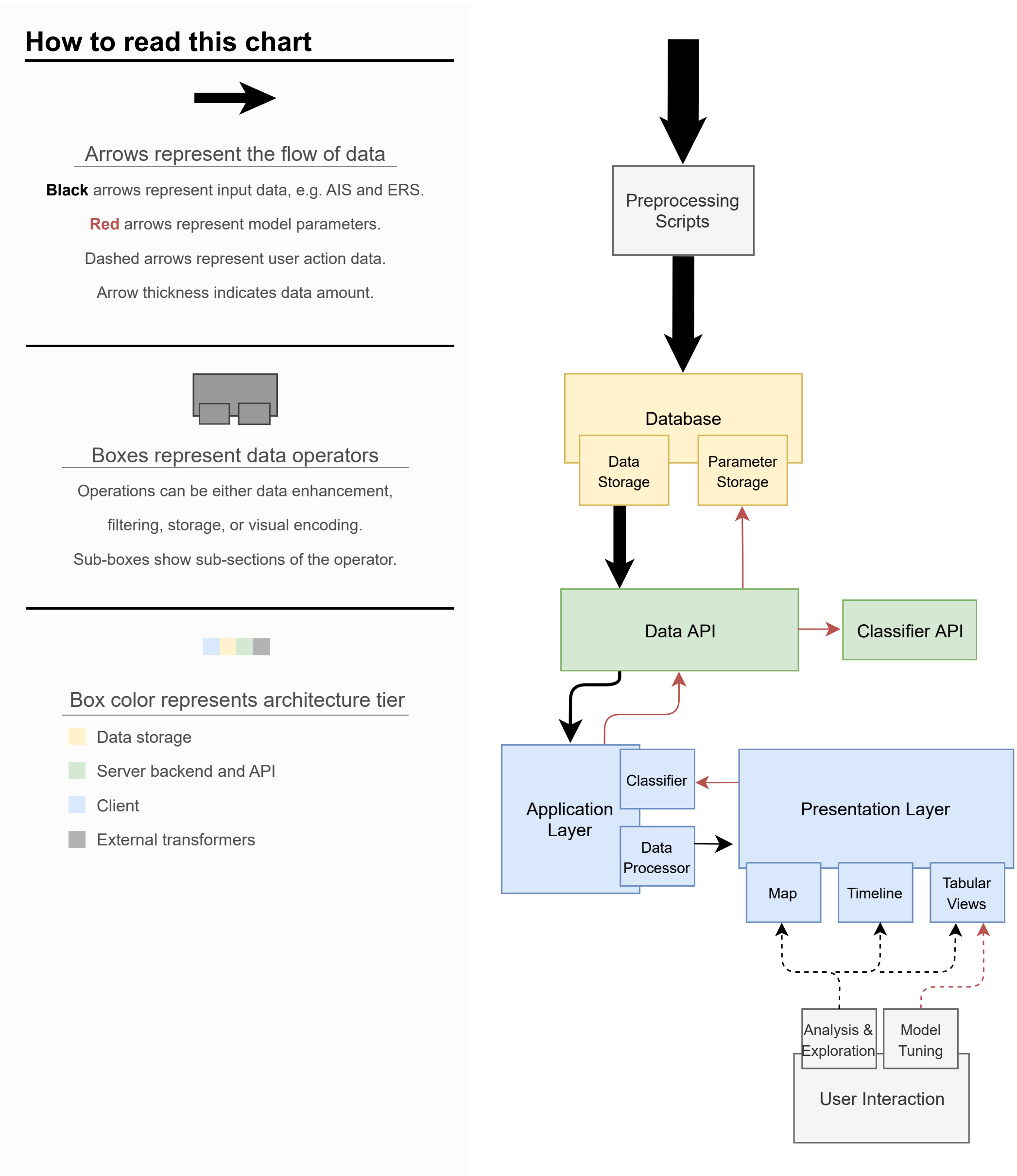

FIGURE 5.1: VA-TRaC 's architecture and data flow

the visual transformation mapping, presents the view stage, and receives view stage operations (such as user input). The analytical abstraction and visualization abstraction stages can be performed by either the server or the client, depending on application needs. VA-TRaC moves as much as possible from these stages to the client. The data API only performs the initial data transformation by querying the values and fetching those that are relevant. All following stages, including analytical stage operations, occur within the client, with the user providing view stage operations. The reasoning for such a "fat client" approach is two-fold: First, since multiple users may want to modify the indicator parameters at the same time, computing changes locally reduces the potential for data races where two users modify the algorithm parameters on the server simultaneously, which could cause unanticipated results for subsequent runs of the algorithm. By keeping such analytical stage operations local, one can be sure that one user's parameter modifications don't interfere with those of others. Second, VA-TRaC may be used by inspectors while on board vessels, where the internet connection can be sub-par. Using a fat client allows the inspectors to open the application while in a location with a good connection, and as long as the application is kept running, it will continue working even if the connection falters.

The disadvantage of using a fat client is the increased requirements for computation. However, as only a small, finite part of the total data (AIS items from a 24 hour period) is ever received by the client, the needed processing power has an upper bound. To further reduce the amount of computation that must be done by the client, a suite of data processing scripts preprocesses and filters the data before it enters the database, as described in the discussion of raw data in Chapter 4.

5.2 Technology Choices

When selecting programming languages, frameworks, algorithms and other technology for any substantial software project, it is important to assess the impact each will have on the resulting project [62]. To avoid a solution with many useless dependencies or one that is hard to understand and hard to change, an overarching set of goals was identified, and each technology was validated based on their suitability for reaching these goals. The following enumeration shows the criteria used for VA-TRaC. At least one item must be true to allow the technology to be used, and the more, the better. Listed in order of importance:

1. The technology must ideally be in use at the Directorate of Fisheries. If not, it must be easy to integrate or replace.
2. The technology must be used in a production environment at multiple other large institutions or companies.

3. The technology must be suitable for the development of a visual analytics application.

4. The technology must allow for rapid development.

The Directorate of Fisheries has a well-established IT and development department, where the personnel is trained in using specific technologies. Since VA-TRaC may see continued use, using technologies, languages, and libraries that the Directorate's developers are used to was considered a priority, based on the recommendations of Sedlmair et al. [71]. There are, of course, cases where using existing technology choices is infeasible, especially when choosing technologies that are specific to the domain of visualization and visual analytics, where the Directorate's developers have little to no prior knowledge. In these cases, choices have been made based on online popularity and documentation quality, in the hope that these attributes will help other developers get up to speed quickly. The other main stakeholders in system development were the domain experts in the analysis group. Certain functionality, such as the pre-processing of data, is done using tools requested by them, as they wanted the possibility of adding more data sources later on and possibly enhance the algorithms themselves without needing to rely on the developers in the IT department.

Any long-running project benefits from stable, proven technologies [62]. VA-TRaC may potentially fall into this category, if it ends up seeing extended development and use. Making sure the underlying technologies are available and stable for the foreseeable future helps facilitate extended use, as other developers will be able to maintain the system without dependencies suddenly being unavailable or drastically changing. The approach to assessing the stability of a technology was to verify that it was seeing production use in other large companies.

As noted earlier in this chapter, Keim et al. recommends using web technologies for creating visual analytics systems [57]. The goal is to ease user setup and on-boarding for the tool, as well as to make the tool available in a wider variety of settings. VA-TRaC is built to be used in an office setting for model development or statistical analysis, but also in a real-time regulatory setting by inspectors wanting to get a fuller picture of the situation of a specific vessel. A web interface allows the same underlying code to enable both use cases, and the use of a server/client architecture provides a robust way to handle the separation of preprocessing and data filtering algorithms and the user interface into independent parts.

Since the system was developed over approximately four and a half months of development time by a single developer, technologies that facilitate rapid development were necessary to get a result that solved the problem on time. While this was not the primary concern, it was considered a plus when evaluating potential technology choices.

The following technologies are employed in the final version of the system, based on the listed criteria:

- Programming Languages
  - TypeScript
  - JavaScript (node.js)
  - R
- Major frameworks and libraries
  - React (performant and lightweight user interface and web application development)
  - D3.js (low level and flexible vector graphics abstraction for the client-side web)
  - leaflet.js (map and cartography library for the client-side web)
  - tidyverse (data processing framework for R)
  - PlumbR (web server for R)
- Other tools
  - SQLite

TypeScript, node.js, R, and React were chosen based on the explicit wishes of the domain experts in the analysis group and the developers in the IT department. The technologies are all stable, offer extensive ecosystems of libraries, and are in use or are soon-to-be used by the Directorate. The other major libraries and frameworks are very common for their respective use cases, and each one is substantially beneficial in terms of development speed. D3.js and leaflet.js deserve a special mention, as they provide the flexibility needed to create customized interactive visualizations while retaining high performance. Both also have a wealth of documentation and usage tutorials, which is important if maintainers are to expand upon the visualizations without having much previous knowledge of the principles of effective visualization. The use of these technologies also allows the system to be hosted online, thus the user does not need to download specific software to use the system.

SQLite is a lightweight database that is very easy to set up and work with on a single machine. The Directorate runs an Oracle database for their data storage, to which there was limited access during the development, so SQLite was chosen as an alternative with many similar properties. As the Oracle database, it is a relational, SQL-based database, and therefore eases the transition of the data into the main database, if desirable.

The indicators were also validated based on how well they fit the criteria for technology choices. Each indicator can be implemented in less than 350 lines of code, making them easy to understand and integrate into existing solutions, even though they are not currently in use at the Directorate of Fisheries. Further, k-Nearest Neighbors classification is very commonly used, and the 1$ algorithm was developed in part by Microsoft Research and thus has at least some degree of corporate backing. All the indicators are also very fast to run, and therefore suitable for interactive systems even when parameters are modified on the fly. Finally, high-quality and well documented libraries for both k-Nearest Neighbors and the 1$ recognizer are provided for JavaScript, which allowed them to be easily modified and facilitated a faster development process.

Chapter 6

Workflows

In this chapter, the intended workflow of VA-TRaC is presented. As described in Figure 5.1 in the previous chapter, the user performs tasks in three levels of operations, namely exploration and identification of interesting vessels, evaluation of the correctness of indications for a particular vessel, and refinement of the indicators' parameters. Though the levels build on one another, the refinement operations are not mandatory for effective use of VA-TRaC, and can be performed separately. The levels of identification and evaluation are intended to be used by inspectors as a tool to verify the automatic identification of potentially unreported catch operations, possibly in a setting where the inspector is constrained on time. On the other hand, parameter refinement requires a deeper understanding of the underlying models and therefore is intended to be performed by for instance the analysis group, in contexts where time is not as much of a constraint and it is possible to go through the task flow of VA-TRaC multiple times. Therefore, this chapter presents two workflows, one based on identification and evaluation, and one on refinement.

6.1 Identifying Unreported Catch Operations

The following case study follows the workflow of an imagined inspector as they look for vessels to investigate. The inspector has launched the application and loaded all the data, and is about to start going through the vessels. Today, they are interested in analyzing the coast fishery vessels, and has therefore filtered the vessels to the first license group of coast fishery vessels: "Combined conventional and pelagic coast fishery".

Their analysis starts by looking for vessels that haven't reported at all during the last day, and where the program has flagged a potential fishing operation. They sort the list view by reporting status. Figure 6.1 illustrates the license group filter and the use of color to encode the vessel's reporting state. In this case, only vessels with a red highlight are of interest.

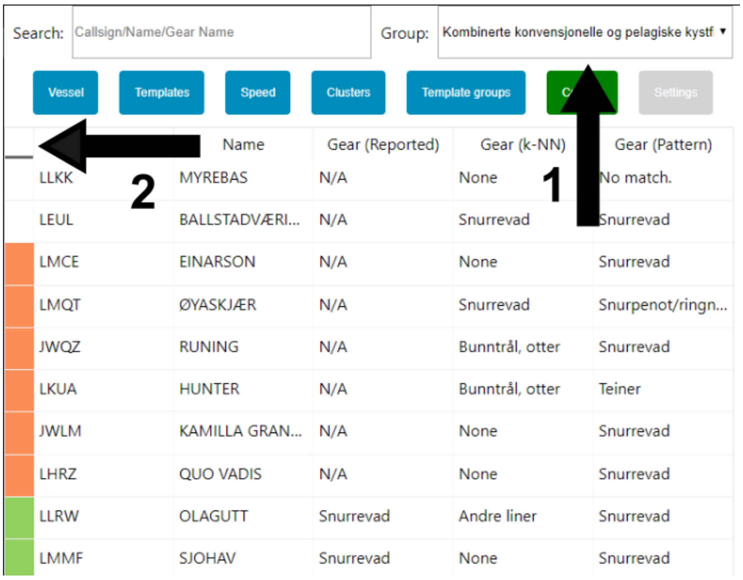

FIGURE 6.1: The inspector selects (1) the license group they want, and (2) sorts the list to see all the vessels that haven't reported DCAs, but have been recognized by the indicators.

One of the combined conventional and pelagic coast fishery vessels has an interesting movement pattern. The vessel has moved close to the coast, then looped around and moved away again. The system has flagged this movement as a potential fishing operation. The scarcity and irregularity of dots on the speed graph shows that the AIS reports have been sporadic, except for the recognized pattern, as illustrated in Figure 6.2. Due to the irregularity in the vessel's AIS reporting, the inspector decides to disregard the case, as the vessel may have technical difficulties and therefore has a valid reason for not sending its daily DCA report.

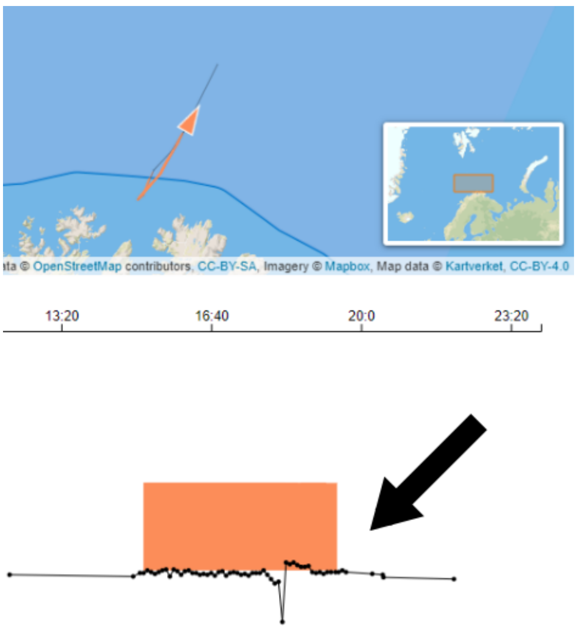

FIGURE 6.2: The vessel's timeline shows that there are lots of missing AIS datapoints.

Having iterated through each vessel in the first group, the inspector chooses the next relevant license group: Conventional coast fishery vessels. Here, they spot another interesting vessel. The vessel, seen in Figure 6.3, has moved around an island over a period of a few hours. During the movement, the vessel has slowed down for a period of time, and has therefore gotten flagged by the speed classifier, identified by red marks in both the map and timeline. The inspector sees that the speed changes of the vessel at the time could plausibly be a fishing operation. However, it would be very uncommon for a vessel to fish that close to a city, and there are plenty of other possible explanations for the behavior, such as the vessel washing its equipment. The inspector decides not to further investigate the vessel.

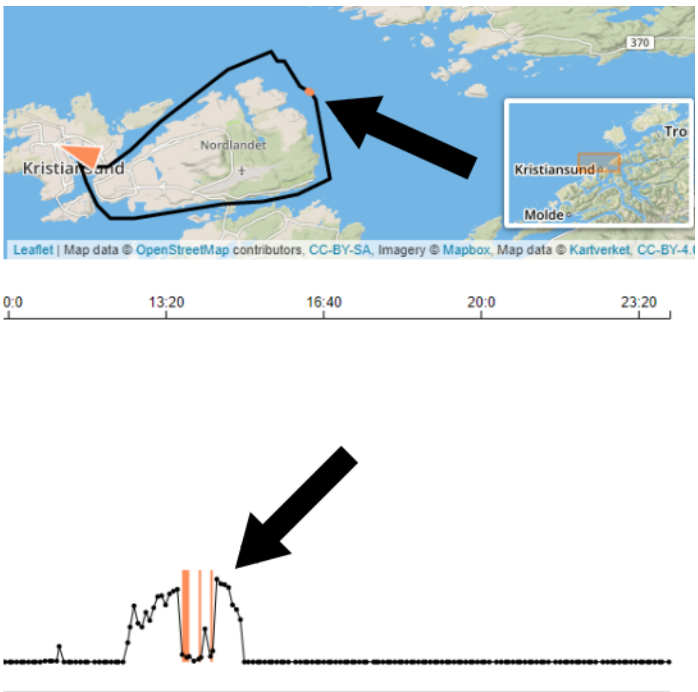

FIGURE 6.3: The vessel has performed a fishing operation-like movement. Due to the proximity to Kristiansund and the airport, the inspector decides this is a false positive.

The inspector finishes verifying all vessels without DCA reports, and moves to verifying the DCA reports of the coast fishery vessels. In the pelagic coast fishery group, they find a vessel that has been flagged by both the speed and 1$ recognition classifiers for a duration, and where no DCA was reported at the time. Figure 6.4 shows the point in the timeline where this occurs. They see that blocks for both speed and pattern indications are present. In Figure 6.5, they check the template in question and verify that it could plausibly look like a part of the pattern of a fishing operation using a seine (a vertical fishing net placed to surround a shoal of fish). The inspector finally checks the k-NN classifier to verify that the vessel probably has been using a seine. They enable the display of fishing operations in the area. In Figure 6.6, dots representing earlier fishing operations (a cluster is outlined with a circle) show that fishing using seines ("not" in Norwegian) is common in the vessels area. The inspector chooses to flag the vessel for further investigation.

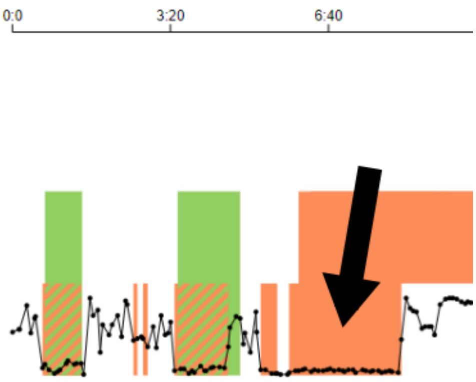

FIGURE 6.4: A change in the vessel's speed suggests that an unreported fishing operation has occurred.

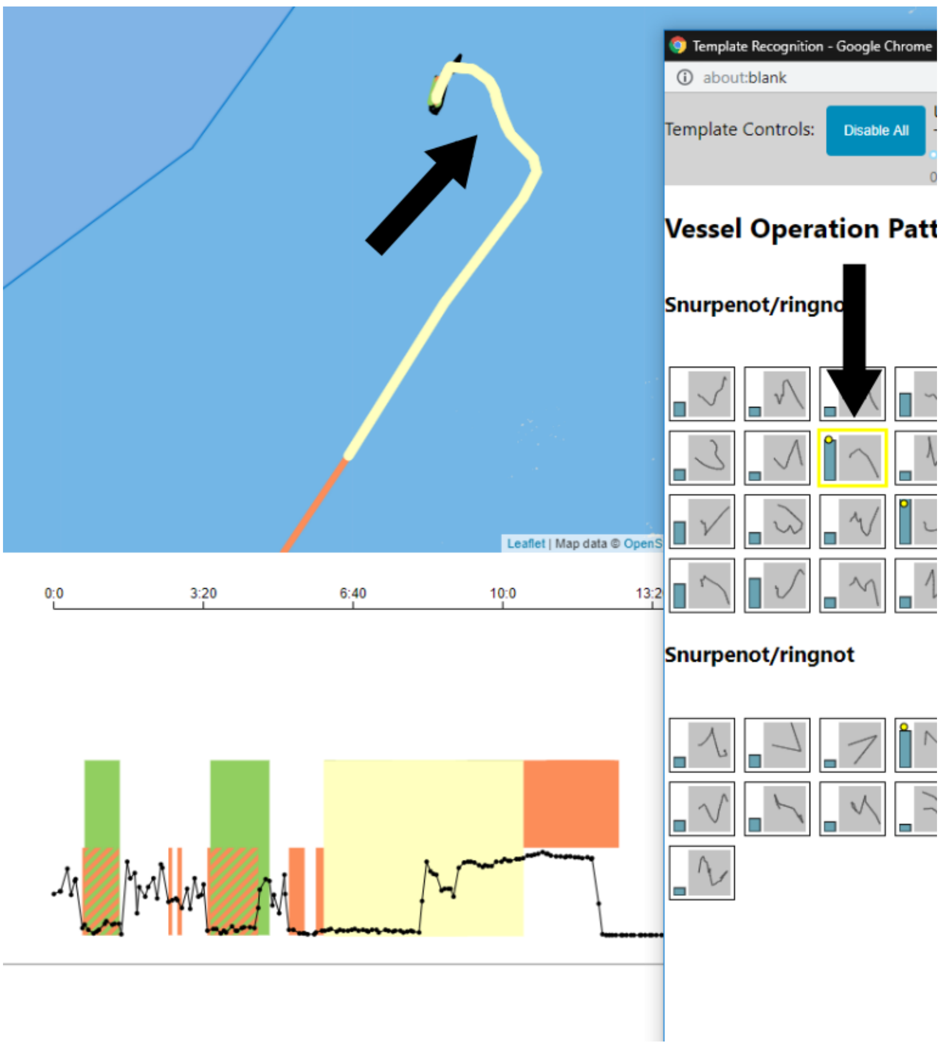

FIGURE 6.5: Hovering the template in the template viewer highlights the template in the timeline and map.

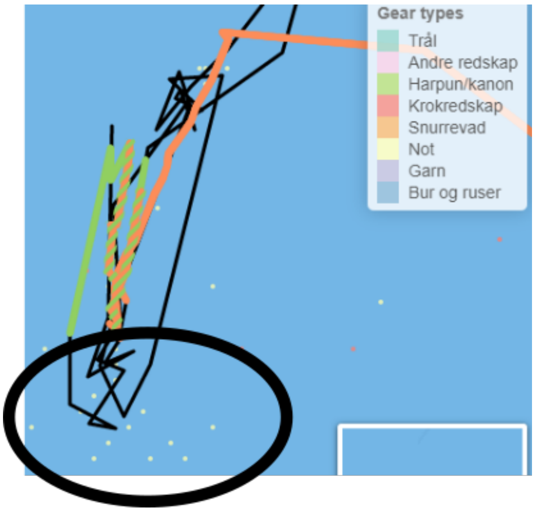

FIGURE 6.6: The black circle shows a cluster of seine operations in the area the vessel has been moving. Other seine operations are found scattered around the immediate surroundings.

In the described use case, the inspector found two vessels that had performed potentially illegal operations. Both cases would have been very difficult to identify by looking at traditional maps of the vessels' trajectories, and benefited from being identified by automatic algorithms and verified using multiple views to explain the vessel's behavior. Due to the inspector's knowledge of the area and different forms of vessel behavior, the first vessel was deemed unworthy of further investigation. This type of knowledge is hard to encode in automatic models, and the advantage of having a human in the loop becomes apparent.

6.2 Building New Templates and Tuning Parameters

The other use of VA-TRaC is refining the indicators used for identifying vessels. In the following section, the typical workflow for indicator enhancement is presented. The analyst's goal for the session is building a recognition template.

The analyst starts their session by opening the Template Group viewer, and looks for license group and gear combinations that currently have no patterns assigned to them. In this imagined situation, the currently available templates are the same as in Figure 4.21. As can be seen in the figure, many license group/gear combinations currently lack templates. The analyst sees that no groups contain templates for line fishery gear, and decides to add one. By sorting the vessel list by reported gear, the analyst finds vessels involved in line fishery. They then open the Template viewer.

The Template viewer displays a list of all the templates available for the vessel. The analyst scrolls through the list, and searches for similar templates that describe the movement during line fishery well. Based on the analyst's knowledge of line fishery, they avoid the templates that are very generic (in this case the short ones with

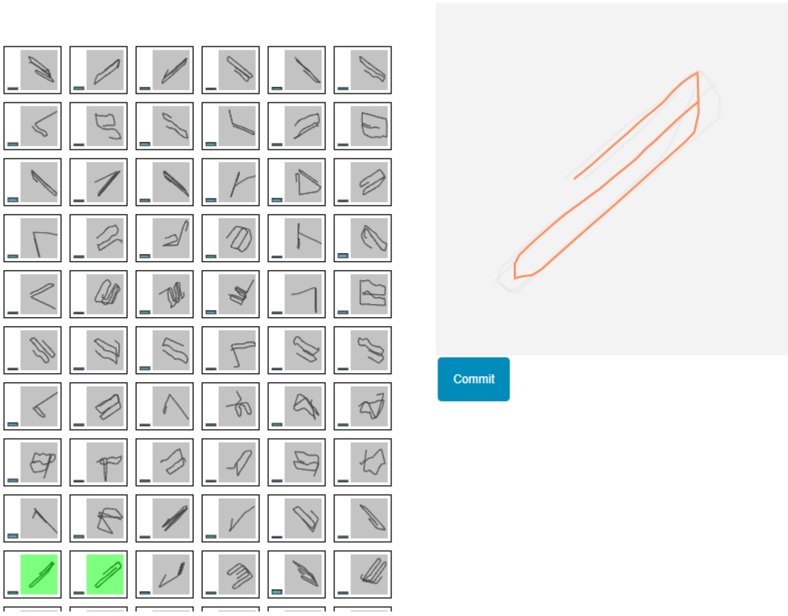

FIGURE 6.7: Building a new template for line fishery. Two similar patterns are chosen, and the average is displayed.

a single angle), and the ones with a very high fractal dimension. They find two patterns that describe a typical line fishery operation. The movement patterns of the operations are characterized by sharp, 90-degree angles and a blocky trajectory. They click on the two candidates to add them to the current average, as illustrated in Figure 6.7, and saves them by pressing commit. From the next line vessel in the list, they decide to add a second template. They identify two patterns they want to use. This time, the average path gets too rounded, and some of the 90-degree angles disappear. To rectify this, they decide to add extra weight to one of the templates by clicking on it an additional time. As can be seen in Figure 6.8, in this case the weighting causes corners to get less rounded, and the sharp angles get preserved better for some regions of the average path.

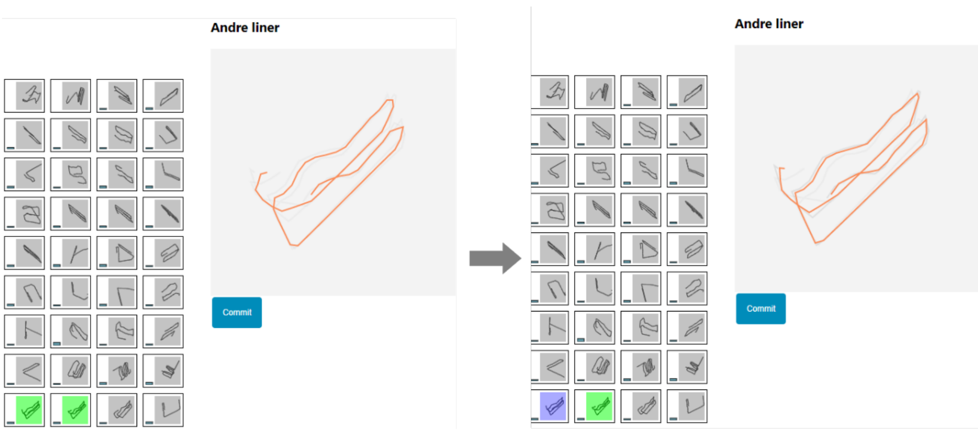

FIGURE 6.8: The blue template gets weighted more, and the average retains more of its features compared to the other template.

Satisfied with the new patterns, the analyst moves to enhance the parameters of the speed classifier. Shown in Figure 6.9, they notice a vessel in the timeline that has sent a DCA report for a period where the speed classifier looks too strict. They select the vessel in the vessel list by searching for its call sign, then opens the speed viewer (1). The vessel's license group is identified by finding the red bar in the histogram list. The analyst notices that many catch operations in the group occur at between 3.0 and 4.0 knots, so they increase the threshold from 3.1 knots. They can see the recognitions update in real-time in the timeline (2), and due to the result decides that 3.9 knots is a good threshold. Afterward, they verify in the evaluation viewer that the algorithm performance across all vessels didn't drop.

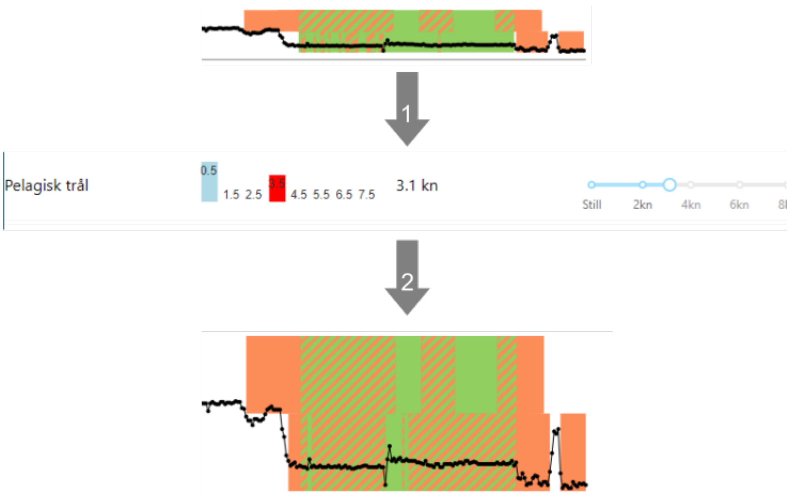

FIGURE 6.9: Tuning the speed threshold of a single group increases the correctly identified regions of a catch operation.

Chapter 7

Validation

The following chapter presents and discusses the validation methods employed during the work on VA-TRaC. The methods are presented both in terms of Elmqvist et al.'s evaluation patterns, giving an intuition to what was each method entailed in practice, and Munzner's nested model of validation, showing what each method actually validated.

7.1 Description of Validation Measures

The following section outlines all the different measures that were taken to validate VA-TRaC, grouped by time of occurrence. Where appropriate, the measures are assigned to the corresponding pattern of Elmqvist et al. [33].

7.1.1 Early Stages

The first meetings with the Directorate of Fisheries were conducted as an informal group interview attended by employees from several different areas of expertise, including the Statistics Department, the IT Department, the Fisheries Monitoring Centre, the Department of Regulations, and the analysis group. The meetings sought to identify a feasible problem to solve. During the discussions, several data analysis problems the Directorate had were mentioned, and the suitability of visualization-based solutions to each one was discussed. Once possible problems had been defined, they were further discussed with a visualization expert (the thesis supervisor), to verify that they were well suited for a visualization-based solution. Based on these discussions, identification and verification of illegal fishing vessel operations was chosen as the best suited candidate.

The design phase consisted of more or less informal discussions of possible algorithms (discussed in depth in the section on indicators in Chapter 4), which culminated in expert reviews of several designs, developed using the "Five Design-Sheet" methodology [68]. The reviews consisted of presenting the sheets separately to two people from the analysis group, who gave their comments. The comments were then compared, and used as a basis for choosing the most suitable design. In terms of Elmqvist et al.'s [33] patterns, the "Five Design-Sheet" methodology is a *paper baseline* pattern. The researcher creates the designs on paper, letting the users study and provide feedback on the design. The methodology also contains aspects of the *do-it-yourself* pattern, since the researcher has the final word in creating a final design.

An additional presentation of the system's design, this time focused on architecture and algorithm choices, was held for members of the analysis group and IT department before development started. Each of the chosen algorithms was discussed, with the goal of verifying that the algorithms were sound choices for fishing operation identification. Additionally, the meeting attendants discussed ways the system could be integrated into the data pipeline at the Directorate, for instance by designing for supporting dynamic data down the road.

7.1.2 Development Stage

Validation in the development phase consisted of continuous justifications of each part of VA-TRaC 's design. The justifications were done in an ad-hoc fashion, as development progressed and new insights were found. The resulting justifications are thoroughly discussed in Chapter 5.

Once the system's design started to settle and initial development had taken place, the system was presented again. The complete IT department and the most of the Statistics Department each attended a walkthrough of each component of the tool, and were encouraged to comment on the tool's usage. Most attendants did not have prior experience with the tool, though they possessed at least some degree of domain knowledge. Such a review by a larger audience is akin to the *complementary participants* pattern [33], especially since the intent of the presentations were to understand how a generalized user, for instance an inspector, would react to the system.

7.1.3 Post-Development Stage

Towards the end of development, one person from the analysis group was appointed as the initial main user and maintainer of VA-TRaC, and various discussions with this person resulted in extra measures taken to ease the transition, such as providing an API for running the indicators directly without the visualization interface. While VA-TRaC was not deployed to production, two meetings with members of the IT department were held to discuss system deployment at a later date, as well as continued development. To simulate working with the deployed system, the previously mentioned person from the analysis group was engaged for a *pair analytics* session, where they controlled VA-TRaC under guidance, and attempted a regular workflow. A similar session was held with another analysis group member that had been central during the system's design as well, though this did not include as thorough a walkthrough of the system. Each session culminated in an interview where the employees were asked about their thoughts on the system and the development process. The interview notes are available in Appendix A.

A final presentation of VA-TRaC was held for five members of the analysis group. The presentation was very hands-on, and sought to accurately display a typical workflow using VA-TRaC, along the lines of the *once upon a time* pattern. The attendees also had the opportunity to guide and direct the workflow through questions and comments, thus providing a close-to-interactive experience without the system needing to be deployed. Finally, the Software Usability Scale (SUS) [17], a quantitative scale of software usability, was employed, with four of the five attendees responding. The results of the questionnaire are found in Figure 7.3, and are discussed in-depth later in this chapter.

7.1.4 Employed Patterns

Table 7.1 lists each identified validation pattern in the previous description, and places them in the categories identified by Elmqvist et al [33]. Note: Validation patterns mean patterns intended for early confirmation of an evaluation study or analysis scheme to avoid wasting time and resources. The term is not to be confused with Munzner's use of the term validation, whose meaning is otherwise the one used.

Category	Pattern Name	Type	Occurrences
Exploration	Do-it-yourself	Qualitative	1
Control	Pair analytics	Qualitative	2
Generalization	Complementary participants	Qualitative	2
	(Expert review)	Qualitative	1
Validation	(Prototype)	Qualitative	1
Presentation	Once Upon a Time	Qualitative	2

TABLE 7.1: The used patterns based on Elmqvist et al.'s taxonomy [33]. Parentheses around an element means the pattern was not followed in full.

The most common patterns of validation employed were *once upon a time* and *complementary participants*. Both studies ease the burden on the domain experts that are to use the system, as less contribution and active engagement on their part is needed [33]. When working as a researcher in organizations where the domain experts have existing workflows and workloads that need to be completed, reduction of disruptions to their day-to-day work makes it easier to organize and conduct validation

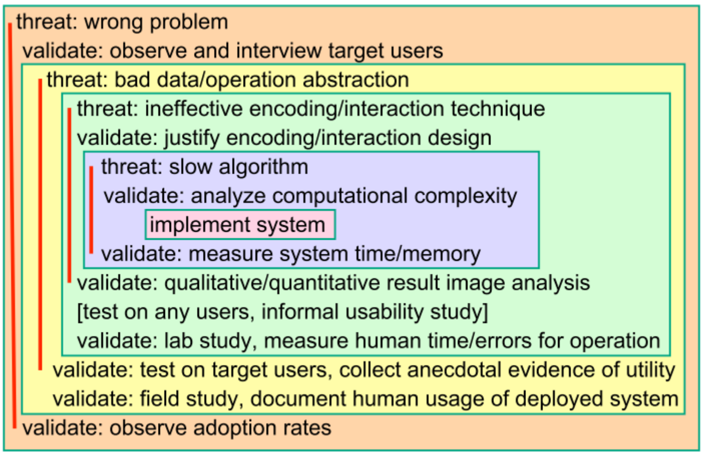

FIGURE 7.1: Munzner's Framework of Visualization Design and Validation [64]

sessions, as noted in Sedlmair et al.'s recommendation of limiting sessions with users to one hour [71]. The disadvantage of the patterns is that they don't maximize the potential for feedback from the core users. In the case of developing VA-TRaC, the advantages were found to outweigh the disadvantages.

All the patterns employed were qualitative in nature. This realization caused the use of the System Usability Scale as an additional, quantitative form of validation. There was also a lack of Validation patterns (Elmqvist's use of the term), discounting the fact that the entirety of VA-TRaC is a prototype.

7.2 Threats and Validations

The following sections frame the validation efforts in terms of *The Nested Model of Visualization Validation* [64]. As shown in Figure 7.1, there are four categories of threats to an effective visualization system. The threats not involving algorithm choices are the main difficulties for the kind of applied research where the researcher works in an external, large organization, as found by Sedlmair et al. Their challenges working with BMW mostly stem from organizational and political issues having to do with domain experts and their workflows in a large organization and little to do with implementation issues [71]. My personal experience at the Directorate mirrors this sentiment, as many of the challenges were similar to the ones found by Sedlmair et al. and occurred due to difficulties with organizational structure and the innate inertia of large organizations. Even so, at least one form of validation was performed for each level during the study, as illustrated in Figure 7.2.

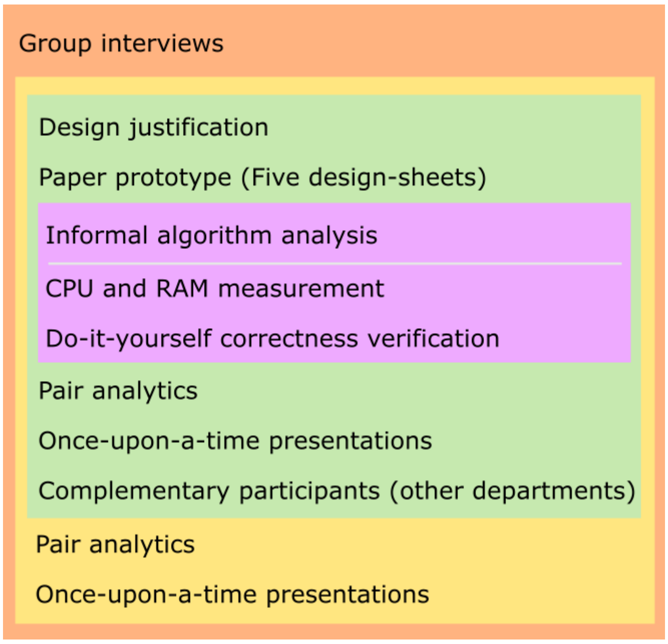

FIGURE 7.2: The validations performed in each step of the nested model. The color scheme follows the ones used by Munzner, as seen in Figure 7.1

7.2.1 Domain Situation

The domain level threat is that of solving the wrong problem. Large organizations have many groups of employees with different, often intersecting problems. Navigating and mapping these problems to find specific issues where applied research applications can deliver value can be a challenge [71]. Therefore, a group interview to identify a problem that should be solved using visualization was performed, to quickly identify multiple possible issues. The interview offered access to a diverse group of people, as recommended by Sedlmair et al.'s *R3-Environment* and *R7-OneHour* recommendations. R3 states that you should identify sub-problems and sub-targets to work with, R7 that you should not use more of the domain expert's time than necessary, as they more often than not have busy schedules [71]. The disadvantage of the method was that it may have caused a more imprecise problem definition than what a more in-depth and personal problem validation process could have created. For instance Sedlmair et al. got a lot of value out of specific discussions with domain experts [71]. In my case, more in-depth interviews with key domain experts, possibly outside the analysis group, may have uncovered the fact that no existing workflows for solving the problem addressed by VA-TRaC existed, which in practice did not become fully apparent until some time into development. Another issue with the method is that possible stakeholders that are less comfortable in group meetings may have stayed silent, making novel problem domains or objections to the ones mentioned go unnoticed.

Munzner recommends observing adoption rates as the downstream validation of the domain situation. Due to time constraints and the fact that VA-TRaC did not get deployed as a production system before the conclusion of the study made this infeasible. Therefore, no downstream domain situation validation was performed. However, the potential user base is quite small, and some of the potential users responded to the System Usability Scale. While the number of participants is too low for meaningful statistical assessment, the above-average results of the statement "I would like to use this system frequently" at least gives an indication that willingness to adopt the system exists (the complete responses are presented later in this chapter).

7.2.2 Data/Task Abstraction

The data and task abstraction threat is that wrong data and task abstractions are chosen, making the system not solve the problem of the user as the generalized visualization tasks don't map to the real world tasks the user tries to solve. Validation of the abstraction level can only occur after the system is at least partially implemented, as the way to find out whether the system solves the user's problem is to let the user use the system for their work. Thus, there are no immediate validation approaches suitable for abstraction validation, and all validations must occur downstream [64]. However, as a pointer to whether the task generalization was on the right track, the reasoning behind the task abstractions underlying the design sheets was discussed with the stakeholders.

Munzner recommends target user tests for downstream data/task validation. The sessions using the *once upon a time* pattern were focused on such validation, by letting users critique the system for the most common workflows. The sessions were complemented by the *pair analytics* sessions, as *once upon a time* by itself gives a result skewed by the narrative of the presenter [33].

The results for both validation types were quite positive, and responses were for the most part focused on small usability improvements rather than fundamental issues in the data and task abstractions. A recurring suggestion was the addition of more data sources and classification algorithms. Due to time constraints and constrained data access, this suggestion could not be followed through. However, the suggesters specified that additional data sources would be an extension to the existing choices, and the data and task abstractions were deemed sound.

The System Usability Scale provided anecdotal evidence of the system's utility. Users answered a questionnaire with ten generic statements about their perception of the system, and responses were converted into a score (Appendix B contains a description of the scale and questionnaire). VA-TRaC received an average total score of 73.125 from four respondents. As can be seen in Figure 7.3, the statements with the best results were "I thought there was too much inconsistency in the system" and

"I found the various functions in this system were well integrated", with respective average scores of 9.375 and 8.75. The lowest scores were given to "I needed to learn a lot of things before I could get going with this system", average score 5.625, followed by "I felt very confident using the system" and "Most people would learn to use this system very quickly", average score 6.25.

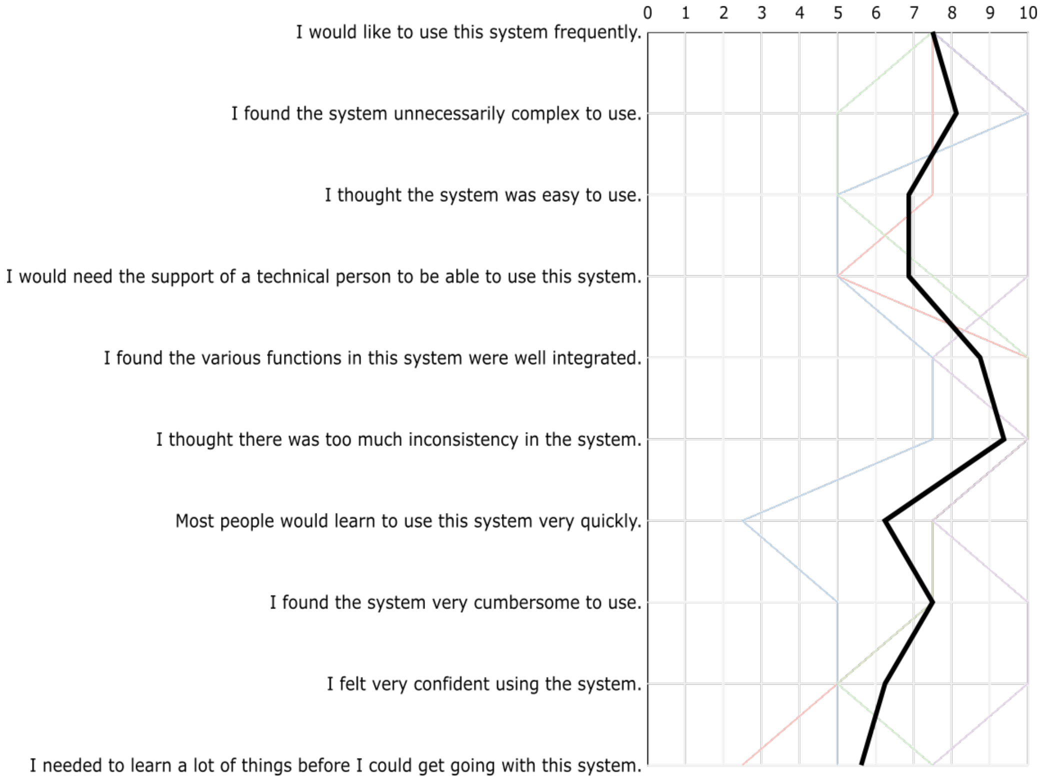

FIGURE 7.3: Results from all four responses of the System Usability Scale. Individual responses are represented by different hues. The thicker, black line denotes the mean.

From the responses, it is clear that the main issue of VA-TRaC is learning to use it and how the underlying algorithms work. For deployment, care must be taken to ensure the users understand the system's workflow, capabilities, and limitations, and that new users are trained to gain confidence in how to navigate and understand the workflows. This prompted additional interactions with the person appointed to be the initial user, to make sure their understanding of the system and the indicators were solid. On the other hand, the respondents seem to think the system is concise and integrated, without many complexities. This bodes well for a system built to let users work together with automatic algorithms, as it hints at the choice of visual encodings and interaction idioms that allow for an effective workflow once learned.

7.2.3 Visual Encoding/Interaction Idiom

The idiom level is threatened by idioms that insufficiently communicate the desired abstraction [64], meaning that users cannot effectively perform the tasks they want to, even though they have been correctly assessed and generalized by the researcher. Immediate validation of visual encoding and interaction idioms means justifying choices with regard to their effectiveness for the data and task at hand [64]. Chapter 4 contains such a justification for VA-TRaC, which will therefore not be repeated here.

Downstream validation of the idiom level was done through the *once-upon-a-time*, *pair analytics*, and especially *paper prototype* and *complementary participants*. The *paper prototype*, in the form of the "Five Design-Sheet" method, was used early to validate the initial choices of visual encodings encodings. In contrast to the *once-upon-a-time* and *pair analytics* sessions, the use of *complementary participants* allowed for feedback from people that had not seen the system before, and therefore were neutral to the chosen encodings at the beginning of the session.

An advantage of using feature presentations such as the *once upon a time* and *complementary participants* sessions in lieu of static images for validating visual encodings is that they allow validation of interaction idioms. For a system such as VA-TRaC, that involved exploration of the data through interaction, this is paramount to give the users a full understanding of the system. Each widget in the system was addressed individually, to encourage a more focused discussion of the idioms used for that specific widget, similarly to how analysis of still images would have occurred. Another advantage of the approach is that it brings some of the benefits of *Wizard-of-Oz* testing [33], in that the researcher can give feedback on what the results of certain interactions would be even though they are yet unimplemented.

The *pair analytics* sessions, though performed after the main development phase was concluded, uncovered weaknesses in both the visual encoding and the interaction idiom. For example, it was found that there was a need for more filtering options, as seeing all vessels at the same time was unnecessary for some tasks, cluttered the views, and reduced performance, leading to the addition of license groups not just as metadata for the indicators, but also as a user-enabled filter. Iterating through the vessel list using textual search was also found insufficient, so additional modes of navigation, through clicks on the vessel list, were added. Similarly, legends for the line charts in the timeline view were added based on these sessions.

7.2.4 Algorithm

As mentioned, the algorithm level was the least rigorously validated. Even so, it is important to keep a high level of interactivity in visual analytics systems, even in the presence of complex automatic algorithms and large amounts of data, lest the user becomes annoyed and loses focus [64]. A heuristic was therefore used: As the span of a single task in a point-and-click system is approximately ten seconds, results should be displayed within that time window in order to not impose delays that degrade the user experience [19]. For VA-TRaC, the heuristic was applied by selecting algorithms that are fast enough to run to completion within the aforementioned ten-second window instead, both for visualization presentation and for the indicators. The heuristic took the place of the recommended immediate validation for the algorithm level: algorithm complexity analysis [64]. This simplified implementation, giving more time for other work, such as validation of the other levels and system development. As the system was created as a relatively full-featured prototype in a new domain, it was deemed the correct trade-off to focus on exploring the design space and presenting the domain experts with the possibilities of a visualization solution instead of verifying the complexity of the employed algorithms.

Munzner notes time and memory measurements as the ubiquitous form of downstream validation at the algorithm level. The approach is common enough that it is nearly mandatory for novel algorithms [64]. In addition to the mentioned 10-second heuristic, an additional heuristic stating that memory should not exceed what can reasonably be expected on a regular work laptop was defined. The memory limit for VA-TRaC was therefore set to one gigabyte. Both heuristics were informally validated regularly during development.

The other downstream approach is the validation of algorithm correctness. When developing systems that may provide grounds for legal action, it goes without saying that algorithm correctness should be considered essential. This applies even in the development of prototype applications. As noted by McConnell [62], prototype applications run the risk of being put into real-world use, and developers should strive to get it right the first time. Effort was therefore put into validating that the algorithms were correct, by analyzing still images and interactions of the system and comparing them to the values found in the raw data, using the *do-it-yourself* pattern. *Do-it-yourself* was chosen since there were no stakeholders that had much insight into both the visual encodings and the inner workings of VA-TRaC, and therefore did not know exactly what to look for.

As Munzner notes, algorithm correctness is notoriously difficult to get right [64]. A case where this became very evident was the discovery that parsing of DCA messages into fishing operation time ranges was incorrect. This led to what seemed like a lack of reporting for the vessel and caused false positives in the recognition system. The discovery occurred during a presentation of idiom choices when one of the domain experts pointed out that fishing activities in the area the vessel was located usually are reported very diligently, and found it odd that this vessel seemed to not have reported anything. The error would probably have gone undiscovered if the meeting hadn't taken place, and highlights the fact that *do-it-yourself* solutions to algorithm correctness validation can be insufficient.

7.3 Takeaways from Validation

Using Munzner's model to independently reason about validation at separate layers turned out to be very helpful for the project, as levels where issues were found or where validation was lacking could be handled in a more focused manner, saving time both for myself and, more importantly, for the stakeholders who oftentimes had a busy schedule. Being able to hold shorter design validation sessions while still getting the wanted feedback meant more sessions could be held. Longer and more thorough approaches such as full-day workshops [58] would in this context be infeasible. Again, this mirrors the findings of Sedlmair et al. and their recommendation to keep sessions short [71]. I also found that for this project, the nested model worked well as a guideline for the chronological ordering of when to perform a certain validation, even though Munzner specifies that this is not the intent of the model [63].

Catch operation analysis, which was initially thought to be a area of work for inspectors relatively well-supported by digital tools, turned out to not have much support, and VA-TRaC thus became part of the greenfield work in the Directorate's nascent data analysis endeavors. The fact that the domain hadn't been explored in much depth earlier could have been uncovered faster if more thorough immediate domain validation had been employed, which may have helped guide the design process early on.

The gradual change in which stakeholders took part in the development and validation process, with gradual identification of domain experts who were interested in the system, was a success. The initial group interviews provided both an opportunity to get a wide range of potential users involved and let the domain experts who considered the project as outside their field of interest to get a cursory overview of the system. They could then decide whether the system was of interest to them and whether they would want to contribute their time to it. As the project's design, requirements, and features got more focused and cemented, so did the group of interested parties. For instance, while most of the IT Department only attended the first, large meeting, some individuals got further involved in the development process and asked for regular updates on how the system was performed, while commenting on technical choices. An example: while most members of the IT Department only attended the initial presentation, some got involved in how to successfully transfer the code base to their department. This led to the two meetings with IT department members which discussed project deployment and continued development. Hopefully, these meetings will help foster VA-TRaC's continued existence and ease the transition to new maintainers.

Two takeaways were found for validating algorithm correctness: First, conferring with the domain experts was critical in the validation. As the earlier example shows, domain experts can identify algorithm errors that look normal to the researcher. This sort of tacit domain knowledge is difficult for the researcher to obtain [71], and close collaboration with the domain experts can help offset this knowledge gap. This was also evident during the *pair analytics* sessions. Second, the occurrences where the algorithms were found to be incorrect happened when validation at another level was taking place. There were some attempts at directly presenting some of the algorithms, but as the domain experts involved were not used to thinking in terms of algorithms, these presentations didn't amount to any real discussions of algorithmic correctness. However, when the domain experts were presented abstractions and idioms, there seemed to be more engagement, as these levels were more intuitive, and this caused the issues at the algorithm level to surface as well.

Chapter 8

Conclusion

This thesis has presented VA-TRaC, a visual analytics tool built to let users visually identify and evaluate fishing operations found by applying automatic algorithms to vessels' trajectories. VA-TRaC provides the Directorate of Fisheries with a new tool to assist in the timely completion of catch operation analysis, and thus helps solve the challenge of regulating fishing vessel activity, which is becoming more and more difficult due to increasing numbers of vessels expected to follow regulations. Though some earlier research has been done on the use of visual analytics for analysis of catch operations, it remains a domain without much scientific research. In contrast to other tools, VA-TRaC uses not only positional data in the form of AIS, but also additional data sources such as ERS reports, and places more weight on the visual encoding of individual catch operations, which eases the identification and evaluation of operations for both the user and the automatic algorithms. VA-TRaC also lets the user modify the parameters of the automatic algorithms, allowing the algorithms to be strengthened even by users without programming knowledge.

VA-TRaC was developed in collaboration with domain experts in the analysis group at the Directorate, who work to implement similar algorithms, called indicators, designed to give the Directorate's inspectors data-driven indications on whether a vessel is performing illegal activities. It was designed using Tamara Munzner's "what-why-how" framework to generate task abstractions, and effective visual encodings were chosen based on analysis of data, tasks, and workflow. It was implemented as a web application, following the recommendations of early works on visual analytics and the wishes of the personnel at the Directorate. VA-TRaC was validated using a combination of qualitative validation approaches such as group interviews, once-upon-a-time presentations, and paper mockups. Preliminary validation showed that VA-TRaC was a success, and was considered to provide a well-integrated and consistent tool for identifying, evaluating and refining the results from the available indicators.

8.1 Recommendations

During the development and validation of VA-TRaC, several impressions were formed about the process of building a visual analytics tool for domain experts in a large organization. The following section presents the main recommendations based on my work at the Directorate. They are by no means applicable to every situation, but could be useful when working in similar domains.

- Using a validation framework such as the nested model of visualization validation made it easier to verify the effectiveness of the system at all times. This includes the period before the system was being designed: thinking about system validation using the nested model even during the initial meetings helped guide the search for a valid domain problem, by making me consider the downstream implications of a specific problem. Furthermore, structured validation efforts allowed me to speak with more confidence during system presentations, and made it easier to understand what the short-term goals should be during the development phase.
- Validation should be extended to successive design iterations. For the design of VA-TRaC, validation of design iterations after the initial one were much more ad-hoc than the initial, five-design sheet based one. My experience working in a greenfield project such as VA-TRaC is that such an approach is quite brittle and that one has less confidence in the ad-hoc design iterations than the the structured one. Software engineering methodologies such as "agile" could be employed to strengthen the design iteration workflow. One can envision changes to the five-design sheet methodology where the layout and content of each sheet is kept, but only a single sheet is created at regular intervals such as a "sprint", incorporating the successes of the last design with any updates to requirements or insights obtained since the last one.
- There are vast amounts of tacit knowledge the Directorate, as is common in large organizations [74]. Being only a single researcher on a limited schedule meant that there was no time to learn every detail of the domain. Instead, I learned the high-level components of fishery analysis, and let the domain experts provide feedback on the suitability of approaches and algorithms during the various meetings and presentations. As an example, one early stakeholder recommended adding a classification algorithm based on the vessel's speed, and others recommended extending this classification to work on different groups of vessels, as size and engine power has implications on speed during fishing operations. These kinds of detail are very hard to come by as an independent researcher from another domain, so letting the domain experts deal with details of fishery analysis meant my time was freed to work on my area of expertise, the design and implementation of visualization software.

- In the Directorate of Fisheries, there are a myriad of different domain experts that may be interested in projects such as visualization design studies. However, many domain experts are not able to contribute in meaningful ways even though they would like to, as they have obligations elsewhere. Furthermore, changes in personnel happen from time to time, and it is therefore necessary to continuously be on the lookout for relevant stakeholders. For example, I initially believed that the IT Department would inherit the project. However, some time into VA-TRaC's development, a newly employed member of the analysis group was appointed as the main user and maintainer instead. Sedlmair et al. recommend spending time identifying the correct stakeholders as well [71], which strengthens the case that the recommendation holds generally for large organizations. In my experience, being aware and taking steps to verify stakeholder involvement when changes happen is crucial for ensuring that the development process is as valuable as possible for both the researcher and the organization.

Appendix A

Appendix: Interview notes and validation results

The following sections contain the notes taken from the interview parts of the pair analytics sessions.

A.1 Interview questions

The following listing contains the interview questions. As the interviews were loosely structured, they followed the direction the domain expert wanted to take the conversation. The questions were therefore not always phrased as below during the actual interview, but similar questions were asked.

A.1.1 System

- Usefulness (Data/operation abstraction):
  - Do you think the system is useful?
  - Do you think the system can be made useful? (Data/operation abstraction)
  - Will you keep using the system?
- Usability (Encoding/interaction)
  - Was the system hard to use? Were there any specific features you struggled with?
  - Do you think seeing the visual representations made it easier to gain insight?
  - What would you change in the system?
  - Did you lose focus while using the system (slow performance etc.)?

A.1.2 Process

- Do you think you've had opportunities to influence the direction of the system? Did you feel you have enough knowledge of both technical and domain issues to adequately guide the process?
- Have you received adequate updates on the development process?
- In what ways would you like to see the project continue?

A.2 Interview with main stakeholder

Date: 17. June 2019

A.2.1 Agenda and purpose

- Pair analytics/Case Study: Assess quality of algorithm results for a single vessel, evaluate parameters for clustering.
- System walk through: Explain all widgets and functionality: Feedback on usability, intuitiveness, possible extensions, difficulties, missing features.

A.2.2 Notes

- Missing legends for certain marks/channels, e.g. speed and AIS precision. Important to focus on the minutae in addition to the big picture
- Displays of certain vessel information in popup views (license groups in speed popup) are missing/could be clearer
- Likes map widget, as it is easy to understand and contains lots of information while being clear
- Finds the kNN map overlays intuitive and useful (both viewing radii and different groups of gears)
- Likes the consistent color scheme, especially how it works with vessel list widget
- System too difficult to use for inspectors or other end users, requires some knowledge of algorithm choices in addition to domain knowledge about fishing vessel activities
- Timeline widget hard to understand without detailed explanations, may be unclear

- Likes the use of different views to maximize information so that both temporal and spatial aspects are nicely displayed
- Nice-to-have: Automatically set distances

Discussion of choices for algorithm parameters and pre-processing steps: k-value choice, template pre-filtering

Walkthrough of data and architecture

Discussion of use cases and potential for system in the Department of Fisheries:

- Include aspects in existing systems in development for inspectors
- Take inspiration from layout and widgets
- Will probably not the system wholesale "in production", as both IT department and domain experts would be needed to improve and control system. Too many stakeholders to keep development going without specific users driving development
- Possibly use results of parameter choices for evaluation of other algorithms
- Discussed use of visual encodings and data/algorithm changes needed to add support for streaming data

A.3 Interview with stakeholder from analysis and statistics department

Date: 21. June 2019

A.3.1 Agenda and purpose

Informal interview to collect anecdotal evidence of utility and discuss future usage areas:

A.3.2 Notes

- **What do you think of the development process?** Has enjoyed being a part of the process, and thinks the project has been a success. Happy with the new ideas that come from having students/people with other backgrounds in the daily work. Also thinks it was a major advantage that I had domain knowledge beforehand, making it easier to let me work by myself (Elmqvist's Do-It-Yourself is less of a risk, though still risky).

- Asks **Do you think you got adequate feedback?** I answer: All-in-all, yes, though it sometimes has been difficult to explain and get relevant feedback on a greenfield application and research field in a large and well-established organization. We discuss the inherent inertia in such organizations, he feels that things haven't been moving slower than they usually do in the organization.
- Feels like there has been enough presentations of progress, and crucially that the presentations have involved the right people, that while they often are very busy have gotten some understanding of the project. I could have done a little more more to identify and include key actors. Could also have done a little more to add more interactivity to presentations, and explain the systems in less formal/rigid/academic terms (even though I tried).
- **Do you see future potential for the application?** Discussed indicators of vessel/master/shipping company/landing facility for inspectors, how the thesis project could be integrated to provide indicators for missed reporting and visual representations of indicators. Also thought it might be helpful to put more indicators into the thesis system, not only other way around. We also discuss potential visual representations for other indicators.
- **Do you think a visualization approach has provided any benefits over automatic methods?** The stakeholder is a statistician, jokes that "Graphics are for cowards". Does see the use of visualization to see developments over time and space, especially the use of maps. Thinks the problem of validating results from automatic methods becomes easier with the use of visualization. We discuss the use of visualization to help inspectors do their job while keeping them accountable for their actions is easier than other approaches, and that training and decision-making may be simpler.
- **Other notable points** The stakeholder feels there has been tremendous value in the "extra-curricular activities" I have engaged in, such as the project presentations to a variety of groups over the course of the implementation period, the knowledge transfer meetings and discussions of data abstraction and visualization approaches for marginally relevant projects has been fundamental in maximizing the probability that the thesis project will be used in the future, either as inspiration to similar tools, or as a partial or wholesale solution.

Appendix B

Appendix: System Usability Scale questionnaire

The System Usability Scale (SUS) was used as part of the validation process, and the questionnaire and scoring rules are included here for reference.

B.1 Questionnaire

The questionnaire consists of the following ten statements.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex to use.
3. I though the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in the system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

For each statement, the user specifies whether she "strongly disagrees", "strongly agrees", or somewhere in between, using a five-point scale:

Strongly disagree				Strongly agree
1	2	3	4	5

B.2 Scoring

The Software Usability Scale gives a total score between 0 and 100. Each item can get a score from 0 to 10. To calculate the score for an item, convert the answer for the item to the corresponding number. For items 1, 3, 5, 7 and 9, subtract 1 from the number, and multiply by 2.5. For items 2, 4, 6, 8, and 10, subtract the score from 5, then multiply by 2.5. The total score of the questionnaire is the sum of all the individual items' scores.

Bibliography

[1] Wolfgang Aigner et al. *Visualization of Time-Oriented Data*. Springer, 2011. ISBN: 978-0-85729-078-6.

[2] Deborah Ancona. 'Sensemaking: Framing and Acting in the Unknown'. In: *The Handbook of Teaching Leadership: Knowing, Doing, and Being*. Ed. by Scott Snook, Nitin Nohria and Rakesh Khurana. SAGE Publishing Inc., 2014, pp. 3–19. ISBN: 978-1412990943.

[3] Gennady Andrienko and Sankt Augustin. 'Visual Analytics Tools for Analysis of Movement Data'. In: *ACM SIGKDD Explorations Newsletter* 9.2 (2007), pp. 38–46. DOI: 10.1145/1345448.1345455.

[4] Gennady Andrienko et al. 'Geovisual analytics for spatial decision support: Setting the research agenda'. In: *International Journal of Geographical Information Science* 21.8 (2007), pp. 839–857. DOI: 10.1080/13658810701349011.

[5] Gennady Andrienko et al. 'Space, time and visual analytics'. In: *International Journal of Geographical Information Science* 24.10 (2010), pp. 1577–1600. DOI: 10.1080/13658816.2010.508043.

[6] Natalia Andrienko and Gennady Andrienko. 'Visual analytics of movement: An overview of methods, tools and procedures'. In: *Information Visualization* 12.1 (2013), pp. 3–24. DOI: 10.1177/1473871612457601.

[7] BarentsWatch. *BarentsWatch Fishery activity*. 2019. URL: https://www.barentswatch.no/en/fisheryactivity/.

[8] Katrina E. Barkwell et al. 'Big data visualisation and visual analytics for music data mining'. In: *Proceedings of the IEEE International Conference on Information Visualization*. 2018, pp. 235–240. DOI: 10.1109/iV.2018.00048.

[9] Leilani Battle et al. 'Evaluating Visual Data Analysis Systems : A Discussion Report'. In: *Proceedings of HILDA: Workshop on Human-In-the-Loop Data Analytics*. Houston, 2018, pp. 1–6. DOI: 10.1145/3209900.3209901.

[10] Lucas Bernardi, Themis Mavridis and Pablo Estevez. '150 Successful Machine Learning Models: 6 Lessons Learned at Booking.com'. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 1743–1751. DOI: 10.1145/3292500.3330744.

[11] Christopher Bonham et al. *Analysing port and shipping operations using big data*. Tech. rep. June. 2018. URL: https://datasciencecampus.ons.gov.uk/wp-content/uploads/sites/10/2018/06/here.pdf.

[12] Anthony J. Booth. 'Incorporating the spatial component of fisheries data into stock assessment models'. In: *ICES Journal of Marine Science* 57.4 (Aug. 2000), pp. 858–865. DOI: 10.1006/jmsc.2000.0816.

[13] R Borgo et al. 'Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications'. In: *Eurographics - State of the Art Reports*. 2013. DOI: 10.2312/conf/EG2013/stars/039-063.

[14] Nick Bostrom and Eliezer Yudkowsky. 'The Ethics of Artificial Intelligence'. In: *The Cambridge handbook of artificial intelligence*. Ed. by Keith Frankish and William M. Ramsey. Vol. 1. Cambridge University Press, 2014, pp. 316–334. ISBN: 978-0-521-87142-6.

[15] Matthew Brehmer and Tamara Munzner. 'A Multi-Level Typology of Abstract Visualization Tasks'. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2376–2385. DOI: 10.1109/TVCG.2013.124.

[16] Cynthia Brewer and Mark Harrower. *ColorBrewer: Color Advice for Cartography*. 2019. URL: http://colorbrewer2.org.

[17] J Brooke. 'SUS—a quick and dirty usability scale'. In: *Usability evaluation in industry*. Ed. by Patrick W. Jordan et al. CRC Press, 1996. ISBN: 978-0748404605.

[18] A. Buja et al. 'Interactive data visualization using focusing and linking'. In: *Proceedings of the IEEE Conference on Visualization*. IEEE Computer Society Press, 1991, pp. 156–163. DOI: 10.1109/VISUAL.1991.175794.

[19] Stuart K. Card, George G. Robertson and Jock D. Mackinlay. 'The information visualizer, an information workspace'. In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. 1991, pp. 181–186. DOI: 10.1145/108844.108874.

[20] Shui-Kai Chang, Tzu-Lun Yuan and Verena Trenkel. 'Deriving high-resolution spatiotemporal fishing effort of large-scale longline fishery from vessel monitoring system (VMS) data and validated by observer data'. In: *Canadian Journal of Fisheries and Aquatic Sciences* 71.9 (2014), pp. 1363–1370. DOI: 10.1139/cjfas-2013-0552.

[21] Ed H Chi. 'A Taxonomy of Visualization Techniques using the Data State Reference Model'. In: *Proceedings of the IEEE Symposium on Information Visualization*. 2000. DOI: 10.1109/INFVIS.2000.885092.

[22] Tsun Chow and Dac-Buu Cao. 'A survey study of critical success factors in agile software projects'. In: *Journal of Systems and Software* 81.6 (2008), pp. 961–971. DOI: 10.1016/j.jss.2007.08.020.

[23] Pak Chung Wong et al. 'The Top 10 Challenges in Extreme-Scale Visual Analytics'. In: *IEEE Computer Graphics and Applications* 32.4 (2012), pp. 63–67. DOI: 10.1109/MCG.2012.87.

[24] Lawrence. Chung et al. *Non-Functional Requirements in Software Engineering*. Springer US, 2000. ISBN: 1461552699.

[25] William S. Cleveland. 'A Model for Studying Display Methods of Statistical Graphics'. In: *Journal of Computational and Graphical Statistics* 2.4 (1993), pp. 323–343. DOI: 10.2307/1390686.

[26] William S. Cleveland and Robert McGill. 'Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods'. In: *Journal of the American Statistical Association* 79.387 (Sept. 1984), p. 531. DOI: 10.2307/2288400.

[27] Andy Cockburn, Amy Karlson and Benjamin B. Bederson. 'A review of overview+detail, zooming, and focus+context interfaces'. In: *ACM Computing Surveys* 41.1 (2009), pp. 1–31. DOI: 10.1145/1456650.1456652.

[28] Colour Blind Awareness. *Types of Colour Blindness*. URL: http://www.colourblindawareness.org/colour-blindness/types-of-colour-blindness/.

[29] Eric Delmelle and Raymond Dezzani. 'Overview, classification and selection of map projections for geospatial applications'. In: *Handbook of Research on Geoinformatics*. Ed. by Hassan A. Karimi. February 2015. IGI Global, 2009, pp. 89–98. ISBN: 9781591409953.

[30] Alan Dix and Geoffrey Ellis. 'Starting Simple - Adding Value to Static Visualisation Through Simple interaction'. In: *Proceedings of the ACM AVI International Conference on Advanced Visual Interfaces*. 1998, pp. 124–134. DOI: 10.1145/948496.948514.

[31] Nicholas D. Ducharme-Barth and Robert N. M. Ahrens. 'Classification and analysis of VMS data in vertical line fisheries: Incorporating uncertainty into spatial distributions'. In: *Canadian Journal of Fisheries and Aquatic Sciences* 1764 (2017), pp. 1–51. DOI: 10.1139/cjfas-2016-0181.

[32] Nicholas D. Ducharme-Barth, Kyle W. Shertzer and Robert N. M. Ahrens. 'Indices of abundance in the Gulf of Mexico reef fish complex: A comparative approach using spatial data from vessel monitoring systems'. In: *Fisheries Research* 198 (Feb. 2018), pp. 1–13. DOI: 10.1016/J.FISHRES.2017.10.020.

[33] Niklas Elmqvist and Ji Soo Yi. 'Patterns for visualization evaluation'. In: *Information Visualization* 14.3 (2015), pp. 250–269. DOI: 10.1177/1473871613513228.

[34] Elsevier B.V. *Most Cited Fisheries Research Articles - Elsevier*. URL: https://www.journals.elsevier.com/fisheries-research/most-cited-articles.

[35] John W. Emerson et al. 'The Generalized Pairs Plot'. In: *Journal of Computational and Graphical Statistics* 22.1 (2013), pp. 79–91. DOI: 10.1080/10618600.2012.694762.

[36] Alex Endert, Patrick Fiaux and Chris North. 'Semantic interaction for visual text analytics'. In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 473–482. DOI: 10.1145/2207676.2207741.

[37] Alex Endert et al. 'The State of the Art in Integrating Machine Learning into Visual Analytics'. In: *Computer Graphics Forum* 36.8 (2017), pp. 458–486. DOI: 10.1111/cgf.13092.

[38] René A Enguehard, Orland Hoeber and Rodolphe Devillers. 'Interactive exploration of movement data: A case study of geovisual analytics for fishing vessel analysis'. In: *Information Visualization* 12.1 (2012), pp. 65–84. DOI: 10.1177/1473871612456121.

[39] Andre Esteva et al. 'Dermatologist-level classification of skin cancer with deep neural networks'. In: *Nature* 542.7639 (2017), pp. 115–118. DOI: 10.1038/nature21056.

[40] Fiskeridirektoratet. *Den aktive fiskeflåten*. URL: https://fiskeridir.no/Yrkesfiske/Statistikk-yrkesfiske/Fiskere-fartoey-og-tillatelser/Den-aktive-fiskeflaaten.

[41] Fiskeridirektoratet. *Forslag til Plan for Utvidet Datainnsamling fra Fiskeflåten*. 2018. URL: https://www.fiskeridir.no/content/download/23171/323126/version/14/file/elektronisk-rapportering-fartoy-under-15-meter-oktober2018.pdf.

[42] Fiskeridirektoratet. *Hva er Fiskeridirektoratets sjøtjeneste?* 2017. URL: https://fiskeridir.no/Yrkesfiske/Regelverk-og-reguleringer/Stenging-og-aapning/Nord-for-62-N/Hva-er-Fiskeridirektoratets-sjoetjeneste.

[43] Fiskeridirektoratet. *J-208-2017: Forskrift om posisjonsrapportering og elektronisk rapportering for norske fiske- og fangstfartøy*. 2017. URL: https://fiskeridirektoratet.no/Yrkesfiske/Regelverk-og-reguleringer/J-meldinger/Gjeldende-J-meldinger/J-208-2017.

[44] Fiskeridirektoratet. *J-244-2018: Forskrift om regulering av fisket etter blåkveite nord for 62°N i 2019*. 2018. URL: https://www.fiskeridir.no/Yrkesfiske/Regelverk-og-reguleringer/J-meldinger/Gjeldende-J-meldinger/J-244-2018.

[45] Fiskeridirektøren. *Nasjonal strategisk risikovurdering*. Tech. rep. 2019. URL: https://www.fiskeridir.no/Yrkesfiske/Dokumenter/Rapporter/2019/Nasjonal-strategisk-risikovurdering-NSRV-for-2019.

[46] Sohaib Ghani et al. 'Visual analytics for multimodal social network analysis: A design study with social scientists'. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2032–2041. DOI: 10.1109/TVCG.2013.223.

[47] Robert B. Haber and David A. McNabb. 'Visualization idioms: A Conceptual Model for Scientific Visualization Systems'. In: *Visualization in Scientific Computing*. Ed. by Gregory M. Nielson and Bruce Shriver. IEEE Computer Society Press, 1990, pp. 74–93. ISBN: 081868979X.

[48] Jeffrey Heer and Michael Bostock. 'Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design'. In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. 2010, pp. 203–212. DOI: 10.1145/1753326.1753357.

[49] Xiaoke Huang et al. 'TrajGraph: A Graph-Based Visual Analytics Approach to Studying Urban Network Centralities Using Taxi Trajectory Data'. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 160–169. DOI: 10.1109/TVCG.2015.2467771.

[50] Alfred Inselberg. 'The plane with parallel coordinates'. In: *The Visual Computer* 1.4 (1985), pp. 69–91. DOI: 10.1007/BF01898350.

[51] Tobias Isenberg et al. 'A systematic review on the practice of evaluating visualization'. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2818–2827. DOI: 10.1109/TVCG.2013.126.

[52] Halld'Or Janetzko et al. 'Feature-driven visual analytics of soccer data'. In: *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*. 2015, pp. 13–22. DOI: 10.1109/VAST.2014.7042477.

[53] Jeffrey Shaffer. *5 Tips on designing text tables*. 2016. URL: https://www.tableau.com/about/blog/2016/4/if-you-must-use-text-table-follow-these-5-design-tips-53766.

[54] Daniel A. Keim. 'Information visualization and visual data mining'. In: *IEEE Transactions on Visualization and Computer Graphics* 8.1 (2002), pp. 1–8. DOI: 10.1109/2945.981847.

[55] Daniel A. Keim et al. *Visual Analytics : Definition , Process and Challenges*. 2008, pp. 154–175. DOI: 10.1007/978-3-540-70956-5{_}7.

[56] Daniel A. Keim et al. 'Visual Analytics: Scope and Challenges'. In: *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*. Ed. by Simeon Simoff, Michael H. Böhlen and Arturas Mazeika. Springer Berlin, 2008, pp. 76–90. ISBN: 978-3-540-71080-6.

[57] Daniel Keim et al., eds. *Mastering the Information Age: Solving Problems with Visual Analytics*. Goslar: Eurographics Association, 2010. ISBN: 978-3-905673-77-7.

[58] Ethan Kerzner et al. 'A Framework for Creative Visualization-Opportunities Workshops'. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2018), pp. 748–758. DOI: 10.1109/TVCG.2018.2865241.

[59] Xiaolei Li, Jiawei Han and Sangkyum Kim. 'Motion-alert: Automatic anomaly detection in massive moving objects'. In: *Lecture Notes in Computer Science* 3975 LNCS (2006), pp. 166–177. DOI: 10.1007/11760146{_}15.

[60] Jock Mackinlay. 'Automating the Design of Graphical Presentations of Relational Information'. In: *ACM Transactions on Graphics* 5.2 (1986), pp. 110–141. DOI: 10.1145/22949.22950.

[61] Mapbox. *Maps*. 2019. URL: https://www.mapbox.com/maps/.

[62] Steve McConnell. *Code Complete*. 2nd ed. Microsoft Press, 2004. ISBN: 978-0-7356-1967-8.

[63] Tamara Munzner. 'A Nested Model for Visualization Design and Validation'. In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 921–928. DOI: 10.1109/TVCG.2009.111.

[64] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, 2014. ISBN: 978-1-4665-0891-0.

[65] Peter Pirolli and Stuart Card. 'The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis'. In:

Proceedings of the International Conference on Intelligence Analysis. Vol. 5. 2005, pp. 2–4.

[66] Matthew D. Plumlee and Colin Ware. 'Zooming versus multiple window interfaces: Cognitive costs of visual comparisons'. In: *ACM Transactions on Computer-Human Interaction* 13.2 (2006), pp. 179–209. DOI: 10.1145/1165734.1165736.

[67] William Ribarsky and Brian Fisher. 'The human-computer system: Towards an operational model for problem solving'. In: *Proceedings of the Annual Hawaii International Conference on System Sciences* 2016-March (2016), pp. 1446–1455. DOI: 10.1109/HICSS.2016.183.

[68] Jonathan C Roberts, Chris Headleand and Panagiotis D Ritsos. 'Sketching designs using the Five Design-Sheet methodology'. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 419–428. DOI: 10.1109/TVCG.2015.2467271.

[69] Dominik Sacha et al. 'Knowledge Generation Model for Visual Analytics'. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1604–1613. DOI: 10.1109/TVCG.2014.2346481.

[70] R.J. Scheepens. 'Visualization for Maritime Situational Awareness'. PhD thesis. Eindhoven University of Technology, 2015. ISBN: 9789038639314.

[71] Michael Sedlmair et al. 'Information visualization evaluation in large companies: Challenges, experiences and recommendations'. In: *Information Visualization* 10.3 (2011), pp. 248–266. DOI: 10.1177/1473871611413099.

[72] Shneiderman. 'Direct Manipulation: A Step Beyond Programming Languages'. In: *Computer* 16.8 (Aug. 1983), pp. 57–69. DOI: 10.1109/MC.1983.1654471.

[73] Ben Shneiderman. 'The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations'. In: *Proceedings of the IEEE Symposium on Visual Languages*. 1996, pp. 336–343. DOI: 10.1109/VL.1996.545307.

[74] Elizabeth A. Smith. 'The role of tacit and explicit knowledge in the workplace'. In: *Journal of Knowledge Management* 5.4 (Dec. 2001), pp. 311–321. DOI: 10.1108/13673270110411733.

[75] Larry W. Smith. 'Stakeholder analysis: A pivotal practice of successful projects'. In: *Proceedings of the Project Management Institute Annual Seminars & Symposium*. Houston, 2000.

[76] Statistics Norway. *Fiske - SSB*. URL: https://www.ssb.no/jord-skog-jakt-og-fiskeri/faktaside/fiske#blokk-3.

[77] Tableau. *Tableau*. URL: https://www.tableau.com/.

[78] James J Thomas and Kristin A Cook, eds. *Illuminating the Path: The Research amd Development Agenda for Visual Analytics*. IEEE, 2005. ISBN: 0769523234.

[79] Yann Tremblay et al. 'Interpolation of animal tracking data in a fluid environment'. In: *Journal of Experimental Biology* 209.1 (2006), pp. 128–140. DOI: 10.1242/jeb.01970.

[80] Edward Tufte. *The Visual Display of Quantitative Information*. 2nd ed. Graphics Press LLC, 2001. ISBN: 978-1930824133.

[81] Jarke J. Van Wijk. 'Bridging the gaps'. In: *IEEE Computer Graphics and Applications* 26.6 (2006), pp. 6–9. DOI: 10.1109/MCG.2006.120.

[82] Jarke J. Van Wijk. 'The value of visualization'. In: *Proceedings of the IEEE Visualization Conference*. 2005, pp. 79–86. DOI: 10.1109/VISUAL.2005.1532781.

[83] Taowei David Wang et al. 'Extracting insights from electronic health records: Case studies, a visual analytics process model, and design recommendations'. In: *Journal of Medical Systems* 35.5 (2011), pp. 1135–1152. DOI: 10.1007/s10916-011-9718-x.

[84] Colin Ware. *Information Visualization: Perception for Design*. 3rd ed. Morgan Kaufmann, 2013. ISBN: 978-0123814647.

[85] Jacob O. Wobbrock, Andrew D. Wilson and Yang Li. 'Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes'. In: *Proceedings of the ACM UIST Annual Symposium on User Interface Software and Technology*. 2007, p. 159. DOI: 10.1145/1294211.1294238.

[86] Pak Chung Wong and Jim Thomas. 'Visual analytics'. In: *IEEE Computer Graphics and Applications* 24.5 (Sept. 2004), pp. 20–21. DOI: 10.1109/MCG.2004.39.

[87] Glenn Xavier and Somayeh Dodge. 'An exploratory visualization tool for mapping the relationships between animal movement and the environment'. In: *Proceedings of the ACM SIGSPATIAL International Workshop on Interacting with Maps*. February. 2014, pp. 36–42. DOI: 10.1145/2677068.2677071.

[88] Ji Soo Yi, Youn ah Kang and John Stasko. 'Toward a Deeper Understanding of the Role of Interaction in Information Visualization'. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (Nov. 2007), pp. 1224–1231. DOI: 10.1109/TVCG.2007.70515.

[89] Haonan Yu et al. 'Video paragraph captioning using hierarchical recurrent neural networks'. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4584–4593. DOI: 10.1109/CVPR.2016.496.

[90] Leishi Zhang et al. 'Visual analytics for the big data era — A comparative review of state-of-the-art commercial systems'. In: *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*. Oct. 2012, pp. 173–182. DOI: 10.1109/VAST.2012.6400554.

[91] Ye Zhao et al. *TrajAnalytics: A Web-Based Visual Analytics Software of Urban Trajectory Data*. 2016. DOI: 10.6084/m9.figshare.4573057.v1.

[92] Hartmut Ziegler et al. 'Visual market sector analysis for financial time series data'. In: *Proeceedings of the IEEE Conference on Visual Analytics Science and Technology* (2010), pp. 83–90. DOI: 10.1109/VAST.2010.5652530.

[93] Torre Zuk et al. 'Heuristics for information visualization evaluation'. In: *Proceedings of the AVI Workshop on BEyond time and errors: novel evaluation methods for information visualization*. 2006, pp. 1–6. DOI: 10.1145/1168149.1168162.